Oracle® APEX

App Builder User's Guide

Release 24.2
G12913-02
February 2025

Oracle APEX App Builder User's Guide, Release 24.2

G12913-02

Copyright © 2003, 2025, Oracle and/or its affiliates.

Primary Authors: Terri Jennings, CM Dietrich

Contributors: Zaid Ali , Zsolt Angyal , Manal Berkane , Ananya Chatterjee, Christina Cho, Carsten Czarski , Stefan Dobre , Hilary Farrell, Alexis Galdamez, Florian Grasshoff, Paige Hanssen, Menno Hoogendijk, Edward Jones, Christian Lara, Carlos Maciel, Steven Muench, Ralf Muller, Frantisek Nagy, Christian Neumueller, Matthew Nolan, Jonathan Rhind, Antonio Rivera, Christian Rokitta, Anton Scheffer , Fanel Secara, John Snyders, Yassine Wahbi, Ronny Weiss, Patrick Wolf

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience xlix
Documentation Accessibility xlix
Diversity and Inclusion xlix
Related Documents l
Conventions l

1 Changes in Release 24.2 for Oracle APEX App Builder User's Guide

2 Quick Start

2.1 About Accessing Your Development Environment 2-1
2.2 Understanding Oracle APEX 2-2
2.2.1 What is Oracle APEX? 2-2
2.2.2 How Oracle APEX Works 2-3
2.2.3 About the Differences Between Runtime and Full Development Environments 2-3
2.3 Understanding the Workspace Home Page 2-4
2.3.1 Development Tools 2-4
2.3.2 Top Apps and Top Users Dashboards 2-6
2.3.3 Summary and Messages Regions 2-6
2.3.4 Resources and Available Updates Regions 2-7
2.4 APEX Menu Bar 2-9
2.4.1 APEX Menus 2-10
2.4.2 Global Search 2-10
2.4.3 Extension Menu 2-15
2.4.4 Administration Menu 2-15
2.4.5 Help Menu 2-17
2.4.6 Account Menu 2-18
2.5 Understanding App Builder 2-20
2.5.1 Accessing App Builder 2-20
2.5.2 Understanding the App Builder Home Page 2-21
2.5.2.1 App Builder Home Page Icons and Navigation Bar 2-22
2.5.2.2 Create Button 2-24

2.5.2.3 App Builder Home Side Bar 2-24
2.5.3 Accessing the Application Home Page 2-25
2.5.4 Understanding the Application Home Page 2-26
2.5.4.1 Edit Application Definition Button 2-26
2.5.4.2 Application Home Page Icons 2-28
2.5.4.3 Create Page Button 2-29
2.5.4.4 Application Home Page Navigation Bar 2-29
2.5.4.5 Application Home Side Bar 2-31
2.6 Changing Your Profile or Password 2-31
2.7 Editing User Preferences and Enabling Dark Mode 2-32
2.8 Using Oracle APEX Documentation 2-34
2.8.1 About Oracle Help Center 2-34
2.8.2 Downloading the Documentation Library 2-36
2.8.3 Redirecting the Help Menu 2-36
2.8.4 Viewing Field-Level and Page Designer Help 2-36
2.8.4.1 Viewing Field-Level Help in Oracle APEX 2-36
2.8.4.2 Viewing Help in Page Designer 2-38
2.9 Using the App Gallery 2-39
2.9.1 About Sample Apps, Starter Apps, Custom Apps, and Utility Apps 2-39
2.9.2 About the Oracle APEX GitHub Repository 2-40
2.9.2.1 Navigating the GitHub Repository 2-40
2.9.3 Installing Apps from the Gallery 2-41
2.9.4 Uninstalling Apps from the Gallery 2-43

3 App Builder Concepts

3.1 Understanding Applications 3-2
3.2 Running an Application or Page 3-3
3.2.1 Run Application and Run Page Buttons 3-4
3.2.2 How Your Browser Impacts the Way Applications Run 3-4
3.2.3 Running an Application from the App Builder Home Page 3-5
3.2.4 Running an Application from the Application Home Page 3-5
3.2.5 Running a Page on the Application Home Page 3-6
3.3 How Does Page Processing and Page Rendering Work? 3-6
3.3.1 How the APEX Engine Renders and Processes Pages 3-7
3.3.2 About Implicit Commit Points 3-7
3.3.3 Understanding Conditional Rendering and Processing 3-7
3.3.4 About Verifying User Identity 3-9
3.3.5 About Controlling Access to Controls and Components 3-9
3.4 About Enabling Support for Bookmarks 3-9
3.5 How Oracle APEX Uses Templates 3-10
3.5.1 Basic Templates 3-10

3.5.1.1 Template Substitution Syntax 3-11
3.5.1.2 Escape Filters 3-11
3.5.1.3 Built-in Substitution Strings 3-12
3.5.1.4 Substitutions in Text Messages (Legacy) 3-12
3.5.1.5 JavaScript in Templates 3-13
3.5.2 Theme Templates 3-13
3.5.3 Accessing the Universal Theme Reference App 3-14
3.5.4 Template Components 3-15
3.5.5 Template Directives 3-15
3.6 Understanding Session State Management 3-16
3.6.1 What is Session State? 3-16
3.6.2 About Session IDs 3-16
3.6.3 What Is a Session? 3-17
3.7 Managing Session State 3-17
3.7.1 Viewing Session State 3-18
3.7.2 Managing Session State Overrides 3-19
3.8 Understanding URL Syntax 3-20
3.8.1 Understanding Friendly URL Syntax 3-20
3.8.1.1 About Friendly URL Syntax 3-21
3.8.1.2 Enabling or Disabling Friendly URL Syntax 3-22
3.8.1.3 Customizing a Friendly URL 3-23
3.8.2 Understanding f?p URL Syntax 3-25
3.8.2.1 About f?p URL Syntax 3-25
3.8.2.2 About Using f?p Syntax to Link Pages 3-26
3.8.2.3 Calling a Page Using an Application Alias or Page Alias 3-27
3.8.2.4 Calling a Page from a Button URL 3-28
3.8.3 Hiding the Session ID on Public Pages 3-28
3.9 Managing Session State Values 3-29
3.9.1 About Referencing Session State 3-30
3.9.2 About Setting Session State 3-31
3.9.3 Clearing Session State 3-32
3.9.3.1 About Clearing Cache for an Item 3-33
3.9.3.2 About Clearing Cache for All Page Items 3-33
3.9.3.3 About Clearing Report Regions 3-34
3.9.3.4 Clearing Cache for an Entire Application 3-34
3.9.3.5 About Resetting an Application Completely 3-35
3.9.3.6 About Clearing Cache for the Current User Session 3-35
3.9.4 Referencing Session State Using Bind Variable Syntax 3-35
3.9.4.1 About Using Bind Variable Syntax 3-35
3.9.4.2 About Using Bind Variables in Regions Based on a SQL Query or LOV 3-36
3.9.4.3 About Using Bind Variables in Regions Based on PL/SQL 3-36
3.9.5 About Session Cloning 3-37

3.10 Using Substitution Strings 3-37
3.10.1 About Using Substitution Strings 3-38
3.10.1.1 Placeholder Substitution Strings within Templates 3-38
3.10.1.2 Session State Substitution Syntax for Page or Application Items 3-39
3.10.1.3 Substitution Strings for Interactive Grid, Cards, and Map Columns 3-39
3.10.1.4 Substitution Strings for Other Report Columns 3-39
3.10.1.5 Using Double Quotation Marks for Special Characters or Case Sensitivity 3-40
3.10.1.6 Substitutions in Text Messages (24.2 or later) 3-40
3.10.2 Controlling Output Escaping in Substitution Strings 3-41
3.10.3 Viewing Supported Substitution Strings for a Template 3-43
3.10.4 Using Built-in Substitution Strings 3-43
3.10.4.1 APEX_CSP_DISPLAY_NONE 3-45
3.10.4.2 APEX_FILES 3-46
3.10.4.3 APEX$ROW_NUM 3-46
3.10.4.4 APEX$ROW_SELECTOR 3-47
3.10.4.5 APEX$ROW_STATUS 3-47
3.10.4.6 APP_ID 3-47
3.10.4.7 APP_ALIAS 3-48
3.10.4.8 APP_AJAX_X01, ... APP_AJAX_X10 3-48
3.10.4.9 APP_BUILDER_SESSION 3-49
3.10.4.10 APP_DATE_TIME_FORMAT 3-49
3.10.4.11 APP_FILES 3-49
3.10.4.12 APP_NLS_DATE_FORMAT 3-50
3.10.4.13 APP_NLS_TIMESTAMP_FORMAT 3-50
3.10.4.14 APP_NLS_TIMESTAMP_TZ_FORMAT 3-50
3.10.4.15 APP_PAGE_ALIAS 3-51
3.10.4.16 APP_PAGE_ID 3-51
3.10.4.17 APP_REGION_ID 3-51
3.10.4.18 APP_REGION_STATIC_ID 3-51
3.10.4.19 APP_REQUEST_DATA_HASH 3-52
3.10.4.20 APP_SESSION 3-52
3.10.4.21 APP_SESSION_VISIBLE 3-53
3.10.4.22 APP_TITLE 3-53
3.10.4.23 APP_UNIQUE_PAGE_ID 3-53
3.10.4.24 APP_USER 3-54
3.10.4.25 APP_VERSION 3-55
3.10.4.26 AUTHENTICATED_URL_PREFIX 3-55
3.10.4.27 BROWSER_LANGUAGE 3-55
3.10.4.28 CURRENT_PARENT_TAB_TEXT 3-55
3.10.4.29 DEBUG 3-56
3.10.4.30 HOME_LINK 3-56
3.10.4.31 JET_BASE_DIRECTORY 3-56

3.10.4.32 JET_CSS_DIRECTORY 3-57
3.10.4.33 JET_JS_DIRECTORY 3-57
3.10.4.34 LOGIN_URL 3-57
3.10.4.35 LOGOUT_URL 3-57
3.10.4.36 MAIN_APP_ID 3-58
3.10.4.37 APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang 3-58
3.10.4.38 OWNER 3-58
3.10.4.39 PRINTER_FRIENDLY 3-58
3.10.4.40 PROXY_SERVER 3-59
3.10.4.41 PUBLIC_URL_PREFIX 3-59
3.10.4.42 REQUEST 3-59
3.10.4.43 Using REQUEST 3-60
3.10.4.44 SCHEMA OWNER 3-61
3.10.4.45 SQLERRM 3-61
3.10.4.46 SYSDATE_YYYYMMDD 3-62
3.10.4.47 THEME_DB_FILES 3-62
3.10.4.48 THEME_FILES 3-62
3.10.4.49 WORKSPACE_FILES 3-63
3.10.4.50 WORKSPACE_ID 3-63
3.11 Using Template Directives 3-64
3.11.1 About Template Directives 3-64
3.11.2 If Condition Directives 3-65
3.11.3 Case Condition Directives 3-66
3.11.4 Loop Directives 3-67
3.11.5 With and Apply Directives 3-67
3.11.6 Built-in Substitutions for Template Directives 3-68

4 Managing Team Development

4.1 Getting Started with Team Development 4-1
4.1.1 Accessing Team Development 4-2
4.1.2 Setting Up Team Development 4-2
4.1.3 Team Development Toolbar 4-3
4.2 Creating and Updating Issues 4-3
4.2.1 Issues Page 4-4
4.2.2 Creating an Issue 4-6
4.2.3 Updating Multiple Issues at Once 4-7
4.2.4 Updating an Issue 4-7
4.2.4.1 Issue Details Page 4-8
4.2.4.2 Updating the Issue Title 4-10
4.2.4.3 Updating the Issue Description 4-11
4.2.4.4 Managing Comments 4-11

4.2.4.5 Setting Assignees 4-13
4.2.4.6 Setting Labels 4-14
4.2.4.7 Setting Milestones 4-15
4.2.4.8 Associating an Issue with an App and Page 4-16
4.2.4.9 Marking an Issue as Duplicate 4-17
4.2.4.10 Closing an Issue 4-17
4.2.4.11 Managing Deleted Issues 4-19
4.3 Viewing Your User Profile 4-20
4.3.1 Viewing Your Issues 4-20
4.3.2 Viewing Your Notifications 4-20
4.4 Managing Labels 4-21
4.4.1 About Labels Groups and Labels 4-21
4.4.2 Creating Label Groups 4-22
4.4.3 Editing Label Groups 4-23
4.4.4 Adding Labels 4-23
4.4.5 Editing or Deleting Labels 4-24
4.5 Managing Milestones 4-25
4.5.1 Creating Milestones 4-25
4.5.2 Editing and Deleting Milestones 4-25
4.6 Managing Templates 4-26
4.6.1 Creating Issue Templates 4-26
4.6.2 Editing or Deleting Issue Templates 4-27
4.7 Using Team Development Utilities 4-27
4.7.1 Team Development Utilities 4-28
4.7.2 Managing Uploaded Files 4-28
4.7.3 Purging Data 4-29
4.7.4 Reassigning Issues to Another Assignee 4-29
4.7.5 Reassigning the Issue Owner 4-29
4.7.6 Removing a Milestone for All Issues 4-30
4.7.7 Removing Labels for All Issues 4-30
4.7.8 Managing Notification Preferences 4-31
4.7.8.1 Editing Your Own Notification Preferences 4-31
4.7.8.2 Adding User Notification Preferences 4-31
4.7.8.3 Editing Notification Preferences for Other Users 4-32
4.8 Managing Feedback 4-33
4.8.1 About Feedback 4-33
4.8.2 Adding Feedback Using the Create Application Wizard 4-34
4.8.3 Adding Feedback Using the Create Page Wizard 4-35
4.8.4 Submitting Feedback Within a Running Application 4-36
4.8.5 Reviewing Feedback Within an Application 4-36
4.8.6 Configuring Feedback to Support Attachments 4-37
4.8.7 Tracking Feedback 4-37

4.8.7.1 Viewing the Feedback Dashboard 4-37
4.8.7.2 Viewing the Feedback Report 4-38
4.8.7.3 Viewing Feedback by Filing User 4-39
4.8.7.4 Viewing the Feedback Calendar 4-39

5 Creating Applications

5.1 Choosing an Application Creation Method 5-2
5.2 Using the Simple Create Application Wizard 5-4
5.3 Using the Full Create Application Wizard 5-4
5.3.1 About the Full Create Application Wizard 5-5
5.3.2 Running the Full Create Application Wizard 5-6
5.3.3 Understanding Page Types, Features, and Settings 5-10
5.3.3.1 Available Page Types in the Create Application Wizard 5-10
5.3.3.2 Available Features in the Create Application Wizard 5-15
5.3.3.3 Available Settings in the Create Application Wizard 5-16
5.3.3.4 About Removing Features Created with the Create Application Wizard 5-19
5.4 Creating an App Using Generative AI 5-20
5.5 Creating an Application from a File 5-22
5.5.1 About Creating an Application from a File 5-22
5.5.2 Creating an Application by Uploading File 5-22
5.5.3 Creating an Application by Copying and Pasting from a File 5-27
5.6 Integrating APEX with Fusion Applications 5-31
5.6.1 About Oracle APEX and Fusion Applications Integration Requirements 5-31
5.6.2 About the Fusion Integration Starter Application 5-32
5.6.3 Creating an Application with Integrations to Fusion Applications 5-32
5.7 Using Application Blueprints 5-34
5.7.1 What Is an Application Blueprint? 5-34
5.7.2 Loading an Application Blueprint 5-35
5.7.3 Viewing an Application Blueprint 5-35
5.8 Copying an Application 5-37
5.8.1 Copying an Application by Running the Create Application Wizard 5-37
5.8.2 Copying an Application from the Application Home Page 5-38
5.9 Creating a Working Copy to Merge, Refresh, or Compare 5-38
5.9.1 About Working Copies 5-39
5.9.2 Creating a Working Copy of an Application 5-40
5.9.3 About Working Copy Menus and Tasks 5-41
5.9.4 Viewing and Editing a Working Copy 5-43
5.9.5 Comparing Changes between a Working Copy and Main 5-44
5.9.6 Comparing Changes Across Working Copies 5-46
5.9.7 Merging Changes from a Working Copy into Main 5-47
5.9.8 Refreshing a Working Copy with Changes from Main 5-49

5.9.9 Deleting a Working Copy 5-50
5.9.10 View or Edit Working Copy Details 5-50
5.10 Creating Applications for Mobile Devices 5-51
5.10.1 Advantages of the Universal Theme 5-51
5.10.2 Creating a Progressive Web App (PWA) 5-52
5.10.2.1 About Progressive Web App Support 5-52
5.10.2.2 Creating a Progressive Web App 5-54
5.10.2.3 Configuring Attributes for a Progressive Web App 5-55
5.11 Deleting an Application 5-59
5.11.1 Deleting an Application from the Application Home Page 5-59
5.11.2 Deleting an Application from the Edit Application Definition 5-59
5.12 Managing Application Attributes 5-60
5.12.1 Editing the Application Definition 5-60
5.12.1.1 Accessing the Application Definition Page 5-61
5.12.1.2 Application Definition Page 5-61
5.12.2 Configuring Security Attributes 5-74
5.12.2.1 Accessing the Security Page 5-74
5.12.2.2 Security Page 5-75
5.12.3 Configuring Globalization Attributes 5-85
5.12.3.1 Accessing the Globalization Page 5-86
5.12.3.2 Globalization Page 5-86
5.12.4 Editing User Interface Attributes 5-89
5.12.4.1 Accessing the User Interface Page 5-90
5.12.4.2 User Interface Page 5-90
5.12.5 Configuring Progressive Web App Attributes 5-98
5.12.5.1 Accessing the Progressive Web App Page 5-98
5.12.5.2 Progressive Web App Page 5-99
5.12.6 Configuring Generative AI Attributes 5-103
5.12.6.1 Accessing the AI Page 5-104
5.12.6.2 AI Page 5-104
5.13 Managing Application Logos 5-105
5.13.1 Managing Logos in the User Interface Attributes 5-105
5.13.2 Editing App Logos at Runtime 5-106
5.14 Managing Application Icons 5-108
5.14.1 About Application Icons 5-108
5.14.2 Managing Icons in User Interface Attributes 5-110
5.14.3 Using Edit App Icon on the Runtime Developer Toolbar 5-111
5.15 Adding Developer Comments 5-111
5.15.1 Locating Developer Comment Button 5-112
5.15.2 Adding Developer Comments to an Application 5-112
5.15.3 Viewing and Editing Developer Comments 5-113
5.15.4 Deleting Developer Comments 5-113

5.15.4.1 Deleting a Specific Developer Comment 5-113
5.15.4.2 Deleting Multiple Developer Comments 5-114
5.15.5 Accessing the Developer Comments from Shared Components 5-114
5.15.6 Viewing the Developer Comments Calendar 5-114
5.16 Using the Find Icon 5-115
5.16.1 Locating the Find Icon 5-115
5.16.2 Searching for Items 5-117
5.16.3 Searching for Pages 5-117
5.16.4 Searching for Queries 5-117
5.16.5 Searching for Tables 5-118
5.16.6 Using the PL/SQL Finder 5-118
5.16.7 Viewing Application Errors 5-119
5.17 Adding Application Comments 5-119
5.18 Managing Application Groups 5-119
5.18.1 About Using Application Groups 5-120
5.18.2 Viewing Application Groups 5-120
5.18.3 Creating an Application Group 5-121
5.18.4 Assigning a Application to an Application Group 5-121
5.18.5 Removing an Application from an Application Group 5-122
5.18.6 Deleting an Application Group 5-122
5.19 Using Application Utilities 5-123
5.19.1 Accessing Application Utilities 5-124
5.19.2 Application Utilities Page 5-124
5.19.3 Viewing the Application Dashboard 5-126
5.19.4 Viewing the Recently Updated Pages Report 5-127
5.19.5 Viewing the Change History Report 5-127
5.19.6 Viewing Database Object Dependencies 5-127
5.19.6.1 About the Database Object Dependencies Page 5-128
5.19.6.2 Generating the Database Object Dependencies Report 5-129
5.19.6.3 Understanding the Database Object Dependencies Report 5-130
5.19.7 Accessing APEX Views 5-134
5.19.8 Managing Application Backups 5-135
5.19.8.1 About Application Backups 5-135
5.19.8.2 Viewing All Application Backups 5-136
5.19.8.3 Managing Backups for a Specific Application 5-137
5.19.9 Using the Embedded Code Utility 5-139
5.19.10 Viewing Page and Application Checksums 5-141
5.19.10.1 Viewing the Application Checksum 5-141
5.19.10.2 Viewing the Page Checksum 5-142
5.19.11 Accessing Font APEX 5-143
5.19.11.1 About Font APEX Icons 5-143
5.19.11.2 Browsing Icons from Application Utilities 5-143

5.19.11.3 Browsing Font Icons from Page Designer 5-145
5.19.12 Accessing Page Specific Utilities 5-146
5.19.13 Accessing Cross Page Utilities 5-147
5.19.14 Using Interactive Report Utilities 5-148
5.19.14.1 Deleting Saved Interactive Reports 5-148
5.19.14.2 Viewing Subscriptions 5-148
5.20 Using Workspace Utilities 5-149
5.20.1 Accessing Workspace Utilities 5-149
5.20.2 Workspace Utilities Page 5-150
5.20.3 Managing Vector Providers 5-151
5.20.3.1 About Vector Providers 5-151
5.20.3.2 Creating a Vector Provider 5-152
5.20.3.3 Editing and Deleting a Vector Provider 5-153
5.20.3.4 Viewing the Vector Providers History Report 5-154
5.20.4 Viewing Cross Application Reports 5-154

6 Managing Pages in an Application

6.1 Adding a New Page to an Application 6-1
6.2 About Copying Application Pages 6-3
6.3 Creating a New Page by Copying 6-4
6.4 Understanding Page Types in the Create Page Wizard 6-6
6.4.1 Component Page Types 6-7
6.4.2 Feature Page Types 6-12
6.4.3 Legacy Page Types 6-13
6.5 Managing Feature Pages 6-14
6.5.1 About Feature Pages 6-14
6.5.2 Adding Features to an Existing App 6-14
6.5.3 About Removing Features Created with the Create Page Wizard 6-15
6.6 Creating Dialog Pages 6-15
6.6.1 About Supported Dialogs 6-16
6.6.2 Creating a Dialog Page Template 6-16
6.6.3 Editing a Dialog Page in Page Designer 6-17
6.6.4 Generating a Cancel Dialog Dynamic Action on the Dialog Page 6-19
6.6.5 Generating a Close Dialog Page Process on the Dialog Page 6-19
6.6.6 About Branching with Dialog Pages 6-20
6.6.7 Handling Refresh Actions on Parent Page 6-20
6.7 About Deleting Application Pages 6-20
6.8 Managing Page Groups 6-21
6.8.1 Viewing Page Groups 6-21
6.8.2 Creating a Page Group 6-22
6.8.3 Editing a Page Group Definition 6-22

6.8.4 Assigning a Page to a Page Group 6-22
6.8.5 Viewing Pages by Page Group 6-23
6.8.6 Deleting a Page Group 6-23
6.9 Locking and Unlocking a Page 6-24
6.9.1 Determining If a Page Is Locked 6-24
6.9.2 Locking a Page 6-24
6.9.2.1 Locking a Page from the Application Home Page 6-25
6.9.2.2 Locking a Page in Page Designer 6-25
6.9.2.3 Locking a Page from Utilities 6-25
6.9.3 Unlocking a Page 6-26
6.9.3.1 Unlocking Pages from the Application Home Page 6-26
6.9.3.2 Unlocking a Page in Page Designer 6-26
6.9.3.3 Unlocking Pages from Utilities 6-27
6.9.3.4 Unlocking Pages As a Workspace Administrator 6-27

7 Editing Pages in Page Designer

7.1 About Page Designer 7-1
7.2 Understanding Page Designer UI Elements 7-3
7.2.1 Page Designer Toolbar 7-3
7.2.2 Left Pane of Page Designer 7-6
7.2.2.1 About the Left Pane of Page Designer 7-6
7.2.2.2 Rendering Tab 7-7
7.2.2.3 Dynamic Actions Tab 7-8
7.2.2.4 Processing Tab 7-9
7.2.2.5 Page Shared Components Tab 7-10
7.2.3 Central Pane of Page Designer 7-11
7.2.3.1 Layout Tab 7-11
7.2.3.2 Page Search Tab 7-13
7.2.3.3 Help Tab 7-14
7.2.3.4 Gallery 7-15
7.2.4 Right Pane of Page Designer (Property Editor) 7-16
7.2.5 Page Designer Code Editor 7-17
7.3 Using Page Designer 7-21
7.3.1 Working with Page Designer 7-21
7.3.1.1 Searching for Pages 7-22
7.3.1.2 Viewing a Page in Page Designer 7-23
7.3.1.3 Accessing Property Editor 7-23
7.3.1.4 Dragging and Dropping Tabs Between Panes 7-24
7.3.1.5 Switching Between Three Pane and Two Pane Mode 7-25
7.3.1.6 About Adding Components Using Gallery Context Menus 7-26
7.3.2 Managing Regions, Items, and Buttons 7-27

7.3.2.1 Adding a Region, Item, or Button by Dragging and Dropping 7-28
7.3.2.2 Adding a Region, Item, or Button from the Rendering Tab 7-29
7.3.2.3 Copying a Region, Item, or Button 7-30
7.3.3 Creating Branches, Validations, Computations, and Processes 7-31
7.3.3.1 Creating Pre-Rendering Branches, Computations, and Processes 7-31
7.3.3.2 Creating Page Submission Branches, Validations, Computations, and Processes 7-32
7.3.4 Copying a Page in Page Designer 7-33
7.3.5 Commenting Out Page Components in Page Designer 7-35
7.4 Managing Page Attributes 7-36
7.4.1 Editing Page Attributes 7-36
7.4.2 Incorporating JavaScript Using Page Designer 7-37
7.4.3 Configuring Rejoin Sessions for a Page 7-38
7.4.4 Configuring Dialog Attributes 7-40
7.4.5 Configuring Pages to Warn Users of Unsaved Changes 7-41
7.5 Running a Page from Page Designer 7-42
7.6 Runtime Developer Toolbar 7-43
7.7 Deleting Pages 7-46
7.7.1 Deleting a Page 7-46
7.7.2 Deleting Multiple Pages 7-47
7.7.3 Deleting a Range of Pages 7-47

8 Developing Reports

8.1 Understanding Report Page Types 8-2
8.1.1 Report Page Type Use Cases 8-2
8.1.2 Comparison of Report Page Types 8-5
8.2 Creating a Report Using the Create Application Wizard 8-11
8.2.1 Report Options When Running the Create Application Wizard 8-11
8.3 Creating a Report Using the Create Page Wizard 8-13
8.3.1 Report Options When Running the Create Page Wizard 8-13
8.3.2 Creating a Report on New Page 8-16
8.4 Creating a Report by Dragging and Dropping from the Gallery 8-17
8.5 Managing Cards 8-18
8.5.1 Example Cards Page 8-19
8.5.2 Creating a Cards Page Using the Create Application Wizard 8-20
8.5.3 Creating a Cards Page Using the Create Page Wizard 8-21
8.5.4 Editing Cards Region Attributes 8-22
8.5.5 Editing Card Attributes 8-24
8.5.6 Using Actions to Link from a Cards Page 8-25
8.6 Managing Smart Filters 8-28
8.6.1 Example Smart Filters 8-28

8.6.2 About Creating Smart Filters Pages 8-30
8.6.2.1 About Creating Smart Filters Pages with a Wizard 8-31
8.6.2.2 Search Filter Types 8-32
8.6.3 Creating Smart Filters Using the Create Application Wizard 8-33
8.6.4 Creating Smart Filters Using the Create Page Wizard 8-34
8.6.5 Creating a Smart Filters Page Manually 8-36
8.6.6 Editing a Smart Filters Page 8-41
8.6.6.1 Reviewing Smart Filters Page Components 8-41
8.6.6.2 Understanding a Smart Filters Page 8-43
8.6.6.3 Adding a New Filter 8-45
8.6.6.4 Editing Filter Attributes 8-48
8.7 Managing Faceted Search 8-52
8.7.1 Example Faceted Search 8-52
8.7.2 About Creating Faceted Search Pages 8-54
8.7.2.1 About Creating Facet Search Pages with a Wizard 8-54
8.7.2.2 Search Facet Types 8-55
8.7.3 Creating Faceted Search Using the Create Application Wizard 8-56
8.7.4 Creating Faceted Search Using the Create Page Wizard 8-57
8.7.5 Creating a Faceted Search Manually 8-58
8.7.6 Editing a Faceted Search Page 8-65
8.7.6.1 Reviewing Faceted Search Page Components 8-66
8.7.6.2 Understanding a Faceted Search Page 8-68
8.7.6.3 Editing Faceted Search Region Attributes 8-70
8.7.6.4 Editing Facet Attributes 8-72
8.7.6.5 Adding a New Facet 8-78
8.7.6.6 Adding a Facet on a Column Containing Multiple Values 8-81
8.7.6.7 Creating Facet Groups 8-84
8.7.6.8 Configuring Single Facet Charts 8-87
8.7.6.9 Editing the Static List of Values for a Range Facet 8-88
8.8 Managing Interactive Grids 8-91
8.8.1 Example Interactive Grid 8-92
8.8.2 Supported Columns in Interactive Grids 8-94
8.8.3 Creating an Interactive Grid Using the Create Application Wizard 8-96
8.8.4 Creating an Interactive Grid Using the Create Page Wizard 8-98
8.8.5 Creating Master Detail from an Existing Interactive Grid 8-99
8.8.6 Managing Interactive Grid Region Attributes 8-103
8.8.6.1 Editing Interactive Grid Region Attributes 8-103
8.8.6.2 Displaying Interactive Grids Conditionally 8-105
8.8.7 Managing Interactive Grid Attributes 8-105
8.8.7.1 Editing Interactive Grid Attributes 8-106
8.8.7.2 Making an Existing Interactive Grid Editable 8-107
8.8.7.3 Controlling Interactive Grid Pagination 8-108

8.8.7.4 Customizing the Interactive Grid Toolbar 8-108
8.8.7.5 Enabling Users to Save Interactive Grid Public Reports 8-110
8.8.7.6 Enabling Flashback and Charts 8-110
8.8.7.7 Configuring Interactive Grid Download Formats 8-111
8.8.7.8 Enabling Icon View in an Interactive Grid 8-112
8.8.7.9 Enabling Detail View in an Interactive Grid 8-114
8.8.8 Managing Interactive Grid Column Attributes 8-116
8.8.8.1 Editing Common Grid Column Attributes 8-116
8.8.8.2 Enabling Hide, Sort, Control Breaks, and Aggregates 8-118
8.8.8.3 Defining a Column as a List of Values for Filtering 8-118
8.8.8.4 Creating a Column Link in an Interactive Grid 8-119
8.8.8.5 About Configuring Interactive Grids to Support Multi Select 8-120
8.8.9 About Customizing Interactive Grids in a Running Application 8-121
8.8.10 Saving Interactive Grid Reports 8-121
8.8.10.1 How the User Type Effects Available Save Options 8-122
8.8.10.2 About Configuration Dependencies When Saving Interactive Grid Reports 8-122
8.8.10.3 Saving a Private Interactive Grid Report 8-123
8.8.10.4 Saving a Public Interactive Grid Report 8-123
8.8.10.5 Saving or Renaming a Primary Interactive Grid Report 8-124
8.8.10.6 Saving an Alternative Interactive Grid Report 8-124
8.8.10.7 Renaming a Saved Interactive Grid Report 8-125
8.8.10.8 Deleting a Saved Interactive Grid Report 8-125
8.8.10.9 Resetting a Saved Interactive Grid Report 8-125
8.8.11 Linking to Interactive Grid Reports 8-126
8.8.11.1 Interactive Grid Request Syntax 8-127
8.8.11.2 Interactive Grid Name and Value Syntax 8-127
8.8.11.3 Interactive Grid ClearCache Syntax 8-128
8.8.11.4 Restrictions When Linking to Interactive Grid Reports 8-129
8.8.11.5 Valid Interactive Grid Linking Examples 8-129
8.8.11.6 Invalid Interactive Grid Linking Examples 8-129
8.8.11.7 Linking to Shared Interactive Grid Reports 8-130
8.9 Managing Interactive Reports 8-131
8.9.1 Example Interactive Report 8-132
8.9.2 Supported Columns in Interactive Reports 8-134
8.9.3 Creating an Interactive Report Using the Create Application Wizard 8-135
8.9.4 Creating an Interactive Report Using the Create Page Wizard 8-136
8.9.5 Managing Interactive Report Region Attributes 8-138
8.9.5.1 Editing Interactive Report Region Attributes 8-138
8.9.5.2 Displaying Interactive Reports Conditionally 8-139
8.9.6 Managing Interactive Report Attributes 8-140
8.9.6.1 Editing Interactive Report Attributes 8-141
8.9.6.2 Managing Link Columns 8-141

8.9.6.3 Customizing an Interactive Report Search Bar 8-144
8.9.6.4 Customizing the Interactive Report Actions Menu 8-147
8.9.6.5 Configuring Interactive Report Download Options 8-148
8.9.6.6 Controlling Interactive Report Pagination 8-153
8.9.6.7 Enabling Icon View 8-154
8.9.6.8 Enabling Detail View 8-156
8.9.6.9 Configuring Advanced Attributes for Interactive Reports 8-156
8.9.7 Managing Interactive Report Column Attributes 8-158
8.9.7.1 Editing Interactive Report Column Attributes 8-159
8.9.7.2 Creating a Column Link in an Interactive Report 8-160
8.9.7.3 Defining a Column as a List of Values in an Interactive Report 8-161
8.9.7.4 Formatting Columns in an Interactive Report 8-161
8.9.7.5 About Filtering on Conditional Links in Interactive Report Columns 8-162
8.9.8 Customizing Interactive Reports in a Running Application 8-163
8.9.8.1 About the Search Bar 8-164
8.9.8.2 About the Actions Menu 8-165
8.9.8.3 Using Interactive Report Filters 8-166
8.9.8.4 Saving Interactive Reports 8-167
8.9.9 Linking to Interactive Reports 8-173
8.9.9.1 Interactive Report Request Syntax 8-174
8.9.9.2 Interactive Report Name and Value Syntax 8-175
8.9.9.3 Interactive Report ClearCache Syntax 8-176
8.9.9.4 Restrictions When Linking to Interactive Reports 8-176
8.9.9.5 Valid Interactive Report Linking Examples 8-177
8.9.9.6 Invalid Interactive Report Linking Examples 8-178
8.9.9.7 Linking to Shared Interactive Reports 8-178
8.10 Managing Classic Reports 8-180
8.10.1 Example Classic Report 8-180
8.10.2 Supported Columns in Classic Reports 8-181
8.10.3 Creating a Classic Report Using the Create Application Wizard 8-182
8.10.4 Creating a Classic Report Using the Create Page Wizard 8-183
8.10.5 Editing Classic Report Region Attributes 8-185
8.10.6 Managing Classic Report Attributes 8-185
8.10.6.1 Editing Classic Report Attributes 8-186
8.10.6.2 Updating Classic Report Appearance Attributes 8-187
8.10.6.3 Editing Classic Report Pagination 8-188
8.10.6.4 Adding Support for XML File or a CSV File Export 8-189
8.10.6.5 Enabling Classic Report CSV Output Option 8-189
8.10.6.6 Controlling Classic Report Column Breaks 8-190
8.10.7 Managing Classic Report Column Attributes 8-190
8.10.7.1 Altering Classic Report Column Layout 8-191
8.10.7.2 Enabling Column Sorting in a Classic Report 8-192

8.10.7.3 Creating a Column Link in a Classic Report 8-193
8.10.7.4 Defining an Updatable Column in a Classic Report 8-194
8.10.7.5 Defining a Column as a List of Values in a Classic Report 8-194
8.10.7.6 Controlling When Classic Report Columns Display 8-195
8.10.7.7 Adding a Download Link to a Classic Report 8-196
8.11 Managing Sorting with Order By Item 8-196
8.11.1 Order By Item Sort Example 8-196
8.11.2 Configuring Sorting with Order By Item 8-197
8.12 Printing Report Regions 8-201
8.12.1 About Report Printing 8-202
8.12.2 About Printing Reports to PDF and Excel 8-203
8.12.3 Configuring Report Printing at the Application Level 8-203
8.12.4 About Classic Report Printing Methods 8-205
8.12.5 Configuring Classic Report Printing Attributes 8-206
8.12.6 About Report Queries 8-206
8.12.7 About Report Layouts 8-207
8.13 Understanding BLOB Support in Forms and Reports 8-207
8.13.1 About BLOB Support in Reports 8-207
8.13.2 Including a Download Link 8-208
8.13.3 Displaying a BLOB Image in a Report 8-209
8.13.4 About Working With BLOBs Procedurally 8-210
8.13.5 About Other Declarative Download Options 8-210

9 Developing Forms

9.1 Understanding Form Types 9-2
9.1.1 Editable Interactive Grid 9-2
9.1.2 Form 9-3
9.1.3 Report with Form 9-4
9.1.4 Master Detail Forms 9-4
9.2 Creating a Form 9-5
9.2.1 Creating a Form Using the Create Application Wizard 9-5
9.2.2 Creating a Form Using the Create Page Wizard 9-7
9.3 Configuring Lost Update Detection 9-8
9.4 About Making an Interactive Grid Editable 9-9
9.5 Creating Master Detail Forms 9-10
9.5.1 About Master Detail Forms 9-10
9.5.2 Creating a Master Detail Form Using the Create Application Wizard 9-14
9.5.3 Creating a Stacked Master Detail Using the Create Page Wizard 9-16
9.5.4 Creating a Side by Side Master Detail Using the Create Page Wizard 9-18
9.5.5 Creating a Drill Down Master Detail Using the Create Page Wizard 9-19
9.6 Validating User Input in Forms 9-22

9.6.1 About Creating Validations and Error Messages 9-22
9.6.2 Associating a Validation with a Specific Item 9-22
9.6.3 About the Error Message Display Location 9-23
9.7 Understanding BLOB Support in Forms 9-23
9.7.1 About BLOB Support in Forms 9-23
9.7.2 About Uploading and Downloading Files into a BLOB 9-24
9.7.3 About Displaying the BLOB 9-24
9.7.4 About Removing the Image Reference 9-25

10 Managing Application Components

10.1 Understanding Oracle JET Integration with Oracle APEX 10-1
10.1.1 About Oracle JET 10-1
10.1.2 How Oracle JET is integrated with Oracle APEX 10-2
10.2 Creating Calendars 10-3
10.2.1 About Supported Calendars 10-4
10.2.2 Sample Calendar 10-5
10.2.3 Creating a Calendar Using the Create Application Wizard 10-6
10.2.4 Creating a Calendar Using the Create Page Wizard 10-7
10.2.5 Creating a Calendar in Page Designer 10-9
10.2.6 Managing Calendar Attributes 10-10
10.2.6.1 Editing Calendar Attributes in the Property Editor 10-11
10.2.6.2 Editing an Existing Calendar to Include Add and Edit Functionality 10-11
10.2.6.3 Adding Calendar Initialization JavaScript Code 10-13
10.2.6.4 Enabling the Dragging and Dropping of Data in an Existing Calendar 10-15
10.2.7 About Dynamic Action Support for Calendar 10-16
10.3 Creating Charts 10-17
10.3.1 About Supported Charts 10-17
10.3.2 Sample Charts 10-19
10.3.3 Creating a Chart Using the Create Application Wizard 10-20
10.3.4 Creating a Dashboard Page Using the Create Application Wizard 10-21
10.3.5 Adding a Chart Using the Create Page Wizard 10-23
10.3.6 Adding a Chart in Page Designer 10-24
10.3.7 Editing Charts 10-26
10.3.7.1 Editing Chart Attributes 10-26
10.3.7.2 Switching Chart Type 10-27
10.3.7.3 Enabling Automatic Refresh 10-27
10.3.7.4 Using Custom JavaScript with Charts 10-29
10.4 Creating Maps 10-30
10.4.1 About Supported Maps 10-31
10.4.2 Sample Maps 10-32
10.4.3 Creating a Map Using the Create Application Wizard 10-34

10.4.4 Adding a Map Region in Page Designer 10-35
10.4.5 Adding a Map Using the Create Page Wizard 10-37
10.4.6 Editing Maps 10-39
10.4.6.1 Viewing and Editing a Map in Page Designer 10-40
10.4.6.2 Adding a New Layer to an Existing Map 10-42
10.4.6.3 Controlling Layer Visibility 10-45
10.4.6.4 Connecting a Map Region with a Report Region 10-47
10.4.7 Creating Custom Map Backgrounds 10-54
10.4.7.1 About Supported Map Background Types 10-54
10.4.7.2 Creating a Custom Map Background 10-54
10.4.7.3 Editing a Custom Map Background 10-56
10.4.7.4 Deleting a Custom Map Background 10-57
10.4.7.5 Copying a Custom Map Background 10-57
10.4.7.6 Accessing Map Background Reports 10-58
10.4.7.7 Using Custom Map Backgrounds 10-58
10.5 Creating Help for Your Application 10-61
10.5.1 Creating Help 10-61
10.5.1.1 About Creating Help 10-62
10.5.1.2 Creating a Help Page and Region 10-62
10.5.1.3 Defining Page Help 10-63
10.5.1.4 Defining Item Help 10-63
10.5.2 Using the Edit Item Help Report 10-64
10.5.3 Seeding Item Help Topics 10-65
10.5.4 Creating a Help Navigation Bar Entry 10-65

11 Using Themes and Theme Styles

11.1 Using Themes 11-1
11.1.1 About Themes 11-2
11.1.2 Understanding the Universal Theme 11-2
11.1.2.1 About the Universal Theme 11-3
11.1.2.2 About Updating or Migrating to the Universal Theme 11-4
11.1.2.3 Refreshing the Universal Theme 11-4
11.1.3 Using Theme Styles 11-5
11.1.3.1 About Theme Styles 11-5
11.1.3.2 Creating a Theme Style 11-6
11.1.3.3 Editing a Theme Style 11-7
11.1.3.4 Selecting a Theme Style on the User Interface Page 11-7
11.1.3.5 Enabling Users to Select a Theme Style 11-8
11.1.4 Using Theme Roller 11-8
11.1.4.1 About Theme Roller 11-8
11.1.4.2 Using the Theme Roller Dialog 11-9

11.1.4.3 About the Redwood Light Theme Style 11-12
11.1.4.4 Importing and Exporting Theme Styles 11-14
11.1.4.5 Style Compatibility 11-16
11.1.4.6 About Developer Console Integration 11-16
11.1.5 Understanding Template Options 11-17
11.1.5.1 About Template Options 11-18
11.1.5.2 About Default Template Options 11-18
11.1.5.3 About Template Option Groups and Presets 11-18
11.2 Managing Themes and Subscriptions 11-19
11.2.1 Managing Theme Subscriptions 11-19
11.2.1.1 About Theme Subscriptions 11-20
11.2.1.2 About Modifying a Theme Subscription 11-20
11.2.1.3 Verifying if a Theme Refresh Is Needed 11-21
11.2.1.4 Refreshing a Subscribed Theme 11-21
11.2.1.5 Unsubscribing to a Theme 11-22
11.2.1.6 Re-subscribing to a Built-in Theme 11-22
11.2.2 Accessing New Themes 11-23
11.2.3 Editing Themes 11-24
11.2.3.1 Editing a Theme 11-24
11.2.3.2 Theme Attributes 11-24
11.2.4 Switching Themes 11-30
11.2.4.1 About Switching the Active Theme 11-30
11.2.4.2 Switching an Active Theme 11-30
11.2.5 Changing a Theme Identification Number 11-32
11.2.6 Copying a Theme 11-32
11.2.7 Deleting a Theme 11-33
11.2.8 Creating a Master Theme Application 11-33
11.2.8.1 Creating a Master Theme Application by Creating a New Application 11-34
11.2.8.2 Creating a Master Theme Application from an Existing Application 11-34
11.2.9 Managing Workspace Themes 11-35
11.2.9.1 Adding a Workspace Theme to the Theme Repository 11-35
11.2.9.2 Viewing an Existing Workspace Theme 11-36
11.2.9.3 Deleting a Workspace Theme 11-36
11.2.9.4 Modifying a Workspace Theme 11-37
11.2.9.5 Exporting a Workspace Theme 11-37
11.2.10 Managing Instance Themes 11-37
11.2.11 About Exporting and Importing Themes 11-38
11.2.12 Viewing Theme Reports 11-38
11.2.12.1 Viewing All Templates in a Theme 11-38
11.2.12.2 Viewing Theme Template Counts 11-39
11.2.12.3 Viewing File References 11-39
11.2.12.4 Viewing Class References 11-40

11.2.12.5 Viewing Template Substitution Strings 11-40
11.3 Creating Custom Themes 11-41
11.3.1 About Cascading Style Sheets 11-42
11.3.2 About Calling the JavaScript File from the Page Template 11-43
11.3.3 About Using Escaping Syntax in Substitution Strings 11-43
11.3.4 Selecting a Default Page Template 11-43
11.3.4.1 Selecting a Page-level Template Within a Theme 11-44
11.3.4.2 Selecting a Page-level Template for a Specific Page 11-44
11.3.5 Creating a New Template 11-45
11.3.6 Using Template Options 11-45
11.3.6.1 Managing Global Template Options 11-45
11.3.6.2 Managing Template Options 11-50
11.3.7 Viewing Template Reports 11-55
11.3.8 Managing Templates 11-56
11.3.8.1 Viewing Templates on the Templates Page 11-56
11.3.8.2 Viewing Templates Associated with a Specific Page 11-57
11.3.8.3 Replacing Templates 11-57
11.3.8.4 Replacing All Templates within an Application 11-58
11.3.8.5 Viewing Region Position Utilization by Page Template 11-58
11.3.8.6 Unsubscribing to Templates 11-59
11.3.8.7 Publishing Templates 11-59
11.3.9 Breadcrumb Templates 11-59
11.3.9.1 About Breadcrumb Style Navigation 11-60
11.3.9.2 Breadcrumb Template Attributes 11-61
11.3.10 Button Templates 11-62
11.3.10.1 Button Template Attributes 11-62
11.3.11 Legacy Calendar Templates 11-64
11.3.11.1 Supported Legacy Calendar Template Substitution Strings 11-64
11.3.11.2 Legacy Calendar Template Attributes 11-64
11.3.12 Label Templates 11-65
11.3.12.1 Label Template Attributes 11-65
11.3.12.2 About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help 11-67
11.3.13 List Templates 11-68
11.3.13.1 About Hierarchical Lists 11-68
11.3.13.2 List Template Attributes 11-69
11.3.14 Page Templates 11-71
11.3.14.1 About Page Templates 11-71
11.3.14.2 Page Template Attributes 11-71
11.3.14.3 Required Dialog Page Template Attributes 11-76
11.3.15 Popup LOV Templates 11-76
11.3.15.1 Popup List of Values Template Attributes 11-77
11.3.16 Template Component Type Plug-ins 11-78

11.3.17 Region Templates 11-79
11.3.17.1 Region Template Attributes 11-79
11.3.18 Report Templates 11-83
11.3.18.1 Generic Column Templates and Named Column Templates 11-83
11.3.18.2 Report Column Template Attributes for Generic Column Templates 11-84
11.3.18.3 Report Column Template Attributes for Named Column Templates 11-88
11.3.18.4 About Using JavaScript in Column Templates 11-91
11.4 Using Custom Cascading Style Sheets 11-92
11.4.1 About Cascading Style Sheets 11-92
11.4.2 Referencing an Uploaded Cascading Style Sheet in the Page Template 11-92

12 Controlling Page Layout

12.1 About Page Layout in Oracle APEX 12-1
12.2 Optimizing a Page for Printing 12-2
12.2.1 Selecting a Printer Friendly Template for an Application 12-2
12.2.2 Using f?p Syntax to Toggle to Printer Friendly Mode 12-3
12.3 Creating a Global Page to Display Components on Every Page 12-3
12.3.1 Creating a Global Page 12-4
12.3.2 Viewing or Editing a Global Page 12-4
12.4 Managing Regions 12-5
12.4.1 About Regions 12-5
12.4.2 Creating a Region in Page Designer 12-5
12.4.2.1 Creating a Region from the Rendering Tab 12-6
12.4.2.2 Creating a Region from the Gallery 12-6
12.4.2.3 Supported Region Types 12-8
12.4.3 Editing Regions 12-11
12.4.3.1 Editing Region Attributes 12-11
12.4.3.2 Updating the Region Name or Title 12-12
12.4.3.3 Controlling Region Positioning 12-13
12.4.3.4 Specifying a Region Header and Footer 12-13
12.4.3.5 Enabling Region Caching 12-13
12.4.3.6 Enabling Users to Customize a Page 12-14
12.4.3.7 Specifying a Static Region ID 12-15
12.4.3.8 Adding a Region Image 12-15
12.4.3.9 Creating a Region Display Selector 12-16
12.4.4 Copying a Region 12-17
12.4.5 Deleting Regions 12-18
12.5 About Incorporating Content from Other Websites 12-18
12.6 About Managing Images 12-19
12.7 Rendering HTML Using Custom PL/SQL 12-19

13 Managing Application Controls

13.1 Managing Page Items 13-1
13.1.1 Understanding Page Items 13-2
13.1.1.1 About the Differences Between Page Items and Application Items 13-2
13.1.1.2 About Item Naming Conventions 13-3
13.1.1.3 Referencing Item Values 13-3
13.1.1.4 About Referencing Items Using JavaScript 13-4
13.1.1.5 Working with Multiple Select List Item 13-5
13.1.2 Viewing Page Items 13-6
13.1.3 Creating Page Items 13-7
13.1.3.1 Creating a Page Item from the Rendering Tab 13-7
13.1.3.2 Adding a Page Item from the Gallery 13-8
13.1.3.3 Creating a Static List of Values 13-9
13.1.3.4 Creating a Cascading List of Values 13-13
13.1.3.5 Creating a Shuttle Item on the Form Page 13-14
13.1.4 Editing Page Items 13-16
13.1.4.1 Editing Page Item Attributes in the Property Editor 13-17
13.1.4.2 Positioning Page Items 13-17
13.1.4.3 Defining Default Values for Page Items 13-18
13.1.4.4 Configuring Page Item Security 13-18
13.1.4.5 Changing an Existing Item to Use Quick Picks 13-20
13.1.4.6 Configuring Multiple Value Attributes 13-20
13.1.4.7 Displaying Conditional Page Items 13-22
13.1.4.8 Displaying Read Only Page Items 13-22
13.1.4.9 Applying a Format Mask to an Item 13-23
13.1.4.10 Configuring Item Attributes to Warn Users of Unsaved Changes 13-23
13.1.4.11 Controlling How Page Items Write Session State 13-24
13.1.4.12 Viewing Item Utilities 13-25
13.2 Managing Dynamic Actions 13-25
13.2.1 About Dynamic Actions 13-26
13.2.2 About Dynamic Action Events 13-27
13.2.3 Creating a Dynamic Action 13-29
13.2.4 Editing Dynamic Actions 13-32
13.2.5 Defining Dynamic Action Event Scope 13-33
13.2.6 Deleting a Dynamic Action 13-34
13.2.7 About Calling JavaScript Using a Dynamic Action 13-35
13.2.8 Debugging Dynamic Actions 13-35
13.2.8.1 About Debugging Dynamic Actions 13-35
13.2.8.2 Debugging Dynamic Actions 13-35
13.3 Managing Buttons 13-36
13.3.1 About Actions Buttons Can Perform 13-37

13.3.2 About Branching with Buttons 13-37
13.3.3 Creating a Button 13-38
13.3.4 About the Relationship Between Button Names and REQUEST 13-39
13.3.5 Editing a Button 13-39
13.3.6 Displaying a Button Conditionally 13-40
13.3.7 Configuring Button Attributes to Warn Users of Unsaved Changes 13-41
13.3.8 Avoiding Multiple Page Submits by Enabling Show Processing 13-41
13.3.9 Displaying a Button Confirmation Dialog 13-42
13.4 Managing Trees 13-43
13.4.1 About Trees 13-43
13.4.2 Creating a Tree Using the Create Page Wizard 13-44
13.4.3 Creating a Tree in Page Designer 13-45
13.4.4 Editing Tree Attributes 13-47
13.5 About Incorporating JavaScript into an Application 13-48

14 Adding Navigation

14.1 About Using Lists as Navigation 14-1
14.2 Controlling Navigation Using Branches 14-2
14.2.1 Creating a Branch 14-3
14.2.2 About Branching Conditionally 14-4
14.3 Creating Breadcrumbs 14-4
14.3.1 About Breadcrumbs 14-5
14.3.2 Creating a Breadcrumb While Creating a Page 14-5
14.3.3 Creating a Breadcrumb Manually 14-6
14.3.3.1 About Creating Breadcrumbs Manually 14-6
14.3.3.2 Creating Breadcrumbs from Page Designer 14-7
14.3.3.3 Creating Breadcrumbs from Shared Components 14-9
14.3.3.4 Adding a Breadcrumb Region with a Wizard 14-11
14.3.3.5 Adding a Breadcrumb Region from the Gallery 14-12
14.3.3.6 About Creating Dynamic Breadcrumbs 14-13
14.3.4 Editing Breadcrumbs 14-13
14.3.4.1 Editing Breadcrumb and Breadcrumb Entries 14-14
14.3.4.2 Adding Breadcrumb Entries 14-15
14.3.4.3 Reparenting Breadcrumb Entries 14-17
14.3.4.4 Deleting Unused Breadcrumb Entries 14-17
14.3.4.5 Editing Breadcrumb Attributes in Page Designer 14-18
14.3.5 Accessing Breadcrumb Reports 14-18
14.4 Managing Tabs 14-19
14.4.1 About Tabs 14-20
14.4.2 About the Tabs Page 14-20
14.4.3 Checking for Tab Template Support 14-21

14.4.4 Creating a Tab 14-22
14.4.5 Editing Tabs 14-24
14.4.5.1 Editing Multiple Tabs Simultaneously 14-24
14.4.5.2 Managing Standard Tabs 14-25
14.4.6 Accessing Tab Reports 14-25
14.5 Managing a Classic Navigation Bar 14-26
14.5.1 About Classic Navigation Bar Entries 14-27
14.5.2 Creating a Classic Navigation Bar Entry 14-27
14.5.2.1 Creating a Classic Navigation Bar Entry from Scratch 14-27
14.5.2.2 Creating a Classic Navigation Bar Entry for Feedback 14-29
14.5.3 Copying a Classic Navigation Bar Entry 14-30
14.5.4 Editing a Classic Navigation Bar Entry 14-30
14.5.5 Editing Multiple Classic Navigation Bar Entries Simultaneously 14-31
14.5.6 Accessing Classic Navigation Bar Entry Reports 14-31

15 Managing Computations, Validations, and Processes

15.1 Understanding Page Computations 15-1
15.1.1 About Page Computations 15-2
15.1.2 Creating a Page Computation 15-2
15.1.3 Editing Page Computation Attributes 15-3
15.1.4 Editing the Computation Point and Source 15-3
15.1.5 Example: Computation Points and Computation Syntax 15-4
15.2 Understanding Validations 15-6
15.2.1 About Validations 15-6
15.2.2 What Happens When a Validation Fails? 15-7
15.2.3 Creating a Validation 15-7
15.2.4 Editing a Validation 15-9
15.2.5 About Determining When Validations Execute 15-9
15.2.6 About Defining How Validation Error Messages Display 15-10
15.2.7 Processing Validations Conditionally 15-11
15.2.8 About Altering the Go to Error Link 15-11
15.3 Understanding Page Processes 15-12
15.3.1 About Page Processes 15-12
15.3.2 Creating Page Processes 15-14
15.3.3 Editing Page Processes 15-16
15.3.3.1 Editing Page Process Attributes 15-16
15.3.3.2 Controlling When a Page Process Executes 15-17
15.3.3.3 Making a Page Process Conditional 15-17
15.3.3.4 Creating Page Process Success and Error Messages 15-18
15.3.4 Understanding Background Page Processing 15-19
15.3.4.1 About Background Page Processes 15-19

15.3.4.2 Creating a Background Page Page Process 15-20
15.3.4.3 Monitoring Currently Running Background Processes 15-22
15.4 Understanding Application Processes 15-23
15.4.1 On Demand Application Processes 15-24
15.4.2 About Application Processes that Execute On New Instance 15-24
15.4.3 About Running an On Demand Process from a Page Request 15-25
15.4.4 Example: Application Process 15-26
15.4.5 Creating an Application Process 15-26
15.4.6 Editing Application Process Attributes 15-27
15.4.7 Copying or Subscribing to Application Processes 15-27
15.4.8 Subscribing to an Application Process 15-28
15.4.9 Creating Application Process Error Messages 15-29
15.4.10 Viewing the Application Processes History Report 15-30
15.5 About Branches 15-31
15.6 Understanding Application Computations 15-31
15.6.1 About Application Computations 15-32
15.6.2 About Application Computations that Execute On New Instance 15-32
15.6.3 Creating an Application Computation 15-32
15.6.4 Copying or Subscribing to Application Computations 15-33
15.6.5 Subscribing to an Application Computation 15-34
15.6.6 Accessing the Application Computation History Report 15-35
15.7 Using the Attribute Dictionary 15-36
15.7.1 About the Attribute Dictionary 15-36
15.7.2 Accessing the Manage Attribute Dictionary 15-36

16 Managing Shared Components

16.1 Accessing the Shared Components Page 16-2
16.2 Shared Components Page 16-2
16.2.1 Application Logic 16-3
16.2.2 Security 16-4
16.2.3 Other Components 16-4
16.2.4 Navigation and Search 16-5
16.2.5 User Interface 16-6
16.2.6 Files and Reports 16-6
16.2.7 Data Sources 16-7
16.2.8 Workflows and Automations 16-7
16.2.9 Globalization 16-7
16.2.10 Generative AI 16-8
16.2.11 Tasks Region 16-8
16.2.12 Workspace Objects Region 16-9
16.3 Using Shared Component Subscriptions 16-9

16.3.1 About Shared Component Subscriptions 16-10
16.3.2 About Automatic Dependency Resolution 16-11
16.3.3 Subscribing to a Shared Component 16-12
16.3.4 About Subscribed Component Subscription Status 16-13
16.3.5 Refreshing a Subscribed Shared Component 16-14
16.3.6 Publishing a Master Shared Component 16-15
16.3.7 Unsubscribing to a Shared Component 16-15
16.4 Managing Application Settings 16-16
16.4.1 Creating an Application Setting 16-16
16.4.2 Editing or Deleting Application Setting 16-17
16.4.3 Copying or Subscribing to Application Settings 16-17
16.4.4 Subscribing to a Application Settings 16-18
16.5 Managing Application Items 16-19
16.5.1 Creating an Application Item 16-20
16.5.2 Editing Application Item Attributes 16-22
16.5.3 Copying or Subscribing to Application Items 16-22
16.5.4 Subscribing to an Application Item 16-23
16.5.5 Accessing Application Item Reports 16-24
16.5.6 About Creating an Application Item that Functions as a Global Variable 16-25
16.6 Managing Component Settings 16-25
16.6.1 Configuring Component Settings 16-26
16.6.2 Configuring Checkbox 16-27
16.6.3 Configuring Color Picker Display 16-28
16.6.4 Configuring Date Picker 16-28
16.6.5 Configuring Geocoded Address 16-30
16.6.6 Configuring Interactive Report Action Menu Structure 16-31
16.6.7 Configuring Vector Tile Layers for Maps 16-32
16.6.8 Configuring Oracle Cloud Applications (BOSS) REST Service 16-33
16.6.9 Configuring an Oracle Cloud Applications (SaaS) REST Service 16-33
16.6.10 Dynamically Adding Sliders to Regions 16-34
16.6.11 Configuring Select Many 16-35
16.6.12 Configuring Server Side Geocoding 16-35
16.6.13 Configuring Show AI Assistant 16-36
16.6.14 Configuring Star Rating 16-37
16.6.15 Configuring Switch Defaults 16-38
16.6.16 Viewing the Component Settings History Report 16-38
16.7 Creating Lists of Values at the Application-Level 16-39
16.7.1 About Lists of Values 16-40
16.7.2 Creating a Static List of Values at the Application-Level 16-40
16.7.3 Creating a Dynamic List of Values at the Application-Level 16-43
16.7.4 Editing an Existing List of Values 16-46
16.7.4.1 Editing a List of Values in Shared Components 16-46

16.7.4.2 Editing List of Values in Bulk 16-48
16.7.4.3 Subscribing to a List of Values 16-49
16.7.5 Copying or Subscribing to a List of Values 16-50
16.7.6 Subscribing to a List of Values 16-50
16.7.7 About Referencing Session State Within a List of Values 16-51
16.7.8 Accessing Lists of Values Reports 16-52
16.8 Creating Lists 16-52
16.8.1 About Static and Dynamic Lists 16-53
16.8.2 Creating Static Lists 16-54
16.8.2.1 About Creating a Static List 16-54
16.8.2.2 About Adding Entries and Sublists to a Static List 16-55
16.8.2.3 Creating a Static List from Scratch 16-55
16.8.2.4 Adding Entries or Sublists Manually 16-56
16.8.2.5 Copying Static List Entries Between Lists 16-58
16.8.2.6 Reparenting Static List Entries 16-59
16.8.2.7 Managing Orphaned Static List Entries 16-59
16.8.2.8 Resequencing Static List Entries 16-60
16.8.3 Creating Dynamic Lists 16-60
16.8.3.1 About Creating a Dynamic List 16-60
16.8.3.2 Understanding Dynamic List Syntax 16-61
16.8.3.3 Creating a Dynamic List from Scratch 16-64
16.8.4 Editing List Attributes 16-65
16.8.4.1 Editing a Static List 16-65
16.8.4.2 Editing a Dynamic List 16-67
16.8.5 Adding a List to a Page 16-68
16.8.6 Deleting a List 16-69
16.8.7 Copying or Subscribing to a List 16-70
16.8.8 Subscribing to a List 16-71
16.8.9 Accessing List Reports 16-72
16.8.10 Managing Navigation Menus 16-72
16.8.10.1 About Navigation Menus 16-73
16.8.10.2 About Editing Navigation Menu Lists 16-75
16.8.10.3 Changing Where and How Navigation Menus Display 16-75
16.8.10.4 About Switching from Tabs to Navigation Menus 16-76
16.8.11 Managing Navigation Bar Lists 16-77
16.8.11.1 About Navigation Bars 16-77
16.8.11.2 Editing Navigation Bar Lists 16-78
16.8.11.3 Editing Navigation Bar Attributes 16-78
16.9 Adding Search to an Application 16-79
16.9.1 Example Searches 16-80
16.9.2 About Creating Application Searches 16-81
16.9.3 Creating a Standard Search 16-82

16.9.3.1 Creating Standard Search Configurations 16-83
16.9.3.2 Editing Search Configurations to Include Links 16-86
16.9.3.3 Creating a Standard Search Page 16-88
16.9.4 Creating an Oracle Text Search 16-91
16.9.4.1 Creating an Oracle Text Index 16-91
16.9.4.2 Creating an Oracle TEXT Search Configuration 16-92
16.9.4.3 Creating an Oracle TEXT Search Page 16-94
16.9.5 Creating a List Search 16-96
16.9.5.1 Creating an APEX List 16-96
16.9.5.2 Creating a List Search Configuration 16-97
16.9.5.3 Creating a List Search Page 16-98
16.9.6 Creating a Ubiquitous Search 16-101
16.9.6.1 Creating a Ubiquitous Search Index 16-102
16.9.6.2 Creating a Ubiquitous Search Configuration 16-102
16.9.6.3 Creating a Ubiquitous Search Page 16-104
16.9.7 Creating an Oracle Vector Search 16-106
16.9.7.1 About Oracle Vector Search Configurations 16-107
16.9.7.2 Adding a Vector Column 16-107
16.9.7.3 Retrieving the Vector Embeddings from a Vector Provider 16-108
16.9.7.4 Creating an Oracle Vector Search Configuration 16-108
16.9.7.5 Creating an Oracle Vector Search Page 16-110
16.9.8 Creating a Search Page Manually 16-111
16.9.9 Editing a Search Configuration 16-114
16.9.10 Copying or Subscribing to a Search Configuration 16-116
16.9.11 Subscribing to a Search Configuration 16-117
16.9.12 Editing a Search Page 16-118
16.9.12.1 Reviewing Search Page Components 16-118
16.9.12.2 Editing Search Results Region Attributes 16-119
16.9.12.3 Editing Search Results Attributes 16-120
16.9.12.4 Editing Search Source Attributes 16-122
16.9.13 Accessing Search Configuration Reports 16-124
16.10 Using Shortcuts 16-124
16.10.1 What is a Shortcut? 16-124
16.10.2 Supported Shortcut Types 16-125
16.10.3 Defining a Shortcut from Scratch 16-126
16.10.4 Editing Existing Shortcuts 16-127
16.10.5 Copying an Existing Shortcut 16-127
16.10.6 Copying or Subscribing to a Shortcut 16-128
16.10.7 Subscribing to a Shortcut 16-129
16.10.8 Accessing Shortcut Reports 16-130
16.11 Using Component Groups 16-130
16.11.1 About Component Groups 16-131

16.11.2 Creating a Component Group 16-131
16.11.3 Copying and Subscribing to a Component Group 16-132
16.11.4 Refreshing a Subscribed Component Group 16-133
16.11.5 Publishing a Component Group 16-134
16.11.6 Unsubscribing from a Component Group 16-135
16.11.7 Deleting a Component Group 16-136
16.12 Managing Email Templates 16-136
16.12.1 Defining an Email Template 16-137
16.12.2 Viewing Sample Email Templates 16-139
16.12.3 About Including Dynamic Text 16-140
16.12.4 Editing or Deleting Email Templates 16-142
16.12.5 Copying or Subscribing to an Email Template 16-142
16.12.6 Subscribing to Email Templates 16-143
16.12.7 Accessing Email Template Reports 16-144
16.13 Managing Static Application Files 16-144
16.13.1 Specifying the Location of Static Application Files 16-145
16.13.2 Referencing Static Application Files 16-145
16.13.3 Managing Static Application Files in the Database 16-146
16.13.3.1 Adding Static Application Files to the Database 16-146
16.13.3.2 Viewing Static Application Files 16-147
16.13.3.3 Downloading Static Application Files 16-147
16.13.3.4 Deleting Static Application Files 16-148
16.13.4 Managing Static Application Files in Remote Storage 16-149
16.13.4.1 Uploading Files to Remote Storage 16-150
16.13.4.2 Viewing or Downloading Files in Remote Storage 16-152
16.13.4.3 Synchronizing Files in Remote Storage 16-153
16.14 Managing Static Workspace Files 16-154
16.14.1 About Referencing Static Workspace Files 16-155
16.14.2 Creating Static Workspace Files 16-155
16.14.3 Viewing the Static Workspace File Report 16-156
16.14.4 Downloading Static Workspace Files 16-156
16.14.4.1 Downloading a Static Workspace File 16-156
16.14.4.2 Downloading All Static Workspace Files in a Zip 16-157
16.14.5 Deleting Static Workspace Files 16-157
16.14.5.1 Deleting a Static Workspace File 16-157
16.14.5.2 Deleting Multiple or All Static Workspace Files 16-158
16.15 Printing a Report Region by Defining a Report Query 16-158
16.15.1 About Report Queries 16-158
16.15.2 Creating a Report Query 16-159
16.15.3 Editing a Report Query 16-161
16.15.4 Copying a Report Query 16-161
16.15.5 Viewing Report Query Utilization 16-162

16.15.6 Viewing Report Query History 16-162
16.16 Formatting a Report Region or Report Query Using Report Layouts 16-162
16.16.1 About Report Layouts 16-163
16.16.2 About Report Layout Options 16-163
16.16.3 Creating a Report Layout 16-164
16.16.4 Editing a Report Layout 16-165
16.16.5 Copying or Subscribing to a Report Layout 16-165
16.16.6 Subscribing to Email Templates 16-166

17 Upgrading Oracle APEX Applications

17.1 Reviewing Release Notes 17-1
17.2 Refreshing the Universal Theme 17-2
17.3 Updating App Compatibility Mode 17-3
17.4 Running Application Upgrade 17-4
17.5 Reviewing Old JavaScript 17-4

18 Managing Application Data

18.1 Creating Applications with Data Loading Capability 18-1
18.1.1 About Creating a Page with Data Loading Capability 18-2
18.1.2 Creating a Data Load Definition 18-3
18.1.3 Creating a Data Loading Page Using the Create Page Wizard 18-7
18.1.4 Editing a Data Load Definition 18-8
18.1.5 Copying or Subscribing to a Data Load Definition 18-10
18.1.6 Example: Log Errors to a DML Error Log 18-11
18.2 About Using Collections 18-13
18.3 Working with Duality Views and JSON Sources 18-14
18.3.1 Managing Duality Views 18-14
18.3.1.1 About Duality Views 18-14
18.3.1.2 Creating a Duality View Source 18-15
18.3.1.3 Editing and Deleting a Duality View Source 18-16
18.3.1.4 Copying or Subscribing to a Duality View 18-17
18.3.1.5 Viewing Duality Views Utilization 18-18
18.3.1.6 Viewing Duality Views History 18-18
18.3.2 Managing JSON Sources 18-19
18.3.2.1 About JSON Sources 18-19
18.3.2.2 Creating a JSON Source 18-20
18.3.2.3 Editing and Deleting a JSON Source 18-21
18.3.2.4 Copying or Subscribing to a JSON Source 18-22
18.3.2.5 Viewing JSON Sources Utilization 18-23
18.3.2.6 Viewing JSON Sources History 18-23

18.4 Managing REST Data Sources 18-24
18.4.1 Understanding REST Data Sources 18-24
18.4.1.1 About REST Data Source Types 18-25
18.4.1.2 Understanding Data Profiles 18-26
18.4.1.3 How REST Data Sources Differ from Legacy Web Service References 18-28
18.4.1.4 About Authentication and Credentials in REST Data Sources 18-28
18.4.2 Creating a REST Data Source 18-29
18.4.3 Example: Creating a REST Data Source on Top of a SQL Query 18-32
18.4.4 Editing or Deleting a REST Data Source 18-33
18.4.4.1 Editing a REST Data Source 18-34
18.4.4.2 Deleting a REST Data Source 18-37
18.4.5 Copying or Subscribing to a REST Data Source 18-38
18.4.6 Copying a REST Data Source 18-38
18.4.7 Working with REST Data Sources for Oracle Cloud SaaS Applications 18-39
18.4.7.1 About REST Data Source Support for Oracle Cloud SaaS Applications 18-39
18.4.7.2 Creating REST Data Source for Oracle Cloud SaaS Apps 18-40
18.4.7.3 REST Data Source Definitions for Oracle Cloud SaaS Apps 18-41
18.4.7.4 REST Data Source Runtime Features for Oracle Cloud SaaS Apps 18-42
18.4.7.5 Filter Parameters for Query By Example and Parent and Child Use Cases 18-44
18.4.7.6 Working Against a Sandbox 18-45
18.4.7.7 Best Practices for Configuring REST Service Base URL 18-46
18.4.7.8 Overriding Default Runtime Headers If Necessary 18-46
18.4.7.9 Data Profile Column Annotations 18-47
18.4.7.10 Example: Creating a REST Data Source Against a Fusion Applications Endpoint 18-47
18.4.8 Working with Oracle Cloud Applications BOSS REST Services 18-49
18.4.8.1 About Oracle Cloud Applications BOSS REST Services 18-50
18.4.8.2 Creating a BOSS REST Data Source 18-51
18.4.8.3 About the BOSS REST Data Source Definition 18-53
18.4.8.4 BOSS REST Data Source Runtime Features 18-55
18.4.8.5 Filter Parameters for Query By Example and Parent and Child Use Cases 18-58
18.4.8.6 Best Practices for Configuring BOSS REST Service Base URL 18-59
18.4.8.7 Working with Parent and Child Data in Regions 18-60
18.4.8.8 Using BOSS REST Data Sources in PL/SQL 18-61
18.4.8.9 Performing DML Operations on Objects with ETags 18-64
18.4.8.10 Reading Business Objects with Nested Collections 18-64
18.4.8.11 Writing Business Objects with Nested Collections 18-68
18.4.8.12 Invoking BOSS Business Object Custom Actions 18-69
18.4.9 Managing REST Source Catalogs 18-69
18.4.9.1 Creating a REST Source Catalog 18-70
18.4.9.2 Creating a REST Source Catalog from OpenAPI 18-71
18.4.9.3 Editing a REST Source Catalog 18-73

18.4.9.4 Managing Catalog Groups 18-74
18.4.9.5 About Exporting and Importing a REST Source Catalog 18-74
18.4.10 Managing Data Synchronization 18-75
18.4.10.1 About Data Synchronization 18-75
18.4.10.2 About Importing and Exporting REST Source Synchronizations 18-76
18.4.10.3 Configuring REST Synchronization 18-77
18.4.10.4 Viewing REST Synchronizations Defined in an App 18-79
18.4.10.5 Clearing Synchronization Settings 18-79
18.4.10.6 Viewing Synchronize Usage 18-80
18.4.10.7 Managing the REST Synchronization Log 18-80
18.4.11 Viewing REST Data Source Utilization 18-81
18.4.12 Viewing REST Data Sources History 18-81
18.5 Managing REST Enabled SQL References 18-81
18.5.1 Understanding REST Enabled SQL Service References 18-82
18.5.1.1 About REST Enabled SQL Service References 18-83
18.5.1.2 REST Enabled SQL Service Reference Requirements 18-83
18.5.1.3 About MySQL Support 18-83
18.5.1.4 Differences between REST Enabled SQL Service References and Database Links 18-84
18.5.1.5 Populating the Endpoint URL Dynamically 18-85
18.5.1.6 Exporting and Importing REST Enabled SQL Services 18-87
18.5.2 Before You Begin: REST Enabled SQL Service Requirements 18-87
18.5.3 Creating a REST Enabled SQL Service Reference 18-88
18.5.4 Example: Creating a REST Enabled SQL Service Reference on Oracle 18-90
18.5.5 Example: Creating REST Enabled SQL Service Reference on MySQL 18-92
18.5.6 Editing a REST Enabled SQL Service Reference 18-95
18.5.7 Deleting a REST Enabled SQL Service Reference 18-97
18.5.8 Viewing REST Enabled SQL Service Utilization 18-98
18.5.9 Viewing REST Enabled SQL Service History 18-98
18.5.10 Creating or Updating Components to Use a REST Enabled SQL 18-99
18.5.10.1 Creating a Report Using a REST Enabled SQL Reference 18-99
18.5.10.2 Creating a Chart Using a REST Enabled SQL Reference 18-101
18.5.10.3 Creating a Calendar Using a REST Enabled SQL Reference 18-102
18.5.10.4 Specifying a REST Enabled SQL Reference in an Existing Component 18-104
18.6 Managing Remote Servers 18-106
18.6.1 Understanding Remote Servers 18-106
18.6.1.1 About Remote Servers 18-107
18.6.1.2 How APEX Stores REST Data Source Information 18-107
18.6.1.3 About Flexible Remote Servers 18-107
18.6.1.4 Exporting and Importing Remote Server Information 18-109
18.6.2 Creating a Remote Server 18-110
18.6.3 Editing or Deleting a Remote Server 18-111

18.6.4 Viewing Remote Server Utilization 18-112
18.6.5 Viewing Remote Server History 18-113
18.7 Managing Web Credentials 18-113
18.7.1 Understanding Web Credentials 18-113
18.7.1.1 Protecting Web Credentials by Defining Valid URLs 18-114
18.7.1.2 About Supported Authentication Types in Web Credentials 18-114
18.7.1.3 Using a Database Credential with Web Credentials 18-115
18.7.1.4 Exporting and Importing Web Credentials 18-115
18.7.2 Creating Web Credentials 18-116
18.7.3 Editing or Deleting Credentials 18-119
18.7.4 Viewing Credential Utilization 18-119
18.7.5 Viewing Credential History 18-120
18.8 Managing Generative AI in APEX 18-120
18.8.1 Understanding Generative AI Support in APEX 18-121
18.8.2 Managing Generative AI Services 18-122
18.8.2.1 Creating a Generative AI Service Object 18-122
18.8.2.2 Editing or Deleting a Generative AI Service 18-126
18.8.2.3 Copying or Subscribing to an AI Configuration 18-127
18.8.2.4 Viewing Generative AI Service Utilization 18-128
18.8.2.5 Viewing Generative AI Service History 18-128
18.8.3 Using APEX Assistant 18-129
18.8.4 Including Generative AI in Applications 18-133
18.8.4.1 About Including Generative AI in Applications 18-133
18.8.4.2 Configuring AI Attributes for an Application 18-136
18.8.4.3 Managing AI Configurations and RAG Sources 18-136
18.8.4.4 Creating a Dynamic Action to Display AI Assistant 18-143
18.8.4.5 Creating a Dynamic Action to Generate Text with AI 18-146
18.8.4.6 Creating an AI Assistant to Suggest Salary 18-149
18.9 Accessing Data with Database Links 18-155
18.9.1 About Database Links 18-156
18.9.2 Creating a Database Link 18-156
18.9.3 Viewing an Existing Database Link 18-156

19 Extending Application Capabilities

19.1 Sending Email from an Application 19-1
19.1.1 About Configuring Email 19-1
19.1.1.1 About Enabling Network Services 19-2
19.1.1.2 About Configuring Email in a Full Development Environment 19-2
19.1.2 About Email Provisioning 19-3
19.1.3 About the Mail Queue 19-3
19.1.4 About the APEX_MAIL Package 19-3

19.2 Implementing Plug-ins 19-4
19.2.1 Understanding Plug-ins 19-4
19.2.1.1 About Plug-ins 19-5
19.2.1.2 About Template Component Type Plug-ins 19-5
19.2.1.3 Example Plug-ins 19-8
19.2.2 Creating Plug-ins 19-8
19.2.2.1 Creating a Plug-in from Scratch 19-8
19.2.2.2 Standard Attributes for Item Type Plug-ins 19-10
19.2.2.3 Copying a Plug-in 19-12
19.2.3 Editing Plug-ins 19-13
19.2.3.1 Editing a Plug-in 19-13
19.2.3.2 Subscribing to a Plug-in 19-14
19.2.3.3 Adding Custom Attributes to a Plug-in 19-15
19.2.3.4 Plug-in Attribute Types 19-17
19.2.3.5 Creating a File to Associate with a Plug-in 19-19
19.2.3.6 Automatically Loading CSS and JavaScript Files 19-20
19.2.3.7 Creating Attribute Groups 19-21
19.2.3.8 Adding Events to a Plug-in 19-22
19.2.4 Copying or Subscribing to a Plug-in 19-23
19.2.5 Deleting a Plug-in 19-24
19.2.6 Viewing the Plug-in Repository 19-24
19.2.7 Resetting the Plug-in Interactive Report 19-25
19.2.8 Viewing the Plug-in Utilization Page 19-25
19.2.9 Viewing the Plug-in History 19-25
19.3 Understanding jQuery Support 19-26
19.3.1 About Upgrading to jQuery 3.6.4 19-26
19.4 Manually Refreshing Oracle APEX Components 19-26
19.5 Managing Automations 19-26
19.5.1 About Automations 19-27
19.5.2 About Importing and Exporting Automations 19-28
19.5.3 Creating an Automation 19-29
19.5.4 Editing an Existing Automation 19-34
19.5.5 Managing the Automation Execution Log 19-37
19.5.5.1 Viewing the Automation Execution Log 19-37
19.5.5.2 Purging the Automation Execution Log 19-39
19.5.6 Viewing the Automation History Report 19-40

20 Managing Workflows and Tasks

20.1 About Workflows 20-2
20.1.1 About Workflow Versions 20-3
20.1.2 About Workflow Definitions 20-3

20.1.2.1 About Workflow Parameters 20-4
20.1.2.2 About Workflow Variables 20-4
20.1.2.3 About Workflow Activities 20-4
20.1.2.4 About Workflow Connections 20-5
20.1.2.5 About Workflow Participants 20-6
20.1.3 About Workflow Designer 20-6
20.1.3.1 The Workflow Designer Tree 20-7
20.1.3.2 The Workflow Diagram Toolbar 20-8
20.1.3.3 The Activities Palette 20-10
20.1.3.4 The Workflow Diagram Builder 20-11
20.1.4 About the Workflow Console, Workflow Details, and Workflow Dashboard 20-15
20.1.5 About the Workflow Runtime Engine 20-16
20.1.6 States and Transitions 20-17
20.1.6.1 Workflow States and Transitions 20-17
20.1.6.2 Activity States and Transitions 20-22
20.1.7 Example: Workflows 20-24
20.2 Creating a Workflow 20-27
20.2.1 Creating a Workflow Parameter 20-28
20.2.2 Creating a Workflow Variable 20-29
20.2.3 Creating a Workflow Activity 20-29
20.2.3.1 Adding a Workflow Start Activity 20-31
20.2.3.2 Adding a Workflow End Activity 20-32
20.2.3.3 Adding a Workflow Switch Activity 20-33
20.2.3.4 Adding a Workflow Wait Activity 20-37
20.2.3.5 Adding a Workflow Human Task Activity 20-38
20.2.3.6 Adding a Custom Process Type Plug-in as an Activity 20-39
20.2.3.7 Adding an Invoke Workflow Activity 20-41
20.2.3.8 Defining an Activity Variable 20-42
20.2.4 Adding a Workflow Connection 20-43
20.2.5 Adding a Workflow Participant 20-44
20.3 Copying a Workflow 20-44
20.4 Creating the Workflow Console, Workflow Details, and Workflow Dashboard Pages 20-45
20.5 Managing Workflow Versions 20-46
20.5.1 Creating a Workflow Version 20-46
20.5.2 Saving and Activating a Workflow Version 20-47
20.5.3 Editing a Workflow Version 20-47
20.5.4 Copying a Workflow Version 20-48
20.5.5 Deactivating a Workflow Version 20-48
20.5.6 Importing a New Application Version Containing Workflows 20-49
20.6 About Using Workflows in Page Processes 20-49
20.6.1 Starting a Workflow Instance 20-49
20.6.2 Terminating a Workflow Instance 20-50

20.6.3 Suspending a Workflow Instance 20-51
20.6.4 Resuming a Workflow Instance 20-51
20.6.4.1 Resuming Suspended Workflows at Any Activity 20-52
20.6.5 Retrying a Workflow Instance 20-52
20.7 Viewing the Workflow Instance ID in Debug Messages 20-53
20.8 About Workflow Reports 20-53
20.8.1 Viewing the Workflow Utilization Report 20-53
20.8.2 Viewing the Workflow History Report 20-54
20.9 Workflow Instance Retention 20-54
20.10 Workflow Substitution Strings 20-55
20.11 Workflow Views 20-55
20.11.1 Workflow Metadata Views 20-56
20.11.2 Workflow Runtime Views 20-56
20.12 About Workflow Keyboard Shortcuts 20-57
20.13 Frequently Asked Questions 20-58
20.14 About Tasks 20-66
20.14.1 About Task States and Transitions 20-67
20.14.2 Example: Approval Tasks and Action Tasks 20-72
20.15 Managing Tasks 20-76
20.15.1 Managing Task Definitions 20-76
20.15.1.1 About Task Definitions 20-77
20.15.1.2 Creating a Task Definition 20-78
20.15.1.3 Copying a Task Definition 20-82
20.15.1.4 Editing a Task Definition 20-83
20.15.1.5 Viewing the Task Definition Utilization Report 20-87
20.15.1.6 Viewing the Task Definition History Report 20-88
20.15.2 Managing Task Details 20-88
20.15.2.1 About the Task Details Page 20-88
20.15.2.2 Editing the Task Details Page 20-89
20.15.2.3 Configuring the Task History 20-89
20.15.3 Managing Unified Task Lists 20-90
20.15.3.1 About Unified Task Lists 20-90
20.15.3.2 Creating Unified Task Lists 20-90
20.15.3.3 Editing Unified Task Lists 20-91
20.16 Creating Tasks from Task Definitions 20-92
20.16.1 Example: Creating a Human Task from a Form Submission 20-93
20.17 About Creating Human Tasks Programmatically 20-94
20.18 About Creating a Task Definition Process Plug-in 20-94
20.19 About Task Retention and Task Purging 20-94
20.19.1 Retrieving Tasks Before Purge 20-95
20.20 Substitution Strings and Bind Variables for Tasks 20-95
20.21 Runtime Views for Tasks 20-96

20.22 Troubleshooting Tasks 20-97

21 Managing Application Security

21.1 Understanding Administrator Security Best Practices 21-1
21.1.1 About Oracle APEX Administrator Roles 21-2
21.1.2 Web Server Security Considerations 21-3
21.1.2.1 About Configuring Oracle REST Data Services with Oracle APEX 21-3
21.1.3 Managing Instance Security 21-3
21.1.3.1 About Creating Login Controls 21-4
21.1.3.2 About Enabling Public File Upload 21-5
21.1.3.3 About Restricting User Access by IP Address 21-5
21.1.3.4 About Specifying an Instance Proxy 21-5
21.1.3.5 About Utilizing Secure Sockets Layer 21-6
21.1.3.6 About Enabling RESTful Access 21-6
21.1.3.7 About Rejoin Sessions 21-6
21.1.3.8 About Isolating Workspaces 21-7
21.1.3.9 About Utilizing Session Timeout 21-8
21.1.3.10 Restricting Password Reuse 21-8
21.1.3.11 About Enabling RESTful Services 21-9
21.1.4 Configuring Instance Settings 21-10
21.1.4.1 About Enabling Automation When Creating Workspaces 21-10
21.1.4.2 About Enabling and Configuring Email 21-11
21.1.4.3 About Configuring Storage Options 21-11
21.1.4.4 About Creating Wallets 21-12
21.1.5 About Configuring Workspace Purge Settings 21-12
21.1.6 Understanding Workspace Management 21-12
21.1.6.1 About Managing Workspace to Schema Assignments 21-13
21.1.6.2 About Managing Developers and Users 21-13
21.1.6.3 About Managing Component Availability 21-13
21.1.7 About Integrating with Oracle BI Publisher 21-14
21.1.8 About the Advantages of the Oracle APEX Runtime Environment 21-14
21.1.9 Enabling Network Services in Oracle Database 21-15
21.2 Understanding Developer Security Best Practices 21-15
21.2.1 About Items of Type Password 21-16
21.2.2 Identifying At Risk Password Items 21-17
21.2.2.1 Viewing the Security Profiles Report 21-17
21.2.2.2 Viewing the Password Items Report 21-18
21.2.3 Understanding Cross-Site Scripting Protection 21-18
21.2.3.1 About Cross Site-scripting Security Breaches 21-18
21.2.3.2 Protecting HTML Regions and Other Static Areas 21-19
21.2.3.3 About Protecting Dynamic Output 21-20

21.2.3.4 About Protecting Report Regions 21-21
21.2.3.5 About Protecting Form Items 21-22
21.2.3.6 About Restricting Characters Entered on an Item 21-23
21.2.4 About Session State and Security 21-23
21.2.5 Preventing URL Tampering 21-24
21.2.5.1 How Session State Protection Works 21-24
21.2.5.2 Enabling Session State Protection 21-25
21.2.5.3 Configuring Session State Protection 21-25
21.2.6 About Securing File Uploads 21-31
21.3 Controlling Access to Applications, Pages, and Page Components 21-32
21.3.1 Understanding Access Control 21-33
21.3.1.1 About Adding Access Control 21-33
21.3.1.2 About Access Control Authorization Schemes 21-34
21.3.1.3 About Configuring Access Control 21-35
21.3.1.4 About Exporting an Application with Access Control 21-36
21.3.2 Creating an Access Control Using the Create Page Wizard 21-37
21.3.3 Managing User Access 21-38
21.3.4 Configuring Access Control 21-39
21.3.5 About Controlling Access for Pages and Page Components 21-39
21.3.6 About Removing an Access Control Created with a Wizard 21-40
21.3.7 Managing Roles and User Assignments 21-40
21.3.7.1 About Application Access Control 21-41
21.3.7.2 Creating Access Control Roles 21-42
21.3.7.3 Copying or Subscribing to Access Control Roles 21-42
21.3.7.4 Subscribing to Access Control Roles 21-43
21.3.7.5 Deleting Access Control Roles 21-44
21.3.7.6 Adding User Role Assignments 21-44
21.3.7.7 Editing User Role Assignments 21-45
21.4 Establishing User Identity Through Authentication 21-46
21.4.1 Understanding Authentication 21-46
21.4.1.1 How Authentication Works 21-47
21.4.1.2 About Support for Deep Linking 21-47
21.4.1.3 About Determining Whether to Include Authentication 21-48
21.4.2 Understanding Preconfigured Authentication Schemes 21-48
21.4.2.7 Open Door Credentials 21-49
21.4.2.1 Builder Extension Sign-in 21-50
21.4.2.2 Custom Authentication 21-51
21.4.2.3 Database Accounts 21-53
21.4.2.4 HTTP Header Variable 21-54
21.4.2.5 LDAP Directory 21-55
21.4.2.6 No Authentication (using DAD) 21-57
21.4.2.8 Oracle APEX Accounts 21-58

21.4.2.9 Oracle Application Server Single Sign-On Server 21-59
21.4.2.10 SAML Sign-In 21-61
21.4.2.11 Social Sign-In 21-62
21.4.3 Viewing the Authentication Scheme Associated with an Application 21-65
21.4.4 Creating an Authentication Scheme 21-65
21.4.5 Configuring an Existing Authentication Scheme 21-66
21.4.6 Changing the Authentication Scheme Associated with an Application 21-67
21.4.7 Copying or Subscribing to an Authentication Scheme 21-67
21.4.8 Subscribing to an Authentication Scheme 21-68
21.4.9 Creating a Login Page 21-69
21.4.10 Using a Procedure to Configure Authentication at Runtime 21-69
21.4.11 Viewing Authentication Scheme Reports 21-71
21.5 Providing Security Through Authorization 21-71
21.5.1 How Authorization Schemes Work 21-72
21.5.2 About Authorization Scheme Types 21-72
21.5.3 Creating an Authorization Scheme 21-73
21.5.4 Attaching an Authorization Scheme to an Application, Page, or Components 21-73
21.5.4.1 Attaching an Authorization Scheme to an Application 21-74
21.5.4.2 Attaching an Authorization Scheme to a Page 21-74
21.5.4.3 Attaching an Authorization Scheme to a Control or Component 21-75
21.5.5 Editing an Authorization Scheme 21-75
21.5.5.1 Editing Attributes of an Existing Authorization Scheme 21-76
21.5.5.2 Changing the Evaluation Point Attribute 21-76
21.5.5.3 Subscribing to an Authorization Scheme 21-77
21.5.5.4 About Resetting Authorization Scheme State 21-78
21.5.6 Copying or Subscribing to an Authorization Scheme 21-78
21.5.7 Viewing Authorization Reports 21-79

22 Managing Application Globalization

22.1 Understanding Application Translation and Globalization Support 22-2
22.1.1 About the Translation Process 22-2
22.1.2 About Language Identification 22-2
22.1.3 About the Rules for Translating Applications 22-3
22.1.4 How Translated Applications Are Rendered 22-3
22.1.5 Making Application Attributes Translatable 22-4
22.1.5.1 About Shortcuts that Support Translatable Messages 22-4
22.1.5.2 About Messages 22-4
22.1.5.3 About Dynamic Translation Text Strings 22-5
22.1.5.4 Marking a Region as Not Translatable 22-5
22.1.5.5 Identifying a Template as Translatable 22-5
22.2 Specifying the Primary Language for an Application 22-6

22.3 Applying Format Masks to Items 22-7
22.4 About Translating Applications for Multibyte Languages 22-7
22.5 Understanding the Translation Process 22-7
22.5.1 Step 1: Define Application Languages 22-8
22.5.2 Step 2: Seed and Download to a Translation File 22-9
22.5.2.1 Seeding Translatable Text 22-9
22.5.2.2 Downloading an XLIFF File 22-10
22.5.3 Step 3: Translate the XLIFF File 22-12
22.5.4 Step 4: Upload and Apply a Translated XLIFF Document 22-12
22.5.4.1 Uploading a Translated XLIFF Document 22-12
22.5.4.2 Applying an Uploaded XLIFF Document 22-13
22.5.4.3 Deleting an Uploaded XLIFF Document 22-14
22.5.5 Step 5: Publish the Application 22-14
22.5.6 Manually Editing a Translation 22-15
22.6 Translating Default Reports for Interactive Reports and Interactive Grids 22-16
22.7 Translating Messages 22-17
22.7.1 About Translating Messages 22-17
22.7.2 About Translating Messages Used in PL/SQL Procedures 22-18
22.7.3 Translating Messages Used Internally by APEX 22-18
22.7.3.1 Internal Messages Requiring Translation 22-19
22.7.3.2 Interactive Grid Messages Requiring Translation 22-23
22.7.3.3 Interactive Report Messages Requiring Translation 22-40
22.7.3.4 Progressive Web App (PWA) Messages Requiring Translation 22-57
22.7.3.5 Search and Smart Filters Messages Requiring Translation 22-65
22.7.4 Managing Text Messages 22-66
22.7.4.1 Workflow for Translating Messages into Multiple Languages 22-66
22.7.4.2 Creating Text Messages 22-67
22.7.4.3 Editing or Deleting Text Messages 22-68
22.7.4.4 Selecting Translation Messages in Page Designer 22-69
22.7.4.5 Copying or Subscribing to Text Messages 22-71
22.7.4.6 Subscribing to Text Messages 22-71
22.8 Translating Data That Supports List of Values 22-72
22.8.1 Defining a Dynamic Translation 22-73
22.9 Understanding Supported Globalization Codes 22-73
22.9.1 Oracle APEX Globalization Codes 22-73

23 Managing Application Performance

23.1 About Best Practices to Improve Performance 23-1
23.2 Identifying Performance Issues 23-2
23.2.1 About Troubleshooting Slow-Running Applications 23-2
23.2.2 Diagnosing Performance Issues 23-3

23.2.3 About Debugging Problematic SQL Queries 23-3
23.2.4 About Addressing Slow Queries 23-3
23.3 Utilizing Logs and Reports 23-4
23.3.1 Accessing the Monitor Activity Page in Workspace Administration 23-4
23.3.2 About Managing Log Files 23-5
23.3.3 Viewing the Dashboard Report 23-5
23.3.4 Instance Activity Reports 23-6
23.4 About Utilizing Database Reporting 23-6
23.5 About Database Parameters that Impact Performance 23-7
23.6 About Limiting Resources 23-7
23.7 About Uploading Static Files to Your Web Server 23-8
23.8 Creating Custom Activity Reports Using APEX_ACTIVITY_LOG 23-8
23.8.1 Enabling Logging for an Application 23-8
23.8.2 APEX_ACTIVITY_LOG 23-9
23.8.3 APEX_ACTIVITY_LOG Sample Query 23-10

24 Debugging an Application

24.1 Reviewing Session State 24-1
24.2 Utilizing Debug Mode 24-2
24.2.1 Enabling and Disabling Debug Mode for an Application 24-2
24.2.2 Available Debug Levels 24-3
24.2.3 Controlling Debug Mode for a Page at Runtime 24-4
24.2.4 Configuring Debug Mode Using the Session Detail Report 24-4
24.2.5 Configuring the Debug Level Programatically 24-5
24.2.6 Using URL Parameters to Control Debug Mode 24-5
24.2.6.1 Using Friendly URL Syntax to Access Debug Mode 24-5
24.2.6.2 Using f?p Syntax to Access Debug Mode 24-5
24.2.7 Controlling the Debug Level for an Instance 24-6
24.2.8 Viewing Debug Messages 24-6
24.2.8.1 Viewing Debug Messages for an Application 24-7
24.2.8.2 Viewing Debug Messages from Utilities 24-7
24.2.8.3 Querying Debug Messages 24-8
24.2.8.4 Available Debug Utility Scripts 24-9
24.2.8.5 Debugging an APEX Session from the User Interface 24-11
24.2.8.6 Tips for Sharing Debug Output 24-13
24.2.9 Using APEX_DEBUG When Developing Your Application 24-14
24.2.9.1 Programmatically Enabling or Disabling Debugging for a Call 24-14
24.2.9.2 Programmatically Enabling or Disabling Debugging for a Specific Session 24-14
24.2.9.3 Logging Messages 24-15
24.3 Removing Controls and Components to Isolate a Problem 24-15
24.3.1 Excluding and Including Page Components 24-16

24.3.2 Using Server-side Conditions 24-16
24.4 About Monitoring Application and Page Resource Use 24-17
24.5 Enabling SQL Tracing and Using TKPROF 24-17
24.6 Running Advisor to Check Application Integrity 24-18
24.6.1 About APEX Advisor 24-18
24.6.2 Running Advisor on an Entire Application 24-18
24.6.3 Running Advisor on a Single Page 24-19

25 Deploying an Application

25.1 System Development Life Cycle Methodologies to Consider 25-1
25.1.1 Differences Between Iterative and Planned Development 25-2
25.1.2 About the Advantages of Creating Prototypes 25-2
25.1.3 Waterfall 25-2
25.1.4 Spiral 25-3
25.1.5 Rapid Application Development 25-3
25.2 Understanding the Deployment Process 25-3
25.2.1 About Suggested Environments 25-3
25.2.2 Deployment Scenarios to Consider 25-4
25.3 Exporting and Importing from App Builder 25-5
25.3.1 Understanding the Export and Import Process 25-5
25.3.2 Exporting Workspaces, Applications, and App Components 25-6
25.3.2.1 Splitting Export Files into Separate Scripts 25-7
25.3.2.2 About Exporting Static Files 25-9
25.3.2.3 Exporting a Workspace 25-9
25.3.2.4 Exporting an Application 25-10
25.3.2.5 Exporting an Application Page 25-13
25.3.2.6 Exporting Application Components 25-17
25.3.2.7 Exporting Themes 25-19
25.3.2.8 Exporting Plug-ins 25-19
25.3.2.9 Exporting UI Defaults 25-20
25.3.2.10 Exporting Team Development Feedback 25-22
25.3.2.11 Exporting a REST Source Catalog 25-23
25.3.2.12 Exporting Script Files from the Script Repository 25-24
25.3.3 Importing Export Files 25-24
25.3.3.1 Importing an Application Export 25-25
25.3.3.2 Importing a Page 25-27
25.3.3.3 Importing a Component 25-28
25.3.3.4 Reimporting an Application Containing Background Processes 25-29
25.3.3.5 About Importing Applications and Application Groups 25-31
25.3.3.6 Importing Themes 25-31
25.3.3.7 Importing Plug-ins 25-32

25.3.3.8 Importing UI Defaults 25-33
25.3.3.9 Importing Team Development Feedback 25-33
25.3.3.10 Importing a REST Source Catalog 25-34
25.3.4 Managing Export Files from the Export Repository 25-35
25.3.4.1 About Installing Export Files 25-35
25.3.4.2 Installing an Application Export from the Export Repository 25-36
25.3.4.3 Installing Other Files from the Export Repository 25-37
25.3.4.4 Deleting Files from the Export Repository 25-37
25.4 How to Create a Custom Application 25-38
25.4.1 How Creating a Custom Application Simplifies Deployment 25-38
25.4.2 Creating a Custom Application 25-39
25.4.2.1 Accessing the Supporting Objects Utility 25-39
25.4.2.2 Supporting Objects Page 25-40
25.4.3 Installing Supporting Objects 25-43
25.4.4 Deleting Supporting Objects Scripts, Messages, and Installation Options 25-43
25.4.5 Upgrading a Custom Application 25-43
25.4.5.1 Defining an Upgrade Script 25-44
25.4.5.2 Upgrading a Custom Application 25-44
25.4.6 Deinstalling Supporting Objects 25-45
25.4.7 Viewing an Install Summary 25-46
25.5 About Publishing an Application URL 25-46
25.6 Using Build Options to Control Configuration 25-47
25.6.4 Copying or Subscribing to Build Options 25-48
25.6.5 Subscribing to Build Options 25-49
25.6.1 About Build Options 25-50
25.6.2 Creating and Editing Build Options 25-51
25.6.3 Including or Excluding Build Options 25-52
25.6.6 Selecting a Build Option 25-52
25.6.7 Deleting Build Options and Associated Components 25-53
25.6.7.1 Removing Components Associated with a Build Options 25-53
25.6.7.2 Deleting Build Options Permanently 25-54
25.6.8 Viewing Build Option Reports 25-54
25.6.8.1 Viewing Build Option Utilization 25-54
25.6.8.2 Viewing Build Option History 25-55
25.6.9 Exporting Build Options or Build Option Status 25-55
25.7 Understanding One-Click Remote Application Deployment 25-57
25.7.1 About One-click Remote Application Deployment 25-57
25.7.2 Remotely Deploying an Application 25-57

26 Managing Application Legacy Components

26.1 Legacy Forms Optimized for Mobile Environments 26-1
26.1.1 Mobile Friendly Legacy Forms 26-2
26.1.2 Supported Columns in Legacy Column Toggle and Reflow Reports 26-2
26.1.3 Creating a Legacy Column Toggle Report 26-3
26.1.4 Creating a Legacy List View Report 26-4
26.1.5 Creating a Legacy Reflow Report 26-5
26.2 Viewing Legacy Components in the Gallery 26-7
26.3 Managing Legacy Calendars 26-7
26.3.1 Editing Legacy Calendar Attributes 26-8
26.3.2 About Supported Legacy Calendar Substitution Strings 26-8
26.4 Managing Applications with Legacy Data Loading Capability 26-9
26.4.1 Comparing Legacy Data Loading and New Data Loading 26-9
26.4.2 About Creating a Legacy Data Load Wizard 26-10
26.4.3 Creating a Legacy Data Load Page 26-11
26.4.4 Re-creating Legacy Data Load Pages 26-13
26.4.5 Editing a Legacy Data Load Definition 26-14
26.5 Managing Legacy Tabular Forms 26-16
26.5.1 Best Practices When Managing Tabular Forms 26-16
26.5.2 How Legacy Tabular Forms Work 26-16
26.5.3 About Running a Legacy Tabular Form 26-17
26.6 Managing Legacy Web Services 26-19
26.6.1 About Legacy Web Services 26-19
26.6.2 Creating Legacy Web Service References 26-20
26.6.2.1 Accessing the Web Service References Page 26-21
26.6.2.2 Specifying an Application Proxy Server Address 26-21
26.6.3 About Working with SSL Enabled Legacy Web Services 26-21
26.6.4 Creating Legacy Web Service References Based on a WSDL 26-22
26.6.4.1 About Creating Web Service References Based on a WSDL 26-22
26.6.4.2 Creating a Web Service Reference by Specifying a WSDL Document 26-22
26.6.4.3 Testing a Web Service Reference Created from a WSDL 26-23
26.6.5 Creating Web Service References Manually 26-23
26.6.5.1 About Creating Web Service References Manually 26-24
26.6.5.2 Creating a Web Service Reference Manually 26-24
26.6.5.3 Testing a Web Service Reference Created Manually 26-25
26.6.6 Creating Legacy RESTful Web Service References 26-25
26.6.6.1 About Creating RESTful Web Service References 26-26
26.6.6.2 Creating a RESTful Web Service Reference 26-26
26.6.6.3 Testing a REST Web Service Reference 26-27
26.6.7 Using the Legacy Web Service Reference Repository 26-28
26.6.8 Creating an Input Form and Report on a Legacy Web Service 26-29

26.6.8.1 About the Create Form and Report on Web Service Wizard 26-29
26.6.8.2 Creating a Form and Report by Adding a New Page 26-29
26.6.9 Creating a Form on a Legacy Web Service 26-30
26.6.9.1 About the Create Form on Web Service Wizard 26-31
26.6.9.2 Creating a Form by Adding a New Page 26-31
26.6.10 Creating a Report on a Web Service 26-32
26.6.11 How to Invoke a Legacy Web Service as a Process 26-34
26.6.11.1 Invoking a Web Service as a Process 26-34
26.6.11.2 Editing a Web Service Process 26-36
26.6.12 Viewing a Legacy Web Service Reference History 26-37

A About Item Types

A.1 Checkbox A-3
A.2 Checkbox Group A-3
A.3 Color Picker A-4
A.4 Combobox A-5
A.5 Date Picker A-7
A.6 Date Picker (jQuery) (Desupported) A-9
A.7 Display Image A-10
A.8 Display Map A-11
A.9 Display Only A-12
A.10 File Upload A-12
A.11 Geocoded Address A-14
A.12 Image Upload A-16
A.13 Hidden A-19
A.14 Markdown Editor A-20
A.15 List Manager A-20
A.16 Number Field A-21
A.17 Password A-22
A.18 Percent Graph A-23
A.19 Popup LOV A-23
A.20 QR Code A-24
A.21 Radio Group A-25
A.22 Rich Text Editor A-25
A.23 Select List A-26
A.24 Select Many A-27
A.25 Select One A-29
A.26 Shuttle A-30
A.27 Textarea A-31
A.28 Star Rating A-31
A.29 Switch A-32

A.30 Star Rating A-33
A.31 Text Field A-33
A.32 Text Field with Autocomplete A-34
A.33 Textarea A-34

B Oracle APEX Limits

B.1 Component Limits B-1

C Directory Structure When Splitting Export Files into Separate Scripts

C.1 Export Directory Structure C-1

D Correlating APEX Sessions to Database Sessions

D.1 About APEX Sessions and Database Sessions D-1
D.2 Session Attributes D-2
D.3 APEX Processes D-3
D.4 SQL Workshop D-4
D.5 Data Workshop D-6
D.6 Automations D-7
D.7 Installer D-8

E Available Conditions

E.1 What Are Conditions? E-1
E.2 Conditions Available in App Builder E-1

F About Granted Privileges

F.1 About Granting Privileges to Database Users F-1
F.2 Understanding Privileges Granted to PUBLIC F-1
F.2.1 Packages F-2
F.2.2 Procedures F-4
F.2.3 Functions F-4
F.2.4 Tables F-5
F.2.5 Views F-5
F.2.6 Sequences F-9
F.2.7 Types F-10

Index

Preface

Oracle APEX App Builder User's Guide describes how to use the Oracle APEX development environment to build and deploy database-centric web applications. Oracle APEX turns a single Oracle database into a shared service by enabling multiple workgroups to build and access applications as if they were running in separate databases.

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Documents
- Conventions

Audience

Oracle APEX App Builder User's Guide is intended for application developers who are building database-centric web applications using Oracle APEX. The guide describes how to use the Oracle APEX development environment to build, debug, manage, and deploy applications.

To use this guide, you must have a general understanding of relational database concepts and an understanding of the operating system environment under which you run Oracle APEX.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, see these Oracle resources:

- *Oracle APEX Release Notes*
- *Oracle APEX Installation Guide*
- *Oracle APEX End User's Guide*
- *Oracle APEX Administration Guide*
- *Oracle APEX SQL Workshop Guide*
- *Oracle APEX API Reference*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- Working with SQLcl in *Oracle SQLcl User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Changes in Release 24.2 for *Oracle APEX App Builder User's Guide*

All content in *Oracle APEX App Builder User's Guide* has been updated to reflect release 24.2 functionality and user interface changes.

New Features and Updates

The following features have been added or updated for this release:

- JSON Sources

  JSON Sources in APEX consist of information about the owner and name of a database table. You can use a plain table with columns containing JSON data or a JSON collection table. After picking the table and column to use, APEX either determines the data profile from the actual data or uses an uploaded JSON Schema file.

  Plain tables with JSON columns are supported in Oracle Database 19c and 21c. JSON collection tables are only available in Oracle Database 23ai or higher.

  See Managing JSON Sources.

- JSON Duality Views

  Developers can now create Duality View objects to retrieve data sourced from a Duality View in a local or a remote database. Duality views are available to page components, including reports, charts, and forms, and shared components, including lists of values and automations.

  Duality Views are only available in Oracle Database 23ai or later.

  See Managing Duality Views.

- REST Source Enhancements
  - Support for Token-Based Pagination in Simple HTTP REST Data Sources

    The **Simple HTTP** REST Data Source type now supports REST services with token-based pagination, enhancing integration with APIs that use this pagination scheme.

    See Creating a REST Data Source and Editing a REST Data Source.

  - Flexible Remote Servers

    When editing a remote server, you can now use placeholders (for example, `#tenant#`) within the endpoint URL as well as within other attributes. At runtime, APEX can derive the placeholder values as well as the Base URL of a Remote Server object (used for REST Data Sources, REST Enabled SQL, and Authentication) dynamically with a PL/SQL **Configuration Procedure**. This allows APEX to invoke the endpoints dynamically based on the current application, tenant, or environment settings.

    See About Flexible Remote Servers.

- Workflow Enhancements:
  - New Invoke Workflow activity enables you to invoke a workflow from a parent workflow.

  - New Copy Workflow functionality enables you to copy a workflow from another application.
  - You can now resume a suspended workflow at any activity.
  - Workflow parameters and workflow variables now support CLOB data types.

  See Managing Workflows and Tasks.

- Generative AI
  - Dynamic Action True Actions Changes:
    * New Dynamic Action, True Action, Generate Text With AI - Use the **Generate Text With AI** True Action to call the configured Generative AI Service and generate a one-time response based on user content.
    * Dynamic Action, True Action Renamed **Show AI Assistant** - The Dynamic Action, True Action **Open AI Assistant** which is used to display the AI-powered chat service, AI Assistant, has been renamed. In release 24,2, this True Action, is now named **Show AI Assistant**.

    See Including Generative AI in Applications.
  - AI Configurations and RAG Sources

    An AI Configuration is a shared component that centralizes key Generative AI settings, including the System Prompt, Welcome Message, and Retrieval-Augmented Generation (RAG) sources. Creating an AI Configuration enables you to reuse these settings across multiple AI-enabled components such as dynamic actions that display AI Assistant or generate a text response from a Generative AI Service.

    See Managing AI Configurations and RAG Sources.
  - AI Test Connection Button

    When you create or edit a Generative AI Service, a new **Test Connection** button enables you to validate the information you enter prior to completing the setup.

    See Creating a Generative AI Service Object and Editing or Deleting a Generative AI Service.

- BOSS REST Service Support

  This release adds support for APEX REST Data Sources for Oracle Cloud Applications (BOSS) REST Services.

  See Working with Oracle Cloud Applications BOSS REST Services.

- Fusion Integrated Starter App Wizard

  New option in the Create App wizard enables your to create a new application pre-integrated with a Fusion Applications instance.

  See Integrating APEX with Fusion Applications.

- Document Generator Enhancements

  Document Generator now supports DOCX to DOCX, XLSX to PDF, and XLSX to XLSX.

  See Viewing Database Object Dependencies.

- Database Object Dependencies Report

  Scan an entire application or just one page to generate a report that details database object dependencies. Database Object Dependencies page features a hierarchical tree control to visualize all the tables, views, packages as well as their components.

  See Viewing Database Object Dependencies.

- Control Breaks for Template Components

  APEX now supports **control breaks** (grouping) for Template Component Reports. .

  See Template Components.

- Unlimited Attributes for Item Plug-ins

  In this release the custom attribute limit for has been extended to Item plug-ins. When your upgrade to APEX 24.2, custom attributes are migrated to include a new Static ID attribute.

  See Adding Custom Attributes to a Plug-in

- List Enhancements

  To edit static list, you now select the list Name. In addition, developers can edit list entries on the Grid Edit List Entries report. Simply navigate to the List Details page by selecting Grid Edit from the Tasks list.

  See Editing a Static List and Accessing List Reports.

- Export App Enhancements

  When you export a translated application, APEX checks the application to ensure that the translations are in sync. If the translations are out of sync, a warning message displays on the Application Export Page.

  See Translating Messages.

- Default RTL Direction Based on Language

  The Document Direction attribute on the Application Globalization page and Translation Language Mappings page now defaults to Language Default instead of left-to-right. When an application set to Language Default runs, the direction is now derived from the language.

  See Document Direction.

- Text Messages Enhancements
  - Comments and Metadata

    The Create/Edit Text Message dialog now includes **Comments** and **Metadata**.

    See Managing Text Messages.

  - Display Text Message Values

    In Page Designer, the new Text Messages Picker (Page Designer, Utilities, Show) controls the visibility of the Text Messages dialog buttons (globe icon). If enabled, this button opens a Text Messages dialog that lists all available Text Messages in the current application.

    See Managing Text Messages.

  - Extended Substitution Syntax for Text Messages

    For applications with a Compatibility Mode 24.2 or later, you can now reference text messages by enclosing the message name in curly braces ({}).

    See Substitutions in Text Messages (24.2 or later).

- Application User Experience Data (OpenTelemetry)

  OpenTelemetry is a standard for collecting user experience data in web applications, such as page load times and user interactions. Enabling OpenTelemetry makes it easier to monitor and analyze how users engage with an application. To configure OpenTelemetry, first set the Client Logging Service URL and the Token Relay URL in Workspace Utilities, OpenTelemetry.

See OpenTelemetry.

- Improved Popup LOV Functionality

  The appearance and behavior of Popup LOVs is now similar to searchable LOV items such as Select One, Select Many, and Combobox. This creates a unified and intuitive user experience when searching.

- Themes and Theme Styles

  – Theme Metadata Decoupled from Application Metadata

    Starting with Universal Theme 24.2, theme metadata (templates, theme styles, template components, and so on.) is now stored in a central workspace, decoupled from applications. This improves performance with faster application export/import, smaller file sizes, and optimized storage in version control systems.

    Related updates include:

    * Refresh Theme alert now displays on the Application home page. Previously, this alert displayed on the Shared Components page.

      Refreshing the Universal Theme.

    * Outdated APEX Builder pages related to theme management have been updated or removed, simplifying the interface.

    * Active theme style selection has moved from the theme style level to the theme level.

      See Using Theme Styles.

    See Using Themes and Theme Styles

- Select Many Improvements

  If you set **Display Values As** to Comma-separated List. there is now a counter icon on the item that displays the current number of selected values. Clicking on the counter opens the dropdown, displaying only the selected values.

- Universal Theme Enhancements

  – Font APEX 2.4

    Font APEX now includes 244 country flag icons to easily display flags within your APEX application. Display the new icons with `.fa-flag-##`, where `##` is the country code..

    See Accessing Font APEX.

  – Password Item Enhancements

    Password type page items now include a **Show Password** button. Users can select Show Password to reveal the entered password, and select Hide Password to conceal the password again. This button is enabled by default for all application using Universal Theme 24.2 and later. To disable this button, edit the template options for the **Password** item, and select **Hide Password Visibility**.

- The Page Designer breadcrumb now includes the page number and name.
- Workspace Utilities now includes a link in the Tasks region to Data Dictionary Cache.

Other Changes

- *Forms Optimized for Mobile Environments* has been renamed to Legacy Forms Optimized for Mobile Environments and moved to Managing Application Legacy Components.

Deprecated and Desupported Features

See Deprecated Features and Desupported Features in *Oracle APEX Release Notes*.

2
Quick Start

"Quick Start" offers a quick introduction to Oracle APEX. This section explains how to access your development environment, describes the Workspace home page, the App Builder home page, and the Application home page, explains how to edit your account profile and user preferences, and how to access product documentation.

> **Tip:**
>
> If you are accessing an on-premises installation, complete the installation process before continuing. See *Oracle APEX Installation Guide*.

- About Accessing Your Development Environment
  How you sign in and access Oracle APEX depends upon your user role and where APEX resides.
- Understanding Oracle APEX
  Oracle APEX provides you with all the tools you need to build apps in a single, extensible platform, which runs as a part of Oracle Database.
- Understanding the Workspace Home Page
  When you sign in to Oracle APEX, the Workspace home page appears.
- APEX Menu Bar
  Learn about the APEX menu bar.
- Understanding App Builder
  Developers use App Builder to create and manage applications and application pages.
- Changing Your Profile or Password
  Edit your account profile to update your profile (email address, first or last name), upload a profile photo, or change your password.
- Editing User Preferences and Enabling Dark Mode
  Click the Account menu to enable or disable Dark Mode or edit user preferences.
- Using Oracle APEX Documentation
  Oracle APEX provides field-level help, a hosted online documentation library, and links to the documentation from the product. From the APEX Documentation page in the Oracle Help Center, you can browse content by publication, search content, or download the entire documentation library.
- Using the App Gallery
  Use the App Gallery to install Sample Apps, Starter Apps, Custom Apps, and Utility Apps.

2.1 About Accessing Your Development Environment

How you sign in and access Oracle APEX depends upon your user role and where APEX resides.

Oracle APEX may reside in a local self-managed Oracle database or in a hosted environment (such as an Oracle Cloud service). The sign in process differs depending on where APEX resides.

Accessing Oracle APEX in a Local Self-managed Oracle Database

If you are running APEX in a local self-managed Oracle database, sign in to your APEX workspace as described in Signing In to Your Workspace in *Oracle APEX Installation Guide*

Accessing Oracle APEX in Oracle Cloud

In Oracle Cloud, APEX is installed and enabled in:

- Oracle APEX Application Development (APEX Service)
- Oracle Autonomous Transaction Processing
- Oracle Autonomous Data Warehouse

APEX is available in Exadata Cloud Service and Database Cloud Service in Oracle Cloud Infrastructure. However, you need to manually customize your databases to install and enable APEX by following on-premises installation process or using cloud tooling such as Terraform.

See Also:

- Get an Environment
- Welcome to Oracle APEX Application Development Service in *Getting Started with Oracle APEX Application Development*
- Creating Applications with Oracle Application Express on Autonomous Database in *Using Oracle Autonomous Database Serverless*

2.2 Understanding Oracle APEX

Oracle APEX provides you with all the tools you need to build apps in a single, extensible platform, which runs as a part of Oracle Database.

- What is Oracle APEX?
  Oracle APEX is an enterprise low-code application platform that enables you to build scalable, secure enterprise apps.
- How Oracle APEX Works
  Oracle APEX provides all the tools you need to build apps in a single, extensible platform, which runs as a part of Oracle Database.
- About the Differences Between Runtime and Full Development Environments
  Install Oracle APEX into a runtime or a full development environment.

2.2.1 What is Oracle APEX?

Oracle APEX is an enterprise low-code application platform that enables you to build scalable, secure enterprise apps.

Oracle APEX is a hosted declarative development environment for developing and deploying database-centric web applications. Thanks to built-in features such as user interface themes,

navigational controls, form handlers, and flexible reports, APEX accelerates the application development process.

The APEX engine renders applications in real time from data stored in database tables. When you create or extend an application, APEX creates or modifies metadata stored in database tables. When the application is run, the APEX engine then reads the metadata and displays the application.

To provide stateful behavior within an application, APEX transparently manages session state in the database. Application developers can get and set session state using simple substitutions and standard SQL bind variable syntax.

2.2.2 How Oracle APEX Works

Oracle APEX provides all the tools you need to build apps in a single, extensible platform, which runs as a part of Oracle Database.

Oracle APEX uses a simple 3-tier architecture where requests are sent from the browser, through a web server, to the database. All processing, data manipulation and business logic is executed in the database. This architecture guarantees zero latency data access, top performance, and scalability, out of the box.

Oracle APEX installs with your Oracle database and consists of data in tables and PL/SQL code. Whether you run the APEX development environment or run an application built using APEX, the process is the same. Your browser sends a URL request that is translated into the appropriate APEX PL/SQL call. After the database processes the PL/SQL, the results are relayed back to your browser as HTML. This cycle happens each time you either request or submit a page.

APEX does not use a dedicated database connection. Instead, each request is made through a new database session, consuming minimal CPU resources. Application session state is managed in the database tables by the APEX engine.

Behind the scenes, the APEX engine renders and processes pages. The APEX engine also performs these tasks:

- Session state management
- Authentication services
- Authorization services
- Page flow control
- Validation processing

2.2.3 About the Differences Between Runtime and Full Development Environments

Install Oracle APEX into a runtime or a full development environment.

When you install Oracle APEX, you can install two different environments:

- **Runtime environment**. In a runtime environment users can run applications but cannot modify them. Select this option for production implementations.
- **Full development environment**. In a full development environment, users can develop, modify, run, and delete APEX applications. This option provides complete access to the APEX environment described in this document.

See Also:

- *Oracle APEX Installation Guide*
- Installing Exported Applications into a Runtime Environment in *Oracle APEX Administration Guide*

2.3 Understanding the Workspace Home Page

When you sign in to Oracle APEX, the Workspace home page appears.

Each Oracle APEX development instance can contain multiple workspaces. A **workspace** is a virtual private database which enables multiple users to work within the same APEX installation while keeping their objects, data and applications private. This section describes the regions that make up the Workspace home page.

- Development Tools
  Access development tools from either the menu bar and icons on the Workspace home page.
- Top Apps and Top Users Dashboards
  Learn about the Top Apps and Top Users regions.
- Summary and Messages Regions
  Learn about the Summary and Messages regions.
- Resources and Available Updates Regions
  Learn about the Resources and Available Updates regions.

2.3.1 Development Tools

Access development tools from either the menu bar and icons on the Workspace home page.

Development tools available on the Workspace home page include:

- **App Builder** - Create an application, composed of a set of HTML pages, based on database objects.
- **SQL Workshop** - Access tools for viewing and managing database objects.
- **Team Development** - Track and manage issues in a conversational fashion. You can create templates to provide users with starter text for issues and comments. You can apply labels and label groups, assign issue owners, and set milestones.
- **Gallery** - The Gallery (or App Gallery) enables you to access to Starter Apps, Sample Apps, and Custom Apps.

  Sample Apps highlight specific functionality and are intended to serve as a developer guide on how to make use of a particular feature. *Starter Apps* are functional apps that provide standalone point-solutions, designed to meet simple requirements that do not require a large and unnecessarily complex solution. You can use these apps "out of the box" or extend them with your own custom functionality. *Custom Apps* are custom applications that can be installed on your workspace if made available by your Instance Administrator.

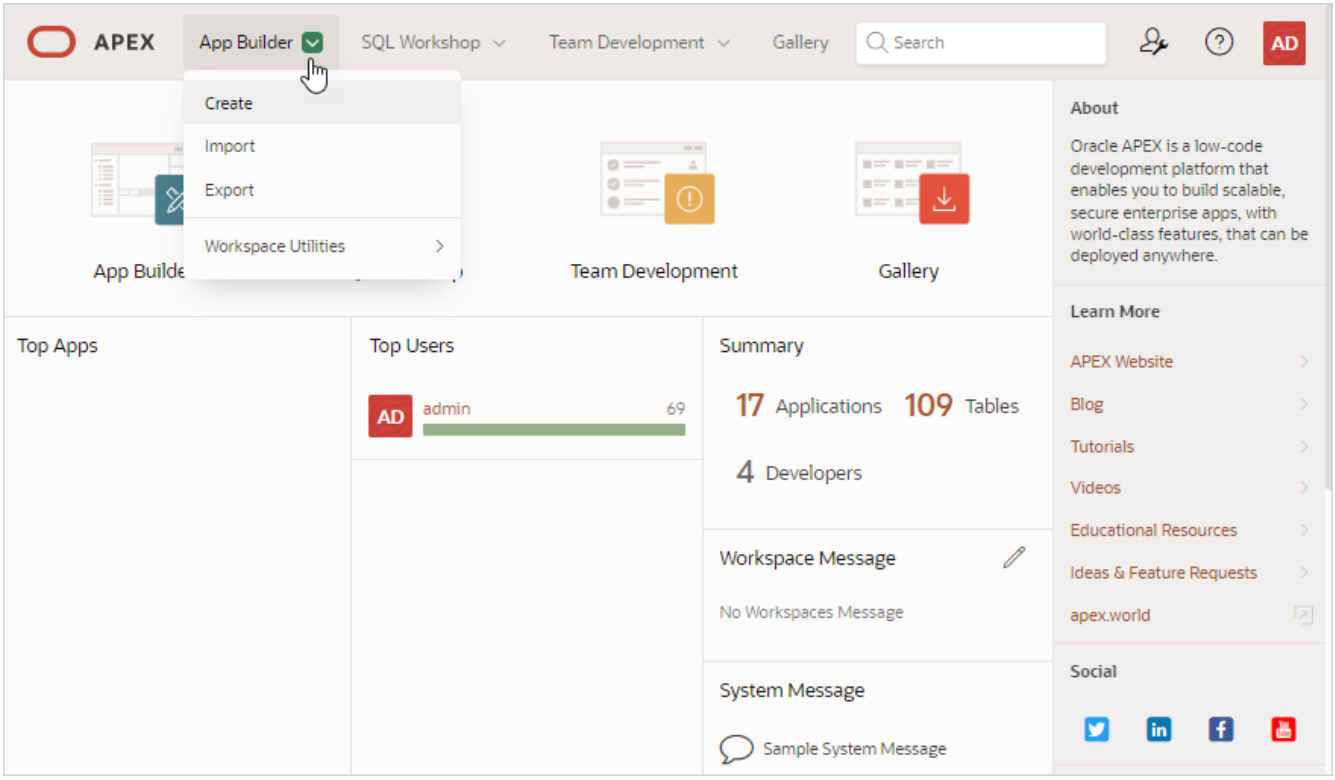
On the Workspace home, you can access these development tools in two ways:

- Selecting options from the APEX menus on the menu bar.

- Clicking development tool icons.

APEX Menus

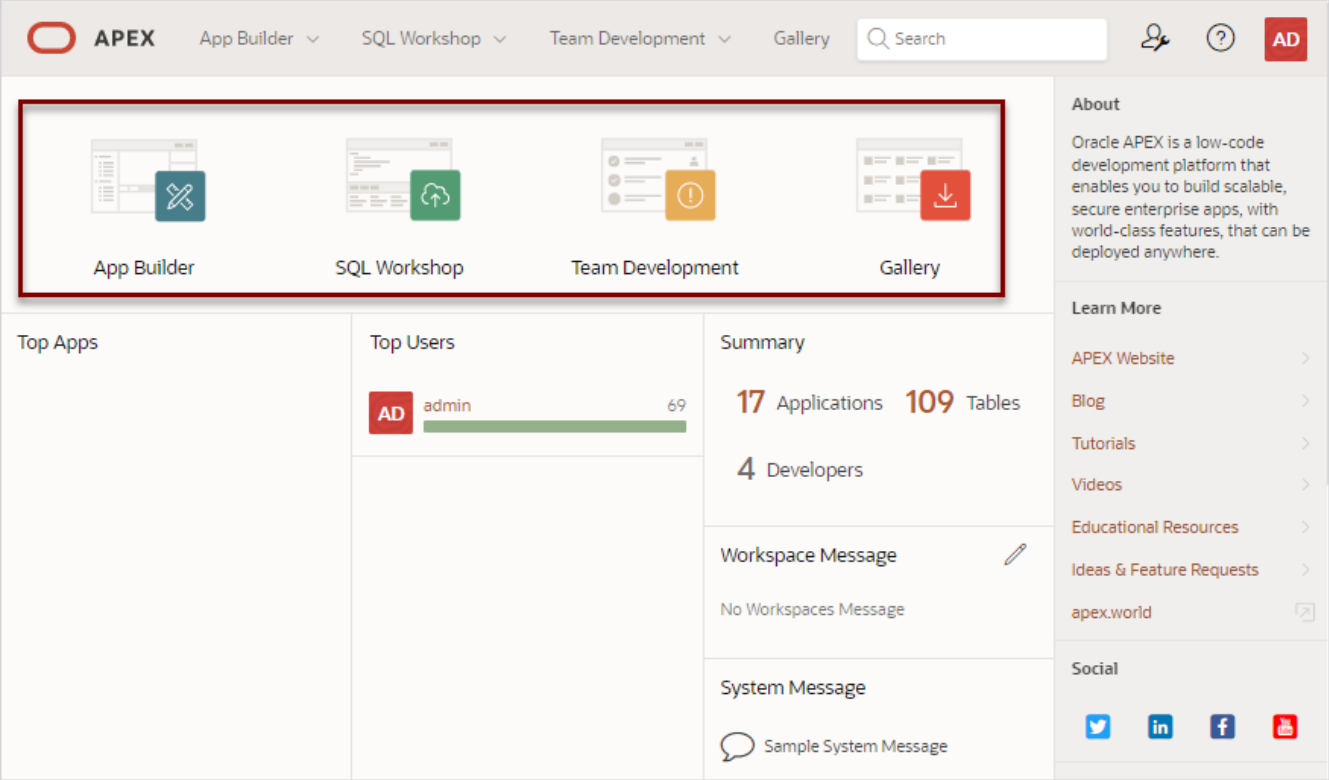
A menu bar displays at the top of the page and includes the following APEX menus: **App Builder**, **SQL Workshop**, **Team Development**, and **Gallery**. You can access these menus on most APEX pages. The example that follows shows the menu bar on the Workspace home page with the **App Builder** menu selected.



> **Tip:**
>
> To learn more about all the options available on the menu bar, see APEX Menu Bar.

Development Tool Icons

The center of the Workspace home page features four large icons that function as alternative navigation. Development tools icons on the Workspace home page include: **App Builder**, **SQL Workshop**, **Team Development**, and **Gallery**.



See Also:

- App Builder Concepts
- Getting Started with SQL Workshop in *Oracle APEX SQL Workshop Guide*
- Managing Team Development
- Using the App Gallery

2.3.2 Top Apps and Top Users Dashboards

Learn about the Top Apps and Top Users regions.

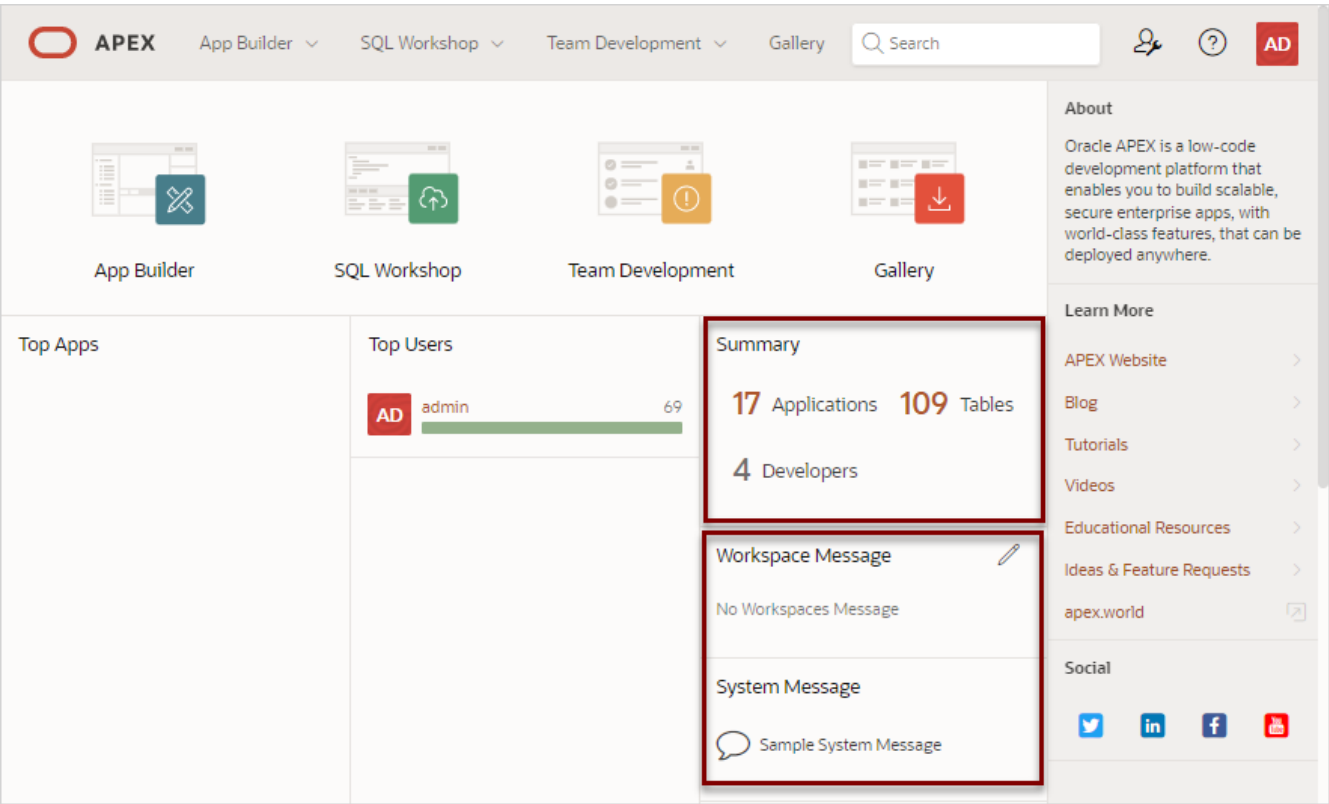
The Top Apps and Top Users regions display real time information about development activities in the current workspace. In the previous image, no apps displays in the Top Apps region, but the Top Users region displays the `admin` user.

2.3.3 Summary and Messages Regions

Learn about the Summary and Messages regions.

Summary Region

The Summary region list total applications, tables, and developers in the current workspace. In the following example, this workspace has 17 application, 109 tables, and 4 developers.



Workspace Message and System Message Regions

The Workspace Message region displays beneath the Summary region. In the previous image, no message it defined so the region reads `No Workspaces Message`. Administrators can update or edit a workspace message by clicking the Edit icon.

The System Message region displays next. Instance administrator can create a System Message to communicate with other users within an instance. In the previous image, the System Message reads `Sample System Message`.

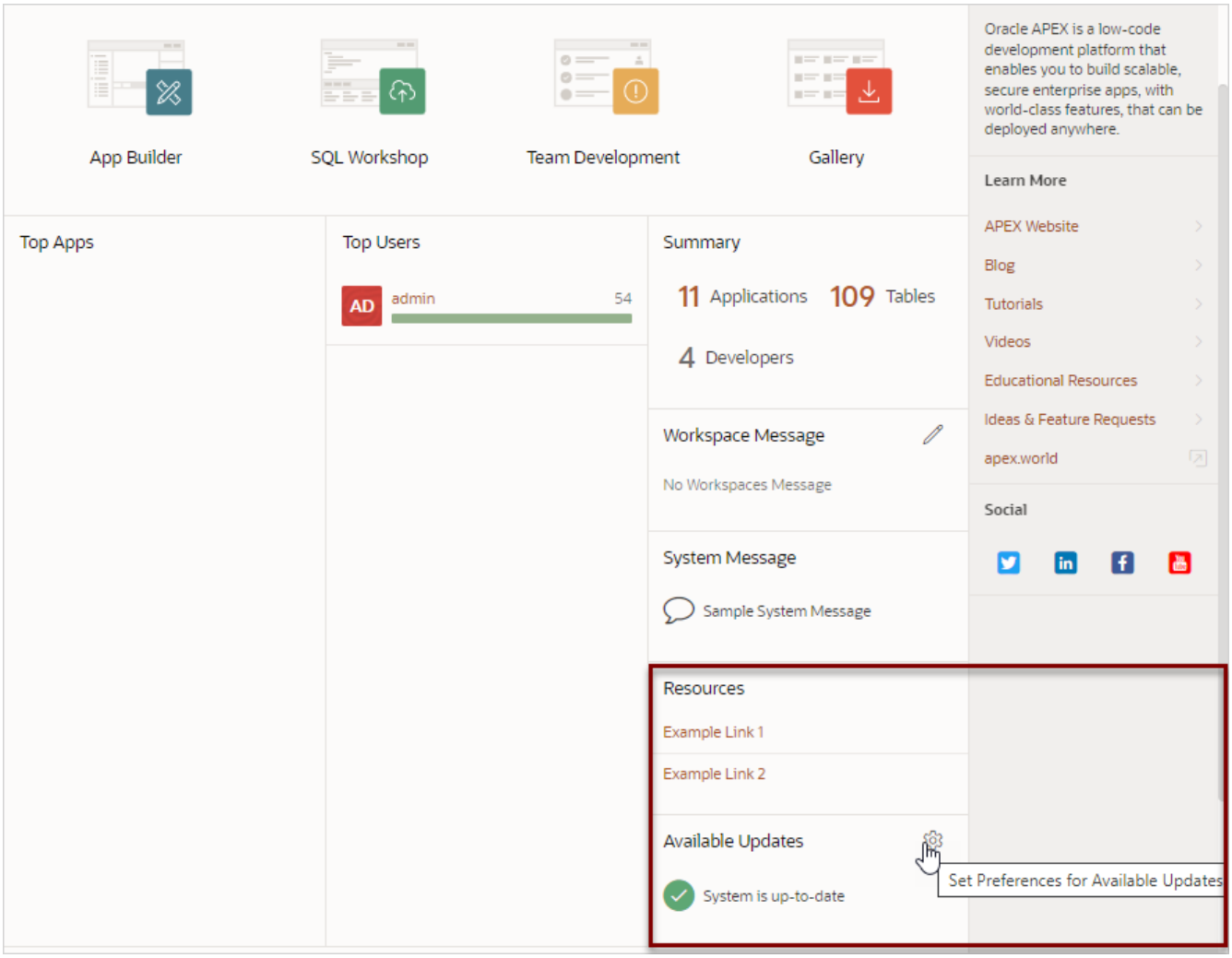
> **See Also:**
>
> Defining a Workspace Message and Defining a System Message in *Oracle APEX Administration Guide*

2.3.4 Resources and Available Updates Regions

Learn about the Resources and Available Updates regions.

Resources

The Resources region displays below System Message. Administrators can add links to the Resources list to communicate important user information such as training, discussion forums, and user feedback applications. The Resources region only appears if resources have been defined in Administration Services.



> **See Also:**
>
> Managing the Site-Specific Resources List in *Oracle APEX Administration Guide*

Available Updates

> **Note:**
>
> APEX updates on Oracle Autonomous Database are managed by Oracle. See Control Oracle APEX Upgrades in *Using Oracle Autonomous Database Serverless*. To learn how to programmatically determine the upgrade status and initiate the upgrade, see `APEX_INSTANCE_ADMIN`, Available Parameter Values in *Oracle APEX API Reference*.

The Available Updates region displays on the lower right side of the page and indicates if updates are available for APEX and Oracle REST Data Services. If your installation is current the following message displays:

```
System is up-to-date
```

To configure updates, click the **Set Preferences for Available Updates** icon which displays on the right side of the region. In the dialog, enable or disable updates by selecting **Yes** or **No** from the Check for Available Updates list and click **Apply Changes**.

> **Note:**
>
> To perform this check, APEX transmits the version number of APEX and other system components to Oracle Corporation. In order to continuously improve our products, Oracle is interested in learning about product usage. To that end, statistics can occasionally be sent to Oracle describing the product features in use. No personally identifiable information is sent. To review Oracle's privacy policy see http://www.oracle.com/us/legal/privacy/overview/index.html.

2.4 APEX Menu Bar

Learn about the APEX menu bar.

You can access development tools from either the menu bar and icons on the Workspace home page. The menu bar displays in the header region on most Oracle APEX pages and enables developers to access APEX menus, the Global Search, Extension menu (if enabled), Administration menu, Help menu, and the Account menu.

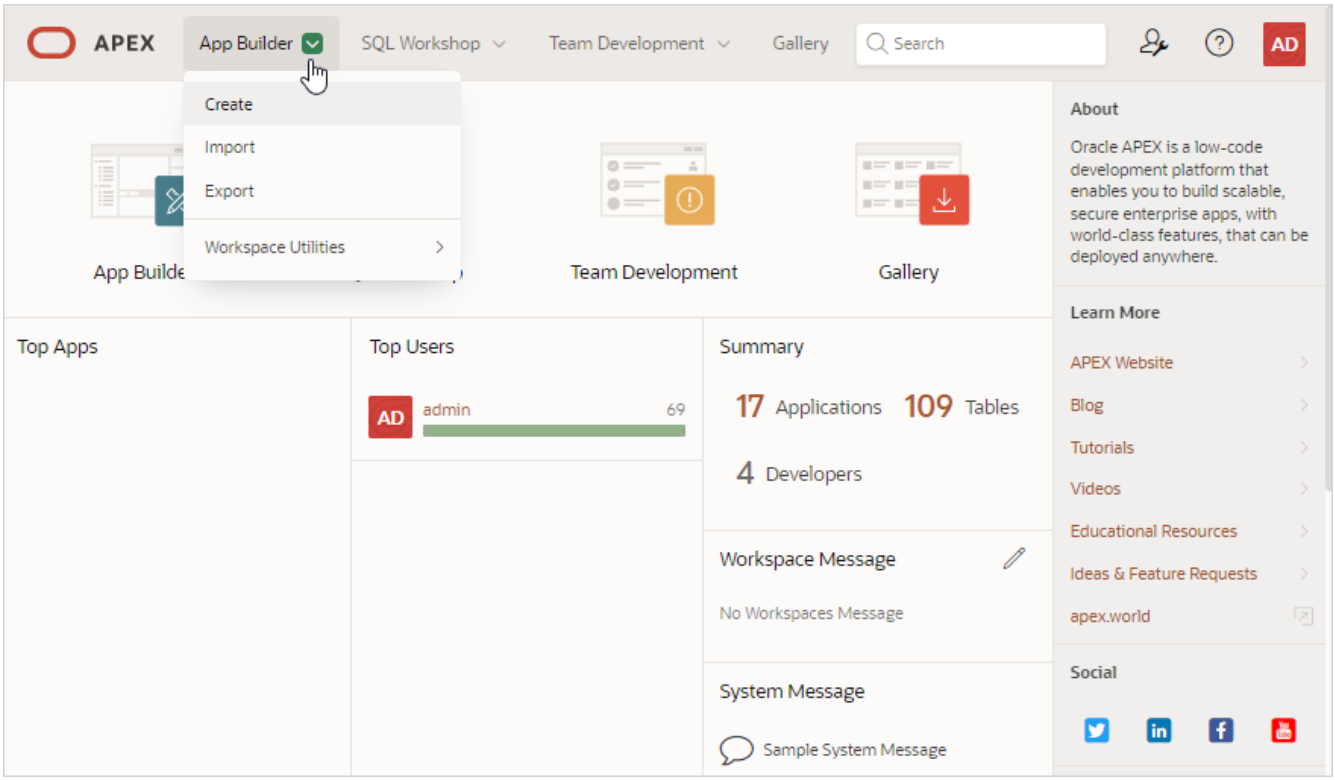
You can access development tools from either the menu bar and icons on the Workspace home page.

- APEX Menus
  Use the APEX menus to access to the following development tools: App Builder, SQL Workshop, Team Development, and the Gallery.
- Global Search
  Search all applications in an APEX workspace using Global Search.
- Extension Menu
  Workspace administrators can add custom links to the Extensions menu on the APEX menu bar.
- Administration Menu
  Click the Administration menu to access Workspace Administration pages such as Manage Service, Manage Users and Groups, Monitor Activity, Dashboards, or Change My Password.
- Help Menu
  Click the Help menu to link to documentation on Oracle Help Center, the Discussion Form, the Oracle APEX page, and the About Oracle APEX dialog.
- Account Menu
  Click the Account menu to configure appearance settings, edit your user profile or preferences, or sign out.

2.4.1 APEX Menus

Use the APEX menus to access to the following development tools: App Builder, SQL Workshop, Team Development, and the Gallery.

Each menu offers quick access to key development tools, components, and functions. In the following example, the **App Builder** menu is selected and displays the menu options **Create**, **Import**, **Export**, and **Workspace Utilities**.



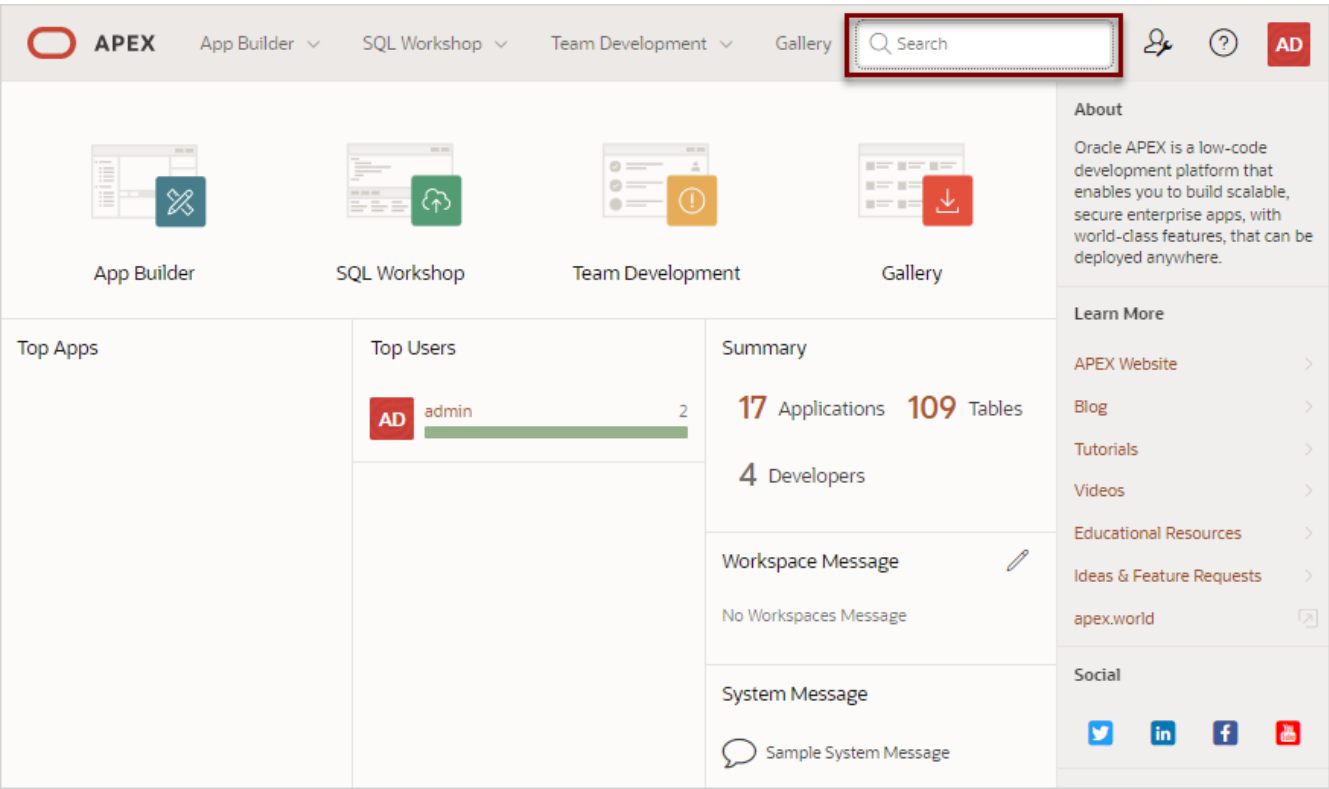
See Also:

- App Builder Concepts
- Getting Started with SQL Workshop in *Oracle APEX SQL Workshop Guide*
- Managing Team Development
- Using the App Gallery

2.4.2 Global Search

Search all applications in an APEX workspace using Global Search.

The Global Search field (also known as Spotlight Search) displays in the header region.



Key features of Global Search include:

- Offers effortless navigation within a workspace enabling you to quickly find applications, pages, Shared Components, APEX menus, and so on.
- Search for component names and immediately set focus to those items in Page Designer.
- Provides a unified search experience throughout APEX.

About Using Global Search

You can activate Global Search in two ways:

- Click the **Global Search** field in the in the header region
- Use the keyboard shortcut (`Ctrl + Quote`) (`'`) (US Keyboards only)

Use Global Search to search all applications in the workspace no matter what your location in the current workspace. For example, if your workspace contains the *Sample Reporting* sample app, entering `Sample` will find it. Similarly, you can access global navigation menus that appear at the top window adjacent to the Oracle logo. For example, enter `SQL Commands` in the Global Search field to instant access SQL Commands.

To close the Global Search dialog, press the `Esc` key, or click anywhere outside the results.

Syntax to Run Searches

Search by page name and page alias. With in an in-app, use a colon (:) or dash (–) between the application ID or alias and the page ID, name, or alias.

Within App Builder and on the Application home page, APEX supports the following search types and syntax:

- **Pages**:

  Page Name, ID, or Alias
- **Applications**:

  App Name, ID, or Alias
- **App Shared Components and Actions**:

  Examples include workflows, lists of values, templates, and so on.
- **Workspace Utilities, SQL Workshop Utilities**
- **Advanced Search**

Advanced Search

For Advanced Search use in the following formats:

```
application_ID:{search term}
application_ALIAS:{search term}
application_ID-{search term}
application_ALIAS-{search term}
```

Search terms can be a page ID, Name, Alias, or Shared Component.

Where:

- `application_ID` - A numeric identification number (or ID) that identifies an application.
- `application_ALIAS` - An alternate alphanumeric application identifier. You can use this identifier in place of the application name.
- `page_ALIAS` - An alternate alphanumeric page identifier. You can use this identifier in place of the page name.
- `page_ID` - Numeric identification number (or ID) that identifies a specific page.
- `page_NAME` - An alphanumeric name that identifies a specific page.

The following example navigates you to page 10 of application 570:

```
10
570-10
570:10
```

You can also use this syntax to navigate to the Shared Components page of a given application:

```
application_ID:s
application_ID-s
```

Where:

- `application_ID` is the numeric identification number (or ID) that identifies an application.

The following example demonstrates how to navigate to the Shared Components page of application 570:

```
570:s
570-s
```

Navigating Search Results

Global Search results display in a dialog. Use the Up and Down arrows keys to navigate the results and press **Enter** to make a selection. Since the the **Global Search** icon displays on nearly every page in APEX, you can continue to narrow your search until you find the app, page, or component (such as the item, button, or region) which contains the keyword.

Other available shortcuts change depending on where you are in App Builder, for example:

- In Page Designer - To search for a page, press `Ctrl + 4`.
- In App - To search within an application, press `Ctrl + 3`.
- All Apps - To search all applications, press `Ctrl + 2`.
- Current Workspace - To search a workspace, press `Ctrl + 1`.

> **Tip:**
>
> Some browsers map these `Ctrl` key shortcuts for quick tab navigation. In that case, the shortcut does not work as described above. Instead, they simply activate a particular open browser tab.

Example 2-1 Search Example

Suppose you need to find an application called *Employees* which includes interactive grid built on the `EMP` table.

> **Note:**
>
> In the following example, references to *templates* appear because they contain the text string `emp`.

1. **Find the application**.
   a. On the Workspace home page, click the Global Search field and enter `emp`.

      Global Search would return results in the following order:

      - **Applications**

        Employees

        App 105 (employees)
      - **Pages**

        Employees

        App 105/Page 2 (employees)
      - **Global**

        T**emp**lates

        Team Development / Templates
      - Shared Components:
        - T**emp**lates

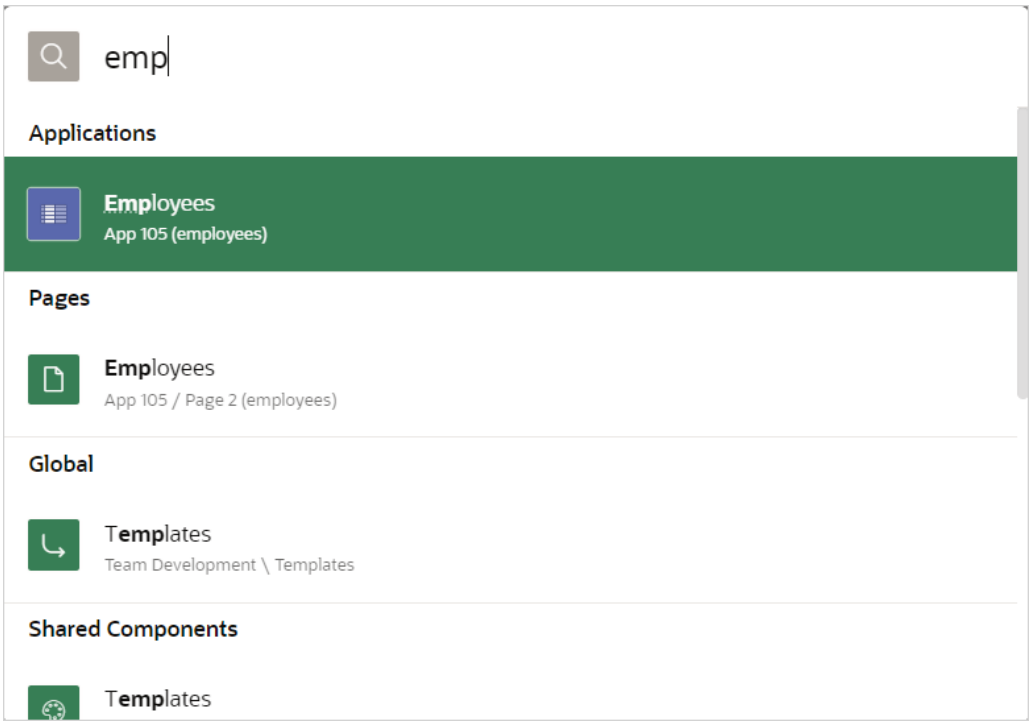
App 105 / Shared Components

– Email T**emp**lates

App 105 / Shared Components

- **Actions**
  - Email T**emp**lates

    App 105 / Shared Components
- **Shortcuts**
  - Search this App for emp (`Ctrl + 3`)
  - Search all applications for emp (`Ctrl + 2`)
  - Search Workspace for emp (`Ctrl + 1`)



b. Under **Applications**, select **Employees** to view that application.

   The Employees application appears.

2. **Search the application**.

   a. On the Application home page, click the Global Search field and enter `emp`.

      Global Search returns the results.

   b. Under **Pages**, select the **Employees** page.

      The page appears in Page Designer.

3. **Search the page**.

   a. In Page Designer, click the Global Search field and enter `emp` again. Then, select the shortcut, **Search Page for emp** (or click **Crtl+4**).

      Global Search returns the results and displays them in the Page Search tab in the Central pane.

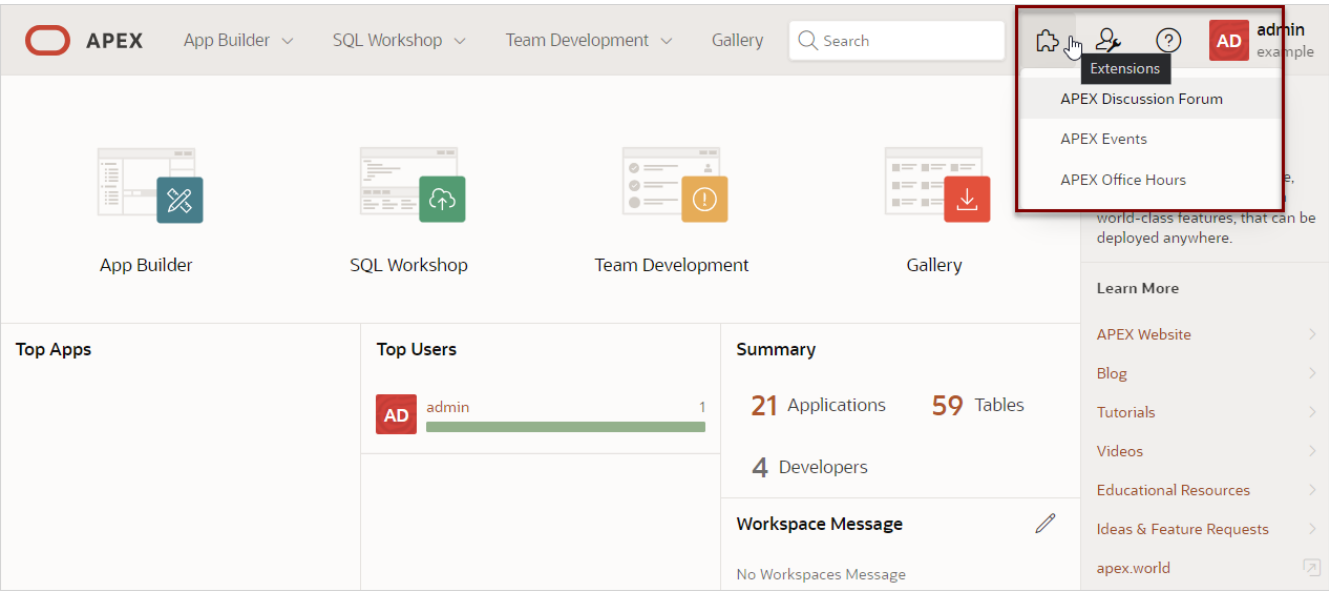
b. Select a component (such as a process, column, region, and so on).

2.4.3 Extension Menu

Workspace administrators can add custom links to the Extensions menu on the APEX menu bar.

The Extension menu only appears on the menu bar if a Workspace Administrator has added links to it. When configured, the Extension menu displays on the menu bar between the **Global Search** field and the **Administration** menu.

In the following example shows an Extensions menu with links to APEX Discussion Forum, APEX Events, and APEX Office Hours.

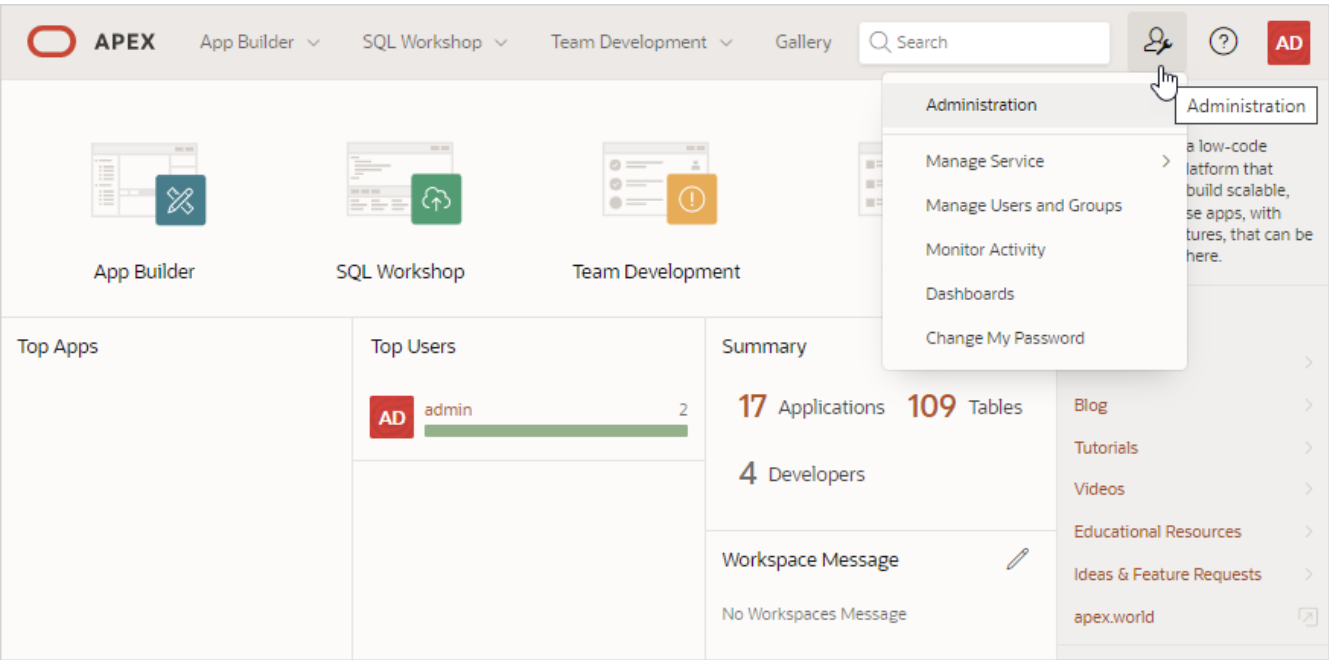


To learn more about the Extension menu, see Managing Links on the Extension Menu in *Oracle APEX Administration Guide*.

2.4.4 Administration Menu

Click the Administration menu to access Workspace Administration pages such as Manage Service, Manage Users and Groups, Monitor Activity, Dashboards, or Change My Password.

The Administration menu displays adjacent to the Help menu and features an outline person with a wrench.



Clicking the Administration menu displays the following options:

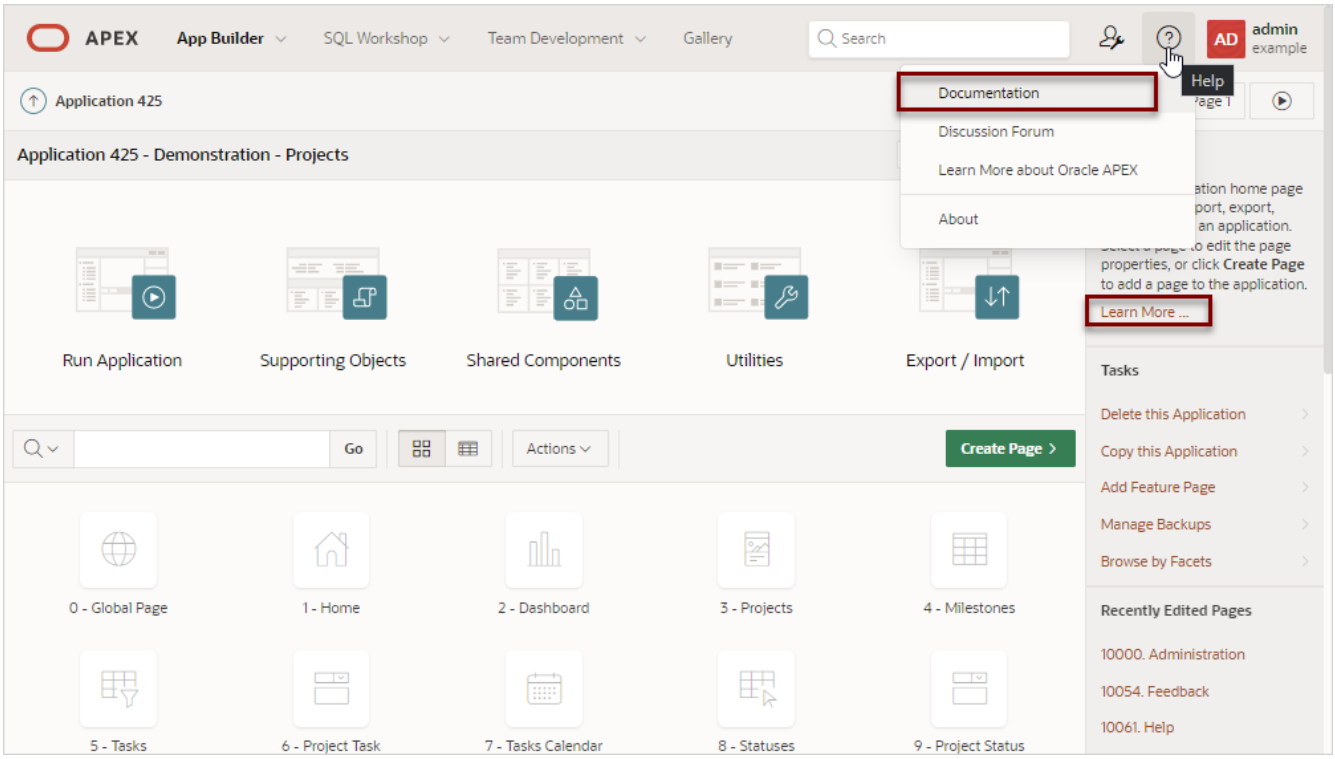
- **Administration** links to the Workspace Administration page. To access the Oracle APEX development environment, users sign in to a shared work area called a **workspace**. Workspace administrators monitor and administer a workspace. Common workspace administration tasks include managing service requests, configuring workspace preferences, creating workspace announcements, managing session state, creating user accounts, and managing groups. See Workspace Administration Page in *Oracle APEX Administration Guide*.
- **Manage Service** displays a submenu of options on the Manage Service page. Administrators use the Manage Service page to manage service requests, configure workspace preferences, define workspace messages, enable a workspace-level banner, view the Workspace Summary report, and configure Extension menu links. See Manage Service Page in *Oracle APEX Administration Guide*.
- **Manage Users and Groups** links to Manage Users and Groups page. Workspace administrators can create new user accounts, manage existing user accounts, and change user passwords. See Managing Users in a Workspace in *Oracle APEX Administration Guide*.
- **Monitory Activity** links to the Monitor Activity page. Use this page to monitor changes to page views, developer activity, page view analysis, and active sessions. It also shows reports on such information as login attempts and external click counts. See Monitoring Activity Within a Workspace in *Oracle APEX Administration Guide*.
- **Dashboards** links to the Dashboard page. Use this page to view details about the current workspace and monitor users, activity, developer activity, performance, application development, and view database details. See Viewing Workspace Dashboards in *Oracle APEX Administration Guide*.
- **Change My Password** links to the Change Password section of the Edit Profile dialog box. See Changing Your Profile or Password.

> **See Also:**
>
> Workspace and Application Administration in *Oracle APEX Administration Guide*

2.4.5 Help Menu

Click the Help menu to link to documentation on Oracle Help Center, the Discussion Form, the Oracle APEX page, and the About Oracle APEX dialog.

The **Help** menu resembles a question mark enclosed with a circle.



Click the Help menu to access the following menu items:

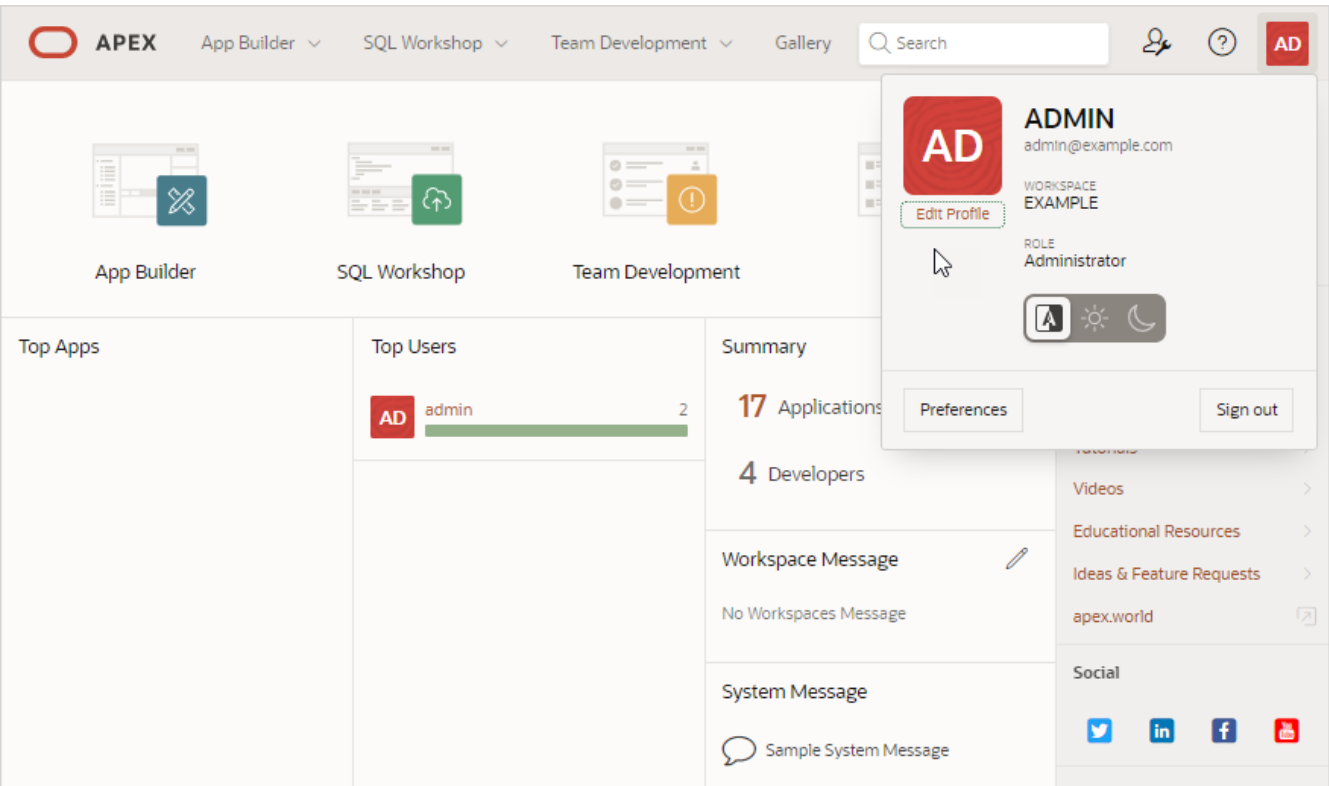
- **Documentation** links to Oracle APEX, Oracle Help Center. To learn more, Using Oracle APEX Documentation.
- **Discussion Forum** links to the APEX page in Oracle Forums.
- **Learn More about Oracle APEX** links to the Oracle APEX page.
- **About** links to the About Oracle APEX page which displays details about the Oracle APEX instance including product details, CGI environment details. database information, currently selected character sets, database initialization parameters, and Oracle legal notices.

  To learn more, see Viewing the About Oracle APEX Page in *Oracle APEX Administration Guide*.

2.4.6 Account Menu

Click the Account menu to configure appearance settings, edit your user profile or preferences, or sign out.

The Account menu displays in the upper right corner of the page. Open the Account menu by clicking on the User icon in the header.



The Account menu displays your username, email address, the current workspace, and your user role. If Single Sign On (SSO) is enabled, the link **Change Workspace** also appears.

> **Tip:**
>
> If you are using Single Sign-on authentication, **Change Workspace** also displays. Click **Change Workspace** to view a list of workspaces you have access to.

Key controls on Account menu include:

- **Edit Profile**. Click **Edit Profile** to edit the options. Click **Apply Changes** to save your changes.
  - Profile Details:
    - Email Address - Specifies the email address for use with general correspondence, notifications about resetting passwords, and so on.

    - First Name - Specify the first or given name to further identify the user. This optional field is used for informational purposes only.
    - Last Name - Specify the last or family name to further identify the user. This optional field is used for informational purposes only.
  - Profile Photo - Select a photo to further identify the user. This optional field is used for informational purposes only.
  - Password - If you wish to change your password, enter a new password. Otherwise, leave the password fields empty and the current password will not be changed.
    - Enter Current Password - Enter your current password. Do not enter a password unless you want to change the current password.
    - Enter New Password - The password used to log in to the system. Do not enter a password unless you want to change the current password.
    - Confirm Password - Enter the password again. Do not enter a password unless you want to change the current password.
- **Appearance**. Configure how the Oracle APEX user interface displays. Options include:
  - **Automatic** (default) detects the appropriate setting based on existing system or browser settings.
  - **Light Mode** renders the user interface using dark-colored text, icons, and graphical user interface elements on a lighter background.
  - **Dark Mode** renders the user interface using light-colored text, icons, and graphical user interface elements on a darker background.
- **Preferences**. Click **Preferences** to configure the options. Click **Apply Changes** to save your changes.
  - **Default Scheme** - Specifies the default schema used for data browsing, application creation, and SQL script execution.
  - **Default Date Format** - Specify the default Oracle date format for the user. This option controls the default date format within SQL Workshop.
  - **Unicode CSV Downloads** - Specify whether to provide SQL Workshop CSV downloads in Unicode. This only affects SQL Workshop and not custom applications.
  - **Run application in new window** - Specify how Oracle APEX manages windows or tabs when running applications from the App Builder when using Mozilla Firefox and Microsoft Internet Explorer.
  - **Use single window to run all applications** - Specify whether different applications reuse the same application runtime window or tab.
- **Sign Out**. Click **Sign Out** to log out of the current workspace.

See Also:

- Changing Your Profile or Password
- Editing User Preferences and Enabling Dark Mode

2.5 Understanding App Builder

Developers use App Builder to create and manage applications and application pages.

The App Builder home page displays all installed applications in the current Oracle APEX instance. When a developer selects an application to edit, the Application home page appears. Use the Application home page to run, edit, import, export, copy, or delete applications.

- Accessing App Builder
  Access App Builder by clicking the App Builder icon on the Workspace home page.
- Understanding the App Builder Home Page
  Use App Builder to create new applications, select an application to edit, import previously exported applications, view the Dashboard, or access workspace utilities.
- Accessing the Application Home Page
  Access the Application home page by clicking the App Builder icon on the Workspace home page and then selecting an application.
- Understanding the Application Home Page
  Use the Application home page to run, edit, import, export, copy, or delete applications.

See Also:

- Quick Start
- App Builder Concepts

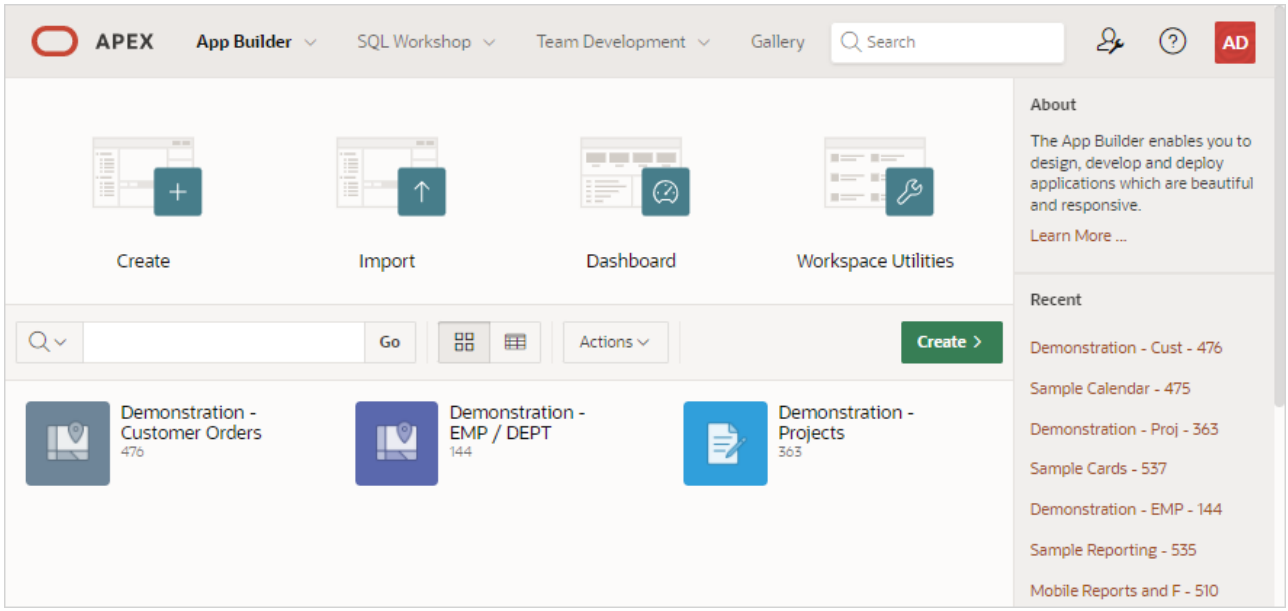
2.5.1 Accessing App Builder

Access App Builder by clicking the App Builder icon on the Workspace home page.

To access App Builder:

1. Sign in to Oracle APEX.
2. On the Workspace home page, click the **App Builder** icon.

   The App Builder home page appears.



Use the App Builder home page create new applications, select an application to edit, import previously exported applications, view the Dashboard, or access workspace utilities. In the above image, currently installed applications display as large icons at the bottom of the page.

> **Tip:**
>
> In a new workspace, the bottom of the page displays two additional regions. Click **Create a New App** to access the Create Application Wizard and create a new application. Click **Install a Starter or Sample App** to access the Gallery and install sample apps, plug-ins, and pre-built apps you can use as is or customize.

> **See Also:**
>
> - Understanding the App Builder Home Page
> - Creating Applications

2.5.2 Understanding the App Builder Home Page

Use App Builder to create new applications, select an application to edit, import previously exported applications, view the Dashboard, or access workspace utilities.

> **Tip:**
>
> In a new workspace, the first time you access the App Builder home page, a **Get Started Now** region appears. Click **Create New App** to access the launch the Create Application Wizard. See Choosing an Application Creation Method. Click **Install a Starter or Sample App**to access the Gallery. See Using the App Gallery.

- App Builder Home Page Icons and Navigation Bar
  Learn about using the icons and navigation bar on the App Builder home page.
- Create Button
  Learn about the App Builder home page Create button.
- App Builder Home Side Bar
  Learn about links on the App Builder home page side bar.

> **See Also:**
>
> - Accessing App Builder
> - APEX Menu Bar

2.5.2.1 App Builder Home Page Icons and Navigation Bar

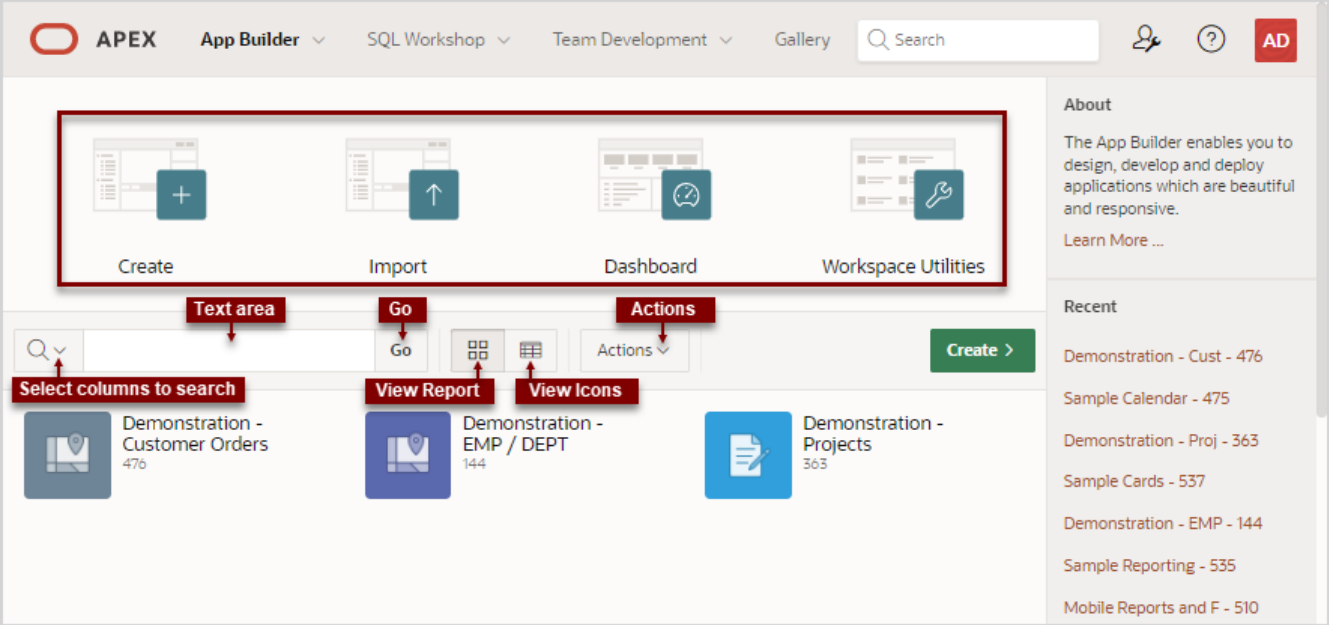
Learn about using the icons and navigation bar on the App Builder home page.

The App Builder home page features four large icons (Create, Import, Dashboard, and Workspace Utilities) and the bottom of the page displays all currently installed applications.

About App Builder Home Page Icons

The top of App Builder home page contains four large icons:

- **Create**. Launches the Create Application Wizard. Use the Create Application Wizard to build a complete application containing multiple pages. See Creating Applications.
- **Import**. Launches the Import Wizard. Use this wizard to import export files from the export repository. See Importing Export Files.
- **Dashboard**. Links to the App Builder Dashboard which displays metrics about applications in the current workspace.
- **Workspace Utilities**. Use the Workspace Utilities page to access REST Enabled SQL Services, Remote Servers, Application Groups, Web Credentials, Workspace Themes, Export, Oracle APEX Views, Manage Backups, and REST Source Catalog. See Using Workspace Utilities.



About the Navigation Bar

The bottom of App Builder home page displays all installed applications. You can customize how the applications display using the navigation bar in the center of page.

Available controls include:

- **Select columns to search** - This control features a magnifying glass button. Click this button to narrow your search to only specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View Icons** (default) - Displays each application as a large icon identified by the application name.
- **View Report** - Displays each application as a line in a report with the following columns: Application (ID), Name (application), Pages, Updated, Working Copies, and Run.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view. For example, display additional columns by selecting **Actions**, **Columns**.

> **Tip:**
>
> In a new workspace, the bottom of the App Builder home page displays two regions: **Create a New App** and **Install a Starter or Sample App**. Follow the on-screen instructions. Once you create or install an application, these region disappear.

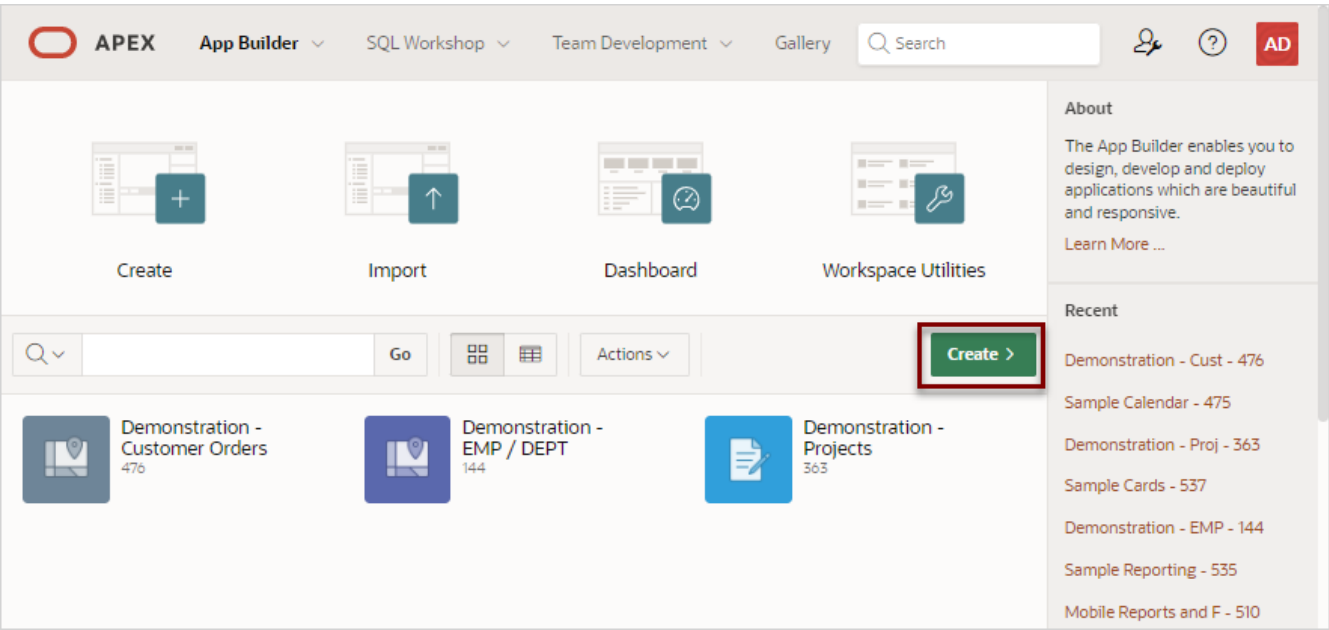
See Also:

- Choosing an Application Creation Method
- Importing Export Files
- About the Actions Menu

2.5.2.2 Create Button

Learn about the App Builder home page Create button.

The Create button displays in the center of the page to the right of the navigation bar.

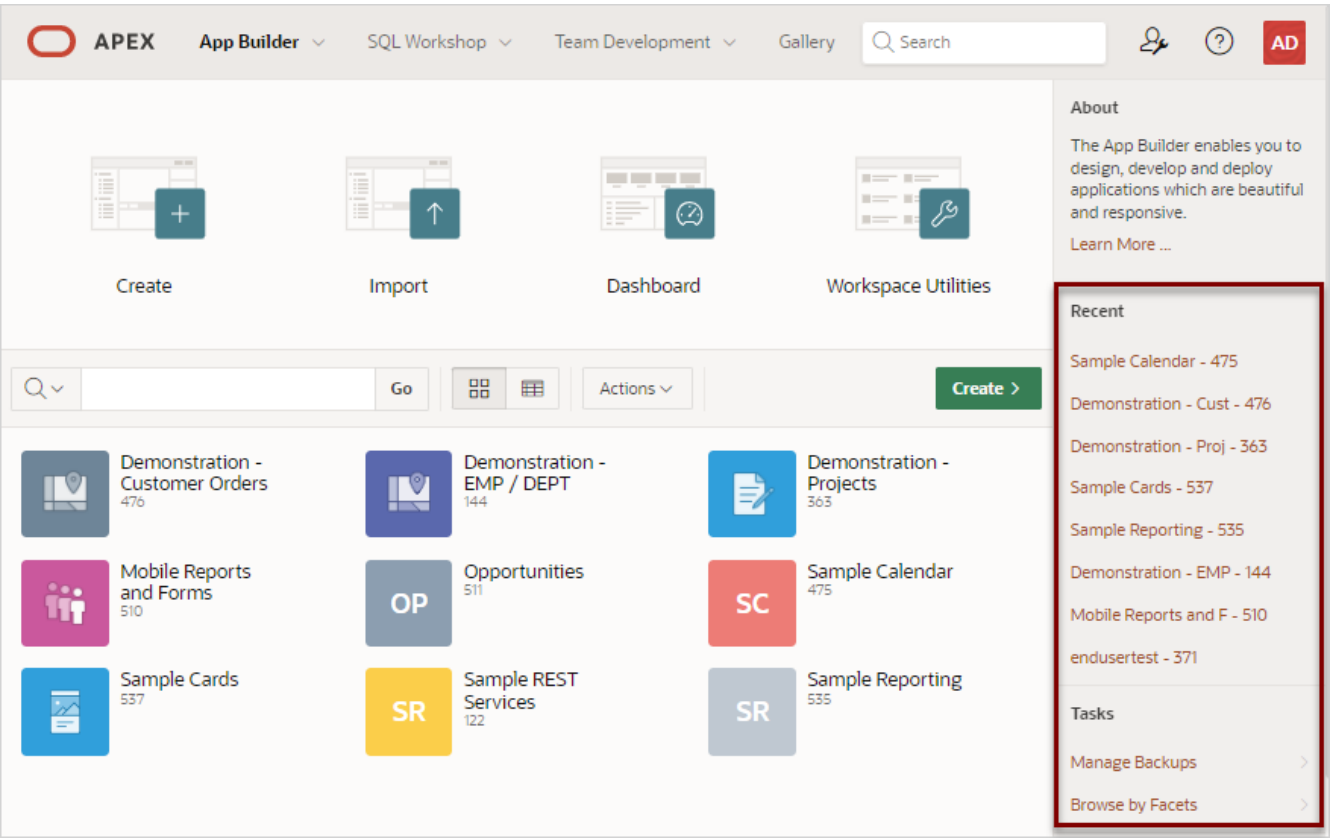


Click **Create** to create a new application. The Create an Application page appears and prompts you to select how to create the application. To learn more, see Choosing an Application Creation Method.

2.5.2.3 App Builder Home Side Bar

Learn about links on the App Builder home page side bar.

The App Builder home page feature two regions on the side bar: Recent and Tasks.



Recent

The **Recent** region contains links to recently viewed applications. Click an application name to view it on Application home page.

Tasks

The **Tasks** region contains the following links:

- **Manage Backups** - Links to the Manage Backups page. See Managing Application Backups.
- **Export Applications** - Links to the Export Application page. See Exporting an Application.
- **Browse by Facets** - Links to the Search Applications page. Use Search Applications find applications in the current workspace.

2.5.3 Accessing the Application Home Page

Access the Application home page by clicking the App Builder icon on the Workspace home page and then selecting an application.

Use the Application home page to run, edit, import, export, copy, or delete applications.

To access the Application home page:

1. Sign in to Oracle APEX.

   The Workspace home page appears.

2. Click the **App Builder** icon.

The App Builder home page appears.

3. Select an application on the Application home page.

   The Application home page appears.

> **See Also:**
>
> Understanding the Application Home Page

2.5.4 Understanding the Application Home Page

Use the Application home page to run, edit, import, export, copy, or delete applications.

When a developer selects an application on the App Builder home page, the Application home page appears.

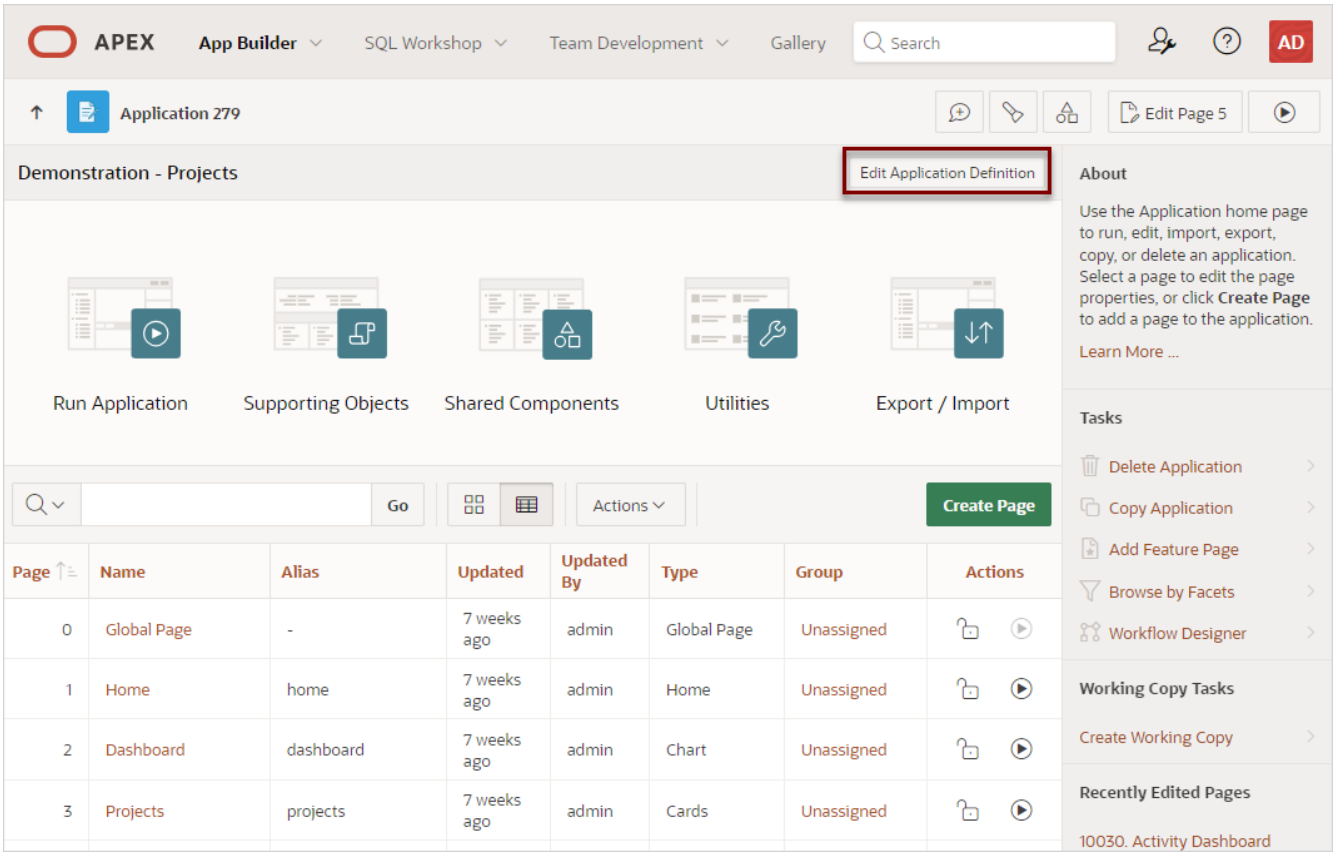
- Edit Application Definition Button
  Learn how to identify the application ID and name and edit the application definition.
- Application Home Page Icons
  Learn about the icons on the Application home page.
- Create Page Button
  Learn about Create Page button on the Application home page.
- Application Home Page Navigation Bar
  Learn about navigation bar on the Application home page.
- Application Home Side Bar
  Learn about links on the Application home page side bar.

> **See Also:**
>
> - Accessing App Builder
> - APEX Menu Bar

2.5.4.1 Edit Application Definition Button

Learn how to identify the application ID and name and edit the application definition.

The application ID and the application name display at the top of the Application home page. The application ID also displays in the breadcrumb at the top of the page.

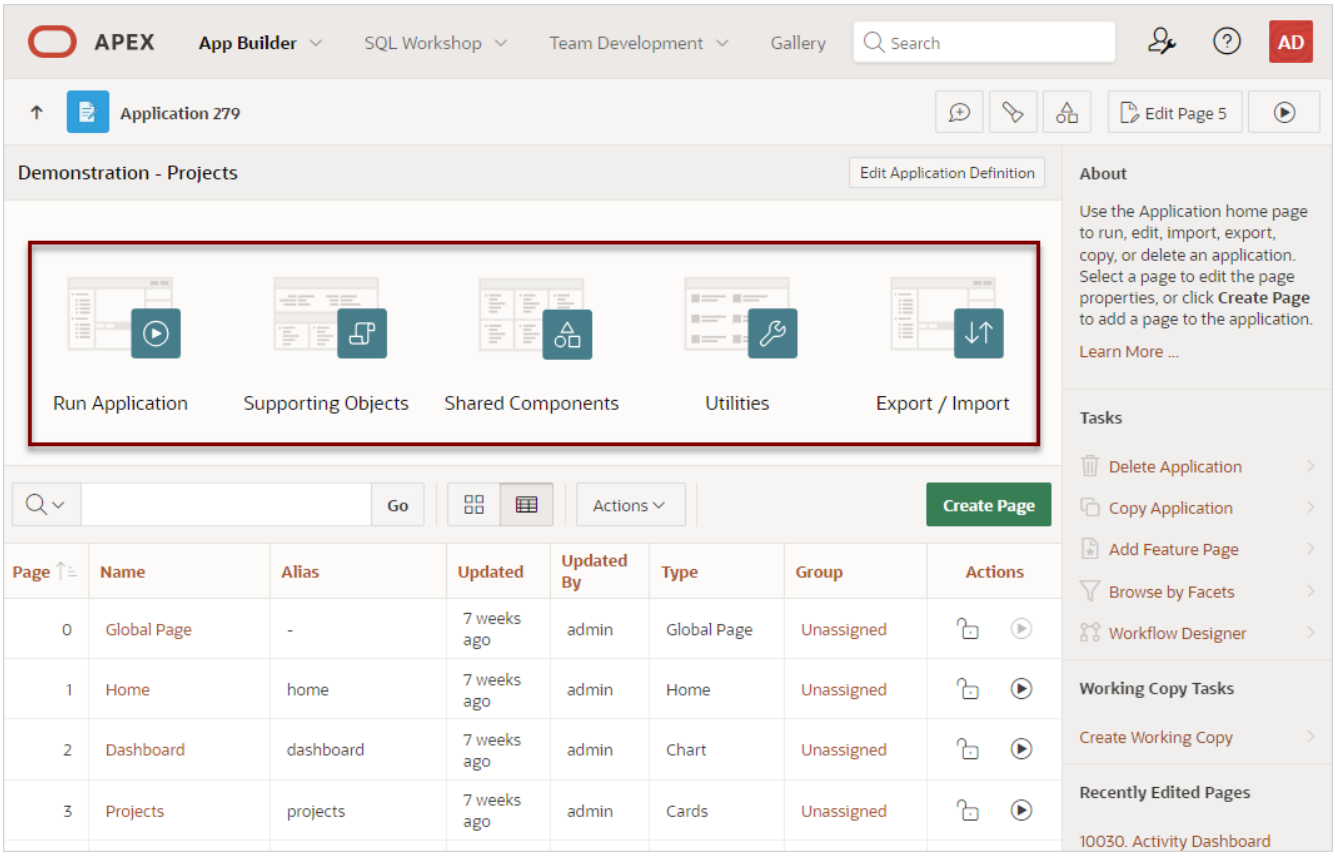


The **Edit Application Definition** button displays to the right of the application ID and name. Click **Edit Application Definition** to edit the application definition, including the application Name and Application Alias.

> **See Also:**
>
> Editing the Application Definition

2.5.4.2 Application Home Page Icons

Learn about the icons on the Application home page.



The following large icons appear beneath the application ID and application name:

- **Run Application** submits the home page in the current application to the Oracle APEX engine to render viewable HTML.

  See Running an Application or Page.

- **Supporting Objects** links to the Supporting Objects page. Use Supporting Objects to define database object installation scripts that are invoked when importing an application. You can also define deinstallation scripts to drop objects when deleting an application.

  See How to Create a Custom Application.

- **Shared Components** links to the Shared Components page. Shared components can display or be applied to any page within the application.

  See Managing Shared Components.

- **Utilities** links to the Utilities page. Use this page to monitor developer activity, view dashboards, run Advisor, and view numerous other reports.

  See Using Application Utilities.

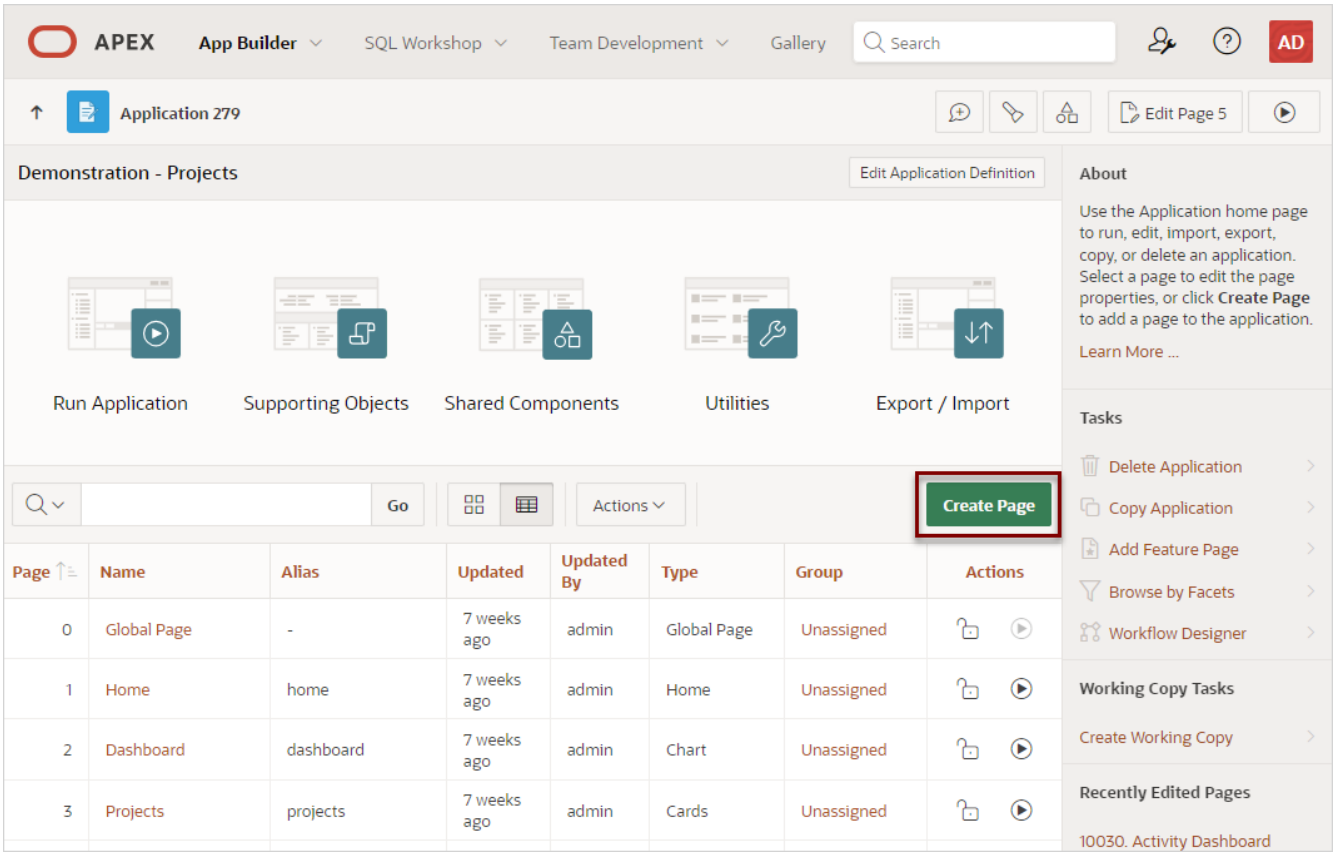
- **Export/Import** links you to the Export/Import page. Click **Import** or **Export** to import or export a workspace, an application, application components, or related files. Select

Remote Deployment to deploy an application to remote Oracle APEX instances using REST Enabled SQL references.

See Exporting and Importing from App Builder and Understanding One-Click Remote Application Deployment.

2.5.4.3 Create Page Button

Learn about Create Page button on the Application home page.



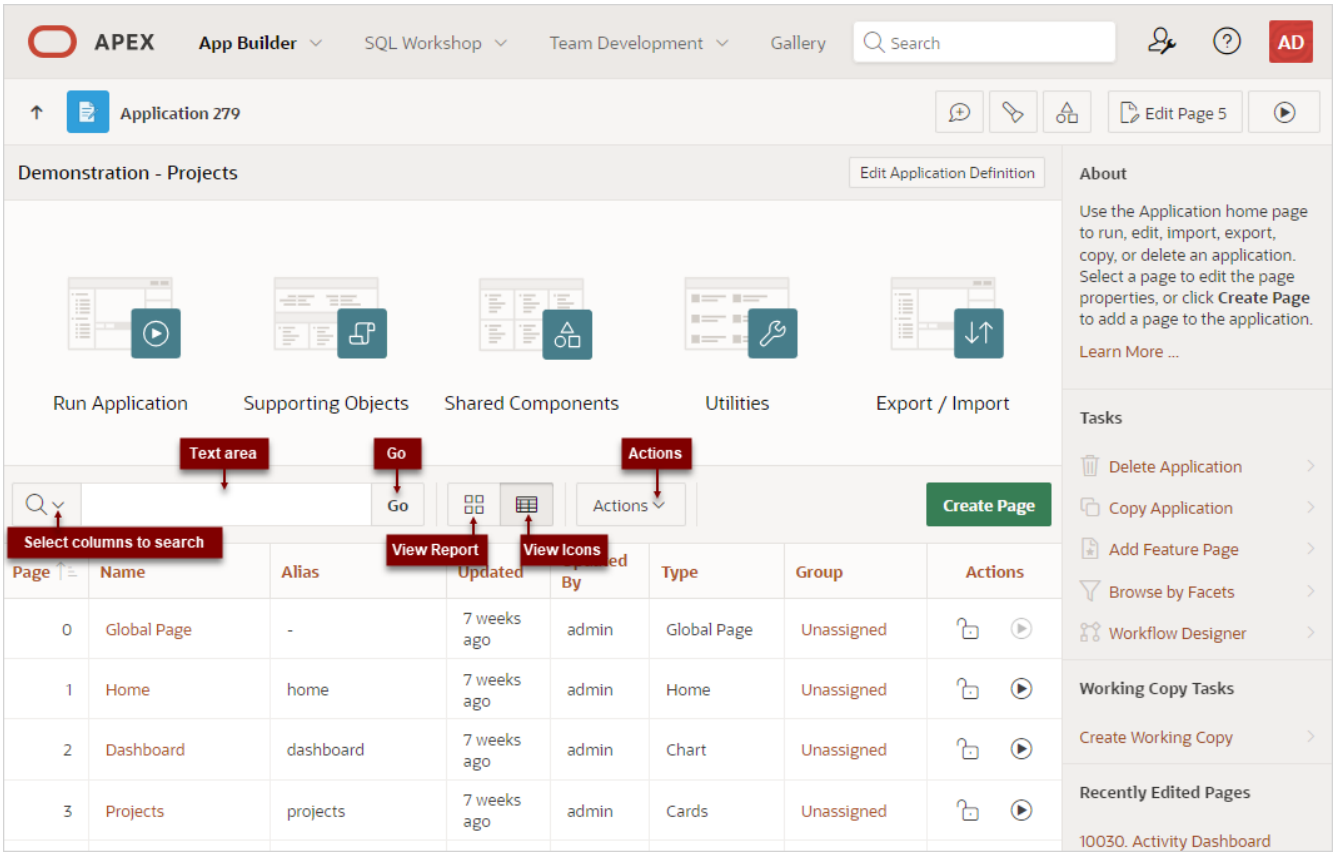
Clicking the **Create Page** button launches a wizard that walks you through creating a page in an application.

> **See Also:**
>
> Adding a New Page to an Application

2.5.4.4 Application Home Page Navigation Bar

Learn about navigation bar on the Application home page.

A navigation bar displays in the center of the Application home page. You can use the navigation to search for pages or customize the appearance of the page.



The Application home page navigation bar contains the following controls:

- **Select columns to search** - This field contains a magnifying glass icon. Click this icon to narrow your search to only specific columns. To search all columns, select **All Columns**.
- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View icons**. Use these icons to toggle between two views:
  - **View Icons** (default) - Displays each page as a large icon identified by the page name.

    To view a page, click the page icon.
  - **View Report** - Displays each page as a line in a report. Each line includes the page number (ID), the page name, when the page was last updated and by whom, the page type, any associated group, the user interface, and lock status.

    To view a page, click the page name. Click the **Lock** icon to prevent conflicts during application development. Click the **Run** icon to run the associated page and render viewable HTML.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.

> **See Also:**
>
> - App Builder Home Page Icons and Navigation Bar
> - Locking and Unlocking a Page

2.5.4.5 Application Home Side Bar

Learn about links on the Application home page side bar.

Application Home Tasks List

A Tasks list displays on the side bar of the Application home page and contains the following links

- **Delete Application** deletes the current application. See Deleting an Application.
- **Copy Application** creates a copy of the current application. See Copying an Application from the Application Home Page.
- **Add Feature Page** launches the Feature page of the Create Page Wizard enabling you to add features to an existing application. See Adding Features to an Existing App.
- **Browse by Facets** - Links to the Search Applications page. Select a Page Type to filter the view.
- **Workflow Designer** - Links to Workflow Designer. Use the Workflow Designer to create or edit Workflow Definitions. See Managing Workflows and Tasks.

Working Copy Tasks

Click **Create Working Copy** to create a special copy of an application, make changes in the copy, and then merge those changes back into the main development application. See Creating a Working Copy to Merge, Refresh, or Compare.

Recently Edited Pages

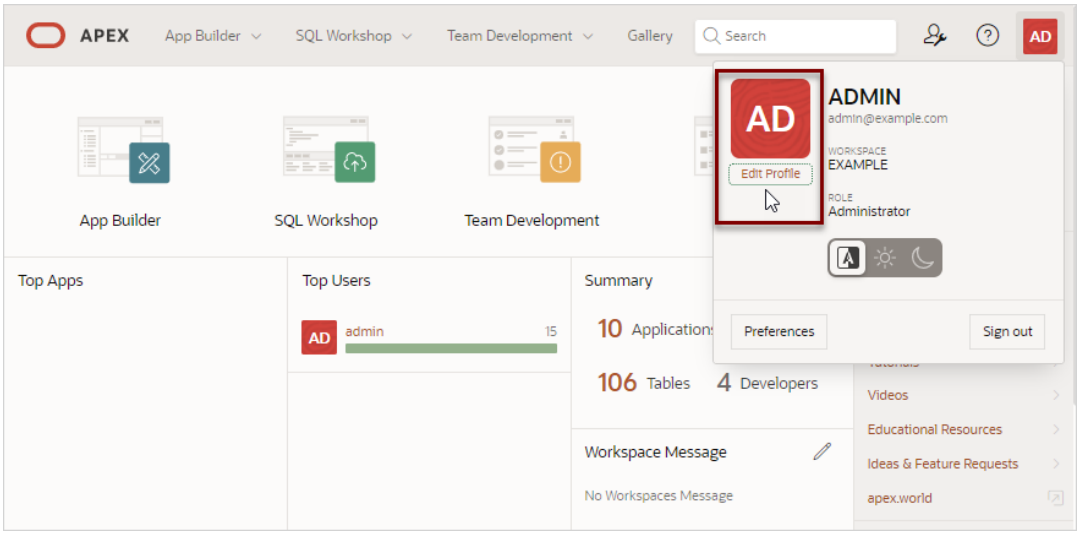
Recently Edited Pages lists links to recently edited pages within the current application. To link to specific page, click the page ID and name.

2.6 Changing Your Profile or Password

Edit your account profile to update your profile (email address, first or last name), upload a profile photo, or change your password.

To edit your account profile:

1. Click the **Account** menu in the header region.
2. Click **Edit Profile**.



The Edit Profile dialog appears.

3. Under Profile Details, edit the following:
   - Email Address - Enter the email address used for general correspondence and notifications about resetting passwords.
   - Email Address - Enter the first or given name to further identify the user. This optional field is used for informational purposes only.
   - Last Name - Enter the last or given name to further identify the user. This optional field is used for informational purposes only.
4. Under Profile Photo, add, change, or remove your photo.
5. Under Change Password, enter a new password.
6. Click **Apply Changes**.

> **Tip:**
>
> You can also reset your password and recover your workspace name from the Sign In page. See Resetting Your Password from the Sign In Page and Recovering Your Workspace Name in *Oracle APEX Installation Guide*.

2.7 Editing User Preferences and Enabling Dark Mode

Click the Account menu to enable or disable Dark Mode or edit user preferences.

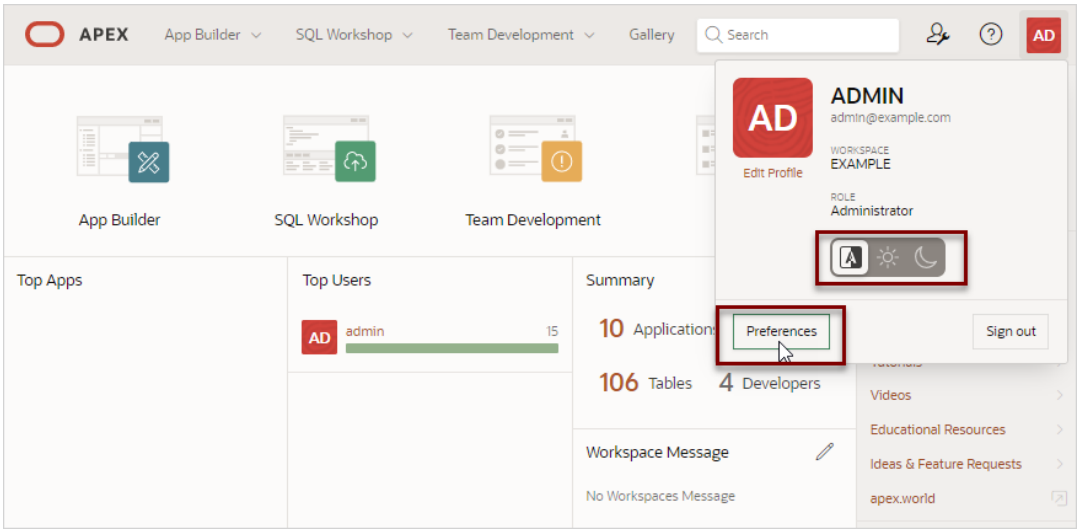
Configure the **Appearance** switch to determine if the Oracle APEX user interface displays in **Dark Mode**, **Light Mode**, or **Automatic Mode**.

Click **Preferences** to change your default schema, default date format, configure whether SQL Workshop CSV downloads are in Unicode, and control how windows display when App Builder runs applications.

To enable Dark Mode or edit preferences:

To configure the **Appearance** switch or user Preferences:

1. Click the **Account** menu in the header region.



2. Configure the **Appearance** switch to determine if the APEX user interface displays in Dark Mode or Light Mode. Options include:
   - **Automatic** detects the appropriate setting based on existing system or browser settings.
   - **Light Mode** renders the user interface using the standard color scheme.
   - **Dark Mode** renders the user interface using an inverted color scheme, applying light-colored text, icons and graphical user interface elements on a darker background.
3. To configure Preferences, click **Preferences**. In the Preferences dialog, edit the following:
   - **Default Schema** - Specifies the default schema used for data browsing, application creation, and SQL script execution. When using workspaces that have more than one schema available, this schema is the default for creating applications, performing SQL queries, and so on. This feature does not control security, only the user's preference.
   - **Default Date Format** - Specify the default Oracle date format for the user. This option controls the default date format within SQL Workshop.
   - **Unicode CSV Downloads** - Specify whether to provide SQL Workshop CSV downloads in Unicode. This does only affect SQL Workshop and not custom applications.
   - **Run application in new window** - Specify how APEX manages windows or tabs when running applications from App Builder when using Mozilla Firefox and Microsoft Internet Explorer. This setting has no effect when using other browsers, such as Google Chrome and Apple Safari.
     If set to **On**, the default, browser preferences are ignored and a new window always opens. If set to **Off**, the runtime application opens in a new window or tab based on Browser preferences. Mozilla Firefox and Microsoft Internet Explorer generally do not set focus to another tab in the same window, except when that tab is first launched.
   - **Use single window to run all applications** - Specify whether different applications reuse the same application runtime window or tab.
     If set to **On**, applications run from the App Builder reuse the window or tab used for running the previous application. If set to **Off**, applications you run display in its own window or tab.
4. Click **Apply Changes**.

2.8 Using Oracle APEX Documentation

Oracle APEX provides field-level help, a hosted online documentation library, and links to the documentation from the product. From the APEX Documentation page in the Oracle Help Center, you can browse content by publication, search content, or download the entire documentation library.

- About Oracle Help Center
  Access Oracle Help Center by clicking Documentation on the Help menu.
- Downloading the Documentation Library
  Download a ZIP file of the online documentation library.
- Redirecting the Help Menu
  Redirect the Help menu at the instance level.
- Viewing Field-Level and Page Designer Help
  Learn about field-level Help and Help in Page Designer.

2.8.1 About Oracle Help Center

Access Oracle Help Center by clicking Documentation on the Help menu.

Oracle Help Center publishes a comprehensive library of publications for each Oracle APEX release. Whether you are new to APEX or an advanced user, you can find useful information about building stunning, scalable, secure apps with Oracle APEX.

Using Oracle Help Center

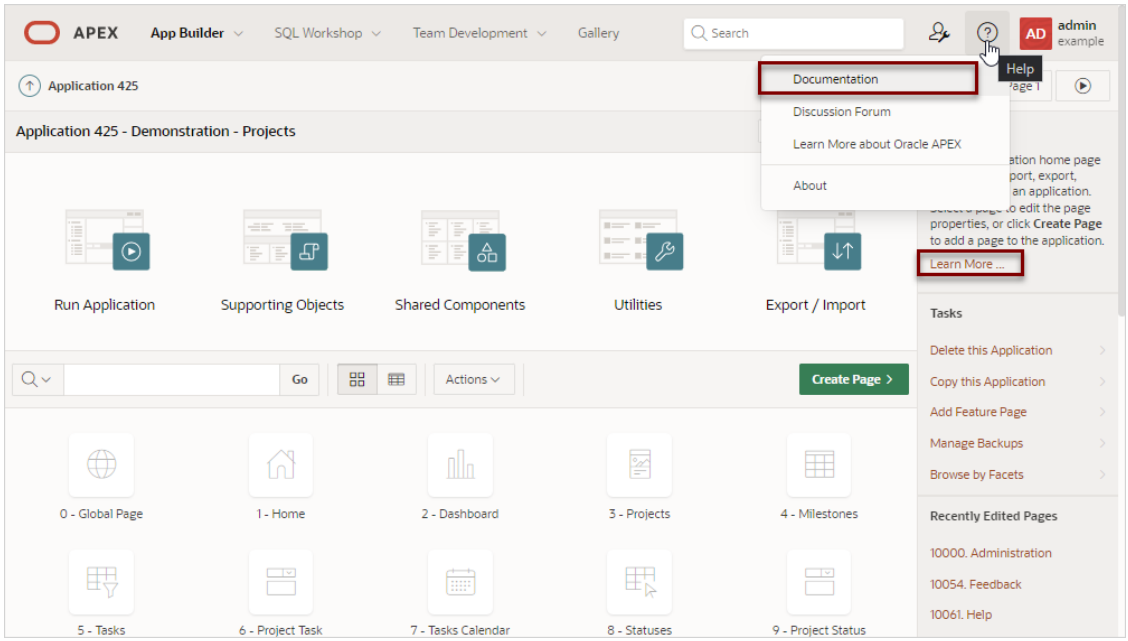
Oracle Help Center contains documentation for Oracle products, with a specific page for Oracle APEX Documentation.

When viewing a publication on Oracle Help Center, you can access the home page for the current Oracle APEX release by clicking the **Release** *number* breadcrumb. When the release page appears, you can:

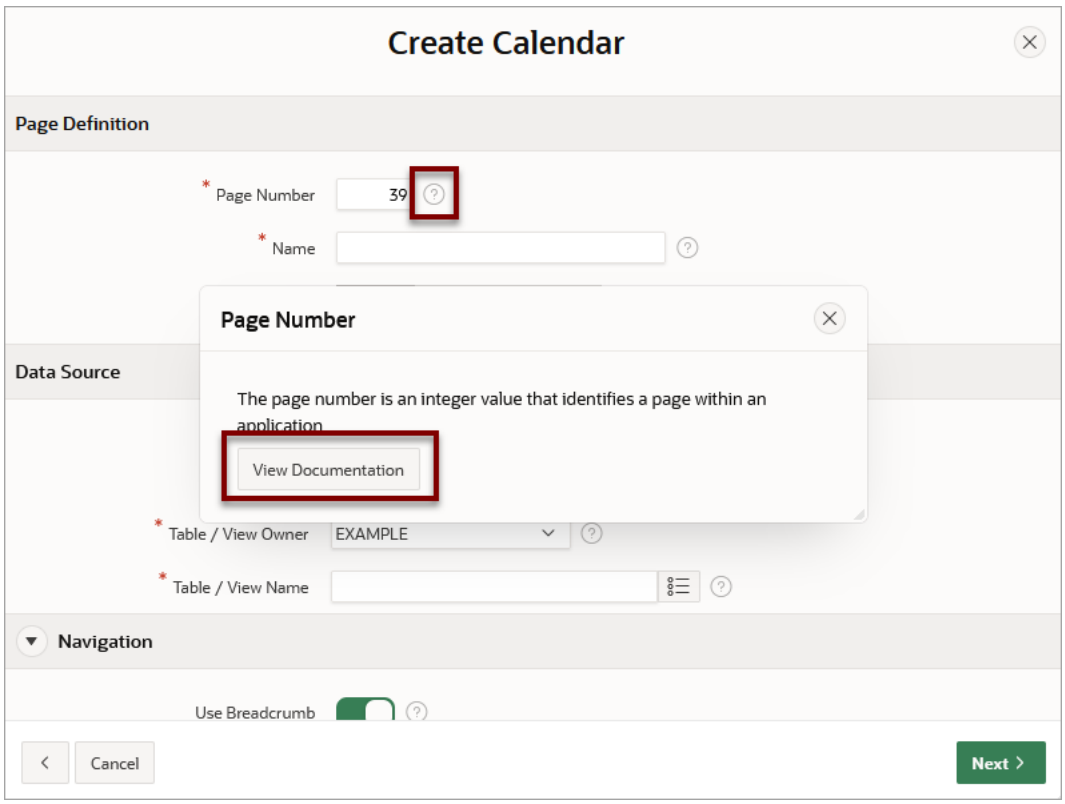
- Browse topics by selecting a subject area.
- View all publications in the library by selecting **Books** in the left navigation menu.
- Download the library locally by selecting **Download**.

Using Documentation Links in Oracle APEX

From Oracle APEX, access Oracle Help Center by clicking the Help menu and selecting **Documentation**, or clicking a **Learn More...** link in the product UI.



You can also access the documentation by clicking the item-level help, and then **View Documentation**.



Clicking a documentation link from the product redirects you to the appropriate book or section of the book.

2.8.2 Downloading the Documentation Library

Download a ZIP file of the online documentation library.

To download the online documentation library:

1. Access Oracle Help Center:
   a. Click the **Help** menu in the upper right corner. The **Help** menu icon is a question mark enclosed in a circle.
   b. Select **Documentation**.

   The documentation opens.
2. Navigate to the product release page. Click the **Release 24.2** breadcrumb.

   The Oracle APEX Release page appears.
3. Under **Books**, select **Download**.
4. Save the ZIP file locally.
5. Download the ZIP file and extract it locally.
6. Open the extracted folder and then open **Readme.html**.
7. Click **Start Here**.

2.8.3 Redirecting the Help Menu

Redirect the Help menu at the instance level.

Administrators can redirect the Help menu at the instance level in **Manage Instance**, **Instances Settings** in the Oracle APEX Administration Services application. APEX Administration Services is a separate application that enables an instance administrator to manage an entire APEX hosted instance. This functionality is not available to some customers who are using a hosted instance.

> **See Also:**
>
> Configuring the Help Menu in *Oracle APEX Administration Guide*

2.8.4 Viewing Field-Level and Page Designer Help

Learn about field-level Help and Help in Page Designer.

Most select lists, checkboxes, items, and fields within APEX include field-level Help. Page Designer also includes Help for all Property Editor attributes.

- Viewing Field-Level Help in Oracle APEX
- Viewing Help in Page Designer

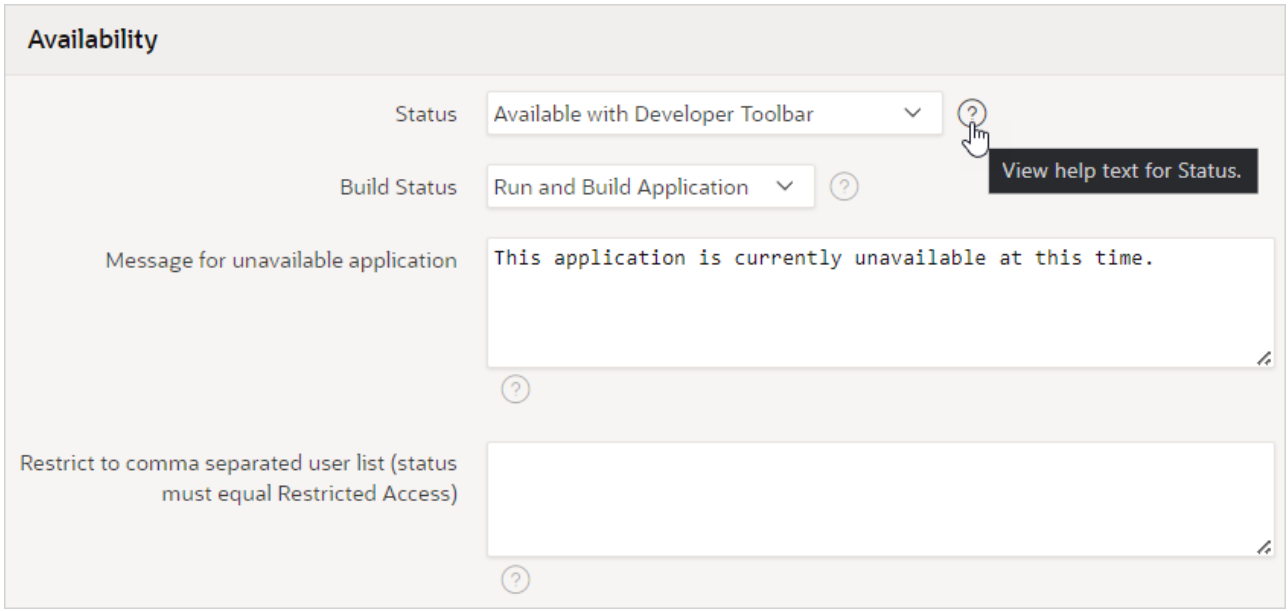
2.8.4.1 Viewing Field-Level Help in Oracle APEX

Most select lists, checkboxes, items, and fields within Oracle APEX include field-level Help.

To view field-level help:

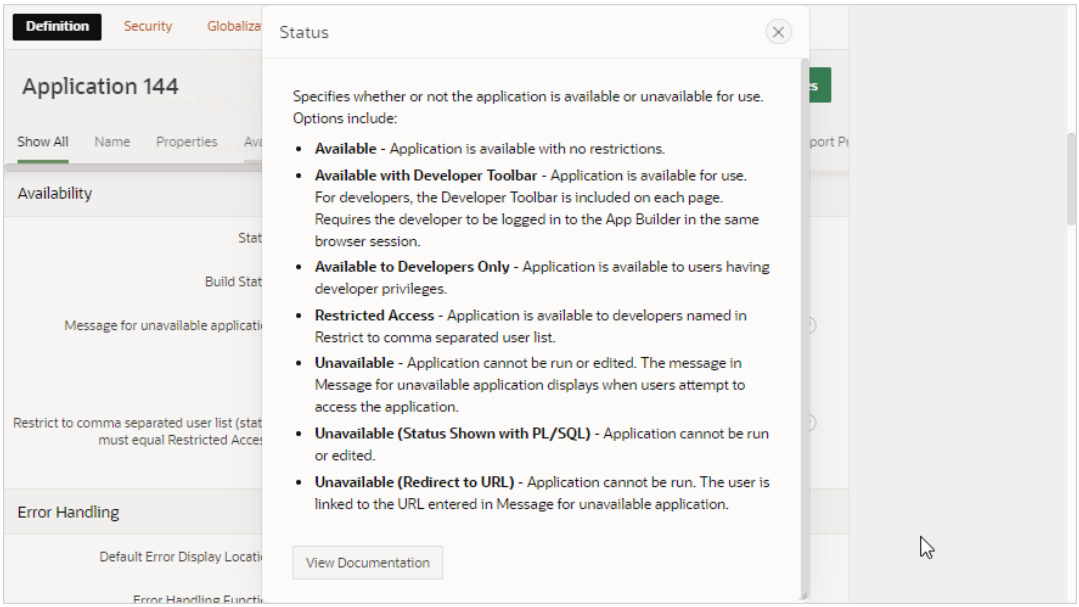
1. Navigate to the attribute you need information about.

   Most attributes in APEX include field-level Help. Attributes with field-level Help have a light gray icon that resembles a question mark (?). Hovering over the icon displays the tooltip, `View help text for....`



2. To access field-level Help, click **View help text for...** .

   A Help window appears.



> **Tip:**
>
> To access field-level help with the keyboard, focus on the correct field and then press **Alt+F1**.

3. For additional documentation, click **View Documentation** at the bottom of the Help window.

> **See Also:**
>
> About Oracle Help Center

2.8.4.2 Viewing Help in Page Designer

Page Designer includes Help for all attributes in the Property Editor.

To view Help in Page Designer:

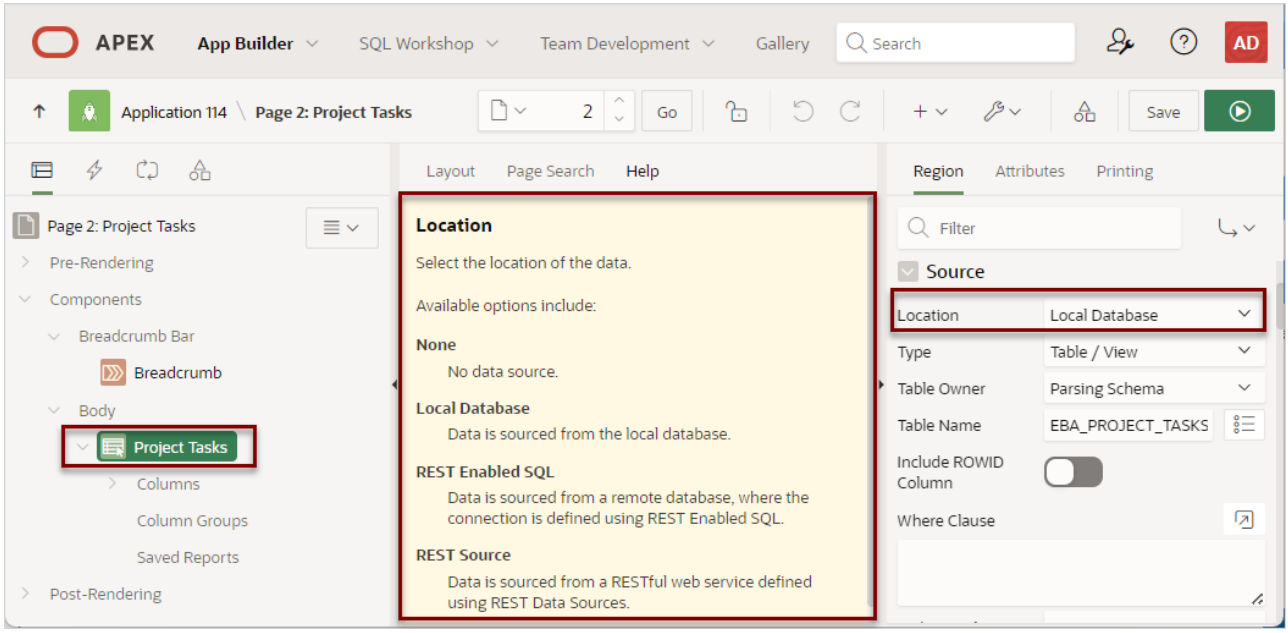
1. Access Page Designer.

   Page Designer appears.

2. Select a component or control in either the left pane or the central pane.

   As you change focus in either view, the Property Editor automatically updates to reflect the currently selected component.

3. In the Property Editor, select an attribute.
4. Click the **Help** tab in the central pane.



Once you activate the Help pane, the content that displays changes every time you select another attribute.

> **Tip:**
>
> To view an online summary that describes just the Page Designer window, click the **Help** menu at the top of the page and select **Getting Started in Page Designer**.

> **See Also:**
>
> Viewing a Page in Page Designer

2.9 Using the App Gallery

Use the App Gallery to install Sample Apps, Starter Apps, Custom Apps, and Utility Apps.

- About Sample Apps, Starter Apps, Custom Apps, and Utility Apps
  Sample Apps and **Starter Apps** are available with every Oracle APEX workspace. **Custom Apps** are only available if your instance administrator has enabled them. **Utility Apps** are ready-to-use apps that simplify common APEX application development tasks.
- About the Oracle APEX GitHub Repository
  The Oracle APEX GitHub repository contains Starter Apps, Sample Apps, sample code, and plug-ins you can use in your workspace.
- Installing Apps from the Gallery
  Install Sample and Starter Apps directly from the Gallery by clicking **Install**.
- Uninstalling Apps from the Gallery
  Uninstall Sample and Starter Apps directly from the Gallery.

> **See Also:**
>
> Configuring the Application Gallery in *Oracle APEX Administration Guide*

2.9.1 About Sample Apps, Starter Apps, Custom Apps, and Utility Apps

Sample Apps and **Starter Apps** are available with every Oracle APEX workspace. **Custom Apps** are only available if your instance administrator has enabled them. **Utility Apps** are ready-to-use apps that simplify common APEX application development tasks.

You can access the Gallery from the top menu or by typing `Gallery` in the Global Search field (also known as Spotlight Search). The Gallery includes four types of applications:

- **Sample Apps** - Built with sample data that highlight specific functionality. You can use them as a developer guide for how particular features work.
- **Starter Apps** - Fully functional apps that provide standalone solutions for simple requirements. You can use them as is or extend them with custom functionality.

- **Custom Apps** - Specific to your workspace. **Custom Apps** are only available if your instance administrator has enabled them.
- **Utility Apps** - Ready-to-use applications that simplify common APEX application development tasks. The Utility Apps heading only displays in APEX if at least one utility application is available.

You can install Sample Apps, Starter Apps, Custom Apps, and Utility Apps directly from the Gallery. You can also download Sample Apps and Starter Apps from GitHub to import into your workspace.

> **See Also:**
>
> - Importing an Application Export
> - Configuring the Gallery in *Oracle APEX Administration Guide*

2.9.2 About the Oracle APEX GitHub Repository

The Oracle APEX GitHub repository contains Starter Apps, Sample Apps, sample code, and plug-ins you can use in your workspace.

- Navigating the GitHub Repository
  Use the GitHub repository to access documentation, license information (if applicable), and installation files for Sample and Starter Apps.

2.9.2.1 Navigating the GitHub Repository

Use the GitHub repository to access documentation, license information (if applicable), and installation files for Sample and Starter Apps.

The Oracle APEX GitHub repository can be accessed at https://github.com/oracle/apex. Under **Oracle APEX on GitHub**, click a branch to view. The repository contains information you can use to customize apps:

- **docs** contains the installation source file.
- **plugins** contains examples for APEX plug-ins.
- **sample-apps** contains the sample apps found in the Gallery.
- **sample-code** contains code examples for APEX functionality.
- **starter-apps** contains the starter apps found in the Gallery.

To access GitHub from an APEX workspace:

1. Navigate to the Workspace home page.
2. Click the **Gallery** icon.

   The **Gallery** page opens, displaying cards for **Sample Apps**. **Starter Apps**, and **Custom Apps**.
3. Choose either **Sample Apps** or **Starter Apps**.

   Each app displays as a card with a description.
4. To access GitHub:

a. Click the **More Info** button ( ) which resembles a vertical ellipses.

b. From the menu, select **More Info** again.

The GitHub directory for the app opens.

5. All files associated with the app including the README, licenses (if applicable), and installation files are available for viewing.

   The README contains instructions for installing the app.

6. Download the files you need.

You can now install the app.

2.9.3 Installing Apps from the Gallery

Install Sample and Starter Apps directly from the Gallery by clicking **Install**.

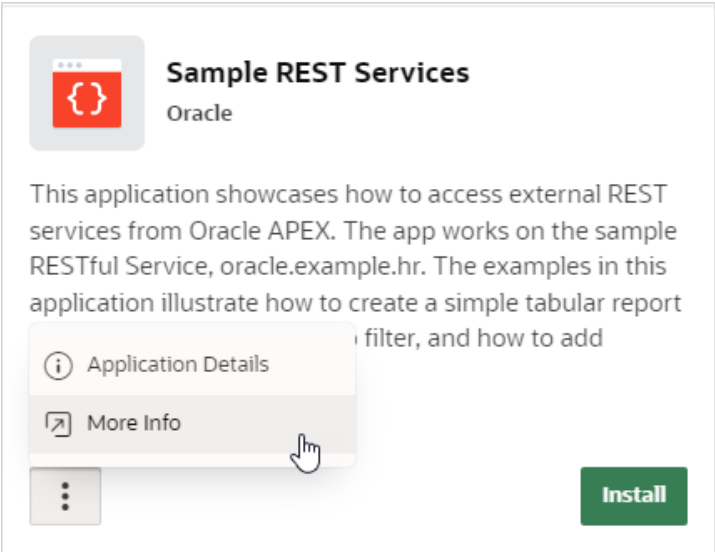
Sample and Starter Apps can be installed from the Gallery. Custom Apps can be installed from the Gallery if they have been enabled by your instance administrator. Utility Apps can be installed from the Gallery if available.

To install apps from the Gallery:

1. Navigate to the Workspace home page.
2. Click the **Gallery** icon.

   The **Gallery** page opens displaying cards. Each card displays a description and buttons. The buttons that display differ depending upon whether the app is installed.

3. The card for an uninstalled app resembles the following example:

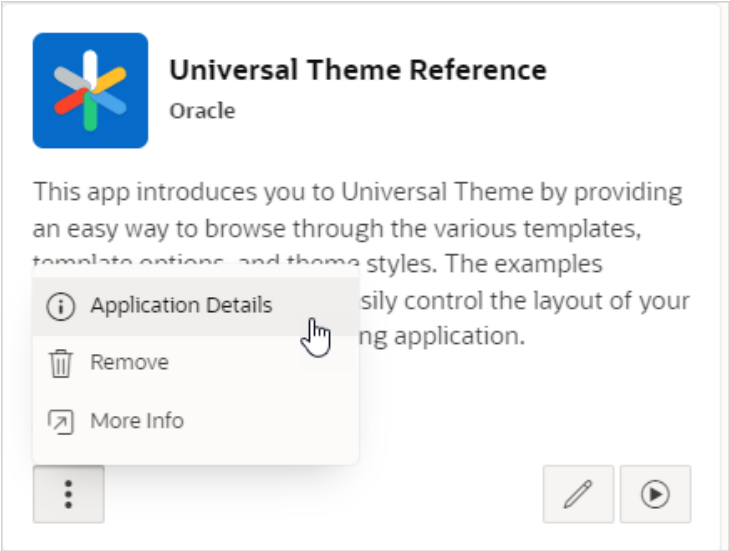


Availlable controls include:

- **More Info** - **More Info** ( ) features an image of a vertical ellipses. Options include:

  - **Application Details** - Displays a dialog with additional information about the app. Expand **Advanced Settings** to install the app but configure the application ID and parsing schema.
    For **Install As Application**, specify whether the Application ID will be assigned automatically (default) or manually. For **Parsing Schema**, select the parsing schema. Click **Install Application** to complete the installation
  - **More Info** - Click **More Info** to open the GitHub repository in a new browser tab. See About the Oracle APEX GitHub Repository.
- **Install** - Click **Install** to perform a quick install of the app.

4. The card for an installed app resembles the following example:



Available controls include:

- **More Info** - **More Info** ( ) features an image of a vertical ellipses. Options include:
  - **Application Details** - Displays a dialog with additional information about the app. Click **Remove Application** to remove the application.
  - **Remove** - Removes the application.
  - **More Info** - Click **More Info** to open the GitHub repository in a new browser tab. See About the Oracle APEX GitHub Repository.
- **Edit** - Click **Edit** to view the application on the Application home page.
- **Run** - Click **Run** to view a rendered version of the application.

See Also:

- Understanding the Application Home Page
- Running an Application or Page
- Importing an Application Export
- Editing Pages in Page Designer

2.9.4 Uninstalling Apps from the Gallery

Uninstall Sample and Starter Apps directly from the Gallery.

To uninstall Sample and Starter Apps directly from the Gallery, click **More Info** ( ) and select **Remove**.

Only apps that were installed from the Gallery can be uninstalled using the procedure described in this topic. Apps downloaded and imported from GitHub will not display a Remove icon and must be deleted manually as described in Deleting an Application.

Note:

Any customizations that were made to an app will be lost if the app is uninstalled. These customizations will not be restored if the app is reinstalled. To preserve customizations, export the application and then import the export file back into your workspace. See Exporting and Importing from App Builder.

To uninstall apps from the Gallery:

1. Navigate to the Workspace home page.
2. Click the **Gallery** icon.

   The **Gallery** page opens.
3. Find the card for the app to be removed.
4. Click **More Info** ( ) and select **Remove**.
5. On the confirmation screen, click **Remove** again.

The app is uninstalled. It can be reinstalled if needed.

See Also:

Deleting an Application

3
App Builder Concepts

To efficiently use App Builder, developers must understand some key concepts including managing user interface design, viewing rendered application pages, understanding page processing and rendering, managing session state, understanding URL syntax, managing session state values, and using built-in substitution strings.

- Understanding Applications
  An Oracle APEX application is an HTML interface that exists on top of database objects such as tables or procedures.
- Running an Application or Page
  To view a rendered version of your application, you run or submit it to the Oracle APEX engine. The APEX engine dynamically renders and processes pages based on data stored in database tables.
- How Does Page Processing and Page Rendering Work?
  Learn how the Oracle APEX engine renders and processes pages, when APEX issues implicit commits, how conditional rendering and processing works, and how to verify user identity or control access to controls and components.
- About Enabling Support for Bookmarks
  Developers can assist users who bookmark application pages by either using zero as the session ID, or enabling **Rejoin Sessions**.
- How Oracle APEX Uses Templates
  Learn how Oracle APEX uses templates to construct the appearance of application pages.
- Understanding Session State Management
  Oracle APEX transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.
- Managing Session State
  View session state and manage session state overrides from the Runtime Developer toolbar which displays at the bottom of any editable page at runtime.
- Understanding URL Syntax
  Oracle APEX applications support two types of URL syntax: **Friendly URL Syntax** and **f?p Syntax**.
- Managing Session State Values
  Manage session state to store and retrieve values for a user as the user navigates between different application pages.
- Using Substitution Strings
  Use substitution strings within templates, region source, or any attribute to pass information and replace a character string with another value.
- Using Template Directives
  Use template directives to control how attributes that support substitution strings are processed.

> **See Also:**
>
> Understanding App Builder

3.1 Understanding Applications

An Oracle APEX application is an HTML interface that exists on top of database objects such as tables or procedures.

About Applications

An application is a collection of pages linked together using navigation menus, tabs, buttons, or hypertext links. Application pages share a common session state and authentication.

To create an application, a developer runs the Create Application Wizard to declaratively assemble pages and navigation. Individual pages are organized using containers called regions. Regions can contain text, custom PL/SQL, reports, charts, maps, calendars, web service content, or forms. Forms are made up of fields (called items) which can be selected from the multitude of built-in types (such as text fields, text areas, radio groups, select lists, checkboxes, date pickers, and popup list of values).

Developers can also create their own custom item types using plug-ins. Session state (or application context) is transparently managed and the user interface presentation is separated from the application logic enabling developers to manage the look and feel of an application by simply selecting a different theme.

About Application Pages

A page is the basic building block of an application. Every application consists of one or multiple pages. Each page can have buttons and fields (called **items**) which are grouped into containers called **regions**. Pages can include application logic (or processes). You can branch from one page to the next using conditional navigation; perform calculations (called **computations**); perform validations (such as edit checks); and display reports, calendars, and charts. You view and edit a page in Page Designer.

About Data Sources and Source Types

You can edit page components in Page Designer. The Source group in Page Designer prompts you to specify the location of the data source. The Source, Location attribute determines where data is sourced from. Possible Locations include:

- **Local Database** - Data is sourced from the local database.
- **REST Enabled SQL** - Data is sourced from a remote database, using REST Enabled SQL.
- **REST Data Source** - Data is sourced from an external REST API.
- **JSON Source** - Data is sourced from a JSON Collection Table or table with JSON columns in the local or a remote database.
- **Duality View** - Data is sourced from a Duality View in either a local or a remote database.

> **Note:**
>
> Both Duality Views and JSON Collection Tables (as a part of a JSON Source) are only available in Oracle Database 23ai or later.

The Source, Type attribute determines how the data is retrieved. Available Source Types include:

- **Table / View** - The data is retrieved based on a table or view. All columns of that table or view are selected. Provide the name of the table or view to use.
- **SQL Query** - The data is retrieved based on a SQL Query. Provide the SQL Query to be executed.
- **Function Body returning SQL Query** - The data is retrieved based on the SQL Query returned by executing a Function Body. Provide a PL/SQL or JavaScript (MLE) block returning a SQL Query. APEX will then execute that SQL query.
- **Property Graph** - Available for local database on Oracle Database 23ai or later. The data is retrieved based on a Property Graph.

> **See Also:**
>
> - Creating Applications
> - Editing Pages in Page Designer
> - Managing Application Data
> - Getting Started with Oracle Property Graphs in *Oracle Database Graph Developer's Guide for Property Graph*

3.2 Running an Application or Page

To view a rendered version of your application, you run or submit it to the Oracle APEX engine. The APEX engine dynamically renders and processes pages based on data stored in database tables.

- Run Application and Run Page Buttons
  Learn how to run pages individually, or run an entire application.
- How Your Browser Impacts the Way Applications Run
  Learn about what determines if a running application displays in a new tab or new browser window.
- Running an Application from the App Builder Home Page
  Run an application from the App Builder home page.
- Running an Application from the Application Home Page
  Run an application from the Application home page.
- Running a Page on the Application Home Page
  Run an application page from the Application home page.

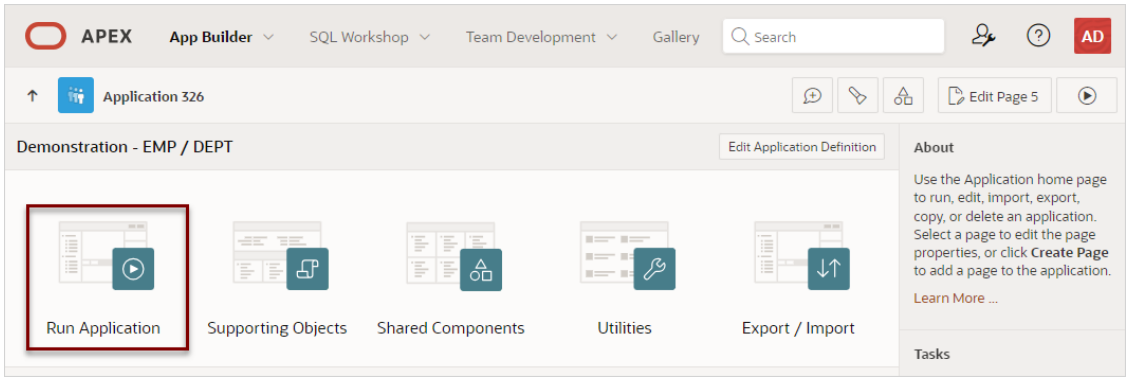
> **See Also:**
>
> Running a Page from Page Designer

3.2.1 Run Application and Run Page Buttons

Learn how to run pages individually, or run an entire application.

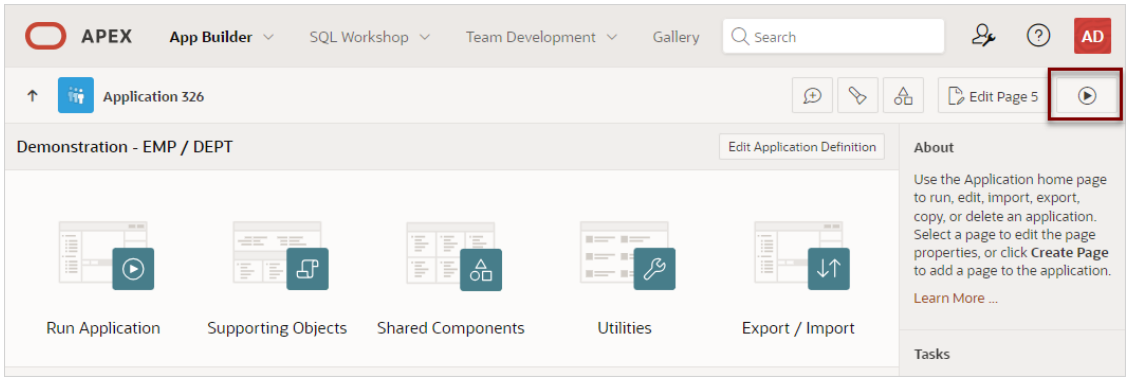
Run Application Button

The **Run Application** button displays on the Application home page and features with a large run (or play) button. To run the application from the application home page, click **Run Application**.



Run Page Button

The Run Page button resembles a small, black play button and displays in the upper right corner of many pages within App Builder. Clicking the **Run Page** button runs the current page or the last selected page.



3.2.2 How Your Browser Impacts the Way Applications Run

Learn about what determines if a running application displays in a new tab or new browser window.

When you run an application, the application displays in a new window. Whether that new window is a new tab or new browser windows depends upon how you have configured your web browser. If you are using Google Chrome or Apple Safari, the running application displays in a new tab by default. If you are using Microsoft Edge or Mozilla Firefox, the application displays in a new browser window by default. You can further control how Oracle APEX manages tabs and windows when running an application by editing your account preferences.

> **See Also:**
>
> Editing User Preferences and Enabling Dark Mode

3.2.3 Running an Application from the App Builder Home Page

Run an application from the App Builder home page.

To run an application from the App Builder home page:

1. On the Workspace home page, click **App Builder**.
2. You can customize the App Builder home page by selecting the following icons in the navigation bar:
   - **View Icons** (default)
   - **View Report**
3. To run the application in Icon View:
   a. On the navigation bar, click the **View Icons** icon:

      Each application displays as a large icon identified by the application name.

   b. Click **Run Application** at the top of the page.
4. To run the application in Report View:
   a. On the navigation bar, click the **View Report** icon:

      Each application displays as a line in a report. Each line includes the application ID, the application name, when the application was last updated, the page count, and who last updated the application.

   b. Click **Run** in the Run column.

> **See Also:**
>
> How Your Browser Impacts the Way Applications Run

3.2.4 Running an Application from the Application Home Page

Run an application from the Application home page.

To run an application from the Application home page:

1. On the Workspace home page, click **App Builder**.

   The App Builder home page appears.

2. Select an application.
3. Click **Run Application** in the middle of page.

   The application displays in a new window.

> **See Also:**
>
> How Your Browser Impacts the Way Applications Run

3.2.5 Running a Page on the Application Home Page

Run an application page from the Application home page.

To run a page from the Pages list:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. Click the **View Report** icon.

   Selecting **View Report** displays each page as a line in a report. Each line includes the page number, the page name, when the page was last updated, who last updated the page, the page type, the selected user interface, a lock icon, and a Run button.
4. Locate the page you want to run.
5. Click **Run** in the Actions column.

   The application displays in a new window.

> **See Also:**
>
> Locking and Unlocking a Page and How Your Browser Impacts the Way Applications Run

3.3 How Does Page Processing and Page Rendering Work?

Learn how the Oracle APEX engine renders and processes pages, when APEX issues implicit commits, how conditional rendering and processing works, and how to verify user identity or control access to controls and components.

- How the APEX Engine Renders and Processes Pages
  The Oracle APEX engine dynamically renders and processes pages based on data stored in Oracle database tables.
- About Implicit Commit Points
  Learn about when Oracle APEX issues implicit commits.
- Understanding Conditional Rendering and Processing
  Use conditions in an application to control how pages and page components display and when processes, computations, and validations execute.

- About Verifying User Identity
  Authentication is the process of establishing users' identities before they can access an application.
- About Controlling Access to Controls and Components
  Learn about controlling access to controls and components.

3.3.1 How the APEX Engine Renders and Processes Pages

The Oracle APEX engine dynamically renders and processes pages based on data stored in Oracle database tables.

To view a rendered version of your application, you request it from the APEX engine with a URL. When you run an application, the APEX engine relies on two processes:

- **Show Page** - Show Page is a page rendering process that assembles all the page attributes (including regions, items, and buttons) into a viewable HTML page. When you request a page using a URL, the engine is running **Show Page**.
- **Accept Page** - Accept Page performs forms page processing, including computations, validations, processes, and branching. When you submit a page, the APEX engine is running **Accept Page** or performing page processing during which it saves the submitted values in the session cache and then performs any computations, validations, or processes.

3.3.2 About Implicit Commit Points

Learn about when Oracle APEX issues implicit commits.

Oracle APEX issues implicit commits at the following points:

- On load, after a page finishes rendering
- On submit, before branching to another page
- On submit, if one or more validations fail, before re-rendering the page
- After a computation
- After changing the value of an item, for example after a PL/SQL process that modifies a bind variable value or when `APEX_UTIL.SET_SESSION_STATE` is called.
- When `APEX_MAIL.PUSH_QUEUE` is called

> **See Also:**
>
> SET_SESSION_STATE Procedure and PUSH_QUEUE Procedure in *Oracle APEX API Reference*

3.3.3 Understanding Conditional Rendering and Processing

Use conditions in an application to control how pages and page components display and when processes, computations, and validations execute.

A condition is a small unit of logic that helps you control the display of regions, items, buttons, and tabs and the execution of processes, computations, and validations. For example, when you apply a condition to a button, the rendering engine evaluates the condition during the

rendering (or Show Page) process. Whether the condition passes or fails determines if the button displays.

You specify a condition by selecting a condition type. You can select a condition type when you first create the control or component, or later when editing the control or component.

Depending upon the type of condition you select, you enter the appropriate values in the fields provided. The condition evaluates to true or false based on the values you enter. Click the select list to view a complete list of all available conditions for a given component or control.

> **See Also:**
>
> Available Conditions for a detailed listing of available condition types

Example 3-1 Current Page is NOT in comma delimited list

`Current Page is NOT in comma delimited list` evaluates to true if the current page number is contained within the comma-delimited list of pages provided. For example:

```
3,100,203
```

If the current page is 100, then this condition evaluates to true and the condition passes.

Example 3-2 PLSQL Expression

Use `PLSQL Expression` to specify an expression in valid PL/SQL syntax that evaluates to true or false. For example:

```
NVL(:MY_ITEM,'NO') = 'YES'
```

If the value of `:MY_ITEM` is YES, as in the previous example, then the condition evaluates as true. If the value of :MY_ITEM is anything other than YES, then the condition evaluates as false.

Example 3-3 Rows returned

`Rows returned` expressed as a SQL query. If the query returns one or more rows, then the condition evaluates as true. For example:

```
SELECT 1 FROM employees WHERE department_id = :P101_DEPTNO
```

This example references item `P101_DEPTNO` as a bind variable. You can use bind variables within application processes and SQL query regions to reference items from session state. If one or more employees are in the department identified by the value of `P101_DEPTNO`, then the condition evaluates as true.

> **See Also:**
>
> Referencing Session State Using Bind Variable Syntax

3.3.4 About Verifying User Identity

Authentication is the process of establishing users' identities before they can access an application.

Authentication may require a user to enter a user name and password, or may involve the use of a digital certificate or a secure key.

Oracle APEX supports modular authentication, making it easy to switch authentication methods when needed. You can establish a user's identity by selecting from several built-in authentication methods, or by using a wizard to create your own custom authentication approach.

> **See Also:**
>
> Establishing User Identity Through Authentication

3.3.5 About Controlling Access to Controls and Components

Learn about controlling access to controls and components.

While conditions control the rendering and processing of specific controls or components on a page, authorization schemes control user access. Authorization is a broad term for controlling access to resources based on user privileges.

Authorization schemes use the identities established by authentication to grant privileges on applications and objects within them. You can specify an authorization scheme for an entire application, a page, or a specific page control such as a region, item, or button. For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bar entries a user sees.

> **See Also:**
>
> Providing Security Through Authorization

3.4 About Enabling Support for Bookmarks

Developers can assist users who bookmark application pages by either using zero as the session ID, or enabling **Rejoin Sessions**.

In previous releases, if the pages within an application were public and did not require authentication, you could assist users in bookmarking pages by using zero as the session ID. Although this approach is still supported, Oracle APEX now includes a new feature called **Rejoin Sessions** that enables developers to control this type of support at the page, application, or instance-level.

> **See Also:**
>
> About Rejoin Sessions

3.5 How Oracle APEX Uses Templates

Learn how Oracle APEX uses templates to construct the appearance of application pages.

The APEX engine constructs the appearance of each page in an application using templates. Templates define how pages, page controls, and page components display. This topic describes the various ways you can use templates to control and refine the appearance of application pages.

- Basic Templates
  Templates in Oracle APEX support the use of substitution strings to pass information and replace any character string with another value.
- Theme Templates
  Learn about the Universal Theme and how to control application layout using theme styles, Theme Roller, and template options.
- Accessing the Universal Theme Reference App
  Run the *Universal Theme Reference* app to learn about features and best practices when using the Universal Theme.
- Template Components
  Use Template Components to create reusable UI components that work in Page Designer like any other plug-in.
- Template Directives
  Template directives are special tokens in a template that control how the text around them is processed.

3.5.1 Basic Templates

Templates in Oracle APEX support the use of substitution strings to pass information and replace any character string with another value.

- Template Substitution Syntax
  Use substitution strings within templates to reference component values.
- Escape Filters
  Learn how to escape special characters in the substitution value.
- Built-in Substitution Strings
  Learn about built-in substitution strings.
- Substitutions in Text Messages (Legacy)
  Learn about using substitutions in text messages.
- JavaScript in Templates
  Learn about using JavaScript in templates.

3.5.1.1 Template Substitution Syntax

Use substitution strings within templates to reference component values.

Pass information and replace any character string with another value using template substitution syntax. To view the most current list supported of substitution strings for a given template, view the template definition and find the Substitution Strings report. See Viewing Templates on the Templates Page.

The following table summarizes two approaches to template substitution syntax.

Type	Description	Example	Result
Placeholder Substitution	Use **Placeholder substitution** string syntax within a template to reference component values. Placeholder substitution strings available within a template are denoted by the number symbol (#).	`Hello #NAME#`	`Hello APEX Community`
Session State Substitution	Reference page or application items using **Session State Substitution** string syntax (also called *`&ITEM.`* syntax): • Reference the page or application item in all capital letters • Precede the item name with an ampersand (&) • Append a period (.) at the end	`Hello &NAME.`	`Hello APEX Community`

Tip:

Use double quotation marks for special characters or case sensitivity. For example:

```
#"Name"#
```

```
&"Name".
```

See Also:

Using Substitution Strings

3.5.1.2 Escape Filters

Learn how to escape special characters in the substitution value.

Escape special characters in the substitution value by appending an exclamation mark (!) followed by a predefined filter name to a page or application item name, report column, or

other substitution string. Output escaping is an important security technique to avoid Cross Site Scripting (XSS) attacks in the browser.

Escape Filter	Description	Classic Syntax	Modern Syntax
`ATTR`	`ATTR` escapes reserved characters in a HTML attribute context.	`#NAME!ATTR#`	`&NAME!ATTR.`
`HTML`	`HTML` escapes reserved HTML characters.	`#NAME!HTML#`	`&NAME!HTML.`
`JS`	`JS` escapes reserved characters in a JavaScript context.	`#NAME!JS#`	`&NAME!JS.`
`STRIPHTML`	`STRIPHTML` removes HTML tags from the output and escapes reserved HTML characters.	`#NAME!STRIPHTML#`	`&NAME!STRIPHTML.`
`RAW`	`RAW` preserves the original item value and does not escape characters.	`#NAME!RAW#`	`&NAME!RAW.`

See Also:

Controlling Output Escaping in Substitution Strings

3.5.1.3 Built-in Substitution Strings

Learn about built-in substitution strings.

App Builder supports many built-in substitution strings. You can reference these substitution strings to achieve specific types of functionality. See Using Built-in Substitution Strings to view a complete list of all availble built-in substitution strings.

3.5.1.4 Substitutions in Text Messages (Legacy)

Learn about using substitutions in text messages.

Note:

Application-defined or system-defined text messages described in this topic have been designated as legacy functionality. If your application's Compatibility Mode is 24.2 or later, Oracle recommends wrapping the message name in curly braces ({}) as described in Substitutions in Text Messages (24.2 or later).

Although Oracle still supports legacy application components, Oracle does not recommend using them.

To access application-defined or system-defined text messages, use the built-in substitution `APP_TEXT$Message_Name`, `APP_TEXT$Message_Name$Lang`, for example:

```
&APP_TEXT$MY_MESSAGE!HTML.
```

```
&APP_TEXT$MY_MESSAGE$NL!HTML.
```

> **See Also:**
>
> APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang

3.5.1.5 JavaScript in Templates

Learn about using JavaScript in templates.

You can also use JavaScript API `apex.util.applyTemplate` to evaluate templates on the client side.

> **See Also:**
>
> applyTemplate in *Oracle APEX JavaScript API Reference*.

3.5.2 Theme Templates

Learn about the Universal Theme and how to control application layout using theme styles, Theme Roller, and template options.

A **theme** is a collection of templates that enables developers to define the layout and style of an entire application. When you create a new application, the Create Application Wizard uses the Universal Theme - 42 (Universal Theme). The Universal Theme features a responsive design, versatile UI components, and easy customization.

You can further modify a theme using:

- **Theme Styles** - A theme style is a CSS that is added to the base CSS. Using the Theme Roller utility, you can change the appearance of an application. Universal Theme - 42 includes theme styles.
- **Theme Roller** - Theme Roller is a live CSS editor available on the Customize menu on the Runtime Developer toolbar. Use Theme Roller to quickly change the theme style, colors, rounded corners and other application attributes.
- **Template Options** - Use template options to declaratively apply CSS modifiers to the templates you have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons.

> **See Also:**
>
> - Understanding the Universal Theme
> - Using Theme Styles
> - Using Theme Roller
> - Understanding Template Options and Using Template Options

3.5.3 Accessing the Universal Theme Reference App

Run the *Universal Theme Reference* app to learn about features and best practices when using the Universal Theme.

The *Universal Theme Reference* app offers an introduction to the Universal Theme and provides details and examples of design features, template components, template icons, and reference information on a variety of topics such as migrating to the latest Universal Theme, building buttons, utilizing color, using CSS variables, and using template directives.

To access the *Universal Theme Reference* app:

1. You can access the *Universal Theme Reference* in two ways:
   - Access a hosted version. Go to `https://apex.oracle.com/ut`.
   - Install the app in your workspace from the Gallery.
2. To install the *Universal Theme Reference* app in your workspace from the Gallery:
   a. Navigate to the Workspace home page.
   b. Click the **Gallery** icon.

      The Gallery page opens, displaying cards for Sample Apps, Starter Apps, and Custom Apps.
   c. Click **Sample Apps**.
   d. Find **Universal Theme Reference** and click **Install**.
   e. Click **Run** to view a rendered version of the application.

      The *Universal Theme Reference* appears.
3. Click the **Main Navigation** menu to explore the Universal Theme. Options include:
   - **Getting Started**
   - **Design** - Highlights key features of the Universal Theme including colors, navigation, grid layout, page templates, theme styles, and mobile patterns.
   - **Components** - Describes and provides examples of available components such as regions, lists, and buttons.
   - **Icons** - Describes and provides examples of available icons.
   - **Reference** - Provides reference information and best practices. Available topics include migrating to the latest Universal Theme, building buttons, utilizing color, using CSS variables, and using template directives to control how substitution strings and content is processed.

See Also:

- Using the App Gallery
- Using Themes and Theme Styles
- Understanding the Universal Theme

3.5.4 Template Components

Use Template Components to create reusable UI components that work in Page Designer like any other plug-in.

Template Component are a new plug-in type for building re-usable UI components with support for actions, menus, and custom attributes within Page Designer. These components can be rendered standalone as regions displaying one or multiple rows of data, and can also be within reports as partials for a column.

Attributes defined in the plug-in are editable in Page Designer for any column or region using the plug-in type. With the introduction of Action Positions and Action Templates for plug-ins, you can define Actions and Menu buttons with row-level conditions for Template Components.

Template Components enable developers to add new region types to APEX that are not based on PL/SQL. New regions based on template components can:

- Display instantly as multiple or single rows
- Remember pagination
- Use faceted search and smart filters
- Support the external Order By
- Support single and multiple row selection as well as keyboard navigation for navigation and selection
- Support a grouping of report data

A Template Components plug-in Type contains an HTML template and includes support for template directives. Template directives are special tokens in a template that control how the text around them is processed.

> **Tip:**
>
> To learn more about Template Component Type plug-ins, see About Template Component Type Plug-ins

3.5.5 Template Directives

Template directives are special tokens in a template that control how the text around them is processed.

Use template directives to control how attributes that support substitution strings are processed. Template directives are only supported by Template Components, specific attributes of email templates, and cards, interactive grid, classic report, interactive report regions. Template directives are processed as part of client or server side substitutions. If an attribute supports template directives, Page Designer Help includes the text `Supports Server-side Template Directives` or `Supports Client-side Template Directives`.

> **See Also:**
>
> Using Template Directives

3.6 Understanding Session State Management

Oracle APEX transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.

- What is Session State?
  Session state enables developers to store and retrieve values for a user as the user navigates between different application pages.
- About Session IDs
  A session ID is a unique number assigned a specific user for the duration of that user's visit (or session).
- What Is a Session?
  A session is a logical construct that establishes persistence (or stateful behavior) across page views.

3.6.1 What is Session State?

Session state enables developers to store and retrieve values for a user as the user navigates between different application pages.

Hypertext Transfer Protocol (HTTP), the protocol over which HTML pages are most often delivered, is a stateless protocol. A web browser is only connected to the server for as long as it takes to download a complete page. Each page request is treated by the server as an independent event, unrelated to any page requests that happened previously or that may occur in the future. To access form values entered on one page on a subsequent page, the values must be stored as session state. Oracle APEX transparently maintains session state and provides developers with the ability to get and set session state values from any page in the application.

3.6.2 About Session IDs

A session ID is a unique number assigned a specific user for the duration of that user's visit (or session).

The Oracle APEX engine establishes the identity of the user for each page request and the session ID to fetch session state from the database. The most visible location of the session ID is in the URL for a page request:

- The third parameter in the f?p URL Syntax:

  ```
  https://apex.example.com/ords/f?p=392:3:13766599855150
  ```

- The `session` parameter in the Friendly URL Syntax::

  ```
  https://apex.example.com/ords/mycompany/r/hr-app/update-employees?
  session=13766599855150
  ```

In both examples, the session ID is `13766599855150`.

Another visible location is in the page's HTML POST data and indirectly in the contents of a session cookie. This cookie is sent by the APEX engine during authentication and is maintained for the life of the application (or browser) session.

APEX assigns new session IDs during authentication processing, records the authenticated user's identity with the session ID, and continually checks the session ID in each page request's URL or POST data with the session cookie and the session record in the database. These checks provide users with flexibility and security.

While the session ID is the key to session state, the session cookie and the session record safeguard the integrity of the session ID and the authentication status of the user.

> **See Also:**
>
> Understanding URL Syntax

3.6.3 What Is a Session?

A session is a logical construct that establishes persistence (or stateful behavior) across page views.

Each session is assigned a unique identifier. The Oracle APEX engine uses this identifier (or session ID) to store and retrieve an application's working set of data (or session state) before and after each page view. Because sessions are entirely independent of one another, any number of sessions can exist in the database at the same time. A user can also run multiple instances of an application simultaneously in different browser programs.

Sessions are logically and physically distinct from Oracle database sessions used to service page requests. A user runs an application in a single APEX session from sign in to sign out with a typical duration measured in minutes or hours. Each page requested during that session results in the APEX engine creating or reusing an Oracle database session to access database resources. Often these database sessions last just a fraction of a second.

> **See Also:**
>
> Viewing Active Sessions in *Oracle APEX Administration Guide*

3.7 Managing Session State

View session state and manage session state overrides from the Runtime Developer toolbar which displays at the bottom of any editable page at runtime.

- Viewing Session State
  View the Session page at runtime by clicking Session and then View Session State on the Runtime Developer toolbar.
- Managing Session State Overrides
  Access the Session Overrides dialog from the Runtime Developer tool bar to manage session state values at runtime.

See Also:

- Runtime Developer Toolbar
- Understanding Page Items
- Managing Application Items
- About Using Collections
- Using Substitution Strings for information about referencing item values

3.7.1 Viewing Session State

View the Session page at runtime by clicking Session and then View Session State on the Runtime Developer toolbar.

The Session page provides valuable information about the session in which the application is currently running. Must of the behavior of an Oracle APEX application is driven by values in session state. For example, a button may display conditionally based on the value of an item session state. This section describes how to view session state for a page.

To view the Session page:

1. Run the application as a developer.
2. Locate the Runtime Developer toolbar. By default, the Runtime Developer toolbar displays at the bottom of the page.

3. On the Runtime Developer toolbar, click **Session** and then **View Session State**.

   The Session page appears and displays the following information displays at the center of the page.

   - **Application** - Identifies the application name.
   - **Session** - Displays the session ID for the current session.
   - **User** - Identifies the current user.
   - **Workspace** - Displays the current workspace ID.
   - **Browser Language** - Identifies the current browser language.
4. Configure the fields at the top of the page and click **Set** to alter the display:
   - **Page** - Locate a page. Enter a page number.
   - **Find** - Enter a case insensitive query. Use Find to search item name and item value, or collection name (depending upon your view).
   - **Rows** - Select the number of items to display.
   - **View** - Select a session state report to view.

3.7.2 Managing Session State Overrides

Access the Session Overrides dialog from the Runtime Developer tool bar to manage session state values at runtime.

The **Session Overrides** switch eanbles you to enable or disable all session values. Likewise, every session value has a switch that enables you control session state value overrides.

For example, if you need to translate an application you can change the Application Language attribute. Another common use case for session overrides is to change the file path values such as `#APP_FILES#`. This type of change would redirect the URL to somewhere else, like `localhost:PORT` for local development.

To access the Session Overrides dialog:

1. Run the application as a developer.
2. Locate the Runtime Developer toolbar. By default, the Runtime Developer toolbar displays at the bottom of the page.
3. On the Runtime Developer toolbar, click **Session** and then **Session Overrides**.
4. In the Session Overrides dialog:
   a. Enable Sessions Overrides - Use this switch to enable or disable all session values.
   b. Globalization:
      - Application Language - Change the language of the current application for the current user.
      - Document Direction- Change the document direction (Left to Right, Right to Left) of the current application for the current user
      - Timezone - Change the timezone of the current application for the current user.

      You can also edit these attributes in Globalization attributes. See Configuring Globalization Attributes.
   c. File Paths:
      - Application Files *`#APP_FILES#`* - Enable this option to change the value of `#APP_FILES#` to point to another location. `#APP_FILES#` determines the virtual path the web server uses to point to the images directory distributed with App Builder. During installation, the virtual path is configured as `/i/` . If you are unsure, contact your administrator.

        You can also edit this attribute in User Interface, Advanced. See Editing User Interface Attributes.
      - Workspace Files *`#WORKSPACE_FILES#`* - Enable this option to change the value of `#WORKSPACE_FILES#` to point to another location. `#WORKSPACE_FILES#` references uploaded images, JavaScript, and cascading style sheets that are shared over many applications within a workspace.
      - Theme Files *`#THEME_FILES#`* - Enable this option to change the value of `#THEME_FILES#` to point to another location. Use the `#THEME_FILES#` to reference files which are stored with your theme definition
   d. Additional Files:
      - JavaScript URLs - Load additional JS file URLs to this application for the current user.

You can also edit this attribute in User Interface, JavaScript. See Editing User Interface Attributes.

- CSS URLs - Load additional CSS file URLs to this application for the current user.

  You can also edit this attribute in User Interface, CSS. See Editing User Interface Attributes.

5. Exit the dialog:
   - Click **Save** to save your changes.
   - Click **Reset** to undo any changes.
   - Click **Cancel** to exist dialog without making any changes.

3.8 Understanding URL Syntax

Oracle APEX applications support two types of URL syntax: **Friendly URL Syntax** and **f?p Syntax**.

> **Tip:**
>
> Applications created using Application Express release 20.1 or later use **Friendly URL Syntax**. You can change existing applications to use Friendly URLs by editing the Friendly URLs attribute in the application definition. See Enabling or Disabling Friendly URL Syntax.

- Understanding Friendly URL Syntax
  Friendly URL Syntax creates a URL structure that identifies the address of Oracle APEX, the application, the page, and uses standard web parameter syntax.
- Understanding f?p URL Syntax
  f?p URL Syntax is a legacy syntax that creates a unique URL structure that identifies the address of Oracle APEX, the application ID, page number, and session ID.
- Hiding the Session ID on Public Pages
  Hide the session ID in a URL by making a page public.

3.8.1 Understanding Friendly URL Syntax

Friendly URL Syntax creates a URL structure that identifies the address of Oracle APEX, the application, the page, and uses standard web parameter syntax.

> **Note:**
>
> The Friendly URL Syntax described in this section is different from a vanity URL or custom domain name. Configuring a vanity URL (or custom domain name) for your APEX instance is typically done at the web server or load balancer-level and is not discussed in this publication.

> **Tip:**
>
> Applications created using Application Express release 20.1 or later use Friendly URL Syntax. You can revert to f?p Syntax by disabling the Friendly URLs attribute.

- About Friendly URL Syntax
  Friendly URL Syntax uses a standard URL hierarchy and passes parameters in a similar fashion to other web applications.
- Enabling or Disabling Friendly URL Syntax
  Determine if an application uses Friendly URL Syntax or f?p Syntax by enabling or disabling the Friendly URLs attribute.
- Customizing a Friendly URL
  Customize a Friendly URL by editing the Path Prefix, Application Alias, or Page Alias.

3.8.1.1 About Friendly URL Syntax

Friendly URL Syntax uses a standard URL hierarchy and passes parameters in a similar fashion to other web applications.

Friendly URL Hierarchy and Syntax

Friendly URL Syntax creates a URL with the following directory hierarchy and syntax:

```
https://hostname:port/ords/r/path_prefix/app_alias/page_alias?parameters
```

Friendly URL Example

```
https://hostname:port/ords/r/mycompany/hr-app/update-employees?
session=13766599855150
```

Where:

- `https:` stands for HyperText Transfer Protocol and operates at the application layer.
- *`hostname`* is the domain name of the server. It can also include a port number or an IP address.
- *`port`* is the port number assigned when configuring Oracle REST Data Services.
- `ords` is the service name defined when configuring Oracle REST Data Services.
- *`r`* is the router shortcut. This value is a constant and should never be changed.
- `mycompany` is the *`path_prefix`* which is URI path prefix used to access RESTful Services. When you create a workspace, this value defaults to workspace name. You can customize the URI path prefix by editing the Path Prefix attribute in Administration, Manage Service, Workspace Preferences, SQL Workshop.
- `hr-app` is the *`app_alias`*. In a new application, the Application Alias defaults to the application Name. You can edit the Application Alias in the application Definition. The Application Alias must be unique within the workspace.
- `update-employees` is the *`page_alias`*. In a new application, the Page Alias defaults to the page Name. You can edit the Page Alias in Page Designer. Page aliases must be unique

within the application. When creating a new page, if a Page Name is already used as a Page Alias, then a numeric value is appended to the new Page Alias to make it unique.

- `?session=13766599855150` identifies the session ID. When you run an application, the Oracle APEX engine generates a session number that serves as a key to the user's session state.

About URL Parameters

The part of the URL after `/?` can list `parameters`.

`http://hostname:port/ords/r/path_prefix/app_alias/page_alias?parameters` (Such as –

`p7_customer_id=377&clear=7&session=1247058356345&cs=3MdkfQa9NLmkyHAbwGPg3U-nmA7XIw5sWjU2rMzSvOy_wP1_z7BBI-Gbh2JgvffARgcYJy8id4VtL3d0l1tYm0w)`

Supported parameters include:

- `session`
- `request`
- `clear`
- `debug`
- `application/page items`
- `printerFriendly`
- `trace`
- `timezone`
- `lang`
- `territory`
- `cs`
- `dialogCs`
- `x01`

See Also:

- Understanding Session State Management
- About Publishing an Application URL

3.8.1.2 Enabling or Disabling Friendly URL Syntax

Determine if an application uses Friendly URL Syntax or f?p Syntax by enabling or disabling the Friendly URLs attribute.

Applications created using Oracle APEX release 20.1 or later use Friendly URL Syntax. You can revert to f?p Syntax by disabling the Friendly URLs attribute.

To enable Friendly URL Syntax:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. Navigate to the Edit Application Definition page. Navigation options include:
   - Click the **Edit Application Definition** button.
   - From Shared Components:
     - Click **Shared Components**.
     - Under Application Logic, click **Application Definition**.

   The Definition page appears.
4. Under Properties, configure the **Friendly URLs** attribute:
   - **On** - If set to **On**, application uses Friendly URLs Syntax in which the URL consists of a directory structure based on the Application Alias, Page Alias, and standard web parameter syntax.
   - **Off** - If set to **Off**, the application uses f?p= Syntax.
5. Click **Apply Changes** to save your changes.

3.8.1.3 Customizing a Friendly URL

Customize a Friendly URL by editing the Path Prefix, Application Alias, or Page Alias.

Friendly URL Syntax creates a URL with the following directory hierarchy and syntax:

```
https://hostname:port/ords/r/path_prefix/app_alias/page_alias?parameters
```

You can customize the follow parts of Friendly URL:

- Path Prefix - *`path_prefix`* is URI path prefix to be used to access RESTful Services. When you create a workspace, this value defaults to workspace name. You can customize the URI path prefix by editing Path Prefix attribute in Workspace Administration.
- Application Alias - *`app_alias`* is an alternate alphanumeric application identifier called the Application Alias. In a new application, the Application Alias defaults to the application Name. You can edit the Application Alias in the application Definition.
- Page Alias - *`page_alias`* is an alphanumeric alias for this page called the Page Alias. In a new application, the Page Alias defaults to the page Name. You can edit the Page Alias in Page Designer.

Friendly URL Customization Example

Suppose you have an application with the following URL:

```
http://hostname:port/ords/r/example/355/2?session=13766599855150
```

Suppose you want to make the following changes:

- *`path_prefix`* - Change `example` to `mycompany`.
- *`app_alias`* - Change `355` to `myapp` .

- *page_alias* - Change `2` to `myinfo`.

The revised URL will display as:

```
http://hostname:port/ords/r/mycompany/myapp/myinfo?session=13766599855150
```

To customize a Friendly URL by editing the Path Prefix, Application Alias, or Page Alias:

1. Sign in to Oracle APEX.
2. To change the Path Prefix:
   a. Navigate to the Workspace Administration page. At the top of the Workspace home page, click the **Administration** menu and select **Administration**.
   b. On the Workspace Administration home page, click **Manage Service** and then click **Set Workspace Preferences**.
   c. Scroll down to SQL Workshop.
   d. In **Path Prefix**, enter a new Path Prefix. In a new workspace, Path Prefix defaults to the workspace name. For example change the Path Prefix to `mycompany`.

      > **Tip:**
      >
      > Path Prefix only displays if **Enabled RESTful Services** is set to **Yes**.

   e. Click **Apply Changes** to save your changes.

      Manage Service page appears.
   f. Return to the Workspace home page. Click the APEX logo (Oracle APEX Home) in the upper left corner.
3. To change the Application Alias:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On Application home page, click the **Edit Application Definition** button.

      The application Definition page appears.
   d. Under Name, edit the **Application Alias** attribute. For example, enter `myapp`.
   e. Click **Apply Changes** to save your changes.
4. To change the Page Alias:
   a. View the page in Page Designer.
   b. In either the Rendering tab, select the page name.
   c. In the Property Editor, find the Identification group.
   d. Under Identification, Page Alias attribute - Enter an alphanumeric alias for this page. For example, enter `myinfo`.
   e. To save your changes, click **Save** or **Save and Run Page**.
5. Run the page. Click or **Save and Run Page**.

   The revised URL displays as:

   ```
   http://hostname:port/ords/r/mycompany/myapp/myinfo?session=13766599855150
   ```

> **See Also:**
>
> Enabling RESTful Services and Defining a Path Prefix for a Workspace in *Oracle APEX Administration Guide*

3.8.2 Understanding f?p URL Syntax

f?p URL Syntax is a legacy syntax that creates a unique URL structure that identifies the address of Oracle APEX, the application ID, page number, and session ID.

- About f?p URL Syntax
  f?p URL Syntax creates a unique URL structure that identifies the location and address of Oracle APEX, the application ID, page number, and session ID.
- About Using f?p Syntax to Link Pages
  Create links between application page using `f?p` syntax.
- Calling a Page Using an Application Alias or Page Alias
  Use URL syntax to call a page using an application alias and/or page alias.
- Calling a Page from a Button URL
  Use URL syntax to call a page from a button URL.

3.8.2.1 About f?p URL Syntax

f?p URL Syntax creates a unique URL structure that identifies the location and address of Oracle APEX, the application ID, page number, and session ID.

f?p URL Structure and Syntax

f?p URL Syntax creates a URL with the following structure and syntax:

```
https://hostname:port/ords/f?p=app_id:page_number:session_id
```

f?p URL Example

```
https://hostname:port/ords/f?p=392:3:13766599855150
```

This example indicates:

- `https:` stands for HyperText Transfer Protocol and operates at the application layer.
- *`hostname`* is the domain name of the server. It can also include a port number or an IP address.
- *`port`* is the port number assigned when configuring Oracle REST Data Services.
- `ords` is the service name defined when configuring Oracle REST Data Services.
- `f?p=`is a prefix used by Oracle APEX to route the request to the correct engine process.
- `392` is the ID of the application being called. The application ID is a unique number that identifies each application.

- `3` is the number of the page within the application. Page number must be unique within the application.
- `13766599855150` is the session number. When you run an application, the APEX engine generates a session number that serves as a key to the user's session state.

> **See Also:**
>
> - Understanding Session State Management
> - About Publishing an Application URL

3.8.2.2 About Using f?p Syntax to Link Pages

Create links between application page using `f?p` syntax.

App Builder includes many wizards that automatically create these references for you. However, you may have to create the syntax yourself in some situations.

To create links between pages in an application, use the following `f?p` syntax:

```
f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly
```

The following table describes the arguments you can pass when using `f?p` syntax.

Table 3-1 f?p Syntax Arguments

Syntax	Description
`App`	Indicates an application ID or alphanumeric application alias. An **application alias** is a unique alternate alphanumeric application identifier that developers can use in place of the application ID. You define an Application Alias editing the Application Definition. **See Also:** Managing Application Attributes
`Page`	Indicates a page number or alphanumeric page alias. A **page alias** is a unique alternate alphanumeric for a page number. Developers can use a page alias in place of the page number ID. You define a Page Alias by editing the page attributes. **See Also:** Managing Page Attributes
`Session`	Identifies a session ID. You can reference a session ID to create hypertext links to other pages that maintain the same session state by passing the session number. You can reference the session ID using the syntax: • Short substitution string: `&SESSION.` • PL/SQL: `V('SESSION')` • Bind variable: `:APP_SESSION`
`Request`	Sets the value of `REQUEST`. Each application button sets the value of `REQUEST` to the name of the button which enables accept processing to reference the name of the button when a user clicks it. You can reference `REQUEST` using the syntax: • Substitution string: `&REQUEST.` • PL/SQL: `V('REQUEST')` • Bind variable: `:REQUEST`

Table 3-1 (Cont.) f?p Syntax Arguments

Syntax	Description
`Debug`	Displays application processing details. Valid values for the DEBUG flag include: • YES. • `LEVEL`*n* • NO Setting this flag to `YES` displays details about application processing. Setting this flag to `LEVEL`*n* (where *n* is between 1 and 9) controls the level of debug detail that displays. The value of YES equals `LEVEL4`. You can reference the Debug flag using the following syntax: • Short substitution string: `&DEBUG`. • PL/SQL: `V('DEBUG')` • Bind variable: `:DEBUG` **See Also**: Debugging an Application
`ClearCache`	Clears the cache. This sets the value of items to null. To clear cached items on a single page, specify the numeric page number. To clear cached items on multiple pages, use a comma-separated list of page numbers. Clearing a page's cache also resets any stateful processes on the page. Individual or comma-separated values can also include collection names to be reset or the keyword `RP`, which resets region pagination on the requested page. The keyword `APP` clears cache for all pages and all application-level items in the current application and removes sort preferences for the current user. The keyword `SESSION` achieves the same result as the `APP` keyword, but clears items associated with all applications that have been used in the current session. **See Also**: Clearing Session State
`itemNames`	Comma-delimited list of item names used to set session state with a URL.
`itemValues`	List of item values used to set session state within a URL. To pass a comma in an item value, enclose the characters with backslashes. For example: `\123,45\` Every character sequence except `backslash comma` (`\,`) can be enclosed with backslash.
`PrinterFriendly`	Determines if the page is being rendered in printer friendly mode. If PrinterFriendly is set to Yes, then the page is rendered in printer friendly mode. The value of PrinterFriendly can be used in rendering conditions to remove elements such as regions from the page to optimize printed output. You can reference the printer friendly preference by using the following syntax: `V('PRINTER_FRIENDLY')` When referenced, the page displays using a printer friendly template. The APEX engine displays all text within HTML form fields as text. The printer friendly template does not need to have the `#FORM_OPEN#` or `#FORM_CLOSE#` tags. The objective is to be able to display information with few tables and in a format suitable for printing.

3.8.2.3 Calling a Page Using an Application Alias or Page Alias

Use URL syntax to call a page using an application alias and/or page alias.

Application and page aliases must consist of valid Oracle identifiers, cannot contain any whitespace, and cannot be case-sensitive. The following example calls a page from within an

application using an application alias and a page alias. This example runs the page *home* of the application *myapp* and uses the current session ID.

```
f?p=myapp:home:&APP_SESSION.
```

Application aliases must be unique within a workspace. If an application in a different workspace has the same application alias, use the `&c` argument to specify the workspace name. For example:

```
f?p=common_alias:home:&APP_SESSION.&c=WORKSPACE_A
```

3.8.2.4 Calling a Page from a Button URL

Use URL syntax to call a page from a button URL.

When you create a button, you can specify a URL to redirect to when the user clicks the button. This example runs page 6001 of application 6000 and uses the current session ID.

```
f?p=6000:6001:&APP_SESSION.
```

Note that this is only one approach to using a button. This method bypasses page submission and acts as a hyperlink on the page. You can also have a button click submit a page. In that approach, clicking the button submits the page for processing, allowing forms to be submitted and session state to be saved.

> **See Also:**
>
> Managing Buttons and APP_SESSION

3.8.3 Hiding the Session ID on Public Pages

Hide the session ID in a URL by making a page public.

By default, all pages require Authentication. However, if a page is public and the user has not signed in, the page URL will not display the session ID.

> **Tip:**
>
> For security reasons, Oracles recommends that administrators disable Rejoin Sessions unless they implement workspace isolation by configuring the Allow Hostname attribute at the workspace or instance-level. See About Isolating Workspaces.

To specify a page as public:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. At the Application-level, edit the **Rejoin Session** attribute:
   a. Click the **Edit Application Definition** button.

      The Application Definition appears.

b. Click **Security**.

c. Session Management, Rejoin Sessions - Select **Enabled for Public Sessions**.

d. Click **Apply Changes** to save your changes.

4. At the page-level, edit the Authentication attribute:

   a. View the page in Page Designer.

      The Application Definition appears.

   b. In the Rendering tab, select the page name.

   c. Security, Authentication - Select **Page is Public**.

   d. Click **Apply Changes** to save your changes.

> **See Also:**
>
> - About Rejoin Sessions
> - About Enabling Support for Bookmarks
> - Session Management
> - Configuring Rejoin Sessions for a Page
> - Configuring Rejoin Sessions for an Instance in *Oracle APEX Administration Guide*

3.9 Managing Session State Values

Manage session state to store and retrieve values for a user as the user navigates between different application pages.

When building interactive, data-driven web applications, the ability to access and manage session state values is critical. In Oracle APEX, session state is automatically managed for every page and easily referenced in static HTML or logic controls such as processes or validations.

- About Referencing Session State
  Reference item values stored in session state in regions, computations, processes, validations, and branches. An item can be a field, a text area, a password, a select list, or a checkbox.
- About Setting Session State
  Set session state using form submissions, bind variables, computations, or f?p syntax. Session state can also be set using the `APEX_SESSION_STATE.SET` APIs.
- Clearing Session State
  Clearing a cached value resets the value to null. You can clear the cached value for specific items, all items on a page, all pages in an application, or the current user session.
- Referencing Session State Using Bind Variable Syntax
  Use bind variable syntax anywhere where you are using SQL or PL/SQL to reference session state of a specified item.

- About Session Cloning
  Use session cloning to open a new browser window in an Oracle APEX application and have it generate a new distinct session identifier and copy the session values from the original APEX session to the new one.

> **See Also:**
>
> Understanding Session State Management and Referencing Item Values

3.9.1 About Referencing Session State

Reference item values stored in session state in regions, computations, processes, validations, and branches. An item can be a field, a text area, a password, a select list, or a checkbox.

The following table describes the supported syntax for referencing item values.

Type	Syntax	Description

SQL	Standard item syntax:
	`:MY_ITEM`
	Syntax for items containing special characters:
	`:"MY_ITEM"`
		For items whose names are no longer than 30 characters, precede the item name with a colon (:). Use this syntax for references within a SQL query and within PL/SQL.
		To reference page items containing special, multibyte, or unicode characters, wrap the page item name in double quotation marks.
		The data type of the bind variable is the Session State Data Type of the item. Most items are `VARCHAR2` by default, but certain item types can be set to `CLOB`.

PL/SQL	`V('MY_ITEM')` or `APEX_SESSION_STATE.GET_VARCHAR2('MY_ITEM')`	Use PL/SQL syntax to reference an item value using the `V` function. You can use the shorthand, `V` function, in place of `APEX_SESSION_STATE.GET_VARCHAR2`. Use this syntax when utilizing Oracle APEX variables directly within an Oracle database object, such as a function, trigger, or Oracle Data Redaction policy. **Tip**: When using `V('MY_ITEM')`, `apex_session_state.get_clob('MY_ITEM')` or similar in a SQL query, Oracle recommends using Fast Dual functionality `=> (select v('my_item') from dual)` to improve performance.
PL/SQL	`APEX_SESSION_STATE.GET_CLOB('MY_CLOB_ITEM')`	This function returns the value of a `CLOB` page item. **Tip**: When using `V('MY_ITEM')`, `apex_session_state.get_clob('MY_ITEM')` or similar in a SQL query, Oracle recommends using Fast Dual functionality `=> (select v('my_item') from dual)` to improve performance.
PL/SQL	`APEX_SESSION_STATE.GET_NUMBER('MY_NUMERIC_ITEM')`	This function returns the value of a page item as a `NUMBER`. The item's Format Mask is used to perform the conversion. **Tip**: When using `V('MY_ITEM')`, `apex_session_state.get_clob('MY_ITEM')` or similar in a SQL query, Oracle recommends using Fast Dual functionality `=> (select v('my_item') from dual)` to improve performance.

Type	Syntax	Description
PL/SQL	`APEX_SESSION_STATE.GET_TIMESTAMP('MY_NUMERIC_ITEM')`	This function returns the value of a page item as a `TIMESTAMP`. The item's Format Mask is used to perform the conversion. **Tip**: When using `V('MY_ITEM')`, `apex_session_state.get_clob('MY_ITEM')` or similar in a SQL query, Oracle recommends using Fast Dual functionality `=> (select v('my_item') from dual)` to improve performance.

Static text (exact)	Standard item syntax:
	`&MY_ITEM.`
	Syntax for items containing special characters:
	`&"MY_ITEM".`
		For static text or an exact substitution, use the convention `&ITEM_NAME` followed by a period (.).
		To reference page items containing special, multibyte, or unicode characters, wrap the page item name in double quotation marks.
		Page items of Session State Data Type `CLOB` cannot be references via this syntax. A different output mechanism must be used to fetch their values, such as a page item in Read Only mode, a Display Only page item, or its manual output using a Dynamic Content region.

> **See Also:**
>
> Changes in Oracle APEX API Reference in *Oracle APEX API Reference*

3.9.2 About Setting Session State

Set session state using form submissions, bind variables, computations, or f?p syntax. Session state can also be set using the `APEX_SESSION_STATE.SET` APIs.

> **See Also:**
>
> Changes in Oracle APEX API Reference in *Oracle APEX API Reference*

Methods for Setting Session State

You can set the value of a page item in your application and therefore set session state using the following methods:

- Form submission. See the next section.
- Bind variable. See Referencing Session State Using Bind Variable Syntax.
- Computation. See Understanding Application Computations.
- f?p syntax. See About Using f?p Syntax to Link Pages.

About Setting Session State with a Form Submission

When a user submits a page, the Oracle APEX engine automatically stores values typed into fields (items) in session state. For example, suppose you have an application containing two pages. The first page of the application contains a form in which a user can enter a phone number. You defined this form by creating an item named *`P1_PHONENO`*. On the second page, you want to display the information the user enters in the form.

When the page is submitted, APEX captures the value entered in the phone number field and stores the value for future use. On the second page, the phone number entered by the user can then be retrieved from session state using the name `P1_PHONE_NO` with the appropriate syntax described in About Referencing Session State.

About Controlling How Items Write Session State

You can control whether a page item writes its session state into persistent (disk) session state or just into memory by configuring the item attribute, **Maintain Session State**. Available Maintain Session State options include:

- **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using the Page Items To Submit attribute.
- **Per Session (Disk)** - Maintain for each session by storing the value in the database, to access it across requests.
- **Per User (Disk)** - Value is saved in a user attribute repository and it is also available for later APEX sessions.

> **Note:**
>
> When creating database items that work with a Form region (for example as part of a wizard), **Per Request (Memory Only)** is the default. Per **User (Disk)** is not available for these items.

> **See Also:**
>
> Controlling How Page Items Write Session State

3.9.3 Clearing Session State

Clearing a cached value resets the value to null. You can clear the cached value for specific items, all items on a page, all pages in an application, or the current user session.

- About Clearing Cache for an Item
  View examples for clearing cache for a single item.
- About Clearing Cache for All Page Items
  View examples for clearing cache for all page items.
- About Clearing Report Regions
  Learn about clearing report cache.
- Clearing Cache for an Entire Application
  View an eample of clearing cache for an application.
- About Resetting an Application Completely
  Learn about resetting an application completely.
- About Clearing Cache for the Current User Session
  View an example of clearing application cache for the current user session.

3.9.3.1 About Clearing Cache for an Item

View examples for clearing cache for a single item.

Clearing cache for a single item resets the value of the item to null. For example, you might use this approach to make sure a specific item's value is null when a page is prepared for rendering.

Example 3-4 Example: Clearing Cache for an Item

The following example uses standard `f?p` syntax to clear the cache for an item. This example calls page 5 of application 100. Placing `MY_ITEM` in the `ClearCache` position of the `f?p` syntax resets the value of `MY_ITEM` to `NULL`.

```
f?p=100:5:&APP_SESSION.::NO:MY_ITEM
```

The following example resets the value of the items `THE_EMPNO` and `THE_DEPTNO`:

```
f?p=100:5:&APP_SESSION.::NO:THE_EMPNO,THE_DEPTNO
```

3.9.3.2 About Clearing Cache for All Page Items

View examples for clearing cache for all page items.

Caching application items is an effective way to maintain session state. However, there are occasions when you may want to clear the cache for all items on a page. For example, suppose you needed to clear all fields on a page when a user clicks a link that creates a new order. By clearing the cache for an entire page, you set the value of all items on the page to null.

Example 3-5 Example: Clearing Cache for Two Pages and Resetting Pagination

This example clears the session cache for two pages and resets pagination.

```
f?p=6000:6003:&APP_SESSION.::NO:RP,6004,6014
```

This example:

- Runs page 6003 of application 6000 and uses the current session ID.
- Indicates to not show debug information (`NO`).
- Clears all values maintained by the current session's cache for items of pages 6004 and 6014.
- Resets region pagination (`RP`) on page 6003 (the requested page).

Example 3-6 Example: Clearing Cache on a Page and Passing an Item Value

This example shows how to implement an update form. It clears existing information and sets the item's value (typically a primary key).

```
f?p=6000:6003:&APP_SESSION.::NO:6003:MY_ITEM:1234
```

This example:

- Runs page 6003 of application 6000 and uses the current session ID
- Indicates to not show debug information (`NO`)
- Clears all values maintained by the current session's cache for items on page 6003
- Sets the session state of an item called `MY_ITEM` to the value `1234`

Example 3-7 Example: Clearing Session Cache on a Page and Passing Values to Multiple Items

This example demonstrates how to implement an update form. It clears existing information, sets the item's value (typically a primary key), and passes values to multiple items.

```
f?p=6000:6004:&APP_SESSION.::NO:6003:MY_ITEM1,MY_ITEM2,MY_ITEM3:1234,,5678
```

This example:

- Runs page 6004 of application 6000 and uses the current session ID
- Clears the current session's cache for items on page 6003
- Indicates to not show debug information (`NO`)
- Sets the value of `MY_ITEM1` to 1234, sets the value of `MY_ITEM2` to null (indicated by the comma used as placeholder), and sets the value of `MY_ITEM3` to 5678

3.9.3.3 About Clearing Report Regions

Learn about clearing report cache.

Report settings can be cached. Depending on the report types, different settings are cached. Use the following to clear report cache settings:

- `RR` - Resets interactive report, interactive grids, or classic report and also resets report pagination. If the target page contains interactive report or interactive grids, the report is returned to the default report settings specified by the developer or saved by the user. If the target page contains classic report, the report sorting resets to developer defined sorting.
- `CR` - This applies only to interactive report. `CR` clears interactive report and resets report pagination. This clears all of the session report settings such as control break, aggregate, flashback, chart, number of rows to display, filter, highlight, computation, group by, and pivot.
- `RP` - Resets interactive report or classic report pagination.

Example 3-8 Resetting Report Settings and Pagination

This example resets report settings and pagination of the target page report regions:

```
f?p=6000:6003:&APP_SESSION.::NO:RR
```

> **Note:**
>
> Starting with Oracle APEX release 2.1, you do not need to define separate clear cache syntax to reset report and reset pagination. `RR` will handle both.

3.9.3.4 Clearing Cache for an Entire Application

View an eample of clearing cache for an application.

This example clears the application's cache by using `f?p` syntax and creating a `Clear Cache` argument using the keyword *APP*.

```
f?p=App:Page:Session::NO:APP
```

3.9.3.5 About Resetting an Application Completely

Learn about resetting an application completely.

Resetting the cache for an entire application does not restore the application to a completely reset state. For example, if an application includes on-new instance computations or on-new instance processes, the Oracle APEX engine runs these computations and processes when the application session is created. Then, it processes the clear cache request and displays the requested page.

To reset an application completely without a session ID (if no cookie is used to track the session ID), you must request it using a URL without a session ID, or by calling `APEX_UTIL.CLEAR_APP_CACHE` from another application. If the session ID is tracked using a cookie, you must logout to reset the state.

3.9.3.6 About Clearing Cache for the Current User Session

View an example of clearing application cache for the current user session.

Another approach to clearing an application's cache is to create a `Clear Cache` argument using the keyword `SESSION`. For example:

```
f?p=6000:6004:12507785108488427528::NO:SESSION
```

3.9.4 Referencing Session State Using Bind Variable Syntax

Use bind variable syntax anywhere where you are using SQL or PL/SQL to reference session state of a specified item.

- About Using Bind Variable Syntax
  Learn about bind variable syntax.
- About Using Bind Variables in Regions Based on a SQL Query or LOV
  Learn about using bind variables in regions defined as a SQL Query, SQL Query (plsql function body returning SQL query), or list of values (LOV).
- About Using Bind Variables in Regions Based on PL/SQL
  Learn about using bind variables in regions based on PL/SQL.

3.9.4.1 About Using Bind Variable Syntax

Learn about bind variable syntax.

In the following example, the search string is a page item.

```
SELECT * FROM employees WHERE last_name like '%' || :SEARCH_STRING || '%'
```

If the region type is defined as SQL Query, you can reference the value using standard SQL bind variable syntax. Using bind variables ensures that parsed representations of SQL queries are reused by the database, optimizing memory usage by the server.

When using bind variable syntax, remember the following rules:

- Bind variable names must correspond to an item name.
- Bind variable names are not case-sensitive.
- Bind variable names cannot be longer than 30 characters (that is, they must be a valid Oracle identifier).

Although page item and application item names can be up to 255 characters, if you intend to use an application item within SQL using bind variable syntax, the item name must be 30 characters or less.

- The data type of bind variables is always `VARCHAR2`. When a bind variable semantically contains a datetime or numeric value, you must perform an explicit conversion to the appropriate data type. For example:

  ```
  SELECT * FROM employees WHERE start_date < to_date(:DATE_STRING, 'DD-MON-
  YYYY')
  ```

3.9.4.2 About Using Bind Variables in Regions Based on a SQL Query or LOV

Learn about using bind variables in regions defined as a SQL Query, SQL Query (plsql function body returning SQL query), or list of values (LOV).

If your region type is defined as a SQL Query, SQL Query (plsql function body returning SQL query), or list of values (LOV), you can reference session state using the following syntax:

```
:MY_ITEM
```

One common way to do this is to incorporate a session state variable in a `WHERE` clause. The following example shows how to bind the value of the item `THE_DEPTNO` into a region defined from a SQL Query.

```
SELECT last_name, job_id, salary
FROM employees
WHERE department_id = :THE_DEPTNO
```

> **See Also:**
>
> About Regions for information about creating regions

3.9.4.3 About Using Bind Variables in Regions Based on PL/SQL

Learn about using bind variables in regions based on PL/SQL.

For region types, processes, computations, validations, conditions, and so on that are defined as PL/SQL dynamic content, regions are constructed using PL/SQL anonymous block syntax. In other words, the beginning and ending keywords are used to enclose the PL/SQL block. For example:

```
IF :P1_JOB IS NOT NULL THEN
  INSERT INTO employees (employee_id, first_name, job_id)
  VALUES (:P1_EMP_ID, :P1_NAME, :P1_JOB)
end if;
```

In this example, the values of the `employee_id`, `first_name`, and `job_id` are populated by the values of `P1_EMP_ID`, `P1_NAME`, and `P1_JOB`.

3.9.5 About Session Cloning

Use session cloning to open a new browser window in an Oracle APEX application and have it generate a new distinct session identifier and copy the session values from the original APEX session to the new one.

In previous releases, opening multiple windows (or browser tabs) in the same APEX application resulted in a number of issues. Typically, all the browser tabs shared the same session and session state which resulted in unpredictable and undesirable results.

To use session cloning, the developer must provide a method for end user to open a new browser tab and specify the `REQUEST` value of `APEX_CLONE_SESSION`. The following is an example URL:

```
f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.:APEX_CLONE_SESSION
```

Best Practices When Using Session Cloning

When using this session cloning, developers should remember:

- The idle time associated with the Oracle APEX session would be affected by any of the APEX sessions, original or cloned ones.
- When a user logs out of one session (original or cloned), all other associated sessions will be logged out.
- The maximum session duration would be a function of the original APEX session and when it was created.
- An administrative user can enable or disable this feature using the procedure:

```
apex_instance_admin.set_parameter(
    p_parameter => 'CLONE_SESSION_ENABLED',
    p_value     => 'Y');
```

> **See Also:**
>
> *Oracle APEX API Reference*

3.10 Using Substitution Strings

Use substitution strings within templates, region source, or any attribute to pass information and replace a character string with another value.

> **Note:**
>
> If an attribute supports substitutions, Page Designer help includes the section called **Additional Information** which describes the supported substitution strings. See Viewing Help in Page Designer.

- About Using Substitution Strings
  Learn about using substitution strings.
- Controlling Output Escaping in Substitution Strings
  Learn about controlling output escaping in substitution strings.
- Viewing Supported Substitution Strings for a Template
  View supported substitution strings in templates in the template definition.
- Using Built-in Substitution Strings
  Learn about built-in substitution strings available in App Builder.

3.10.1 About Using Substitution Strings

Learn about using substitution strings.

> **Tip:**
>
> App Builder supports many built-in substitution strings. You can reference these substitution strings to achieve specific types of functionality. To learn more, see Using Built-in Substitution Strings.

- Placeholder Substitution Strings within Templates
  Include a placeholder substitution string within a template to reference component values.
- Session State Substitution Syntax for Page or Application Items
  Reference page or application items using Session State Substitution syntax (also called `&ITEM.` syntax).
- Substitution Strings for Interactive Grid, Cards, and Map Columns
  Reference columns in an interactive grid, cards page, or map page using substitution variables.
- Substitution Strings for Other Report Columns
  Reference columns in other reports using substitution variables.
- Using Double Quotation Marks for Special Characters or Case Sensitivity
  If item or column names use a special character or are case sensitive, enclose the name with double quotation marks.
- Substitutions in Text Messages (24.2 or later)
  For applications with a Compatibility Mode of 24.2 or later, you can reference text messages by enclosing the message name in curly braces ({}).

3.10.1.1 Placeholder Substitution Strings within Templates

Include a placeholder substitution string within a template to reference component values.

Placeholder substitution strings available within a template are denoted by the pound sign (#). For example:

```
#ABC#
```

> **Tip:**
>
> You can also use JavaScript API `apex.util.applyTemplate` to evaluate templates on the client side. See applyTemplate in *Oracle APEX JavaScript API Reference*.

3.10.1.2 Session State Substitution Syntax for Page or Application Items

Reference page or application items using Session State Substitution syntax (also called *`&ITEM.`* syntax).

To reference page or application items use Session State Substitution string syntax (also called *`&ITEM.`* syntax):

1. Reference the page or application item in all capital letters.
2. Precede the item name with an ampersand (&).
3. Append a period (.) to the item name.

You can use Session State Substitution string syntax in a region, a region title, an item label, or in any of numerous other contexts in which static text is used.

The following example references to an application item named `F101_X` :

```
&F101_X.
```

Notice that item name ends with a required trailing period. When the page is rendered, APEX engine replaces the value the substitution string with the value of the item `F101_X` .

3.10.1.3 Substitution Strings for Interactive Grid, Cards, and Map Columns

Reference columns in an interactive grid, cards page, or map page using substitution variables.

To use substitution variables to reference columns in an interactive grid, cards page, or map page:

1. Reference the column name in all capital letters.
2. Precede the column name with an ampersand (&).
3. Append a period (.) to the item name.

The following example references the interactive grid column `ENAME`:

```
&ENAME.
```

Notice that the column name ends with a required trailing period.

3.10.1.4 Substitution Strings for Other Report Columns

Reference columns in other reports using substitution variables.

To use substitution variables to reference columns in other reports:

1. Reference the column name in all capital letters.
2. Precede the column name with a pound sign (#).
3. Append a pound sign (#) to the item name.

The following example references the interactive or classic report column `ENAME`:

```
#ENAME#
```

3.10.1.5 Using Double Quotation Marks for Special Characters or Case Sensitivity

If item or column names use a special character or are case sensitive, enclose the name with double quotation marks.

If the name contains special characters or is case sensitive, enclose items or columns with double quotation marks. For example:

```
&"ITEM-OR-COLUMN-NAME".
```

Use the following syntax:

```
&"<name>"[!<format>].
```

Where:

- *"<name>"* is an application item or page item or column name. If not quoted, then it must be in set A-Z–0-9_$#
- *"<name>"* is `APP_TEXT$`*<message-name>*
  Where *<message-name>* is an APEX text message name.
- *<format>* is a predefined filter name: `HTML`, `ATTR`, `JS`, `RAW`, `STRIPHTML` as described in Controlling Output Escaping in Substitution Strings.

3.10.1.6 Substitutions in Text Messages (24.2 or later)

For applications with a Compatibility Mode of 24.2 or later, you can reference text messages by enclosing the message name in curly braces ({}).

> **Tip:**
>
> The short substitution syntax described in this section is only available if the application's Compatibility Mode is 24.2 or later. See Editing the Application Definition.

To reference text messages, wrap the message name in curly braces ({}), also known as curly brackets. The following example reference a text message named `TEXT.MSG`:

```
&{TEXT.MSG}.
```

The message name, for example `TEXT.MSG`, can contain special characters. However, white space and a closing curly brace are not supported.

Text message substitutions also support parameters. Before the closing brace, add `name=value pairs`, separated by white space. Note that the name `$lang` is special since it controls the message language. Names can contain any character except white space and "=".

Values can be one of the following:

- Unquoted substitution, HTML-escaped by default. For example:

  ```
  &P1_NAME.
  ```

- Quoted substitution, HTML-escaped by default. For example:

  ```
  &"X-1".
  ```

- Unquoted literal value. Anything except double quotes. For example:

  ```
  he&she+it?
  ```

  White space and closing curly brace are not supported.

- Quoted literal value. Enclose the value in double quotes. To insert a double quote, type it twice. For example:

  ```
  "Hello ""APEX"" world"
  ```

The following is an example for a complex text message with parameters:

```
&{TEXT->MSG
Unquoted_Substitution=&P1_NAME!RAW.
Quoted_Substitution=&"Quoted name"!RAW.
Unquoted_Literal=how/do.you!do?
Quoted_Literal="Jerry Garcia
Ron ""Pigpen"" McKerhan"
$lang=FR}!HTML.
```

> **See Also:**
>
> Translating Messages

3.10.2 Controlling Output Escaping in Substitution Strings

Learn about controlling output escaping in substitution strings.

You can escape special characters in the substitution value by appending an exclamation mark (`!`) followed by a predefined filter name to a page or application item name, report column, or other substitution string. Output escaping is an important security technique to avoid Cross Site Scripting (XSS) attacks in the browser. Oracle APEX already makes a best effort to automatically escape characters in a HTML or JavaScript context. With this extended syntax, developers have fine-grained control over the output.

Available Escape Filters

Available escape filters include:

- `HTML` escapes reserved HTML characters, for example:
  - Item:

    ```
    &P1_DEPTNO!HTML.
    ```

  - Interactive grid column:

    ```
    &ENAME!HTML.
    ```

  - Columns in other reports:

    ```
    #ENAME!HTML#
    ```

> **See Also:**
>
> APEX_ESCAPE in *Oracle APEX API Reference*

- `ATTR` escapes reserved characters in a HTML attribute context, for example:
  - Interactive grid column:

    `&ENAME!ATTR.`
  - Columns in other reports:

    `#ENAME!ATTR#`

> **See Also:**
>
> HTML_ATTRIBUTE Function in *Oracle APEX API Reference*

- `JS` escapes reserved characters in a JavaScript context, for example:
  - Item:

    `&P1_DEPTNO!JS.`
  - Interactive grid column:

    `&ENAME!JS.`
  - Columns in other reports:

    `#ENAME!JS#`

> **Note:**
>
> The JS escape filter is not supported by substitutions performed on the client.

> **See Also:**
>
> JS_LITERAL Function in *Oracle APEX API Reference*

- `RAW` preserves the original item value and does not escape characters, for example:
  - Item:

    `&P1_DEPTNO!RAW.`
  - Interactive grid column:

    `&ENAME!RAW.`
  - Columns in other reports:

    `#ENAME!RAW#`
- `STRIPHTML` removes HTML tags from the output and escapes reserved HTML characters, for example:
  - Item:

```
&P1_DEPTNO!STRIPHTML.
```

– Interactive grid column:

```
&ENAME!STRIPHTML.
```

– Columns in other reports:

```
#ENAME!STRIPHTML#
```

3.10.3 Viewing Supported Substitution Strings for a Template

View supported substitution strings in templates in the template definition.

You can use substitution strings within templates to reference component values. To view the most current list of supported substitution strings for a given template, view Substitution Strings report in the template definition.

To view the view Substitution Strings report:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. To view a template definition, click the template name.

   The template definition appears.
3. Find the **Substitution Strings** report.

> **See Also:**
>
> - Creating Custom Themes
> - Managing Templates

3.10.4 Using Built-in Substitution Strings

Learn about built-in substitution strings available in App Builder.

App Builder supports many built-in substitution strings. You can reference these substitution strings to achieve specific types of functionality.

> **Tip:**
>
> Note that bind variable `:USER` has special meaning within the database. Also, the term **Direct PL/SQL** refers to PL/SQL that can be used in stored database objects such as procedures and functions.

- APEX_CSP_DISPLAY_NONE
- APEX_FILES
- APEX$ROW_NUM
- APEX$ROW_SELECTOR
- APEX$ROW_STATUS
- APP_ID
- APP_ALIAS
- APP_AJAX_X01, ... APP_AJAX_X10
- APP_BUILDER_SESSION
- APP_DATE_TIME_FORMAT
- APP_FILES
- APP_NLS_DATE_FORMAT
- APP_NLS_TIMESTAMP_FORMAT
- APP_NLS_TIMESTAMP_TZ_FORMAT
- APP_PAGE_ALIAS
- APP_PAGE_ID
- APP_REGION_ID
- APP_REGION_STATIC_ID
- APP_REQUEST_DATA_HASH
- APP_SESSION
- APP_SESSION_VISIBLE
- APP_TITLE
- APP_UNIQUE_PAGE_ID
- APP_USER
- APP_VERSION
- AUTHENTICATED_URL_PREFIX
- BROWSER_LANGUAGE
- CURRENT_PARENT_TAB_TEXT
- DEBUG
- HOME_LINK
- JET_BASE_DIRECTORY
- JET_CSS_DIRECTORY
- JET_JS_DIRECTORY
- LOGIN_URL
- LOGOUT_URL
- MAIN_APP_ID
- APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang
- OWNER

- PRINTER_FRIENDLY
- PROXY_SERVER
- PUBLIC_URL_PREFIX
- REQUEST
- Using REQUEST
- SCHEMA OWNER
- SQLERRM
- SYSDATE_YYYYMMDD
- THEME_DB_FILES
- THEME_FILES
- WORKSPACE_FILES
- WORKSPACE_ID

> **See Also:**
>
> Substitutions and Establishing User Identity Through Authentication

3.10.4.1 APEX_CSP_DISPLAY_NONE

Content Security Policy (CSP) guidelines prohibit the use of or direct inline CSS. The `APEX_CSP_DISPLAY_NONE` substitution string ensures that Oracle APEX only renders safe CSS inline exceptions using a centralized substitution string.

Use the `APEX_CSP_DISPLAY_NONE` substitution string to render the following inline CSS:

```
style=display:none;
```

Supported syntax for referencing `APEX_CSP_DISPLAY_NONE`:

```
 #APEX_CSP_DISPLAY_NONE#
```

APEX_CSP_DISPLAY_NONE Example

The following example demonstrates the use of `APEX_CSP_DISPLAY_NONE` in the Universal Theme region template, Inline Dialog.

```
<div id="#REGION_STATIC_ID#_parent">
<div id="#REGION_STATIC_ID#" class="t-DialogRegion #REGION_CSS_CLASSES# js-
regionDialog" #REGION_LANDMARK_ATTRIBUTES# #REGION_ATTRIBUTES#
#APEX_CSP_DISPLAY_NONE# title="#TITLE!ATTR#">
  <div class="t-DialogRegion-wrap">
    <div class="t-DialogRegion-bodyWrapperOut"><div class="t-DialogRegion-
bodyWrapperIn"><div class="t-DialogRegion-body">#BODY##SUB_REGIONS#</div></
div></div>
    <div class="t-DialogRegion-buttons">
       <div class="t-ButtonRegion t-ButtonRegion--dialogRegion">
         <div class="t-ButtonRegion-wrap">
```

```
            <div class="t-ButtonRegion-col t-ButtonRegion-col--left"><div
class="t-ButtonRegion-buttons">#PREVIOUS##DELETE##CLOSE#</div></div>
            <div class="t-ButtonRegion-col t-ButtonRegion-col--right"><div
class="t-ButtonRegion-buttons">#EDIT##CREATE##NEXT#</div></div>
          </div>
        </div>
     </div>
   </div>
</div>
</div>
```

3.10.4.2 APEX_FILES

> **Note:**
>
> `IMAGE_PREFIX` has been renamed to `APEX_FILES`. Although the legacy name, `IMAGE_PREFIX`, is still supported, Oracle recommends using the new name.

The value of `APEX_FILES` determines the virtual path the web server uses to point to the images directory distributed with Oracle APEX. To reference uploaded images, use `WORKSPACE_FILES` and `APP_FILES`.

Reference Type	Syntax
Bind variable	`:APEX_FILES`
Direct PL/SQL	**Note**: Direct PL/SQL is an exception. Use `G_IMAGE_PREFIX` as described in this example. `APEX_APPLICATION.G_IMAGE_PREFIX`
PL/SQL	`V('APEX_FILES')`
Substitution string	`&APEX_FILES.`
Template Substitution	`#APEX_FILES#`

> **See Also:**
>
> - APP_FILES
> - WORKSPACE_FILES
> - Editing the Application Definition

3.10.4.3 APEX$ROW_NUM

`APEX$ROW_NUM` refers the currently processed row number of a submitted tabular form data. You can use this placeholder in validations, processes, and conditions associated with a tabular form to refer to the row number of the currently processed tabular form row.

Reference Type	Syntax
Bind variable	`:APEX$ROW_NUM`
PL/SQL	`V('APEX$ROW_NUM')`
Substitution string	`&APEX$ROW_NUM.`

3.10.4.4 APEX$ROW_SELECTOR

Use `APEX$ROW_SELECTOR` in validations, processes, and conditions associated with a tabular form to refer to the row selector checkbox in a tabular form. This placeholder returns `X` if the tabular form row selector checkbox of the currently processed tabular form row is checked and `NULL` if it unchecked.

Reference Type	Syntax
Bind variable	`:APEX$ROW_SELECTOR`
PL/SQL	`V('APEX$ROW_SELECTOR')`
Substitution string	`&APEX$ROW_SELECTOR.`

3.10.4.5 APEX$ROW_STATUS

Use `APEX$STATUS` in validations, processes, and conditions associated with a tabular form to refer to the row status in a tabular form. This placeholder returns the status of `C` if created, `U` if updated, or `D` if deleted for the currently processed tabular form row.

Reference Type	Syntax
Bind variable	`:APEX$ROW_STATUS`
PL/SQL	`V('APEX$ROW_STATUS')`
Substitution string	`&APEX$ROW_STATUS.`

3.10.4.6 APP_ID

`APP_ID` identifies the application ID of the currently executing application.

Reference Type	Syntax
Bind variable	`:APP_ID`
Direct PL/SQL	`APEX_APPLICATION.G_FLOW_ID (A NUMBER)`
PL/SQL	`NV('APP_ID')`
Substitution string	`&APP_ID.`
SYS_CONTEXT variable	`SYS_CONTEXT('APEX$SESSION', 'APP_ID')`

Substitution String Reference Example

The following is an example of a substitution string reference:

```
f?p=&APP_ID.:40:&APP_SESSION.
```

SYS_CONTEXT Variable Example

The following is a `SYS_CONTEXT` variable example:

```
SELECT ... WHERE application_id = SYS_CONTEXT('APEX$SESSION', 'APP_ID')
```

APEX sets up the `APEX$SESSION` context when it starts to process an incoming request. For example, you can use the value of `'APP_ID'` to access the current application ID in queries and Virtual Private Database (VPD) security policies that protect your table data.

3.10.4.7 APP_ALIAS

`APP_ALIAS` is an alphanumeric name for the current application. `APP_ALIAS` is different from the `APP_ID` in that the `APP_ID` must be unique over all workspaces and all applications hosted in one database. In contrast, `APP_ALIAS` must be unique within a workspace. For example, by using the same `APP_ALIAS` you can create the application, ABC, in two different workspaces. You can use `APP_ALIAS` almost anywhere `APP_ID` can be used. For example, `f?p` syntax can use an `APP_ALIAS` or an application ID as demonstrated in this example:

```
f?p=ABC:1:&APP_SESSION.
```

This example runs application ABC, page 1 using the current session.

Reference Type	Syntax
Bind variable	`:APP_ALIAS`
PL/SQL	`V('APP_ALIAS')`
Substitution string	`&APP_ALIAS.`

The following is an HTML example:

```
Click me to go to page 1 <a href="f?p=&APP_ALIAS.:1:&APP_SESSION."> of the current
application</a>
```

3.10.4.8 APP_AJAX_X01, ... APP_AJAX_X10

`APP_AJAX_Xnn` specifies the values of the `APP_AJAX_X01, ... APP_AJAX_X10` URL parameters most recently passed to or set within the show or accept modules. You typically use these variables in On Demand AJAX processes.

Reference Type	Syntax
Bind variable	`... :APP_AJAX_X01, ... :APP_AJAX_X10`
PL/SQL	`... v('APP_AJAX_X01'), ... v('APP_AJAX_X10')`
Substitution string	`... &APP_AJAX_X01., ... &APP_AJAX_X10.`

See Also:

`G_X01, ... G_X10` variables in APEX_APPLICATION in *Oracle APEX API Reference*

3.10.4.9 APP_BUILDER_SESSION

If the user is also logged in to the workspace as a developer, `APP_BUILDER_SESSION` contains the current session ID of the development environment. Otherwise, `APP_BUILDER_SESSION` is null.

Reference Type	Syntax
Bind variable	`:APP_BUILDER_SESSION`
PL/SQL	`V('APP_BUILDER_SESSION')`
Substitution string	`&APP_BUILDER_SESSION.`

3.10.4.10 APP_DATE_TIME_FORMAT

`APP_DATE_TIME_FORMAT` is the application date time format of the application. This value reflects the format specified in the Application Date Time Format attribute of the Globalization settings of an application. If the Application Date Time Format is not set in an application, then a reference to `APP_DATE_TIME_FORMAT` returns the database session NLS date format and the NLS time format.

Reference Type	Syntax
Bind variable	`:APP_DATE_TIME_FORMAT`
PL/SQL	`V('APP_DATE_TIME_FORMAT')`
Substitution string	`&APP_DATE_TIME_FORMAT.`

3.10.4.11 APP_FILES

> **Note:**
>
> `APP_IMAGES` has been renamed to `APP_FILES`. Although the legacy name, `APP_IMAGES`, is still supported, Oracle recommends using the new name.

Use this substitution string to reference uploaded images, JavaScript, and cascading style sheets that are specific to a given application and are not shared over many applications. If you upload a file and make it specific to an application, then you must use this substitution string, or bind variable.

Reference Type	Syntax
Bind variable	`:APP_FILES`
Direct PL/SQL	Not available.
PL/SQL	`V('APP_FILES')`
Substitution string	`&APP_FILES.`
Template substitution	`#APP_FILES#`

> **See Also:**
>
> APEX_FILES, WORKSPACE_FILES and About Managing Images

3.10.4.12 APP_NLS_DATE_FORMAT

`APP_NLS_DATE_FORMAT` is the application date format of the database session. This value reflects the format specified in the Application Date Format attribute of the Globalization settings of the application. However, if the Application Date Format is not set, then `APP_NLS_DATE_FORMAT` returns the `NLS_DATE_FORMAT` value of the database session at the start of the request to the APEX engine.

Reference Type	Syntax
Bind variable	`:APP_NLS_DATE_FORMAT`
PL/SQL	`V('APP_NLS_DATE_FORMAT')`
Substitution string	`&APP_NLS_DATE_FORMAT.`

3.10.4.13 APP_NLS_TIMESTAMP_FORMAT

`APP_NLS_TIMESTAMP_FORMAT` is the application timestamp format of the database session. This value reflects the format specified in the Application Timestamp Format attribute of the Globalization settings of the application. However, if the Application Timestamp Format is not set, then `APP_NLS_TIMESTAMP_FORMAT` return the `NLS_TIMESTAMP_FORMAT` value of the database session at the start of the request to the APEX engine.

Table 3-2 APP_NLS_TIMESTAMP_FORMAT Syntax

Reference Type	Syntax
Bind variable	`:APP_NLS_TIMESTAMP_FORMAT`
PL/SQL	`V('APP_NLS_TIMESTAMP_FORMAT')`
Substitution string	`&APP_NLS_TIMESTAMP_FORMAT.`

3.10.4.14 APP_NLS_TIMESTAMP_TZ_FORMAT

`APP_NLS_TIMESTAMP_TZ_FORMAT` is the application timestamp time zone format of the database session. This value reflects the format specified in the Application Timestamp Time Zone Format attribute of the Globalization settings of an application. However, if the Application Timestamp Time Zone Format is not set, then `APP_NLS_TIMESTAMP_TZ_FORMAT` returns the `NLS_TIMESTAMP_TZ_FORMAT` value of the database session at the start of the request to the APEX engine.

Reference Type	Syntax
Bind variable	`:APP_NLS_TIMESTAMP_TZ_FORMAT`
PL/SQL	`V('APP_NLS_TIMESTAMP_TZ_FORMAT')`
Substitution string	`&APP_NLS_TIMESTAMP_TZ_FORMAT.`

3.10.4.15 APP_PAGE_ALIAS

`APP_PAGE_ALIAS` is an alphanumeric name for the current application page. A page alias is not case-sensitive and it is an optional page attribute. `APP_PAGE_ALIAS` is unique within an application. You can use `APP_PAGE_ALIAS` almost anywhere `APP_PAGE_ID` can be used.

Reference Type	Syntax
Bind variable	:APP_PAGE_ALIAS
PL/SQL	v('APP_PAGE_ALIAS')
Substitution string	&APP_PAGE_ALIAS.

The following is an HTML example:

The alias of the current page is: `&APP_PAGE_ALIAS.`

3.10.4.16 APP_PAGE_ID

`APP_PAGE_ID` is the current application page ID. For example, if your application was on page 3, then the result would be 3. Using this syntax is useful when writing application components that must work generically in multiple applications.

Reference Type	Syntax
Bind variable	`:APP_PAGE_ID`
PL/SQL	`:APP_PAGE_ID`
PL/SQL and Direct PL	`NV('APP_PAGE_ID')`
Substitution string	`&APP_PAGE_ID.`

The following is an example of a substitution string reference:

```
f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.
```

3.10.4.17 APP_REGION_ID

`APP_REGION_ID` identifies the ID of the current executing region.

Reference Type	Syntax
Bind variable	`:APP_REGION_ID`
PL/SQL	`v('APP_REGION_ID')`
Substitution string	`&APP_REGION_ID.`

3.10.4.18 APP_REGION_STATIC_ID

`APP_REGION_STATIC_ID` identifies the static ID of the current executing region. If no static ID has been entered by a developer, NULL is returned.

Table 3-3 APP_REGION_STATIC_ID Syntax

Reference Type	Syntax
Bind variable	`:APP_REGION_STATIC_ID`
PL/SQL	`v('APP_REGION_STATIC_ID')`
Substitution string	`&APP_REGION_STATIC_ID.`

3.10.4.19 APP_REQUEST_DATA_HASH

`APP_REQUEST_DATA_HASH` is a hash value of the request, item name, and item value parts in the URL. It is primarily useful to detect whether two browser requests passed different parameters to APEX.

Table 3-4 APP_REQUEST_DATA_HASH Syntax

Reference Type	Syntax
Bind variable	`:APP_REQUEST_DATA_HASH`
PL/SQL	`V('APP_REQUEST_DATA_HASH')`
Substitution string	`&APP_REQUEST_DATA_HASH.`

3.10.4.20 APP_SESSION

`APP_SESSION` is the most commonly used built-in substitution strings. You can use this substitution string to create hypertext links between application pages that maintain a session state by passing the session number. Note that you can also use the substitution string `SESSION` in place of `APP_SESSION`.

Table 3-5 APP_SESSION Syntax

Reference Type	Syntax
Bind variable	`:APP_SESSION`
PL/SQL	`V('APP_SESSION')`
Short PL/SQL	`V('SESSION')`
Substitution string	`&APP_SESSION.`
SYS_CONTEXT variable	`SYS_CONTEXT('APEX$SESSION', 'APP_SESSION')`

Consider the following examples:

- From within an HTML region:

  ```
  <a href="f?p=100:5:&APP_SESSION.">click me</a>
  ```

- Using PL/SQL:

  ```
  htf.anchor('f?p=100:5:'||V('APP_SESSION'),'click me');
  ```

- Using a SQL query:

  ```
  SELECT htf.anchor('f?p=100:5:'||:APP_SESSION,'click me') FROM DUAL;
  ```

- Using the `SYS_CONTEXT` variable:

```
SELECT ... WHERE apex_session_id = SYS_CONTEXT('APEX$SESSION', 'APP_SESSION')
```

APEX sets up the `APEX$SESSION` context when it starts to process an incoming request. For example, you can use the value of `'APP_SESSION'` to access the current application session in queries and VPD (Virtual Private Database) security policies that protect your table data.

3.10.4.21 APP_SESSION_VISIBLE

`APP_SESSION_VISIBLE` is similar to the built-in substitution `APP_SESSION`. Use this substitution string to create hypertext links between application pages that maintain a session state by passing the session number. `APP_SESSION_VISIBLE` always returns '0' when users are not authenticated to an application and they are using the Zero Session ID feature of APEX.

Reference Type	Syntax
Bind variable	`:APP_SESSION_VISIBLE`
PL/SQL	`V('APP_SESSION_VISIBLE')`
Substitution string	`&APP_SESSION_VISIBLE.`

Consider the following examples:

- From within an HTML region:

```
<a href="f?p=100:5:&APP_SESSION_VISIBLE.">click me</a>
```

- Using PL/SQL:

```
sys.htf.anchor('f?p=100:5:'||V('APP_SESSION_VISIBLE'),'click me');
```

- Using a SQL query:

```
SELECT sys.htf.anchor('f?p=100:5:'||:APP_SESSION_VISIBLE,'clickme') FROM DUAL;
```

3.10.4.22 APP_TITLE

`APP_TITLE` is an alphanumeric title for the current application. The title is derived from an application substitution string called `APP_TITLE`. If not defined the Logo attribute will be used if it is of type text. The last fallback is the application name.

Reference Type	Syntax
Bind variable	`:APP_TITLE`
PL/SQL	`v('APP_TITLE')`
Substitution string	`&APP_TITLE.`

The following is an HTML example:

```
The title of the current application is: &APP_TITLE!HTML.
```

3.10.4.23 APP_UNIQUE_PAGE_ID

`APP_UNIQUE_PAGE_ID` is an integer generated from an Oracle sequence which is unique for each page view. This number is used by applications to prevent duplicate page submissions and can be used for other purposes. For example, to make a unique URL and avoid browser caching issues, you can embed this number in the request or debug column in calls to the `f` procedure.

Reference Type	Syntax
Bind variable	`:APP_UNIQUE_PAGE_ID`
PL/SQL	`V('APP_UNIQUE_PAGE_ID')`
Substitution string	`&APP_UNIQUE_PAGE_ID.`

The following is an HTML example:

```
SELECT 'f?p=100:1:'||:APP_SESSION||':'||:APP_UNIQUE_PAGE_ID||
    ':::P1_EMPNO:'||employee_id,
   first_name,
    job_id
FROM employees
```

Note the use of the `APP_UNIQUE_PAGE_ID` in the request column. This makes this URL unique and may avoid excessive browser caching problems.

3.10.4.24 APP_USER

`APP_USER` is the current user running the application. Depending upon your authentication model, the value of the user is set differently. If the application is running using database authentication, then the value of the user is the same as the database pseudo column USER. If the application uses an authentication scheme that requires the user to authenticate, the value of `APP_USER` is set by the authentication scheme, usually to the user name used during authentication.

Reference Type	Syntax
Bind variable	`:APP_USER`
PL/SQL	`V('APP_USER')`
Substitution string	`&APP_USER.`
`SYS_CONTEXT` variable	`SYS_CONTEXT('APEX$SESSION', 'APP_USER')`

Consider the following examples:

- From within an HTML region:

  ```
  Hello you are logged in as &APP_USER.
  ```

- Using PL/SQL:

  ```
  htp.p('Hello you are logged in as'||V('APP_USER'));
  ```

- As a bind variable:

  ```
  SELECT * FROM some_table WHERE user_id = :APP_USER
  ```

- Using the `SYS_CONTEXT` variable:

  ```
  SELECT ... WHERE username = SYS_CONTEXT('APEX$SESSION', 'APP_USER')
  ```

  APEX sets up the `APEX$SESSION` context when it starts to process an incoming request. For example, you can use the value of `'APP_USER'` to access the current application user in queries and VPD (Virtual Private Database) security policies that protect your table data.

> **See Also:**
>
> Authentication for information about the Public User attribute

3.10.4.25 APP_VERSION

Use `APP_VERSION` to get the entered version of your application. You can use `APP_VERSION` to display it in your application or append it to external static files to automatically invalidate the browser cache when the application gets updated.

Reference Type	Syntax
Substitution string	`#APP_VERSION#`

3.10.4.26 AUTHENTICATED_URL_PREFIX

This application-level attribute identifies a valid authenticated prefix (that is, a logged in URL prefix). You can use a relative path or a full path beginning with `http`. This item is useful if your application can be run in both authenticated (logged in) and public (not logged in) modes. You can use `AUTHENTICATED_URL_PREFIX` to construct a link to an authenticated page. This item is most useful when using basic database authentication because changes to the URL can require authentication.

Reference Type	Syntax
Bind variable	`:AUTHENTICATED_URL_PREFIX`
PL/SQL	`V('AUTHENTICATED_URL_PREFIX')`
Substitution string	`&AUTHENTICATED_URL_PREFIX.`

3.10.4.27 BROWSER_LANGUAGE

`BROWSER_LANGUAGE` refers to the web browser's current language preference.

Reference Type	Syntax
Bind variable	`:BROWSER_LANGUAGE`
Direct PL/SQL	`APEX_APPLICATION.G_BROWSER_LANGUAGE`
PL/SQL	`V('BROWSER_LANGUAGE')`
Substitution string	`&BROWSER_LANGUAGE.`

3.10.4.28 CURRENT_PARENT_TAB_TEXT

`CURRENT_PARENT_TAB_TEXT` is most useful in page templates, but is only relevant for applications that use two-level tabs (that is, parent and standard tabs). Use this string to reference the parent tab label. This substitution string enables you to repeat the currently selected parent tab within the page template.

Reference Type	Syntax
Bind variable	Not Available.

Reference Type	Syntax
Substitution string	`&CURRENT_PARENT_TAB_TEXT.`

3.10.4.29 DEBUG

Valid values for the `DEBUG` flag are `Yes` or `No`. Turning debug on shows details about application processing. If you write your own custom code, you may want to generate debug information only if the debug mode is set to `Yes`.

Reference Type	Syntax
Bind variable	`:DEBUG`
Direct PL/SQL	`APEX_APPLICATION.G_DEBUG`
PL/SQL	`V('DEBUG')`
Substitution string	`&DEBUG.`

The following is an example of a substitution string reference that preserves the current value of `DEBUG`:

```
f?p=100:1:&APP_SESSION.::&DEBUG
```

3.10.4.30 HOME_LINK

`HOME_LINK` is the home page of an application. The APEX engine redirects to this location if no page is given and if no alternative page is dictated by the authentication scheme's logic. You define the Home URL on the User Interface page, under **Attributes**.

Reference Type	Syntax
Direct PL/SQL	`APEX_APPLICATION.G_HOME_LINK`
PL/SQL	`V('HOME_LINK')`
Template Reference	`#HOME_LINK#`
Substitution String	`&HOME_LINK.`

> **See Also:**
>
> Editing User Interface Attributes and Attributes

3.10.4.31 JET_BASE_DIRECTORY

Use the `JET_BASE_DIRECTORY` substitution string to reference the base directory of the Oracle JavaScript Extension Toolkit (JET) which ships with APEX. Supported syntax for referencing `JET_BASE_DIRECTORY` :

```
#JET_BASE_DIRECTORY#
```

3.10.4.32 JET_CSS_DIRECTORY

Use the `JET_CSS_DIRECTORY` substitution string to reference the base directory of the Oracle JavaScript Extension Toolkit (JET) which ships with APEX. Supported syntax for referencing `JET_CSS_DIRECTORY`:

```
#JET_CSS_DIRECTORY#
```

3.10.4.33 JET_JS_DIRECTORY

Use the `JET_JS_DIRECTORY` substitution string to reference the JavaScript directory of the Oracle JavaScript Extension Toolkit (JET) components which ships with APEX. Supported syntax for referencing `JET_JS_DIRECTORY`:

```
#JET_JS_DIRECTORY#
```

3.10.4.34 LOGIN_URL

Use `LOGIN_URL` to display a link to a login page for users that are not currently logged in.

> **See Also:**
>
> Authentication and Security Page

Reference Type	Syntax
Bind variable	`:LOGIN_URL`
Direct PL/SQL	`APEX_APPLICATION.G_LOGIN_URL`
PL/SQL	`V('LOGIN_URL')`
Substitution string	`&LOGIN_URL.`
Template Substitution	`#LOGIN_URL#`

3.10.4.35 LOGOUT_URL

`LOGOUT_URL` is an application-level attribute used to identify the logout URL. This is a URL that navigates the user to a logout page or optionally directly logs out a user. To create a logout navigation bar entry, add a trailing period to `&LOGOUT_URL` (`&LOGOUT_URL.`). If you are coding a page template, use `#LOGOUT_URL#`.

Reference Type	Syntax
Bind variable	`:LOGOUT_URL`
PL/SQL	`V('LOGOUT_URL')`
Substitution string	`&LOGOUT_URL.`
Template substitution	`#LOGOUT_URL#`

3.10.4.36 MAIN_APP_ID

`MAIN_APP_ID` is an application-level attribute used to identify the main application ID if the current application is a working copy. Otherwise, it shows the current application ID.

Reference Type	Syntax
Bind variable	`:MAIN_APP_ID`
PL/SQL	`NV('MAIN_APP_ID')`
Substitution string	`&MAIN_APP_ID.`

3.10.4.37 APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang

> **Note:**
>
> The built-in substitutions described in this section have been designated as legacy. If your application's Compatibility Mode is 24.2 or later, Oracle recommends using the short substitution syntax described in Substitutions in Text Messages (24.2 or later).

With `APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang` built-in substitution, you can access application defined or system defined text messages, like `APEX_LANG.MESSAGE`. `Message_Name` is the name of the text message, `Message_Name` has to be a valid identifier (`A-Z, 0-9, $, #, _`). The optional `$Lang` parameter can be used to access a specific translation, `$Lang` defaults to the current language for the page request.

Reference Type	Syntax
Bind variable	`:APP_TEXT$Message_Name , :APP_TEXT$Message_Name$Lang`
PL/SQL	`V('APP_TEXT$Message_Name') , V('APP_TEXT$Message_Name$Lang')`
Substitution string	`&APP_TEXT$Message_Name. , &APP_TEXT$Message_Name$Lang.`

The following is an example for the default and the French text message `MY_MESSAGE`, using HTML escaping:

```
Default text: &APP_TEXT$MY_MESSAGE!HTML.
Pardon my French: &APP_TEXT$MY_MESSAGE$FR!HTML.
```

3.10.4.38 OWNER

The `#OWNER#` substitution returns the parsing schema of the APEX application. Use `#OWNER#` to reference this value in SQL queries, PL/SQL or within substitutions.

3.10.4.39 PRINTER_FRIENDLY

The value of `PRINTER_FRIENDLY` determines if the APEX engine is running in print view mode. This setting can be referenced in conditions to eliminate elements not desired in a printed document from a page.

Reference Type	Syntax
Direct PL/SQL	`APEX_APPLICATION.G_PRINTER_FRIENDLY (VARCHAR2 DATATYPE)`
PL/SQL	`V('PRINTER_FRIENDLY')`
Substitution string	`&PRINTER_FRIENDLY.`

3.10.4.40 PROXY_SERVER

`PROXY_SERVER` is an application attribute. The attribute may be used by regions whose source comes from a URL. The following is the correct syntax for a direct PL/SQL reference used when you are writing PL/SQL to access remote web servers from within the database (for example, when using the `utl_http` package shipped with the database).

```
APEX_APPLICATION.G_PROXY_SERVER
```

3.10.4.41 PUBLIC_URL_PREFIX

`PUBLIC_URL_PREFIX` is an application-level attribute that identifies a URL to toggle out of a logged in mode to a public view.

Reference Type	Syntax
Bind variable	`:PUBLIC_URL_PREFIX`
PL/SQL	`V('PUBLIC_URL_PREFIX')`
Substitution string	`&PUBLIC_URL_PREFIX.`
Template substitution	`#PUBLIC_URL_PREFIX#`

3.10.4.42 REQUEST

Each application button sets the value of `REQUEST` to the name of the button or to the request value attribute associated with the button, enabling accept processing to reference the name of the button when a user clicks it. In the `f?p` syntax, `REQUEST` may be set using the fourth argument.

`REQUEST` is typically referenced during Accept processing (that is, the processing that occurs when you post a page).

Reference Type	Syntax
Bind variable	`:REQUEST`
Direct PL/SQL	`APEX_APPLICATION.G_REQUEST`
PL/SQL	`V('REQUEST')`
Substitution string	`&REQUEST.`

> **See Also:**
>
> Using REQUEST

3.10.4.43 Using REQUEST

`REQUEST` is typically referenced during Accept processing (that is, the processing that occurs when you post a page). This section describes additional information about how to use the REQUEST substitution string.

- About Scope and Value of REQUEST for Posted Pages
- About the When Button Pressed Attribute
- About Referencing REQUEST Using Declarative Conditions
- About Using REQUEST for Show Processing
- About Using BRANCH_TO_PAGE_ACCEPT

> **See Also:**
>
> REQUEST

3.10.4.43.1 About Scope and Value of REQUEST for Posted Pages

When you post a page, you initiate Accept processing. Accept processing consists of computations, validations, processes, and branches. The value of `REQUEST` is available during each phase of the Accept processing. Once an application branches to a different page then `REQUEST` is set to NULL.

The value of `REQUEST` is the name of the button the user clicks, or the name of the tab the user selects. For example, suppose you have a button with a name of `CHANGE`, and a label `Apply Change`. When a user clicks the button, the value of `REQUEST` is `CHANGE`.

3.10.4.43.2 About the When Button Pressed Attribute

Validations, processes, and branches have a When Button Pressed attribute. This attribute displays as a select list and contains the names of buttons that exist on the current page. If you make a selection from When Button Pressed, you associate the button's `REQUEST` value with the validation, process, or branch.

When you use a button to submit a page, the `REQUEST` value is passed to the page. The Accept processing logic evaluates each validation, process, and branch that uses a When Button Pressed attribute to determine whether the component should run (or fire). When one of these components runs, do not assume that a user actually clicked the associated button and caused the page to be submitted. Keep in mind, that another button using the same request value may have submitted the page. Similarly, JavaScript on the page can also submit the page and pass in a request value.

3.10.4.43.3 About Referencing REQUEST Using Declarative Conditions

Many developers reference `REQUEST` using conditions. For example, you may want to reset pagination when a user clicks **Go** on a report page. You can reset pagination by creating an on-submit page process. The page process can be made conditional using the condition `Request = Expression 1`.

To conditionalize an on-submit page process:

1. Under Condition, select the condition type **Request = Expression 1**.
2. In Expression 1, enter **GO**.

3.10.4.43.4 About Using REQUEST for Show Processing

You can also use `REQUEST` for Show processing when navigating to a page using `f?p` syntax. For example:

```
f?p=100:1:&APP_SESSION.:GO
```

Remember that the fourth argument in the `f?p` syntax is `REQUEST`. This example goes to application 100, page 1 for the current session, and sets the value of `REQUEST` to `GO`. Any process or region can reference the value of `REQUEST` using Show processing.

The following is a similar example using PL/SQL:

```
IF V ('REQUEST') = 'GO' THEN
   htp.p('hello');
END IF;
```

Note that `htp.p('hello')` is a call to a PL/SQL Web Toolkit package to print the specified text string.

> **See Also:**
>
> *Oracle Database Development Guide*

3.10.4.43.5 About Using BRANCH_TO_PAGE_ACCEPT

You can use a special request `BRANCH_TO_PAGE_ACCEPT` for Show processing to automatically submit the page. For example:

```
f?p=100:1:&APP_SESSION.:BRANCH_TO_PAGE_ACCEPT|SAVE:::P1_DATA:value
```

Using `BRANCH_TO_PAGE_ACCEPT` is the same as navigating to page 1, entering a value into the item P1_DATA, and clicking a button that submits the page with a SAVE request.

3.10.4.44 SCHEMA OWNER

If you are generating calls to applications from within your PL/SQL code, you must reference the owner of the APEX schema. The following describes the correct syntax for a direct PL/SQL reference:

```
APEX_APPLICATION.G_FLOW_SCHEMA_OWNER
```

You may also use `#FLOW_OWNER#` to reference this value in SQL queries and PL/SQL (for example, in a region or a process).

3.10.4.45 SQLERRM

`SQLERRM` is a template substitution only available in the Applications Region Error Message. Supported syntax for a region template substitution reference:

```
#SQLERRM#
```

3.10.4.46 SYSDATE_YYYYMMDD

`SYSDATE_YYYYMMDD` represents the current date on the database server, with the *`YYYYMMDD`* format mask applied. You may use this value instead of repeated calls to the `SYSDATE()` function. The following list describes the supported syntax for referencing `SYSDATE_YYYYMMDD`.

- Bind variable

  ```
  :SYSDATE_YYYYMMDD
  ```

- PL/SQL

  ```
  V('SYSDATE_YYYYMMDD')
  ```

- Direct PL/SQL

  ```
  APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)
  ```

Reference Type	Syntax
Bind variable	`:SYSDATE_YYYYMMDD`
Direct PL/SQL	`APEX_APPLICATION.G_SYSDATE (DATE DATATYPE)`
PL/SQL	`V('SYSDATE_YYYYMMDD')`

3.10.4.47 THEME_DB_FILES

> **Note:**
>
> `THEME_DB_IMAGES` has been renamed to `THEME_DB_FILES`. Although the legacy name, `THEME_DB_IMAGES`, is still supported, Oracle recommends using the new name.

Use the `THEME_DB_FILES` substitution string to always reference files which are stored with your theme definition in the database, regardless of the "File Prefix" setting of your theme. Supported syntax for a template substitution:

```
#THEME_DB_FILES#
```

3.10.4.48 THEME_FILES

> **Note:**
>
> `THEME_IMAGES` has been renamed to `THEME_FILES`. Although the legacy name, `THEME_IMAGES`, is still supported, Oracle recommends using the new name.

Use the `THEME_FILES` substitution string to reference files which are stored with your theme definition. Supported syntax for a template substitution:

```
#THEME_FILES#
```

3.10.4.49 WORKSPACE_FILES

> **Note:**
>
> `WORKSPACE_IMAGE` has been renamed to `WORKSPACE_FILES`. Although the legacy name, `WORKSPACE_IMAGE`, is still supported, Oracle recommends using the new name.

Use this substitution string to reference uploaded images, JavaScript, and cascading style sheets that are shared over many applications within a workspace.

Reference Type	Syntax
Bind variable	`:WORKSPACE_FILES`
Direct PL/SQL	Not available
PL/SQL	`V('WORKSPACE_FILES')`
Substitution string	`&WORKSPACE_FILES.`
Template substitution	`#WORKSPACE_FILES#`

> **See Also:**
>
> APP_FILES and APEX_FILES

3.10.4.50 WORKSPACE_ID

Use this substitution string to reference the workspace ID.

Reference Type	Syntax
Bind variable	`:WORKSPACE_ID`
PL/SQL	`V('WORKSPACE_ID')`
Substitution string	`&WORKSPACE_ID.`
`SYS_CONTEXT` variable	`SELECT ... WHERE workspace_id = SYS_CONTEXT('APEX$SESSION', ''WORKSPACE_ID')`

Consider the following examples:

- From within an HTML region:

  ```
  Hello your workspace id is &WORKSPACE_ID.
  ```

- Using PL/SQL:

  ```
  htp.p('Hello your workspace id is '||V('WORKSPACE_ID'));
  ```

- Using a bind variable:

  ```
  SELECT * FROM some_table WHERE workspace_id = :WORKSPACE_ID
  ```

- Using the `SYS_CONTEXT` variable:

  ```
  SELECT ... WHERE workspace_id = SYS_CONTEXT('APEX$SESSION', 'WORKSPACE_ID')
  ```

APEX sets up the `APEX$SESSION` context when it starts to process an incoming request. For example, you can use the value of `'WORKSPACE_ID'` to access the current workspace ID value in queries and VPD (Virtual Private Database) security policies that protect your table data.

3.11 Using Template Directives

Use template directives to control how attributes that support substitution strings are processed.

- About Template Directives
  Learn about template directives.
- If Condition Directives
  Use the `if` directive to conditionally show text based on if an item or column has a value.
- Case Condition Directives
  Use the `case` directive to show text based on the value of an item or column.
- Loop Directives
  Use the `loop directive` to repeat text once for each item in a multi-value (character delimited) item or column value.
- With and Apply Directives
  Use the `with` directive to assign values to placeholders for the Template Component that is specified in the `apply` directive.
- Built-in Substitutions for Template Directives
  Learn about built in substitution strings available for template directives.

3.11.1 About Template Directives

Learn about template directives.

Template directives are only supported by Template Component Plug-ins, specific attributes in email templates, and the following regions: cards, interactive grid, classic report, application search, and interactive report region. Directives can be nested which means any of the template texts can contain another directive. These directives are processed as part of client or server-side substitutions.

To determine if an attribute supports template directives, Page Designer Help includes the text "Supports Client-side Template Directives" or "Supports Server-side Template Directives." For more details on directive syntax, see apex.util.applyTemplate in *Oracle APEX JavaScript API Reference*.

> **Note:**
>
> Syntax description text in this topic is noted by the use of square brackets and is optional. You do not actually type the square brackets. Upper case text represents something as described in the description.

3.11.2 If Condition Directives

Use the `if` directive to conditionally show text based on if an item or column has a value.

Syntax

```
{if [!][?|=]NAME/}
TRUE_TEMPLATE_TEXT
{elseif [!][?|=]NAME2/}
ELSE_TRUE_TEMPLATE_TEXT
{else/}
FALSE_TEMPLATE_TEXT
{endif/}
```

Where:

- The *NAME* is an item, column name, or template component attribute.
- If the value is true then the text that follows is output.
- The value is false if it is an empty string or ‘F’, ‘N’, or ‘0’.
- Any value that is not false is true.
- If the *NAME* is proceeded by an exclamation mark (!) then the logic is negated and the following template text is processed if the value is false.
- A value is false if after trimming leading and trailing spaces it is an empty string.

The `if` condition directive handles both empty (or not empty) tests and Boolean true/false tests (using the convention of character true/false values such as `'Y'/'N'`) at the same time. This results in confusion for rare case where the intention is to test for not empty but the actual value is 'N', which is not empty but still considered false. The optional `'?'` prefix operator can be used to explicitly test if the value is empty. The optional `'='` prefix operator can be used to explicitly test if the value is true or false.

The following table summarizes the available condition directives.

Condition Directive	Description	PL/SQL Expression
`{if MYITEM/}`	`MYITEM` has to contain a value and the value can not be `'F'`, `'N'`, or `'0'`.	`:MYITEM is not null and upper( :MYITEM ) not in ( 'F', 'N', '0' )`
`{if ?MYITEM/}`	`MYITEM` has to contain a value.	`:MYITEM is not null`
`{if !MYITEM/}`	`MYITEM` can be empty or contain `'F'`, `'N'`, or `'0'`.	`:MYITEM is null or upper( :MYITEM ) in ( 'F', 'N', '0' )`
`{if !?MYITEM/}`	`MYITEM` is empty.	`:MYITEM is null`
`{if =MYITEM/}`	`MYITEM` is empty or does NOT contain `'F'`, `'N'`, or `'0'`.	`:MYITEM is null or upper( :MYITEM ) not in ( 'F', 'N', '0' )`
`{if !=MYITEM/}`	`MYITEM` contains a value which equals to `'F'`, `'N'`, or `'0'`.	`upper( :MYITEM ) in ( 'F', 'N', '0' )`

Condition Directive	Description	PL/SQL Expression
`{if MYITEM%assigned/}`	Useful when consistent output is required for multiple rows of data. When a value like `&COLUMN_NAME.` is assigned to `MYITEM`, it might be empty for some rows after applying substitutions. The `%assigned` condition evaluates to true for all rows in this case.	n/a

There can be zero or more `elseif` directives. The `elseif` and `else` directive are optional. The directives must go in the order shown.

Example

A cards report contains a column named `DESCRIPTION`. The following HTML Expression attribute will display the description if it is not null (empty string) and “No description.” otherwise.

```
{if DESCRIPTION/}&DESCRIPTION.{else/}No description.{endif/}
```

3.11.3 Case Condition Directives

Use the `case` directive to show text based on the value of an item or column.

Syntax

```
{case NAME/}
{when STRING1/}
TEMPLATE_TEXT1
{when STRING2/}
TEMPLATE_TEXT2
{otherwise/}
TEMPLATE_TEXT
{endcase/}
```

The *`NAME`* is an item, column name, or template component attribute. The value is compared with the strings after each `when` directive and if they are equal then the following text is output. If no `when` directive matches then the text after the `otherwise` directive, if there is one, is output. The value and each string is trimmed of leading and trailing spaces before comparison. The comparison is case sensitive.

Example

This example using the sample `EMP` table displays the compensation differently depending on the `JOB`. For sales people it shows both the salary and commission. For the president it shows “--” rather than salary and for all other jobs it shows just the salary.

```
{case JOB/}
{when SALESMAN/}
&SAL. (&COMM.)
{when PRESIDENT/}
```

```
--
{otherwise/}
&SAL.
{endcase/}
```

3.11.4 Loop Directives

Use the `loop directive` to repeat text once for each item in a multi-value (character delimited) item or column value.

Syntax

```
{loop ["SEP"] NAME/}
TEMPLATE_TEXT
{endloop/}
```

The *`NAME`* is an item or column name that has multiple values separated by the character given by `SEP`. The default separator is `":"`. If `SEP` is more than one character it is treated as a regular expression.

The template text within the loop can use these substitutions:

- `APEX$ITEM` - This is the value of the current item in the list.
- `APEX$I` - This is 1 based index of the current item in the list.

Example

The following example takes a column called `TAGS` that contains a comma (,) separated list of tags such as "apples,cherries,pears" and turns it into an HTML list that can be nicely styled with CSS.

```
<ul class="tags">{loop "," TAGS/}
  <li class="tag-item">&APEX$ITEM.</li>
{endloop/}</ul>
```

3.11.5 With and Apply Directives

Use the `with` directive to assign values to placeholders for the Template Component that is specified in the `apply` directive.

Syntax

```
{with/}
PLACEHOLDER1:=EXPRESSION1
PLACEHOLDER2:=EXPRESSION2
{apply TEMPLATE_INTERNAL_NAME/}
```

Requirements when using this directive include:

- Use the `with` directive to assign values to placeholders for the Template Component plug-in that is specified in the `apply` directive.
- Template Component plug-ins use Server-side rendering.

- The use of Template Directives within value expressions is supported.

Example

```
{with/}
TYPE:=IMAGE
IMAGE:=&IMAGE_URL.
ALT:={if DESCRIPTION/}&DESCRIPTION.{else/}No description.{endif/}
{apply THEME$AVATAR/}
```

In this example, `THEME$AVATAR` is the internal name of a Template Component plug-in installed in the application. This template has three placeholders, `TYPE`, `IMAGE` and `ALT` to which values are assigned.

3.11.6 Built-in Substitutions for Template Directives

Learn about built in substitution strings available for template directives.

> **Note:**
>
> Template Directives in Template Component plug-ins also support a number of additional built-in substitutions. To learn more, see item help.

Substitution String	Description
`APEX$ITEM`	Available within a loop directive. Specifies the value of the current item in the list.
`APEX$I`	Available within a loop directive. Specifies the 1 based index of the current item in the list.
`APEX$DOM_ID`	Generates a unique random `DOM ID` for the current execution of the Template Directive snippet. Example usage: `<h1 id="#APEX$DOM_ID#_heading">#HEADING#</h1><someelement aria-labelledby="#APEX$DOM_ID#_heading">`

4

Managing Team Development

Use Team Development to track and manage issues in a conversational fashion. You can create templates to provide users with starter text for issues and comments. You can apply labels and label groups, assign issue owners, and set milestones.

- Getting Started with Team Development
  Access Team Development, complete the setup process, and learn about the Team Development toolbar.
- Creating and Updating Issues
  Create and update issues by either updating issues one at a time, or performing bulk updates of multiple issues at once.
- Viewing Your User Profile
  Click the Notifications icon on the Team Development toolbar view and manage your own notifications and view all issues you own.
- Managing Labels
  Create and update label groups and labels to classify issues.
- Managing Milestones
  Create and update issues milestones to track events.
- Managing Templates
  Create and update templates for issues and issue comments.
- Using Team Development Utilities
  Use Utilities page to manage deleted issues, manage templates, milestones, and labels, view uploaded files, purge data, reassign issues to another user, reassign issue owners, remove a milestone or label from all issues, and manage user notifications.
- Managing Feedback
  Feedback provides a mechanism for end users to post general comments for application administrators and developers.

4.1 Getting Started with Team Development

Access Team Development, complete the setup process, and learn about the Team Development toolbar.

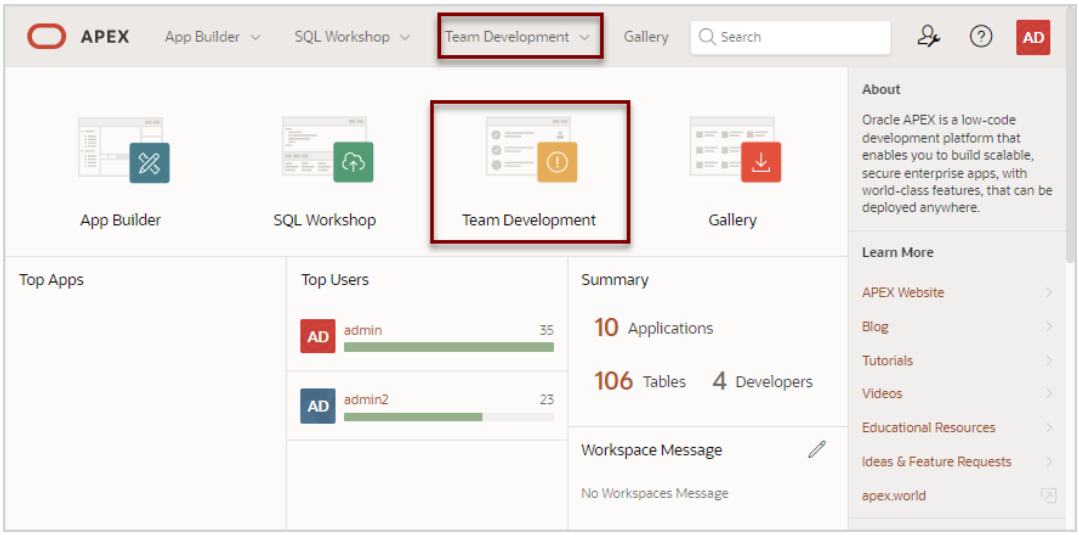
- Accessing Team Development
  Access Team Development by clicking the Team Development menu, or by clicking the Team Development icon on the Workspace home page.
- Setting Up Team Development
  Complete the setup process by installing sample labels, sample templates, sample milestones, and selecting a default language.
- Team Development Toolbar
  Use the toolbar at the top of the most Team Development pages to link to notifications, jump to other pages, or create a new issue.

4.1.1 Accessing Team Development

Access Team Development by clicking the Team Development menu, or by clicking the Team Development icon on the Workspace home page.

You can access Team Development in two ways:

- Click the **Team Development** menu at the top of most pages in Oracle APEX.
- On the Workspace home page, click the **Team Development** icon.



The Issues page appears.

> **See Also:**
>
> Issues Page

4.1.2 Setting Up Team Development

Complete the setup process by installing sample labels, sample templates, sample milestones, and selecting a default language.

When you access Team Development for the first time in a new workspace, a welcome screen appears. Oracle recommends you follow the on-screen instructions and completing the setup process.

To setup Team Development:

1. Access Team Development.

   The Team Development Welcome page appears.

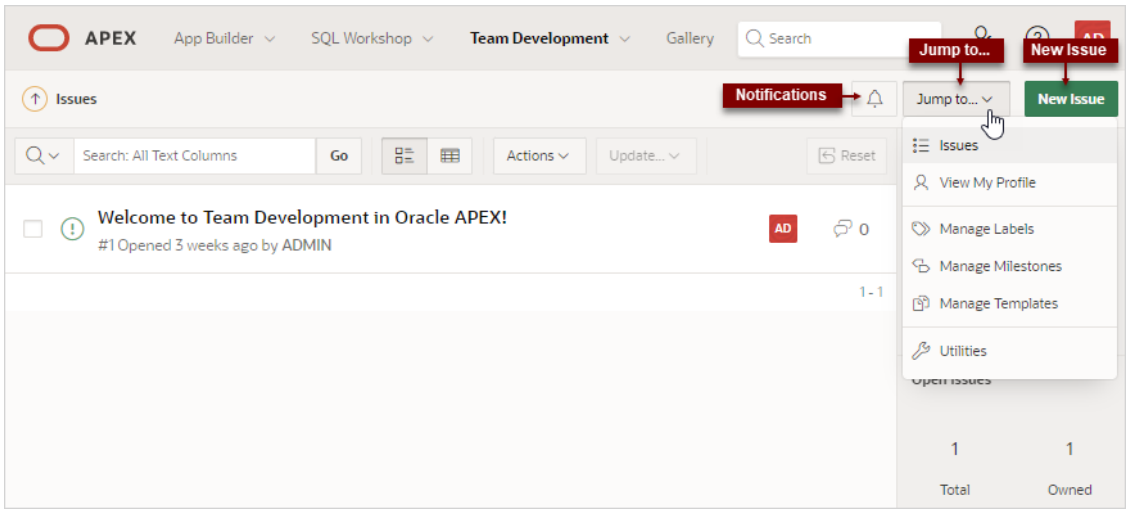
2. For Install Sample Labels, Install Sample Templates, and Install Sample Milestones, select **Yes**.
3. From Sample Data Language, select a language.
4. Click **Complete Setup**.

   The Issues page appears.

4.1.3 Team Development Toolbar

Use the toolbar at the top of the most Team Development pages to link to notifications, jump to other pages, or create a new issue.

Team Development Toolbar



The Team Development toolbar contains the following controls:

- **Notifications** - Links the Notifications page where you can view or clear notifications for all issues which you own or are monitoring. See Viewing Your Notifications.
- **Jump to...** - Access other Team Development pages. Available links include:
  - **Issues** - Links to the Issues page. See Issues Page.
  - **View My Profile** - Links to the User Profile page where you can view a report of all issues you own or are monitoring. See Viewing Your Issues.
  - **Manage Labels** - Links to the Labels page. Use labels and label groups to classify issues. See Managing Labels.
  - **Manage Milestones** - Links to the Milestones page. Create milestones to track events. See Managing Milestones.
  - **Manage Templates** - Links to the Templates page. Templates provide starter text for issues and issue comments. See Managing Templates.
  - **Utilities** - Links to the Team Development Utilities page. See Using Team Development Utilities.

4.2 Creating and Updating Issues

Create and update issues by either updating issues one at a time, or performing bulk updates of multiple issues at once.

- Issues Page
  Use the Issues page to view issues.
- Creating an Issue
  Create new issues on the Issues page by clicking the New Issue button.

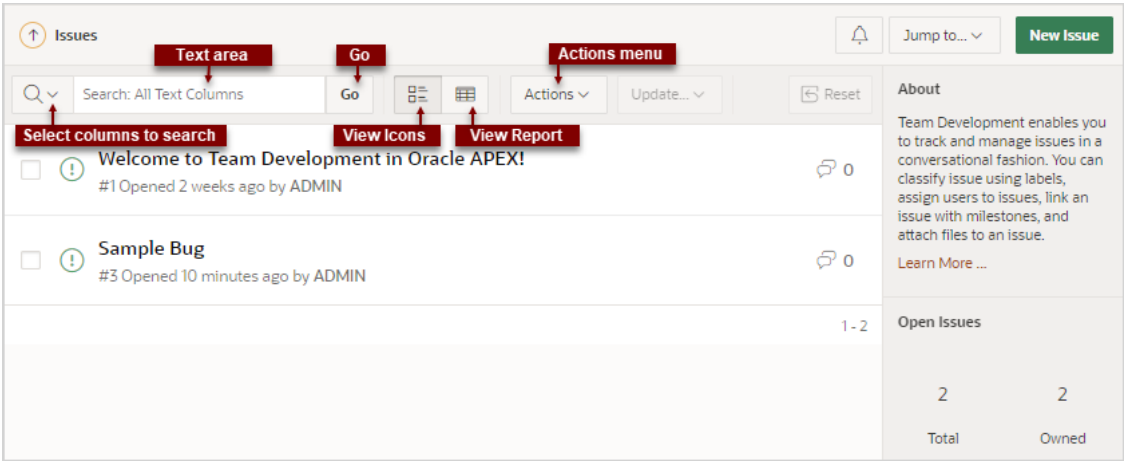
- Updating Multiple Issues at Once
  Update status, assignees, labels, and milestones for multiple issues at once by selecting the issues on the Issues page and then selecting options from the Update menu.
- Updating an Issue
  Update an issue by viewing the Issue Details and then updating the Issue Timeline.

4.2.1 Issues Page

Use the Issues page to view issues.

The Issues page displays all issues being tracked in Team Development.

Navigation Bar



A navigation bar displays at the top of the Issues page and contains the following controls.

- **Select columns to search** - This control features a menu with a magnifying glass. Click this menu to narrow your search to only specific columns. To search all columns, select **All Text Columns**.
- **Search: All Text Columns** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go** button - Executes a search or applies a filter.
- **View icons**. Use these icons to toggle between two views:
  - **Cards** (default) - Displays issue details as card.
  - **Grid** - Displays issues in an interactive grid report.

  To view issue details, click the issue title. To update status, assignees, labels, or milestones in either view, select the issue and then appropriate option on the Update menu.
- **Actions** menu - Use the Actions menu to customize the report.
- **Update**. Update multiple issues at once by selecting them and then selecting an option from the Actions menu. Update menu options include:
  - **Status** - Apply a status (**Open** or **Closed**) to all selected issues.
  - **Assignees** - Add or remove assignees for all selected issues.
  - **Labels** - Assign or remove labels for all selected issues.
  - **Milestones** - Assign or remove milestones for all selected issues.

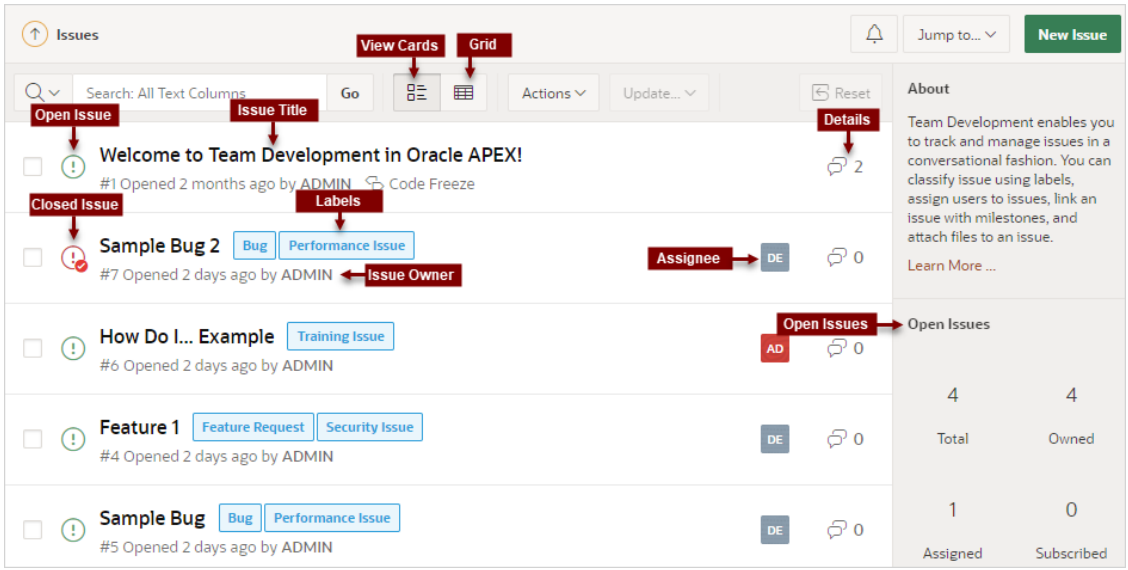
> **Tip:**
>
> To view or edit an issue (including adding issue comments), click issue title.

- **Reset** - Click the **Reset** button to remove any customizations, such as filters or sorting, Select this option to reload the report definition from the server.

> **See Also:**
>
> Team Development Toolbar

List of Issues

Issues appear next and can be viewed as Cards or as an interactive grid. Click the **View Cards** and **Grid** icons to change the display. This example shows **View Cards**.



Key features of the Cards view include:

- **Issue Title** - Click the issue title to view and update details.
- **Open Issue/Closed Issue** icons - Displays the status of the issue. The **Open Issue** icon appears as a green circle around an exclamation point. The **Closed Issue** icon displays as a red circle around an exclamation point and a check mark.
- **Issue Owner** - Displays beneath the issue title.
- **Labels** - Assigned labels display to the right of the issue title. The color indicates the associated label group.
- **Assignee** - The assignees or an issue appear to the right of the labels.
- **Issue Details** - Click the Comment icon/number on the far right to link to the Issue Details. The number indicates the number of issue comments.

Open Issues and Feedback Regions

Summary regions display on the right-side of the page. The **Open Issues** summary region includes the following:

- **Total** - Total issues in the workspace.
- **Owned** - Total issues owned by the current user.
- **Assigned** - Total issues assigned to the current user.
- **Subscribed** - Total issues to which the current user subscribes.

The **Feedback** summary region displays on the bottom right of the page and includes the following:

- **Total** - Total feedback entries.
- **Open** - Total open feedback entries.

> **See Also:**
>
> - Creating and Updating Issues
> - Team Development Toolbar
> - Managing Feedback

4.2.2 Creating an Issue

Create new issues on the Issues page by clicking the New Issue button.

> **Tip:**
>
> To create a new issue, you must select a template. Templates provide starter text for issues and issue comments. Use the sample templates provided when you completed Team Development setup, or create your own. See Managing Templates.

To create an issue:

1. Click the **Team Development** menu.
2. On the Issues page, click **New Issue**.
3. In the New Issues dialog:
   a. Select a template, or select **Continue without a template**. Templates provide users with starter text for issues and comments.
   b. Title - Enter a descriptive title.
   c. Issue Text - Enter a description of the issue. Use the Markdown tools above the field to add formatting.
   d. To preview the HTML, click **Preview**.
4. Click **Create Issue**.

See Also:

- Adding Issue Comments or Attaching a File
- Managing Templates

4.2.3 Updating Multiple Issues at Once

Update status, assignees, labels, and milestones for multiple issues at once by selecting the issues on the Issues page and then selecting options from the Update menu.

To update multiple issues at once:

1. Click the **Team Development** menu.

   The Issues page appears.
2. Select the issues to update by clicking the checkbox to the left the issue title.
3. To update status:
   a. From the Update menu, select **Status**.
   b. Status - Select a new status (that is, **Open** or **Closed**).
   c. Click **Apply Changes**.
4. To update assignees:
   a. From the Update menu, select **Assignees**.
   b. For each assignee, select **Assign** or **Remove**.
   c. Click **Apply Changes**.
5. To update labels:
   a. From the Update menu, select **Labels**.
   b. For each label, select **Assign** or **Remove**.
   c. Click **Apply Changes**.
6. To update milestones:
   a. From the Update menu, select **Milestones**.
   b. For each milestone, select **Assign** or **Remove**.
   c. Click **Apply Changes**.

4.2.4 Updating an Issue

Update an issue by viewing the Issue Details and then updating the Issue Timeline.

Only an issue owner or an administrator can update an issue title or description, or delete an issue.

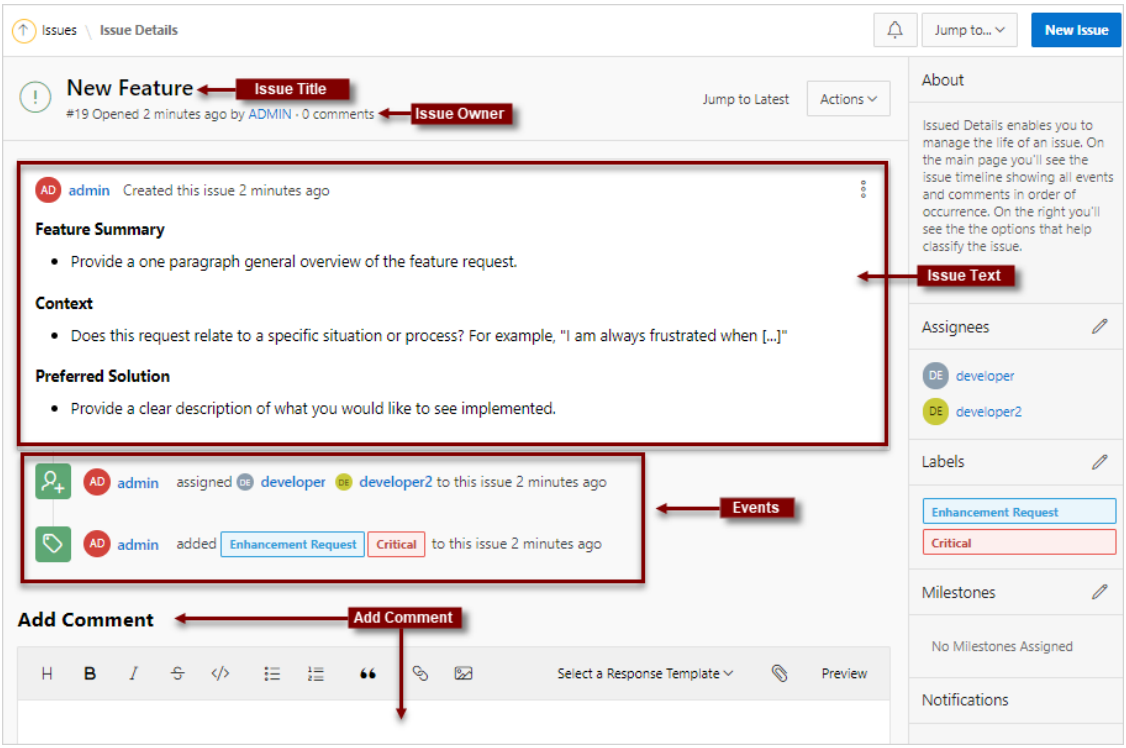
- Issue Details Page
  Use the Issue Details page to update issues.
- Updating the Issue Title
  View an issue and then update the issue title.

- Updating the Issue Description
  View an issue and update the issue description (also known as the Issue Text).
- Managing Comments
  Add or edit issue comments on the Issue Details page.
- Setting Assignees
  View an issue and then set assignees.
- Setting Labels
  View an issue and then set labels.
- Setting Milestones
  View an issue and then set milestones.
- Associating an Issue with an App and Page
  View an issue and then associate it with an application and page.
- Marking an Issue as Duplicate
  View an issue and then mark it as duplicate.
- Closing an Issue
  Close an issue on the Issues Details page.
- Managing Deleted Issues
  Issue owners or administrators can delete an issue on the Issue Details page. Manage deleted issue (that is, delete them forever or reinstate them) by navigating to Utilities, Manage Deleted Issues.

4.2.4.1 Issue Details Page

Use the Issue Details page to update issues.

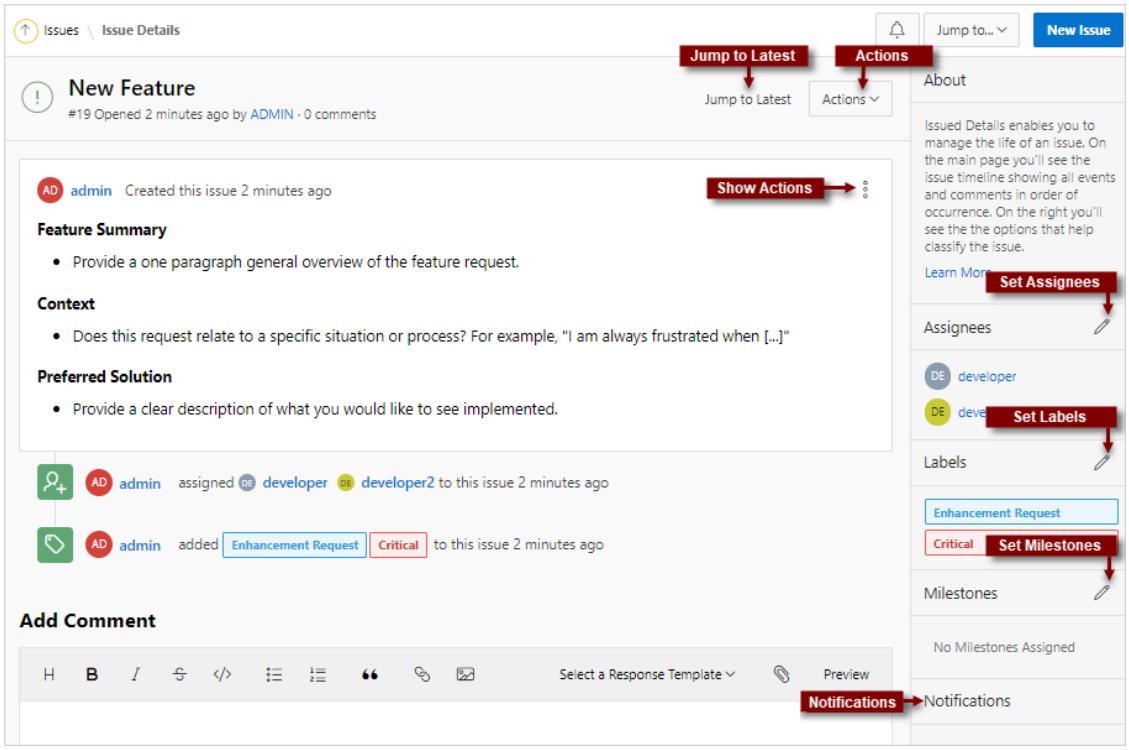
Issues Timeline



The Issue Details page contains an **Issue Timeline**. Updates to the Issue Timeline are called **events**. Newer events display at the top of the timeline. The Issue Timeline is divided into the following sections:

- **Issue Title** - Displays at the top of the page.
- **Issue Text** field - The text that describes the issue.
- **Events** - Each update to the Issue Timeline is logged as an event. Events displays after the Issue Text.
- **Add Comment** - Displays at the bottom of the Issue Timeline. Use this field to record comments about the issue. The Add Comment region includes formatting tools, enables you to select a Response Template, and Preview the text.

Issue Details Controls



General controls on the Issue Details page include:

- **Jump to Latest** - Links to the Add Comment section at the bottom of the page.
- **Actions** - Use Actions menu options to update the issue. Actions menu options include:
  - **Edit Title**
  - **Edit Issue Text**
  - **Set Assignees**
  - **Set Labels**
  - **Set Milestones**
  - **Associate App or Page**
  - **Mark as Duplicate**
  - **Delete Issue**
  - **Close Issue**

> **Note:**
>
> Only an issue owner or an administrator can update an issue title or description, or delete an issue.

- **Show Actions** - Shortcut menu containing two options: **Edit** and **Delete Issue**. **Edit** functions the same as **Edit Issue Text** on the Actions menu.
- **Set Assignees** - Shortcut menu in the right region that functions the same as **Set Assignees** on the Actions menu.
- **Set Labels** - Shortcut menu in the right region that functions the same as **Set Labels** on the Actions menu.
- **Set Milestones** - Shortcut menu in the right region that functions the same as **Set Milestones** on the Actions menu.
- **Notifications** - Enable **Subscribe** to receive notifications regarding updates to the current issue.
- **Attachments** - Click **Attachments** to view attachments associated with this issue.

4.2.4.2 Updating the Issue Title

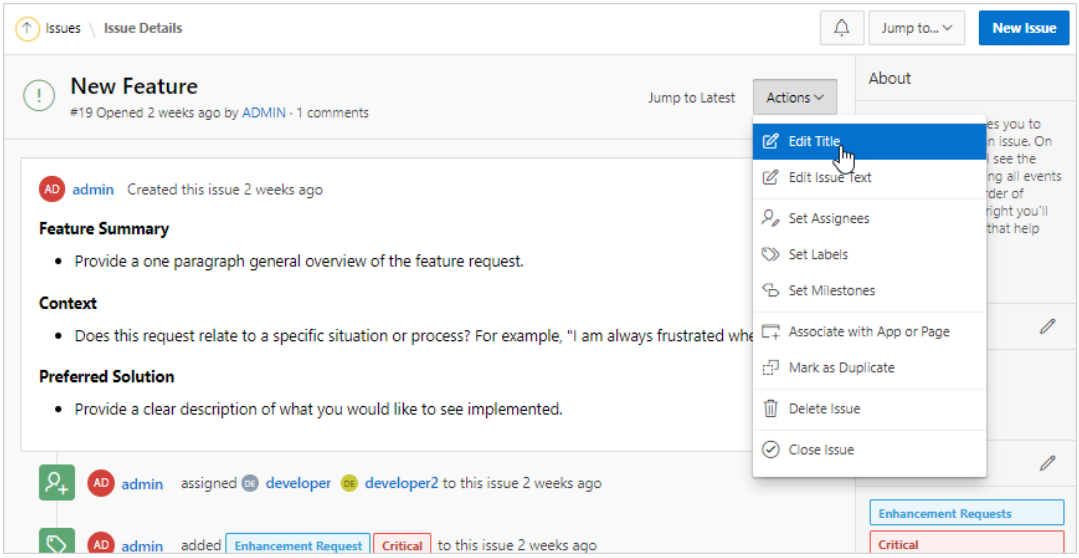
View an issue and then update the issue title.

Only an issue owner or an administrator can update an issue title.

To update the issue title:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. From the **Actions** menu, select **Edit Title**.



3. In Edit Issue Title dialog:
   a. Issue Title - Update the title.

b. Click **Apply Changes**.

4.2.4.3 Updating the Issue Description

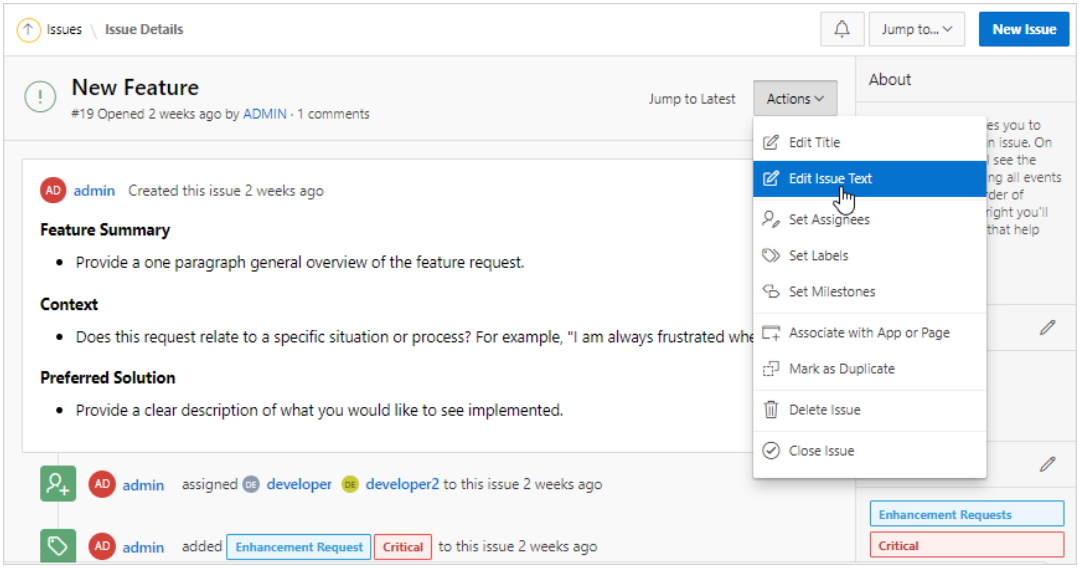
View an issue and update the issue description (also known as the Issue Text).

Only an issue owner or an administrator can update an issue description (or text).

To update the Issue Text:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears. You can update an Issue Text by either selecting **Edit Issue Text** from the Actions menu, or by clicking **Show Actions**, **Edit**.
2. From the **Actions** menu, select **Edit Issue Text**.



3. In the Edit Issue dialog:
   a. Update the description of the issue. Use the Markdown tools above the field to add formatting.
   b. Click **Preview** to preview the HTML.
   c. Click **Apply Changes**.

4.2.4.4 Managing Comments

Add or edit issue comments on the Issue Details page.

- Adding Issue Comments or Attaching a File
  View an issue and then add comments.
- Editing Issue Comments
  View an issue to edit, hide, or delete existing comments.

4.2.4.4.1 Adding Issue Comments or Attaching a File

View an issue and then add comments.

To add issue comments:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. Scroll down to **Add Comment**.

   > **Tip:**
   >
   > You can also click the **Jump to Latest** button adjacent to the issue title

3. Enter a comment. Under **Add Comment**:
   a. Select a Response Template - Select a template if appropriate.
   b. Enter comments in the field provided.
   c. To add formatting (such as bold or italic), click the icons in the toolbar. To preview the rendered HTML, click the **Preview** button.
4. Attach a file. Under **Add Comment**, click **Attach Files**.

   > **Tip:**
   >
   > The **Attach Files** icon does not appear unless you have enabled the workspace-level attribute, **Enable Issue File Repository**.

5. To save your changes:
   - If the issue is open:
     - Click **Comment** to save your changes.
     - Click **Comment and Close** to save your changes and close the issue.
   - If the issue is closed:
     - Click **Comment** to save your changes.
     - Click **Comment and Close** to save your changes and close the issue.

> **See Also:**
>
> - Closing an Issue
> - Configuring Team Development in an Existing Workspace in *Oracle APEX Administration Guide*

4.2.4.4.2 Editing Issue Comments

View an issue to edit, hide, or delete existing comments.

To edit issue comments:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. Scroll down to the comment.
3. Find the **Show actions** menu which resembles three stacked circles. Click the **Show actions** menu and select one of the following:
   - **Edit** - Only available for administrators or the comment creator.
   - **Hide/Unhide** - Only available for administrators.
   - **Delete**- Only available for administrators or the comment creator.
4. To save your changes, click **Apply**.
5. To exit without saving your changes, click the **Close** icon (that is, the "X" enclosed with a circle) in the upper right.

> **See Also:**
>
> Closing an Issue

4.2.4.5 Setting Assignees

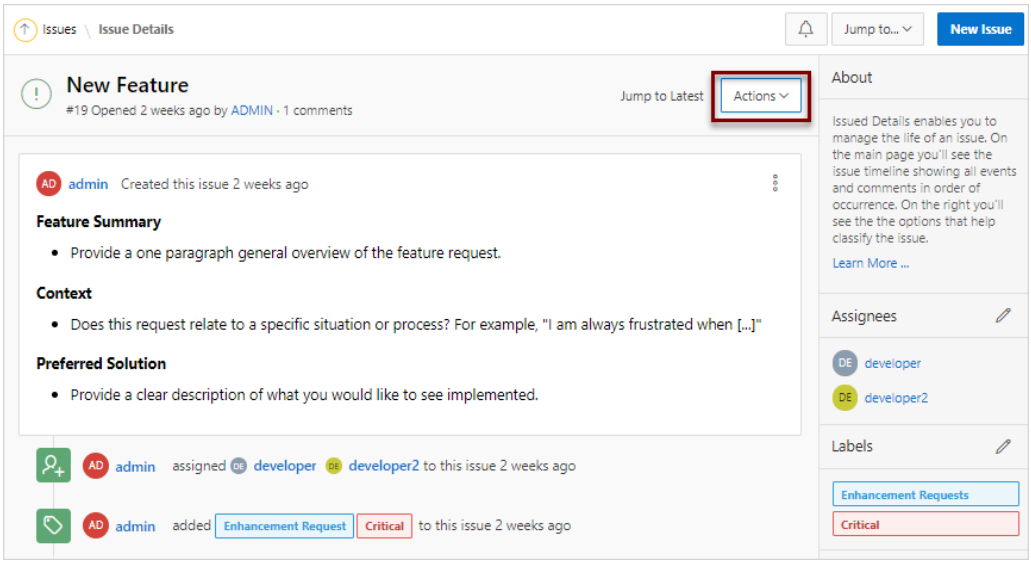
View an issue and then set assignees.

Assignees are users in the Oracle APEX workspace. To create a new assignee, a workspace administrator or Instance administrator must create a new user.

To set assignees:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.



2. From the **Actions** menu, select **Set Assignees**.

   You can also click the **Set Assignees** icon on far right side of the page.

3. In the Assignees dialog:

   a. Select assignees to add to the issue.

   b. Click **Apply Changes**.

   A new event is added to the Event Timeline and the new assignees display under **Assignees** in the far right region.

> **See Also:**
>
> Managing Users in a Workspace and Managing Users Across an APEX Instance in *Oracle APEX Administration Guide*

4.2.4.6 Setting Labels

View an issue and then set labels.

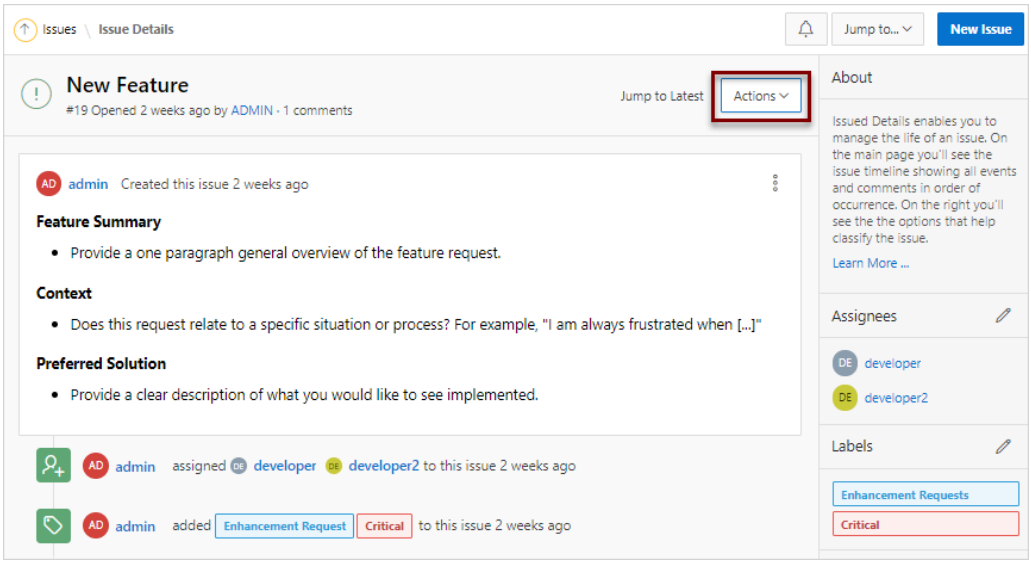
To set labels:

1. View the issue.

   a. Click the **Team Development** menu.

   b. Click the issue title.

   The Issue Details page appears.



2. From the **Actions** menu, select **Set Labels**.

   You can also click the **Set Labels** icon on far right side of the page.

3. In the Labels dialog:

   a. Select labels to add to the issue.

   b. Click **Apply Changes**.

   A new event is added to the Event Timeline and the new labels display under **Labels** in the far right region.

> **See Also:**
>
> Managing Labels

4.2.4.7 Setting Milestones

View an issue and then set milestones.

Use Milestones to track events. You can assign one or more milestones to an event.

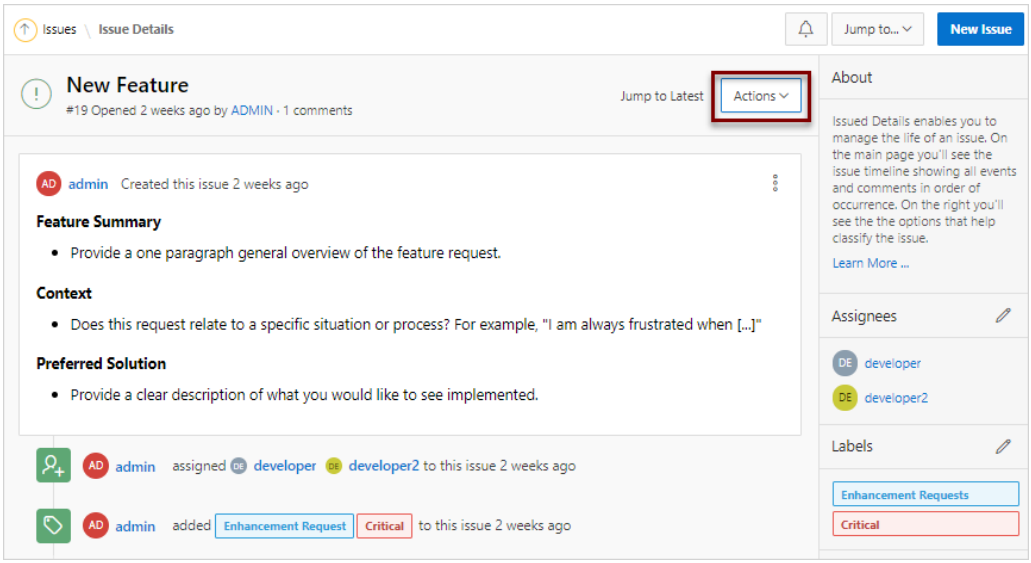
To set milestones:

1. View the issue:

   a. Click the **Team Development** menu.

   b. Click the issue title.

   The Issue Details page appears.



2. From the **Actions** menu, select **Set Milestones**.

   You can also click the **Set Milestones** icon on far right side of the page.

3. In the Milestones dialog:

   a. Select milestones to add to the issue.

   b. Optionally select **Show Closed Milestones**.

   c. Click **Apply Changes**.

   A new event is added to the Event Timeline and the new milestones display under **Milestones** in the far right region. The color of the milestone indicates the following:

   - Red - The milestones date is in the past.
   - Yellow - The milestones date is within two weeks.
   - Green - The milestones date is more than two weeks 2 weeks in the future.

> **See Also:**
>
> Managing Milestones

4.2.4.8 Associating an Issue with an App and Page

View an issue and then associate it with an application and page.

To associate an issue with an application and page:

1. View the issue:

   a. Click the **Team Development** menu.

   b. Click the issue title.

   The Issue Details page appears.

2. From the **Actions** menu, select **Associate with App and Page**.

3. In the Associate App/Page dialog:

a. Application & Page Search - Select an application
b. Page - Select a page.
c. Comments - Enter any comments.
d. Click **Apply**.

A new event is added to the Event Timeline.

4.2.4.9 Marking an Issue as Duplicate

View an issue and then mark it as duplicate.

To mark an issue as duplicate:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. From the **Actions** menu, select **Mark as Duplicate**.
3. In the Mark Duplicate dialog:
   a. Duplicate Issue - Select the duplicate issue.
   b. Click **Apply**.

   A new event is added to the Event Timeline identified by a an icon which resembles two green pages.

4.2.4.10 Closing an Issue

Close an issue on the Issues Details page.

- Closing an Issue
  View an issue and then close it.
- Viewing or Reopening a Closed Issue
  View or reopen a closed issue on the Issues page.

4.2.4.10.1 Closing an Issue

View an issue and then close it.

To close an issue:

1. View the issue:
   a. Click the **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. From the **Actions** menu, select **Close Issue**.

   The icon adjacent to the issue title changes from green to red and contains a red check mark. In addition, a new event is added to the Event Timeline which reads:

   ```
   set status for this issue to CLOSED Now
   ```

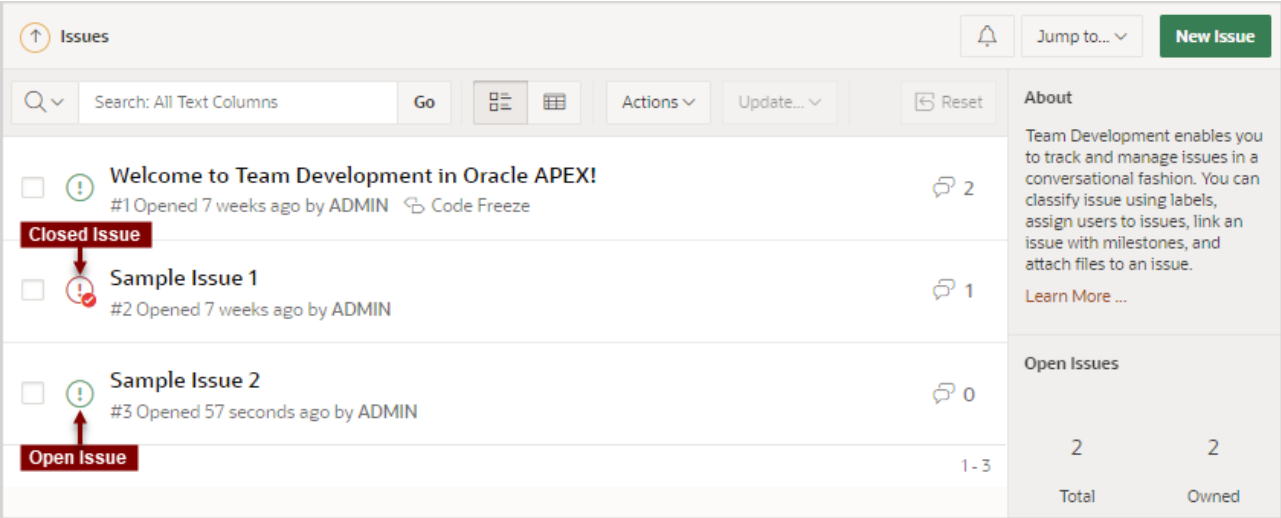
4.2.4.10.2 Viewing or Reopening a Closed Issue

View or reopen a closed issue on the Issues page.

To view or reopen a closed issue:

1. Click the **Team Development** menu.

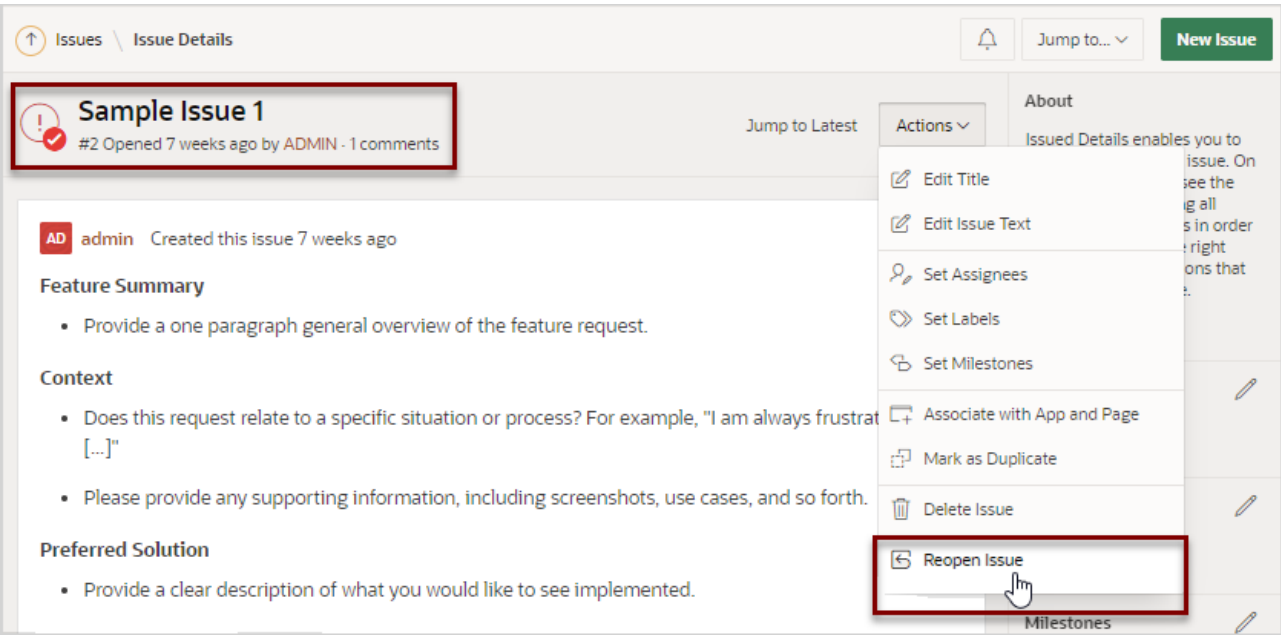
   The Issues page appears. Open issues have a green icon and closed issues have a red icon and check mark.



2. Find the closed issue and click the issue title.

   The Issue Details page appears.

3. From the **Actions** menu, select **Reopen Issue**.



The icon adjacent to the issue title changes from red to green and the red check mark disappears.

A new event is added to the Event Timeline which reads:

```
set status for this issue to OPEN Now
```

4.2.4.11 Managing Deleted Issues

Issue owners or administrators can delete an issue on the Issue Details page. Manage deleted issue (that is, delete them forever or reinstate them) by navigating to Utilities, Manage Deleted Issues.

- Deleting an Issue
  Issue owners or administrators can delete an issue.
- Reinstating Deleted Issues
  Reinstate deleted issues on the Manage Deleted Issues Page.
- Permanently Remove Deleted Issues
  Permanently remove deleted issues on the Manage Deleted Issues page.

4.2.4.11.1 Deleting an Issue

Issue owners or administrators can delete an issue.

To delete an issue:

1. View the issue:
   a. **Team Development** menu.
   b. Click the issue title.

      The Issue Details page appears.
2. From the **Actions** menu, select **Delete Issue**.

   A confirmation page appears asking if you are sure you want to delete the issue.
3. On the confirmation page, click **OK**.

4.2.4.11.2 Reinstating Deleted Issues

Reinstate deleted issues on the Manage Deleted Issues Page.

To reinstate a deleted issue:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Manage Deleted Issues**.
3. Select the issue.
4. From the **Update** menu, select **Reinstate Selected**.

   Confirm your selection, and click **Apply**.

4.2.4.11.3 Permanently Remove Deleted Issues

Permanently remove deleted issues on the Manage Deleted Issues page.

To permanently remove deleted issues:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Manage Deleted Issues**.
3. Select the issue.
4. To permanently delete selected issue:
   a. Select the issues to be removed.
   b. From the **Update** menu, select **Permanently Delete Selected**.

      Confirm your selection, and click **Apply**.
5. To purge all deleted issues, click **Purge All**.

4.3 Viewing Your User Profile

Click the Notifications icon on the Team Development toolbar view and manage your own notifications and view all issues you own.

- Viewing Your Issues
  View the User Profile, Issue page to view a report of all issues you own or are monitoring.
- Viewing Your Notifications
  View and manage your notifications on the Notifications page.

4.3.1 Viewing Your Issues

View the User Profile, Issue page to view a report of all issues you own or are monitoring.

To view issues you own:

1. Click the **Team Development** menu.
2. On the Issues page, click the **Notifications** icon on the toolbar at the top of the page.

   The User Profile, Notifications page appears displaying notifications.
3. Click the **Issues** tab.

   The Issues page appears displaying all issues you own or are monitoring.
4. To view an issue, click the issue title.
5. To configure the type and frequency of your notifications, click **Notification Preferences**.

> **See Also:**
>
> - Editing Your Own Notification Preferences

4.3.2 Viewing Your Notifications

View and manage your notifications on the Notifications page.

Whenever a users interacts with an issue, Team Development creates an event. For example, an **event** can be creating issue or involve updating the issue. **Notifications** are a record of these events.

To view your own user notifications:

1. Click the **Team Development** menu.
2. On the Issues page, click the **Notifications** icon on the toolbar at the top of the page.

   The User Profile, Notifications page appears displaying notifications for all issues you own or are monitoring.
3. To delete all notifications, click **Clear Notifications**
4. To configure the type and frequency of your notifications, click **Notification Preferences**.

> **See Also:**
>
> Editing Your Own Notification Preferences

4.4 Managing Labels

Create and update label groups and labels to classify issues.

If you completed the setup process, you can view sample labels and label groups by going to **Team Development**, **Utilities**, **Manage Labels**.

- About Labels Groups and Labels
  Use label groups and the associated labels to classify issues.
- Creating Label Groups
  Create label groups clicking Create on the Labels page.
- Editing Label Groups
  Edit label groups on the Manage Labels page.
- Adding Labels
  Create labels on the Manage Labels page.
- Editing or Deleting Labels
  Edit labels on the Manage Labels page.

> **See Also:**
>
> Setting Labels

4.4.1 About Labels Groups and Labels

Use label groups and the associated labels to classify issues.

Label Groups and Labels

Labels are organized into label groups and enable you to classify issues. For example, the Label Group **Level of Effort** could contain the labels **Easy Fix**, **Moderate Effort**, and **Large Development Effort**.

Each **label group** has an identifying color to enable you to quickly identify labels associated with the same group. You can define label groups as exclusive or non-exclusive. In **exclusive** label groups, only one label from the group may be assigned to an issue at a time. Applying

another label from the same exclusive group removes any current label from that group. In **non-exclusive** label groups, you can assign multiple labels to an issue from the same label group.

Sample Label Groups and Labels

The following are two sample label groups and labels included in Team Development when you complete the setup process and install Sample Labels:

- Importance (Label Group)
  - Critical (Label)
  - Important (Label)
  - Normal (Label)
  - Backlog (Label)
  - Will Not Address (Label)

  This Label Group is **exclusive** meaning that only one label can be applied to an issue.
- Functional Area (Label Group)
  - Bug (Label)
  - Enhancement Request (Label)
  - Feature Request (Label)
  - Security Issue (Label)
  - Performance Issue (Label)
  - Documentation Issue (Label)
  - Training Issue (Label)

  This Label Group is **non-exclusive** meaning that only one label can be applied one or more labels.

If you completed the setup process, you can view the other sample label groups and labels by clicking the **Team Development** menu and selecting **Labels**. To view the labels within a label group, click a label group name.

> **Tip:**
>
> You also access Labels page from the Utilities page. Click the **Team Development** menu and select **Utilities** and then **Manage Labels**.

4.4.2 Creating Label Groups

Create label groups clicking Create on the Labels page.

To create a label group:

1. Click the **Team Development** menu and then select **Labels.**

   The Labels page appears.
2. On the Labels page, click **Create**.
3. In Create/Edit Label Group dialog:

a. Group Name - Enter a unique name for the label group.
b. Group Label Color - Select a color to display all labels within the group.
c. Values are Exclusive - Determines if the values of the group are exclusively assigned to an issue. Options include:
   - **On** - Only one label from the group can be assigned to an issue at any given time.
   - **Off** - Multiple labels from the same group may be assigned to an issue.
d. **Group Description** - Enter a description of the group.

4. Click **Create**.

4.4.3 Editing Label Groups

Edit label groups on the Manage Labels page.

Use label groups to classify labels.

To edit a label group:

1. Click the **Team Development** menu and then select **Labels**.

   The Labels page appears.

   Note that each a label group has an identifying color making it and the associated labels easier to identify.

2. On the Labels page, select the label group name.

   The Manage Labels page appears, displaying labels in the current group.

3. Click **Edit Group**.

   The Create/Edit Label Group dialog appears.

4. To edit the label group, edit the following attributes:
   a. Group Name - Enter a unique name for the label group.
   b. Group Label Color - Select a color to display all labels within the group.
   c. Values are Exclusive - Determines if the values of the group are exclusively assigned to an issue. Options include:
      - **On** - Only one label from the group can be assigned to an issue at any given time.
      - **Off** - Multiple labels from the same group may be assigned to an issue.
   d. **Group Description** - Enter a description of the group.
   e. Click **Apply Changes**.
5. To delete the label group, click **Delete** in the Create/Edit Label Group dialog.

4.4.4 Adding Labels

Create labels on the Manage Labels page.

To add a new label to a label group:

1. Click the **Team Development** menu and then select **Labels**.
2. On the Labels page, select the label group name.

   The Manage Labels page appears, displaying labels in the current group.

3. On the Manage Labels page, click **Add Row**.

   A new row appears.
4. In the new row, enter the **Sequence**, **Label Name**, **Label Slug**, and **Label Desc**.
5. Click **Save**.

4.4.5 Editing or Deleting Labels

Edit labels on the Manage Labels page.

To edit existing labels:

1. Click the **Team Development** menu and then select **Labels**.
2. On the Labels page, select the label group name.

   The Manage Labels page appears, displaying an editable interactive grid of labels in the current group.
3. To edit a single cell:
   a. Double-click the cell you want to edit.
   b. Enter your changes and click **Save**.
4. To edit a multiple rows:
   a. Select rows by selecting checkboxes in the checkbox column.

      > **Tip:**
      >
      > To select all rows, click the checkbox in the column. With all rows selected, click it again to deselect all rows.

   b. Select the cells you wish to edit and enter your changes.
   c. Click **Save**.
5. To delete multiple rows:
   a. Select the rows to be deleted.
   b. Click the **Actions** menu and then select **Selection**, **Delete Rows**.

      > **Tip:**
      >
      > To delete a single row, click **Row Actions** icon (to the right of the checkbox) and select **Delete Row**.

   c. Click **Save**.

> **See Also:**
>
> Using an Editable Interactive Grid in *Oracle APEX End User's Guide*

4.5 Managing Milestones

Create and update issues milestones to track events.

If you completed the setup process, you can view sample milestones clicking the **Team Development** menu and selecting **Milestones**.

- Creating Milestones
  Create milestones by clicking Create on the Milestones page.
- Editing and Deleting Milestones
  Edit or delete an existing milestone by selecting the milestone name on the Milestones page.

> **See Also:**
>
> Setting Milestones

4.5.1 Creating Milestones

Create milestones by clicking Create on the Milestones page.

To create a milestone:

1. Click the **Team Development** menu and then select **Milestones**.
2. On the Milestones page, click **Create Milestone**.
3. On the Milestone page:
   a. Milestone Name - Enter a unique name for the milestone.
   b. Slug - A unique string identifier for the milestone that will be used in URLs.
   c. Date - Select a date for the milestone.
   d. Milestone Status - Select **Open** or **Closed**.
   e. **Description** - Enter a description of the milestone.
4. Click **Create**.

4.5.2 Editing and Deleting Milestones

Edit or delete an existing milestone by selecting the milestone name on the Milestones page.

To edit or delete a milestone:

1. Click the **Team Development** menu and then select **Milestones**.

   The Milestones page appears.
2. To edit a milestone:
   a. On the Milestones page, click the milestone name.
   b. Click **Edit Milestone**.
   c. On the Milestone page:

- Milestone Name - Enter a unique name for the milestone.
- Slug - A unique string identifier for the milestone that will be used in URLs.
- Date - Select a date for the milestone.
- Milestone Status - Select **Open** or **Closed**.
- Description - Enter a description of the milestone.

d. Click **Apply Changes**.

3. To delete a milestone:
   a. On the Milestones page, click the milestone name.
   b. Click **Edit Milestone**.
   c. Click **Delete**.
   d. In the confirmation dialog, confirm the deletion by clicking **OK**.

4.6 Managing Templates

Create and update templates for issues and issue comments.

Templates enable you to provide users with starter text for issues and comments. Once you create a new template, you can select it when you create a new issue or add a comment.

If you completed the setup process, you can view sample template by going to **Team Development**, **Utilities**, **Manage Templates**.

- Creating Issue Templates
  Create an issue template by clicking Create on the Templates page.
- Editing or Deleting Issue Templates
  Edit existing templates by selecting the template title on the Templates page.

4.6.1 Creating Issue Templates

Create an issue template by clicking Create on the Templates page.

The text of a template defines the outline of what a user should enter.

To create an issue template:

1. Click the **Team Development** menu and then select **Manage Templates**.
2. On the Templates page, click **Create Template**.
3. On Edit Template page:
   a. Template Name - Enter a unique name for the template.
   b. Template Description - Describe the purpose of template, including what it should be used for.
   c. Template Type - Select a template type:
      - **Issue** - Select **Issue** when creating a new issue from scratch.
      - **Response** - Select **Response** when entering a comment.
   d. Enter the template text and provide an outline of what a user should enter.

      Use the Markdown tools above the field to add formatting. Click **Preview** to preview the HTML.

4. Click **Create**.

4.6.2 Editing or Deleting Issue Templates

Edit existing templates by selecting the template title on the Templates page.

To edit or delete a template:

1. Click the **Team Development** menu and then select **Manage Templates**.
2. On the Templates page, click the template name.
3. Click **Edit Template**.

   The Edit Template page appears.
4. Edit the template attributes:
   a. Template Name - Enter a unique name for the template.
   b. Template Description - Describe the purpose of template, including what it should be used for.
   c. Template Type - Select a template type:
   d. Enter the template in the field provided. To add formatting (such as bold or italic), click the icons in the toolbar. To preview the rendered HTML, click the **Preview** button.
   e. Click **Apply Changes**.
5. To delete the template:
   a. Click **Delete**.
   b. In the confirmation dialog, confirm your selection and click **OK**.

4.7 Using Team Development Utilities

Use Utilities page to manage deleted issues, manage templates, milestones, and labels, view uploaded files, purge data, reassign issues to another user, reassign issue owners, remove a milestone or label from all issues, and manage user notifications.

- Team Development Utilities
  Access the Utilities page by selecting Team Development, Utilities.
- Managing Uploaded Files
  View and delete uploaded files attached to issues in the workspace.
- Purging Data
  Users with administrator privileges can purge data for selected components.
- Reassigning Issues to Another Assignee
  Users with administrator privileges can reassign issues from one assignee to another for all issues, or only issues having an Open status.
- Reassigning the Issue Owner
  Users with administrator privileges can reassign the issue owner.
- Removing a Milestone for All Issues
  Users with administrator privileges can emove a specific milestone from all issues.
- Removing Labels for All Issues
  Users with administrator privileges can remove specific labels for all issues.
- Managing Notification Preferences
  Enables administrators to manage notification preferences.

4.7.1 Team Development Utilities

Access the Utilities page by selecting Team Development, Utilities.

User with administrator rights can see the following links on the Utilities page:

- **Manage Deleted Issues** - Purge or reinstate issues that have been deleted. See Managing Deleted Issues.
- **Manage Templates** - Manage templates for issues and comments. See Managing Templates.
- **Manage Milestones** - Create and update issues milestones used to track events. See Managing Milestones.
- **Manage Labels** - Manage Label Groups and Label used to classify issues. See Managing Labels.
- **View Files** - View and delete uploaded files attached to issues in the workspace. See Managing Uploaded Files.
- **Purge Data** - Purge all data for selected components. See Purging Data.
- **Update Assignments** - Perform bulk reassignment of issues to a given user. See Reassigning Issues to Another Assignee.
- **Reassign Owner** - Perform bulk reassignment of issue owners to a given user. See Reassigning the Issue Owner.
- **Remove Milestone from All Issues** - Remove a milestone from all issues. See Removing a Milestone for All Issues.
- **Remove Labels from all Issues** - Remove a label from all issues. See Removing Labels for All Issues.
- **Manage User Notifications Preferences** - Manage issue notification preferences of all users within the workspace. See Managing Notification Preferences.

4.7.2 Managing Uploaded Files

View and delete uploaded files attached to issues in the workspace.

The Issues Files page displays a report of uploaded files. Each file displays on a separate row and includes the file name, associated issue, file size, date the file was added, and the associated user.

To view and delete uploaded files:

1. From the **Team Development** menu, select **Utilities**.
2. Click **View Files**.

   The Issues Files page appears.
3. On the Issues Files page you can:
   - **Download the File** - Under Name, click the file name.
   - **Link to the Issue** - Click the **Issue Title** to link to the Issue Details page.
   - **Delete the File** (Administrators only) - Select the file and click **Delete Checked**.

4.7.3 Purging Data

Users with administrator privileges can purge data for selected components.

> **WARNING:**
>
> Purging data removes all the data for the selected component.

To purge data:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Purge Data**.
3. On the Purge Data page you can:
   a. Select the components for which you wish to purge data.
   b. Click **Delete**.

4.7.4 Reassigning Issues to Another Assignee

Users with administrator privileges can reassign issues from one assignee to another for all issues, or only issues having an Open status.

To reassign issues:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Update Assignees**.

   The Update Assignees page appears.
3. On Update Assignee:
   a. Change From - Select current assignee that you wish to reassign. Users marked as **Orphan** are current assignees whose related APEX user has been removed.
   b. Change To - Select the new assignee.
   c. For Open Issues Only? - Choose whether assignments should be changed for Open issues only, or for all issues.
4. Click **Next**.

   A report appears detailing the issues to be reassigned.
5. Click **Change Assignee**.

4.7.5 Reassigning the Issue Owner

Users with administrator privileges can reassign the issue owner.

To reassign the issue owner:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Reassign Owner**.

   The Reassign Owner page appears.

3. On Reassign Owner:
   a. Change From - Select current owner that you wish to reassign. Users marked as **Orphan** are current assignees whose related APEX user has been removed.
   b. Change To - Select the new owner.
   c. For Open Issues Only? - Choose whether assignments should be changed for Open issues only, or for all issues.
4. Click **Next**.

   A report appears detailing the issues to be reassigned.
5. Click **Change Owner**.

4.7.6 Removing a Milestone for All Issues

Users with administrator privileges can emove a specific milestone from all issues.

To remove a specific milestone from all issues:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Remove Milestone from All Issues**.

   The Remove Milestone page appears.
3. On Remove Milestone:
   a. Milestone - Select the milestone that you wish to disassociate from the selected issues.
   b. For Open Issues Only? - Choose whether the disassociation should occur for Open issues only, or for all issues.
4. Click **Next**.

   A report appears detailing the issues to be reassigned.
5. Click **Remove Milestone**.

4.7.7 Removing Labels for All Issues

Users with administrator privileges can remove specific labels for all issues.

To remove specific labels for all issues:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Remove Labels from All Issues**.

   The Remove Labels page appears.
3. On Remove Labels:
   a. Label - Select the Label or Label Group to be removed.
   b. For Open Issues Only? - Choose whether Labels should be removed for Open issues only, or for all issues.
4. Click **Next**.

   A report appears detailing the issues to be reassigned.
5. Click **Remove Labels**.

4.7.8 Managing Notification Preferences

Enables administrators to manage notification preferences.

Whenever a users interacts with an issue, Team Development creates an event. For example, an **event** can be creating an issue, closing an issue, or adding a comment. **Notifications** are a record of these events. Users can view their own notifications on the **User Profile**, **Notification** page. Administrators can configure user notifications preferences by going to **Utilities**, **Manage User Notification Preferences**.

- Editing Your Own Notification Preferences
  Edit your own notification preferences.
- Adding User Notification Preferences
  Users with administrator privileges can add notification preferences for any user.
- Editing Notification Preferences for Other Users
  Users with administrator privileges can edit notification preferences for other users.

4.7.8.1 Editing Your Own Notification Preferences

Edit your own notification preferences.

To edit your notification preferences:

1. Click the **Team Development** menu.
2. On the Issues page, click the **Notifications** icon.

   The Notifications page appears displaying notifications for all issues you own or are monitoring.
3. Click **Notification Preferences** to specify which notifications you would like to receive.

   The Notification Preferences dialog appears.
4. On Notification Preferences:
   a. For Receives Emails? - Choose whether the user receives emails for subscribed events.
   b. Frequency - If the user has opted to receive emails, this setting determines how often emails are sent. Select a Frequency:
      - Immediately
      - Hourly Summary
      - Daily Summary
   c. Receive Notifications When - Select the types of notifications the user is interested in.

      If the user is receiving emails, the selected notifications are sent at the selected Frequency. If the user has opted not to receive emails, notifications are viewable using the user's profile page.
5. Click **Save Settings**.

4.7.8.2 Adding User Notification Preferences

Users with administrator privileges can add notification preferences for any user.

To add user notification preferences for any user:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Manage User Notification Preferences**.

   The Manage Notification Preferences page appears.
3. To add a new notification, click **Add Preference for User**.
4. On Manage Notification Preference:
   a. Add User - Indicates the user whose notification preferences are being added. Only users who do not already have notification preferences set will appear in this list.
   b. For Receives Emails? - Choose whether the user receives emails for subscribed events.
   c. Frequency - If the user has opted to receive emails, this setting determines how often emails are sent. Select a Frequency:
      - Immediately
      - Hourly Summary
      - Daily Summary
   d. Receive Notifications - Select the types of notifications the user is interested in.

      If the user is receiving emails, the selected notifications are sent at the selected Frequency. If the user has opted not to receive emails, notifications are viewable using the user's profile page.
5. Click **Save Settings**.

4.7.8.3 Editing Notification Preferences for Other Users

Users with administrator privileges can edit notification preferences for other users.

To user notification preferences for other users:

1. From the **Team Development** menu, select **Utilities**.
2. Click **Manage User Notification Preferences**.

   The Manage Notification Preferences page appears.
3. Select a user to edit.
4. On Manage Notification Preference:
   a. Edit User - Displays the current user.
   b. For Receives Emails? - Choose whether the user receives emails for subscribed events.
   c. Frequency - If the user has opted to receive emails, this setting determines how often emails are sent. Select a Frequency:
      - Immediately
      - Hourly Summary
      - Daily Summary
   d. Receive Notifications When - Select the types of notifications the user is interested in.

      If the user is receiving emails, the selected notifications are sent at the selected Frequency. If the user has opted not to receive emails, notifications are viewable using the user's profile page.
5. Click **Save Settings**.

4.8 Managing Feedback

Feedback provides a mechanism for end users to post general comments for application administrators and developers.

- About Feedback
  Adding Feedback enables you to gather and track real-time comments, enhancement requests, and bugs from application users. Feedback enhances communication between the user community and the application owners.
- Adding Feedback Using the Create Application Wizard
  Add a Feedback page when running the Create Application Wizard.
- Adding Feedback Using the Create Page Wizard
  Use the Create Page Wizard to add Feedback to an existing application.
- Submitting Feedback Within a Running Application
  Users can submit feedback by clicking the Feedback navigation bar icon in the upper right of a running application.
- Reviewing Feedback Within an Application
  View feedback within a running application by clicking User Feedback on the Administration page.
- Configuring Feedback to Support Attachments
  Configure feedback to add support for attachments.
- Tracking Feedback
  Track all feedback within a workspace on the Feedback page.

4.8.1 About Feedback

Adding Feedback enables you to gather and track real-time comments, enhancement requests, and bugs from application users. Feedback enhances communication between the user community and the application owners.

Feedback provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from. Providing such an easy feedback loop enhances communication between the user community and the application owners, which leads to greatly improved end user satisfaction.

Creating Feedback:

- Adds a Feedback navigation bar icon in the running application which users can click to leave feedback.
- Creates a report for viewing and triaging feedback.
- Captures the application name, page ID and name, feedback comments, date and time, and user information.

Choosing How to Add Feedback

You can create a Feedback page automatically using a wizard when you create a new application, or by adding a page to an existing application. In both the Create Application Wizard and the Create Page Wizard, you simply select the Feature, **Feedback**.

How Customers Submit Feedback

Users can submit feedback by clicking the **Feedback** navigation bar icon in the upper right of a running application. In the Feedback window, they can rate their user experience by selecting an emoji, entering feedback comments, or uploading files.

Viewing Feedback within an Application

Administrators and developers can track and manage Feedback at the application level by running the application and accessing the Feedback region on the Administration page. To control whether users can upload attachments, click **Feedback Settings**. To view feedback entries, select **User Feedback** to view the Manage Feedback report.

Tracking Feedback in Team Development

Administrators and developers can track all feedback within a workspace in Team Development, Feedback. Each feedback entry tracks the application and page ID and name, displays any feedback comments. Administrators and developers can assign a status and create an issue.

4.8.2 Adding Feedback Using the Create Application Wizard

Add a Feedback page when running the Create Application Wizard.

To create an application which includes a Feedback page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. For Name, enter the name used to identify the application to developers.
6. For Appearance, accept the default Theme Style and menu layout (Vita, Side Menu) or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
7. To add a new page (such as a blank page, calendar, chart, form, report, and so on), click **Add Page** and select the desired page type.
8. Under Features, select **Feedback**.
9. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
10. Click **Create Application**.

> **See Also::**
>
> Understanding Page Types, Features, and Settings

4.8.3 Adding Feedback Using the Create Page Wizard

Use the Create Page Wizard to add Feedback to an existing application.

> **Tip:**
>
> This section describes how to create a feedback by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To add a feedback page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.

   The Application home page appears.
3. Click the **Create Page** button.
4. For Create a Page:
   a. Page Type - Select **Feature** and then **Feedback**.

   > **Tip:**
   >
   > **Component** pages provides page-level functionality and can be added multiple times within a given application such as reports, forms, charts, or calendars. **Feature** pages provide application-level functionality and can only be added once per application.

   The Create Feedback Page appears.

   b. Click **Next**.
5. On the Create Feedback Pages:
   a. Starting Page Number - Enter an unused starting page number. This feature will create multiple pages starting with the number specified.
   b. Page Group - Identify the name of the page group to associate with this page. Page groups help developers manage the pages within an application. To use an existing page group, select the name from the list.
   c. Build Option - The listed build option(s) to be created to support this feature.
   d. Include in Navigation Bar - Select **On** to create a navigation bar entry within the current application to invoke your new feedback page.
   e. Navigation Bar Label - Enter the name of the new navigation bar entry used to access the new feedback page.
   f. Administration Page Preference - Specify the administration page to which a list will be added for this feature.

      Options include:

      - **Create a new page**

- **Identify an existing page**

  Depending upon your selection, the UI changes.

g. Follow the on-screen instructions.

6. Click **Create**.

4.8.4 Submitting Feedback Within a Running Application

Users can submit feedback by clicking the Feedback navigation bar icon in the upper right of a running application.

To submit feedback managed with the application:

1. Run the application for which you want to submit feedback.
2. Click the **Feedback** navigation bar icon in the upper right corner.

   A Feedback window appears.
3. In Feedback:
   a. Experience - Select an emoji that reflects your user experience (**Positive**, **Neutral**, or **Negative**).
   b. Feedback - Enter a comment.
   c. Attachment - Click **Choose File** to upload an attachment.

   > **Tip:**
   >
   > Administrators can enable and disable attachments by selecting **Administration** and editing **Feedback Settings**.

4. Click **Submit Feedback**.

4.8.5 Reviewing Feedback Within an Application

View feedback within a running application by clicking User Feedback on the Administration page.

To view feedback within an application:

1. Run the application.
2. Click **Administration** in the left navigation bar.
3. Click **User Feedback**.

   The Manage Feedback page displays an interactive report.
4. Click the Edit icon adjacent to the entry you wish to view.
5. On the Feedback page:
   a. Review the user feedback.
   b. Response - Respond to the feedback.
   c. Status - Select one of the following:
      - **No Action**
      - **Acknowledged**

- **Open**
- **Closed**

6. To delete the feedback, click **Delete**.
7. To save your changes, click **Apply Changes**.

4.8.6 Configuring Feedback to Support Attachments

Configure feedback to add support for attachments.

> **Note:**
>
> For this feature to work, an administrator must set **Enable Feedback File Repository** to **Yes** for the workspace. See Configuring Team Development in an Existing Workspace in *Oracle APEX Administration Guide*.

Developer and administrators can also configure Feedback Settings to enable support for attachments.

To configure Feedback Settings:

1. Run the application.
2. Click **Administration** in the left navigation bar.
3. Click **Feedback Settings** .

   Feedback Settings appears.
4. For Enable Attachments, select **On** to enable users to upload attachments or **Off** to disable this functionality.
5. To save your changes, click **Apply Changes**.

4.8.7 Tracking Feedback

Track all feedback within a workspace on the Feedback page.

Administrators and developers can track all feedback within a workspace in Team Development. Each feedback entry tracks the application and page ID and name, user, user email, and user comments. Administrators and developers can review the feedback, assign a type, create an issue and assign a status.

- Viewing the Feedback Dashboard
- Viewing the Feedback Report
- Viewing Feedback by Filing User
- Viewing the Feedback Calendar

4.8.7.1 Viewing the Feedback Dashboard

To access the Feedback report:

1. Click the **Team Development** menu and select **Feedback**.

   The Feedback Dashboard appears.

2. Click the **Report** tab.

   The Feedback Dashboard appears.

3. To link from the Feedback Dashboard:

   - **Open Feedback**, **Feedback Users**, or **Feedback Entries** - Click the number enclosed by a circle to link to the appropriate page or report.
   - **By Status**, **By Application**, or **By Filer** - Lists feedback details. Click the link to view more details.
   - **Summary** - Summaries feedback entries. Click the number above to summary entry to link the appropriate report or page.

4.8.7.2 Viewing the Feedback Report

To access the Feedback report:

1. Click the **Team Development** menu and select **Feedback**.

   The Feedback Dashboard appears.

2. Click the **Report** tab.

   The Feedback Report appears.

   To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** to return the report to the default view.

3. Select a number in the Feedback column.

   The View/Edit Feedback dialog appears.

4. On View/Edit Feedback:

   a. Review the user feedback.

   b. Type - Classify the feedback. Select one of the following:

      - **General Comment**
      - **Enhancement Request**
      - **Bug**

   c. Disposition - Select one of the following:

      - **Log as Bug**
      - **Log as To Do**
      - **Log as Feature**
      - **Log as Issue**

5. For Status, select a status of the feedback entry:

   - **0. No Status**
   - **1. Acknowledged**
   - **2. Additional information requested**
   - **3. Open Processing Feedback**
   - **4. Closed**

6. To delete a feedback entry, click **Delete**.

7. To save your changes, click **Apply Changes**.

4.8.7.3 Viewing Feedback by Filing User

To view feedback by filing user:

1. Click the **Team Development** menu and select **Feedback**.

   The Feedback Dashboard appears.

2. Click the **By Filing User** tab.

   The By Filing User report appears.

   To customize the report, use the Actions menu in Search bar at the top of the page. Click **Reset** to return the report to the default view.

3. To view the feedback, click the user in **Filing User** column.

4.8.7.4 Viewing the Feedback Calendar

To view the Feedback Calendar:

1. Click the **Team Development** menu and select **Feedback**.

   The Feedback Dashboard appears.

2. Click the **Calendar** tab.

   The Calendar appears.

5
Creating Applications

An application is a collection of pages linked together using navigation menus, tabs, buttons, or hypertext links. Application pages share a common session state and authentication.

To create an application, a developer runs wizards to declaratively assemble pages and navigation. Individual pages are organized using containers called **regions**. Once created, developers can modify an application by editing application attributes and add new pages using the Create Page Wizard. Developers can further manage the development process by creating page groups, locking and unlocking pages, adding developer and application comments, and accessing a variety of application, page, and cross page utilities and reports.

- Choosing an Application Creation Method
  Learn about available creation methods when creating a new application.
- Using the Simple Create Application Wizard
  Run simple Create Application Wizard to create an application with one click using pre-defined defaults and add pages to it manually.
- Using the Full Create Application Wizard
  Run the full Create Application Wizard to create an application with multiple pages and advanced options.
- Creating an App Using Generative AI
  Create an application using Generative AI and enter natural language.
- Creating an Application from a File
  Learn how to upload a file or copy and paste CSV data and then create an application on the uploaded data.
- Integrating APEX with Fusion Applications
  Use Oracle APEX to build extensions to Fusion Applications.
- Using Application Blueprints
  Each iteration of the Create Application Wizard saves an application blueprint which you can use to start adding pages to an application, or edit the application definition as a JSON document.
- Copying an Application
  Create a copy of an existing application.
- Creating a Working Copy to Merge, Refresh, or Compare
  Create a Working Copy of an application to merge, refresh or compare changes.
- Creating Applications for Mobile Devices
  Build applications for mobile devices with mobile-specific design patterns that can provide an optimal user experience for small screens.
- Deleting an Application
  You can delete an application from the Application home page, or while editing application attributes.
- Managing Application Attributes
  Application attributes control the behavior of an entire application and are divided into the categories: Definition, Security, Globalization, User Interface, Progressive Web App, and AI.

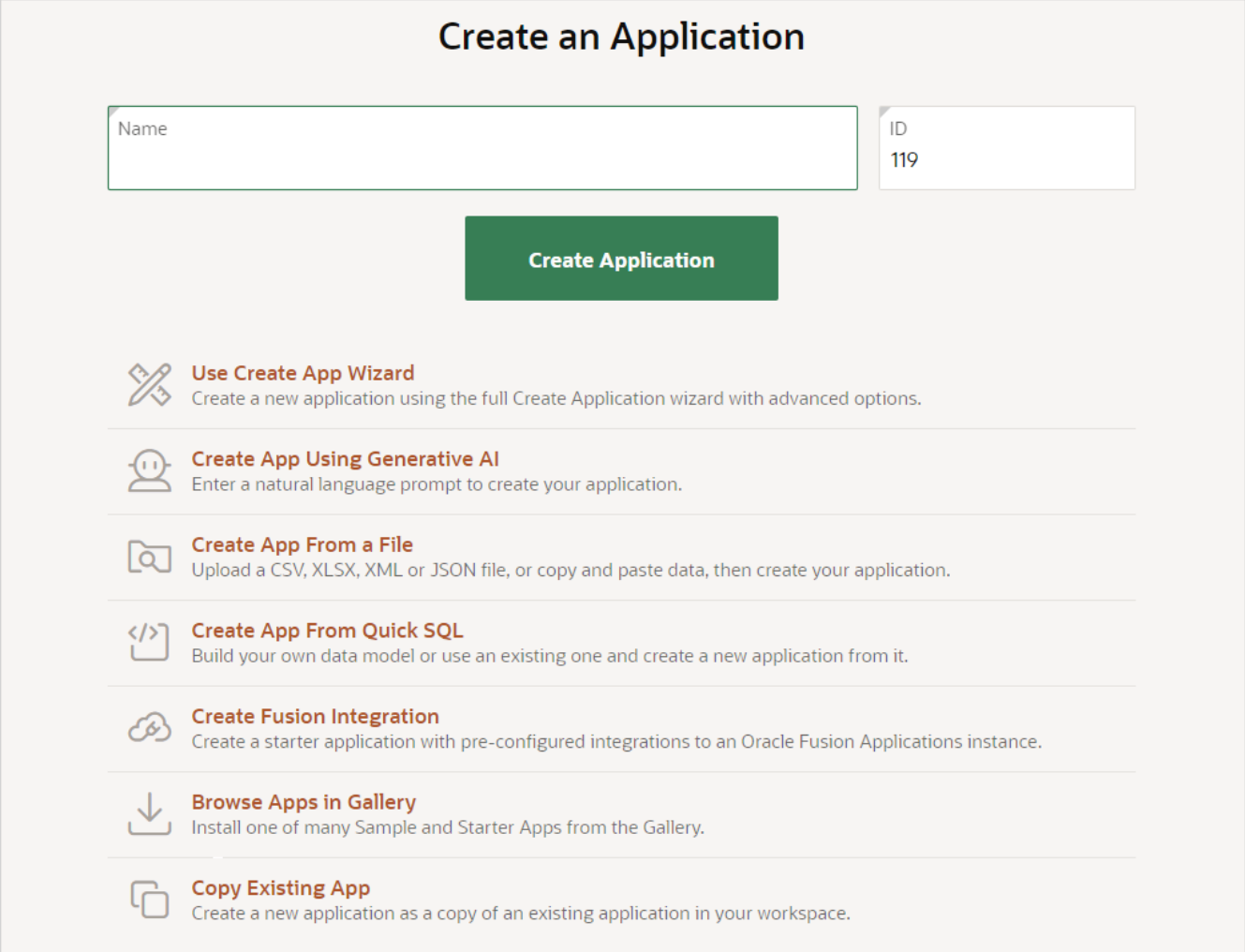
- Managing Application Logos
  Add and update an application logo in an Oracle APEX application.
- Managing Application Icons
  Add and update the icon that displays as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon in Oracle APEX applications.
- Adding Developer Comments
  Add comments to an application, a page, or a group of pages using the Developer Comment, Bug, or To Do button. You can use developer comments to communicate application changes, report issues, or record developer suggestions.
- Using the Find Icon
  Click the Find icon to search for items, pages, queries, tables, PL/SQL code, view debug reports, view session state, and view errors.
- Adding Application Comments
  Use Application Comments to describe an application or track developers involved in the application development.
- Managing Application Groups
  Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.
- Using Application Utilities
  Learn about the application summaries and reports available from Utilities page.
- Using Workspace Utilities
  Use the Workspace Utilities page to acccess REST Enabled SQL Services, Generative AI Services, Remote Servers, Application Groups, Web Credentials, Workspace Themes, Export, Oracle APEX Views, Manage Backups, and REST Source Catalogs.

5.1 Choosing an Application Creation Method

Learn about available creation methods when creating a new application.

> **Tip:**
>
> The first time you access the App Builder home page, a **Get Started Now** region appears. Click **Create New App** to access the launch the Create Application Wizard. Click **Install a Starter or Sample App** to access the Gallery. See Using the App Gallery.

To create a new application, click **Create** on the App Builder home page. The Create an Application page appears and prompts you to select a method of creating your application.



Available options include:

- **Create Application** - Create a simple application with one click. Enter the application Name and click **Create Application**. This option creates an application which contains only a Global Page, Home page, and Login page. Add pages later by running the Create Page Wizard.

  See Using the Simple Create Application Wizard.

- **Use Create App Wizard** - Create a new application using the full Create Application Wizard. The full Create Application Wizard creates a fully functional application based on tables you select or by providing a valid SQL. Add pages that include various components, add application-level features, and configuration options to enable or disable specific functionality.

  See Using the Full Create Application Wizard and Understanding Page Types, Features, and Settings.

- **Create App Using Generative AI** - Enter a natural language prompt to create your application. This option only displays if you have configured a Generative AI Service and enabled the **Used by App Builder** flag.

  See Managing Generative AI Services and Creating an App Using Generative AI.

- **Create App From a File** - Upload a CSV, XLSX, XML, or JSON file, or copy and paste data and then create your application.

  See Creating an Application from a File.

- **Create Fusion Integration** - Create a starter application with pre-configured integration to an Oracle Fusion Applications instance.

  See Integrating APEX with Fusion Applications.

- **Create App from Quick SQL** - Generate the SQL required to create a relational data model from an indented text document. Quick SQL is designed to reduce the time and effort required to create SQL tables, triggers, and index structures.

  See Using Quick SQL in *Oracle APEX SQL Workshop Guide*.

- **Browse Apps in Gallery** - Install one of many Starter Apps and Sample Apps from the Gallery.

  See Using the App Gallery.

- **Copy Existing App** - Create a copy of an existing application.

  See Copying an Application.

5.2 Using the Simple Create Application Wizard

Run simple Create Application Wizard to create an application with one click using pre-defined defaults and add pages to it manually.

The simple Create Application Wizard creates an application containing a Global Page, Home page, and Login page. Once your application is created, you can add component and feature pages manually using the Create Page Wizard.

To run the simple Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating your application.
3. On the Create an Application page:
   a. Name - Enter the name used to identify the application to developers.
   b. Click **Create Application**.

      The Application home pages appears and contains three pages: Global Page, Home page, and Login page.

> **Tip:**
>
> To add pages manually, run the Create Page Wizard. To learn more, see Managing Pages in an Application.

5.3 Using the Full Create Application Wizard

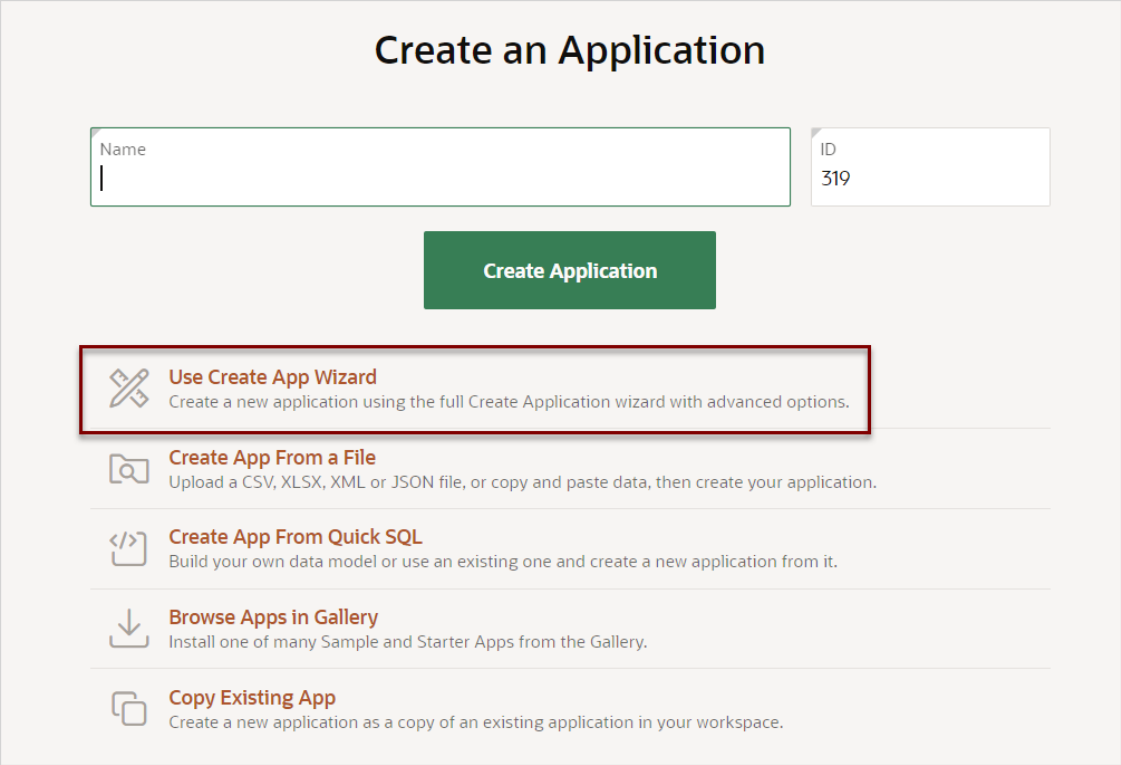
Run the full Create Application Wizard to create an application with multiple pages and advanced options.

- About the Full Create Application Wizard
  Learn about how the full Create Application Wizard works.
- Running the Full Create Application Wizard
  Run the full Create Application Wizard to create an application containing multiple pages based on the tables you select, or by providing valid SQL queries.
- Understanding Page Types, Features, and Settings
  Learn about the supported page types, features, and configurable settings available in the Create Application Wizard.

5.3.1 About the Full Create Application Wizard

Learn about how the full Create Application Wizard works.

Creating a new application can a single or multiple step process. Once you click **Create** on the App Builder home page, the wizard prompts you to select an application type. To launch the full Create Application Wizard, click **Use Create App Wizard** .



> **Note:**
>
> To learn more about other options on this page, see Choosing an Application Creation Method.

The full Create Application Wizard appears. You specify the application name and appearance and then you add pages. Applications can contain multiple pages and can include various components such as calendars, cards, charts, dashboards, forms, interactive grids, master detail or editable grids, and reports. Once created, you can edit the page names, alter the page order, or delete them.

Features provide application-level functionality and can only be added once per application. Available features include Install Progressive Web App, Push Notifications, About Page, Access Control (to configure role-based user authorization), Activity Reporting, Configuration Options (enable or disable specific functionality), Feedback, and Theme Style Selection.

Then, you configure **Settings**. Settings are used in the generation of the application and include the Application ID, Schema, Authentication, Language, Advanced Settings (which include application definition settings and preferences as well as security and globalization attributes), and UI Defaults.

After you create an application using the Create Application Wizard, you can modify pages and add additional pages using the Create Page Wizard. You can also take advantage of application blueprints. Each time you run the Create Application Wizard, it saves an application blueprint which you can use to start adding pages to an application, or edit the application definition as a JSON document.

See Also:

- Running the Full Create Application Wizard
- Managing Pages in an Application

5.3.2 Running the Full Create Application Wizard

Run the full Create Application Wizard to create an application containing multiple pages based on the tables you select, or by providing valid SQL queries.

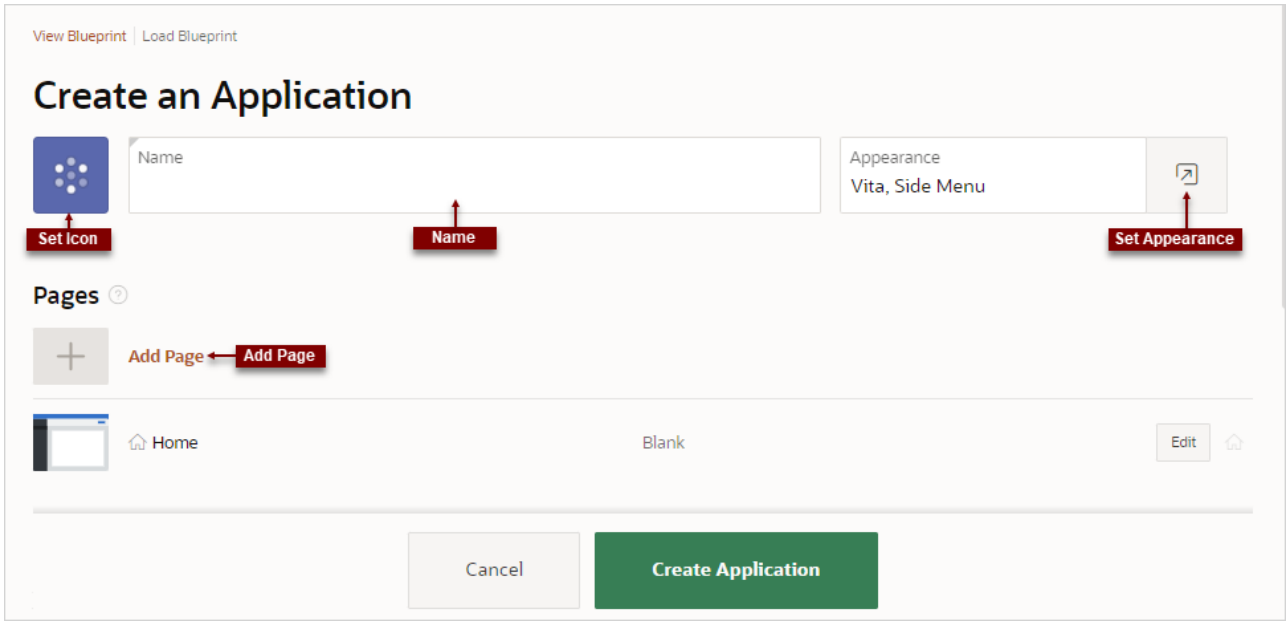
Run the Create Application Wizard by clicking **Create** on the App Builder home page, or by selecting **Create** from the App Builder menu.

To run the full Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating your application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.

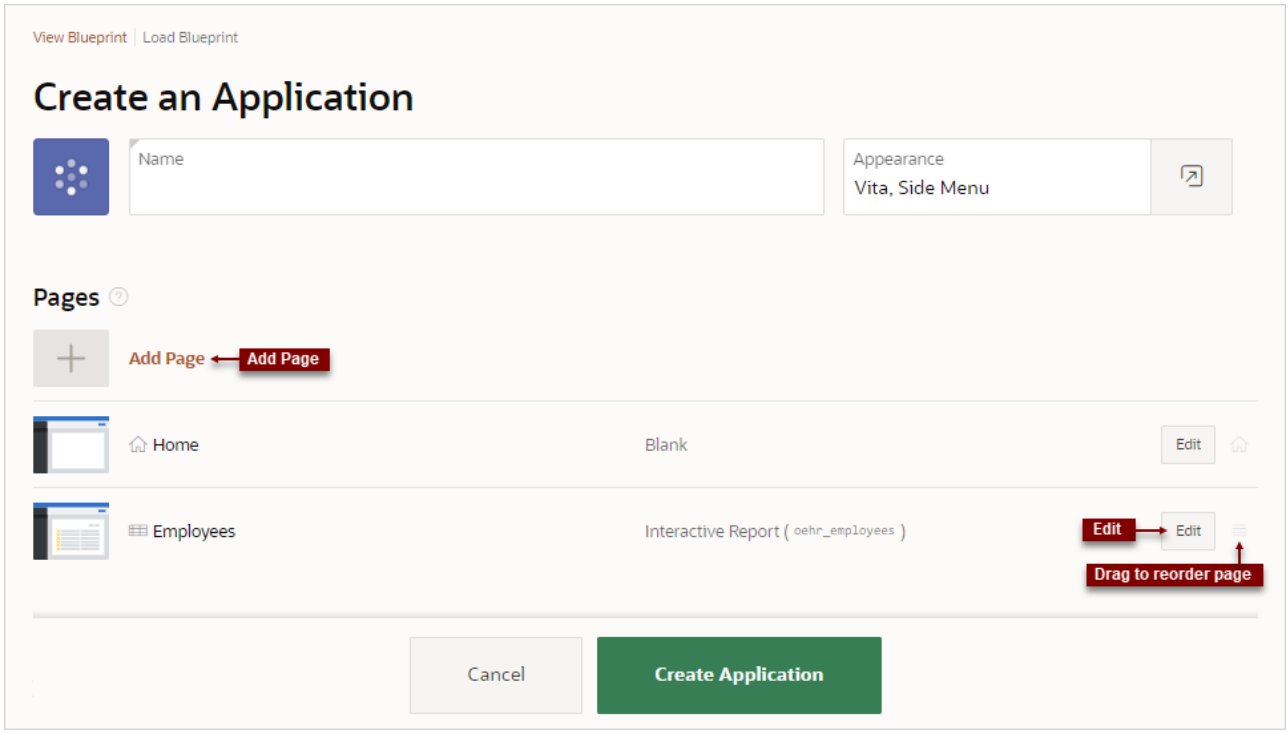


4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon. In the Choose Application Icon dialog you can include:
   - **Upload an icon**. Click the plus (+) sign and navigate to the icon, or drag and drop the icon to the **Upload your own icon** region.
   - **Select an icon from the library**. Select the icon color and then the icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.

   In the Appearance dialog, configure the following:
   - **Theme Style** - Select the default theme style used to display the application.

     > **Tip:**
     >
     > You can change the style to another pre-defined style, or modify it post generation. To change the style run the application and then click **Theme Roller** in the Developer toolbar.
   - **Navigation** - Select a type of navigation menu. Options include:
     - **Side Menu** - Renders a navigation tree on the side of the screen that can be collapsed when not in use.
     - **Top Menu** - Renders the navigation as a menu bar in the application header.
     - **Mega Menu** - Renders the navigation in an collapsible floating panel that displays all navigation items at once.
   - **Application Icon** - Click **Choose New Icon** to change the icon. Your selection displays on the home page of the application.
   - Click **Save Changes**.
7. To add a page, click **Add Page** and select the desired page type. The user interface changes based on the selected page type.

Pages you create display under **Pages**.

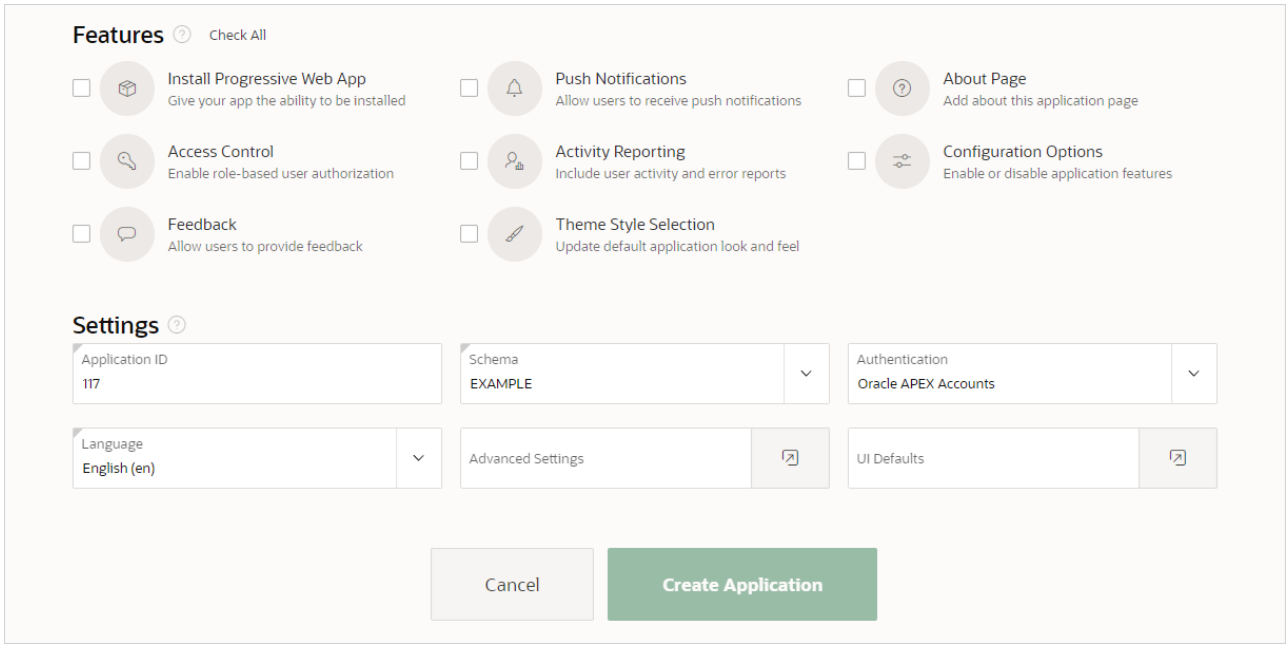


You can edit existing pages as follows:

- **Change the page order**. To change the order in which pages appear in your application, click and hold the **Drag to reorder page** icon and drag and drop it to a new location in the list.

  The Home page always displays first and cannot be reordered. For administrative pages, the page order dictates the order they appear in the Application Administration list on the Administration page.
- **Edit a page**. To edit a page click **Edit**. In the dialog, edit the attributes and click **Save Changes**.
- **Delete a page**. To delete a page, click **Edit** and the click **Delete** .

8. For Features, select features to include with your application.



Features provide application-level functionality and can only be added once per application. To learn more, see Help. To select all features, click **Check All**.

9. For Settings:
   a. Application ID - Enter a unique, numeric identifier for your application. This field contains an automatically generated identifier by default. Application IDs between 3000 to 9000 are reserved for internal use by Oracle APEX.
   b. Schema - Select the database schema which stores the database objects you want to use in this application. Each application obtains its privileges by parsing all SQL as a specific database schema.
   c. Authentication - Select how you want users to authenticate into your application.
   d. Languages - The primary language used in the app. To change the primary language, click the **Select Languages** icon.
   e. Advanced Settings - Click the icon adjacent to the Advanced Settings to edit application definition settings and preferences and security and globalization attributes. To learn more, see Help.
   f. UI Defaults - Click the icon to apply UI Defaults to this application.
10. Click **Create Application**.

See Also:

- Available Page Types in the Create Application Wizard
- Available Features in the Create Application Wizard
- Available Settings in the Create Application Wizard
- Managing Navigation Menus

5.3.3 Understanding Page Types, Features, and Settings

Learn about the supported page types, features, and configurable settings available in the Create Application Wizard.

- Available Page Types in the Create Application Wizard
  Learn about available page types when running the Create Application Wizard.
- Available Features in the Create Application Wizard
  Learn about available features when running the Create Application Wizard. Features provide application-level functionality and can only be added once per application.
- Available Settings in the Create Application Wizard
  Learn about Settings available when running the Create Application Wizard. Settings are used in the generation of the application and include the application ID, the database schema, Advanced Settings (such as the application definition, security, and globalization attributes), and application authentication.
- About Removing Features Created with the Create Application Wizard
  Remove features created with a wizard using build options.

See Also:

- Running the Full Create Application Wizard
- Creating Applications for Mobile Devices

5.3.3.1 Available Page Types in the Create Application Wizard

Learn about available page types when running the Create Application Wizard.

Available Page Types

Page Type	Description	To Learn More
Blank	Creates a page with no built-in functionality. Once you create the application, you can create regions on the page in Page Designer.	See online Help when creating this page.
Calendar	Generates a calendar with monthly, weekly, and daily views.	See Creating Calendars
Cards	A **cards** page features an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row). Features: • Customize every aspect of a cards region's UI (including layout, appearance, icon, badge, and media). • Declaratively create links from a cards page by adding actions. Action types include Button, Full Card, Title, Subtitle, or Media. • Supports advanced HTML expressions including template directives for client-side conditional formatting.	See: Managing Cards

Page Type	Description	To Learn More
Chart	Creates a new page containing a chart. Supported chart types include area, bar, box plot, bubble, combination, status meter gauge, donut, funnel, gantt, line, line with area, pie, polar, pyramid, radar, scatter, and stock charts.	See Creating Charts
Dashboard	Creates a dashboard page. Choose from four different chart layouts which you can easily customize in Page Designer. Available chart layouts include **Primary and 2 secondary charts**, **Grid of 4 charts**, **Primary and 3 secondary charts**, and **Grid of 6 charts**.	See online Help when creating this page.
Faceted Search	A **faceted search** page displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). Displays search results as cards or a classic report. Features: • Select options from facets in the left Search region to narrow down search results. • Set filters using facets that show possible values together with the occurrence count within the result set. • Built-in search capability. • Optionally include a chart for any facet. • Toggle between a bar chart (which is the default) and pie chart type. • Create facet groups for data models having multiple flag columns.	See: Managing Faceted Search
Smart Filters	A **smart filters** page features single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a Smart Filter behaves similarly to faceted search, it features a more space efficient layout. Features: • Includes a search field at the top of the page with filters that display as suggestion chip. • The Search Results can display as a classic report, cards, map, or calendar. • Clicking a filter name displays a list of values. • Clicking to right of a suggestion chip changes it to an applied filter chip and moves to the Applied Filters Area. Clicking the applied filter chip again displays a list. • Built-in search capability.	See: Managing Smart Filters
Form	Create a page containing a form which enables end users to maintain data. Select the table on which to build the form and specify whether or not to include a report.	See: Creating a Form

Page Type	Description	To Learn More
Interactive Grid	An **interactive grid** presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. Features: • Extensive customization capabilities using menus or inline editing. • Turn editing on or off using the region Attribute, Edit, Enabled. • Built-in search capability. • For both editable and non-editable reports, users can change the report layout and create private reports: – Column Heading menu - Change the report layout with Control Break, Aggregations, Freeze, Hide, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Refresh, Flashback. * Format - Control Break, Highlight, Stretch Column Widths. * Selection - Cell Selection, Copy to Clipboard, Refresh Rows. * Chart - Create chart. * Report - Create, edit, and manage saved Private or Alternative reports. * Download - Download or email the report as a CSV, HTML, PDF, or Excel file • If editable, users can edit the data directly: – Row Actions menu - Single Row View, Add Row, Duplicate Row, Delete Row, Refresh Row, Revert Changes. – Add Rows - Add new rows by clicking the **Add Row** button. – Edit Rows - Edit a row by selecting or double-clicking a cell with the mouse, editing the content, and clicking the **Save** button.	See: Managing Interactive Grids

Page Type	Description	To Learn More
Interactive Report	An **interactive report** page features a searchable, easily customizable report. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Built-in search capability. • Change the report layout and create private reports using menus: – Column Heading menu: Change the report layout with Sort Ascending, Sort Descending, Hide Column, Control Break, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Compute, Flashback. * Format - Control Break, Highlight, Rows Per Page. * Chart - Create a chart. * Group By - Group sets of results by one or more columns with Group By. * Pivot - Pivot reports transpose rows into columns to generate results in a cross tab format. * Report - Create, edit, and manage saved Named reports. * Download - Download or email the report as a CSV, HTML, Excel, or PDF. * Subscription - Subscribe to report updates in CSV, HTML, Excel, or PDF format.	See: Managing Interactive Reports
Map	The map component consists of a background map and spatial geometry objects (or layers) sourced from data in a local table. Supported spatial geometry objects (or layers) include: Points, Lines, Polygons, Heat Map, or Extruded Polygons. Spatial geometry objects can be either: • **Geometry Column** - Supported datatypes include `SDO_GEOMETRY`, `VARCHAR2`, or `CLOB`. `VARCHAR2` and `CLOB` columns must contain geometry information in GeoJSON format. • **Two Numeric Columns** - These columns must contain longitude and latitude values. This option only applies to Point and Heat Map layers.	See Creating Maps

Page Type	Description	To Learn More
Master Detail	A master detail reflects a one-to-many relationship between two tables in a database. Master detail forms enable users to insert, update, and delete values from two tables or views. Master Detail options include: • **Stacked** - Creates a single page master detail with editable interactive grids. Users select a row in the master grid to update the detail grids. • **Side by Side** - Create a single page master detail utilizing side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report. • **Drill Down** - Creates a two page (or Drill Down) master detail. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grids for the detail.	See Creating Master Detail Forms
Additional Pages, Classic Report	Expand the **Additional Pages** region to view this option. A **classic report** features a simple report page based on the formatted result of a SQL query. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Enable column sorting in Page Designer by selecting a column and configuring Column, Sorting attributes. • The most versatile report as it is template-driven. Choose from the predefined report templates, including Media List, Comments, Timeline, Badge List, Value Attributes Pairs, and so on. Or, create your own template to customize and visualize data. • Download the report as a CSV, HTML, Excel, or PDF. • Supports Report Layouts to format PDF exports.	See: Managing Classic Reports
Additional Pages, Multiple Reports	Creates multiple report pages containing an interactive report with form. For each page, you choose a table on which to build the interactive report with form. Click **Edit** to change the default options (such as providing a custom **SQL Query**), specifying another report type (such as, selecting**Classic Report** instead of **Interactive Report**). Select **Include Form** to include a form page for creating or updating records.	See: Managing Interactive Reports See online Help when creating this page.
Additional Pages, Timeline	Expand the **Additional Pages** region to view this option. Create a page which displays a timeline. Especially useful for displaying updates.	See online Help when creating this page.
Additional Pages, Wizard	Create a collection of pages to be used as a wizard. Generally wizards are used for entering data across multiple steps.	See online Help when creating this page.

5.3.3.2 Available Features in the Create Application Wizard

Learn about available features when running the Create Application Wizard. Features provide application-level functionality and can only be added once per application.

This table lists available Features when running the Create Application Wizard.

Page Type	Description	To Learn More
Install Progressive Web App	Progressive Web Applications (PWA) are designed to be capable, reliable, and installable. A Progressive Web App feels like a platform-specific application. Enabling this option: • Adds a new navigation bar entry, `Install App`. • Enables users to install the app on devices.	See Creating a Progressive Web App (PWA)
Push Notifications	Web push notifications enable web apps to send notifications to a user's device even when the user is not currently on the website. Notifications can be sent to a user's desktop or mobile device and can be used to alert the user to new content or events on the website, or to re-engage the user with the website. Enabling this option: • Add a new navigation bar entry, `User Settings`. • Enables users to subscribe and unsubscribe to push notifications. • When subscribed, users can receive push notifications from the APEX app.	See Creating a Progressive Web App (PWA) and Configuring Attributes for a Progressive Web App
About Page	Include an About page in the application which displays the application description.	See online Help when creating feature.
Access Control	Incorporate role-based user authentication within your application. Users can be defined as Administrators, Contributors, or Readers. You can then readily define different access to different roles for various components throughout your application, such as pages, menu entries, regions, columns, items, buttons and so forth.	See Controlling Access to Applications, Pages, and Page Components
Activity Reporting	Include numerous reports on user activity for your application such as the most active users, the most used pages, the performance of pages, and errors raised. Includes the following: • Top Users report • Application Error Log report • Page Performance, activity and performance by page • Application activity by page report • Page Views detail report • Automations Log (if the application contains automations)	Not applicable.

Page Type	Description	To Learn More
Configuration Options	Enables application administrators to enable or disable specific functionality within the application. This feature is useful if you select features that need additional development effort before they can be used by end users. This feature can also be expanded to application-specific features. If developers define additional build options and associate them with specific functionality throughout the application, then they can be added to the configuration settings for administrators. For example, within the *Customer Tracker* productivity app, administrators can turn on or off such features as Contacts, Data Loading, Geography, and more.	Not applicable.
Feedback	Feedback provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from. Creating Feedback: • Creates a Navigation bar icon which users can click to leave feedback. • Creates a report for viewing and updating feedback. • Captures the application and page ID, feedback comments, date and time, and user information.	See Managing Feedback
Theme Style Selection	Enables administrators to select a default color scheme (theme style) for the application. Administrators determine whether end users can choose their own theme style by enabling and disabling **Allow End Users to choose Theme Style is enabled**. If enabled, end users simply click on the **Customize** link at the bottom of the home page and select from the available theme styles. For example, users with visual impairment may prefer to utilize the Vita theme style which has a much higher color contrast	Not applicable.

See Also:

About Removing Features Created with the Create Application Wizard

5.3.3.3 Available Settings in the Create Application Wizard

Learn about Settings available when running the Create Application Wizard. Settings are used in the generation of the application and include the application ID, the database schema, Advanced Settings (such as the application definition, security, and globalization attributes), and application authentication.

This table lists available Settings when running the Create Application Wizard.

Page Type	Description	To Learn More
Application ID	Unique, numeric identifier for your application. This field contains an automatically generated identifier by default.	Not applicable.

Page Type	Description	To Learn More
Schema	Select the database schema which stores the database objects you want to use in this application.	Not applicable.
Authentication	Select how you want users to authenticate into your application.	See Establishing User Identity Through Authentication
Language	Select the primary language for this application.	Not applicable.

Page Type	Description	To Learn More
Advanced Settings	Set additional definition, security, and globalization settings. These settings can be applied when creating the application and can be edited post application creation. **Tip**: To learn more about these attributes, see field-level Help. **User Interface Attributes**: • **Apply UI Defaults** - Select **On** to have attribute defaults based on existing UI Defaults defined within this workspace. • **Copy from Existing Apps** - Select **On** to have attribute defaults to be based on existing Form pages, based on the same table, in current applications defined within this workspace. • **Table Prefixes** - Enter table prefixes currently used by existing database tables on which pages are based. These prefixes may be different from the framework table prefix (for database objects created by this application when generated). **Description**: • **Short Description** - Enter text to be displayed as a subtitle, under the application name on the home page. • **Description** - Enter text to be displayed on the About This Application page, accessed from the help icon in the navigation bar. **Settings**: • **Add "Built with APEX" to Footer** - When set to **On**, Oracle APEX adds the text "Built with Love using Oracle APEX" to the footer of every page. • **Version** - Enter the application version. The version is displayed by default on the bottom of each page. • **Logging** - Determines whether or not user activity is recorded in the activity log. When set to **On**, every page view is logged, enabling an administrator to monitor user activity for each application. Disabling logging may be advisable for high volume applications.. **Note**: You can only modify this attribute if the feature configuration **Application Activity Logging** in Oracle APEX Internal Administration is set to **Use Application Setting**. • **Debugging** - Determines whether debug mode can be enabled using the browser. **On** enables the application to enable debug mode using the browser at runtime. **Off** disables the application from enabling in debug from the browser. Debug can be enabled programmatically regardless of this debug setting. If the application is run from the APEX development environment, debugging can always be enabled. **Security**:	Not applicable.

Page Type	Description	To Learn More
	• **Deep Linking** - Enables or prevents deep linking to an application. If **On**, the URL to a specific page ultimately redirects there, possibly after the user has to logged in. If **Off**, the URL does not contain a valid session ID, APEX redirects to the application's home page. • **Maximum session length in seconds** - Defines how long (in seconds) sessions can exist and be used by this application. • **Maximum session idle time in seconds** - Defines the time between the last page request and the next page request. **Globalization**: • **Document Direction** - Set document direction left-to-right or right-to-left. • **Date Format** - Determines the date format to be used in the application. • **Date Time Format** - Specify the date time format to be used in the application. • **Timestamp Format** - Determines the timestamp format to be used in the application. • **Timestamp with Timezone Format** - Determines the timestamp with time zone format to be used in the application.	
UI Defaults	Edit UI Defaults.	See Managing UI Defaults in *Oracle APEX SQL Workshop Guide*

5.3.3.4 About Removing Features Created with the Create Application Wizard

Remove features created with a wizard using build options.

When you add a feature using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Oracle APEX engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the APEX engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features by first removing the components associated with the build option and then deleting the build option.

See Also:

- Including or Excluding Build Options
- Deleting Build Options and Associated Components

5.4 Creating an App Using Generative AI

Create an application using Generative AI and enter natural language.

Note:

This feature requires you create a Generative AI Service and enable the **Used by App Builder** setting. If no Generative AI Services are available in the workspace, the **Create App Using Generative AI** option in the Create Application Wizard is hidden. To learn more, see Managing Generative AI Services.

When you click **Create App Using Generative AI** option in the Create Application Wizard, the APEX Assistant appears and guides your through the process of creating an application.

Tip:

If you have made changes to the tables in your workspace, the APEX Assistant may prompt you to refresh the data dictionary cache to work with the latest tables. Follow the on-screen instructions. If you want to send a sample prompt from APEX Assistant, you can install any of the available sample dataset. Once installed, APEX uses two of the most recently installed sample datasets and APEX Assistant provides sample prompts as quick pick buttons. Click on one of these buttons to send the sample request to the AI Service.

The example that follows builds an application on the sample dataset, Project Data. To learn more about installing a sample dataset, see Using Sample Datasets in *Oracle APEX SQL Workshop Guide*.

To create an app using Generative AI:

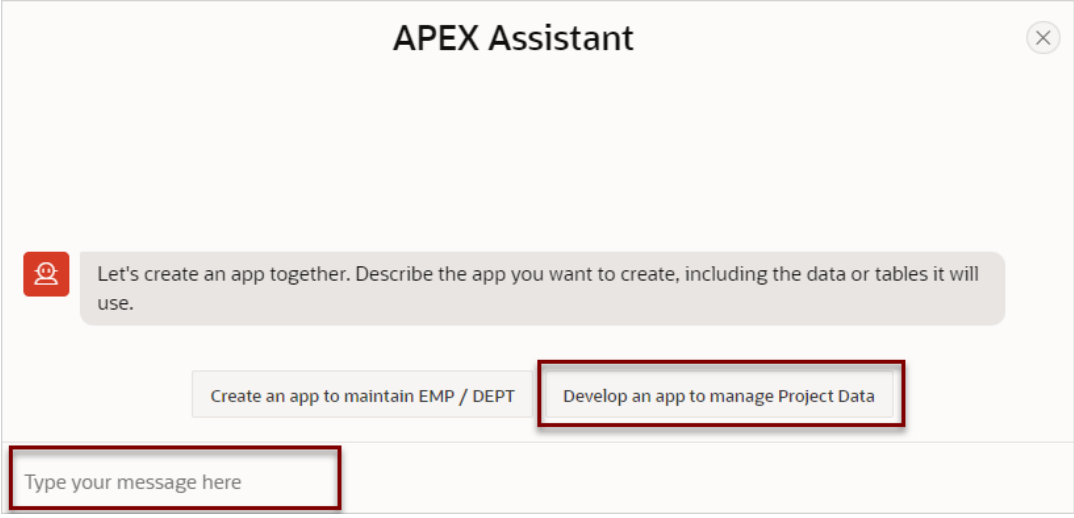
1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating your application.
3. Click **Create App Using Generative AI**.

   The APEX Assistant appears.

   Tip:

   If APEX Assistance does not detect any tables, it may prompt you to browse the sample datasets included with APEX. If prompted, click **Browse Sample Datasets** and install a sample dataset.



4. APEX Assistant offers two options:
   - Select an pre-defined option.

     Based on the Sample Datasets installed, APEX Assistant displays possible options. In the previous image, the pre-defined option, **Develop an app to manage Project Data** is highlighted.
   - Type your message here - In the field provided, enter text describing the application you want to build and click **Send Message** (or press **Enter**).

     For example, if you enter `create an app on project tables`, APEX Assistant provides a summary of the app to be created, including a list of pages and enabled features. Click **Create Application** to continue.
5. In APEX Assistant, click **Create an app to maintain Project Data**.

   APEX Assistant displays a brief summary outlining the features and pages to be created.
6. Enter more messages to iterate through application design using natural language. You can add, remove, or rename pages. You can add or remove features. You can also change to use different table or data.
7. Once satisfied, click **Create Application** to continue.

   The Create an Application page appears with the Name and Pages included and Features and Setting configured.
8. Review the Create an Application page. Update or change any pages or options. To learn more, see Running the Full Create Application Wizard.
9. Click **Create Application**.

See Also:

- Available Page Types in the Create Application Wizard
- Available Features in the Create Application Wizard
- Available Settings in the Create Application Wizard

5.5 Creating an Application from a File

Learn how to upload a file or copy and paste CSV data and then create an application on the uploaded data.

- About Creating an Application from a File
  Upload a CSV, XLSX, XML, TXT, or JSON file, or copy and paste CSV data and then create your application.
- Creating an Application by Uploading File
  Upload a file to the database and then create an application based on the new table.
- Creating an Application by Copying and Pasting from a File
  Copy and paste column delimited data from a CSV file, or selecting a sample and then create an application based on the new table.

5.5.1 About Creating an Application from a File

Upload a CSV, XLSX, XML, TXT, or JSON file, or copy and paste CSV data and then create your application.

When you run the Create Application Wizard and select **Create App From a File**, the Load Data Wizard appears. On the Load Data Wizard, you can choose to:

- **Upload a File**. Select this option to load data from a CSV, XLSX, XML, TXT, or JSON file.
- **Copy and Paste**. Select this option to copy and paste column delimited data. You have the option of selecting a sample data set from the **Select Sample** list.

Once you load data into the database table, the wizard provides an option to Continue to the Create Application Wizard. When you continue, a report with form and dashboard pages based on the new table displays under **Pages**. You then finish the application creation process by naming the application.

> **See Also:**
>
> Loading Data in *Oracle APEX SQL Workshop Guide*

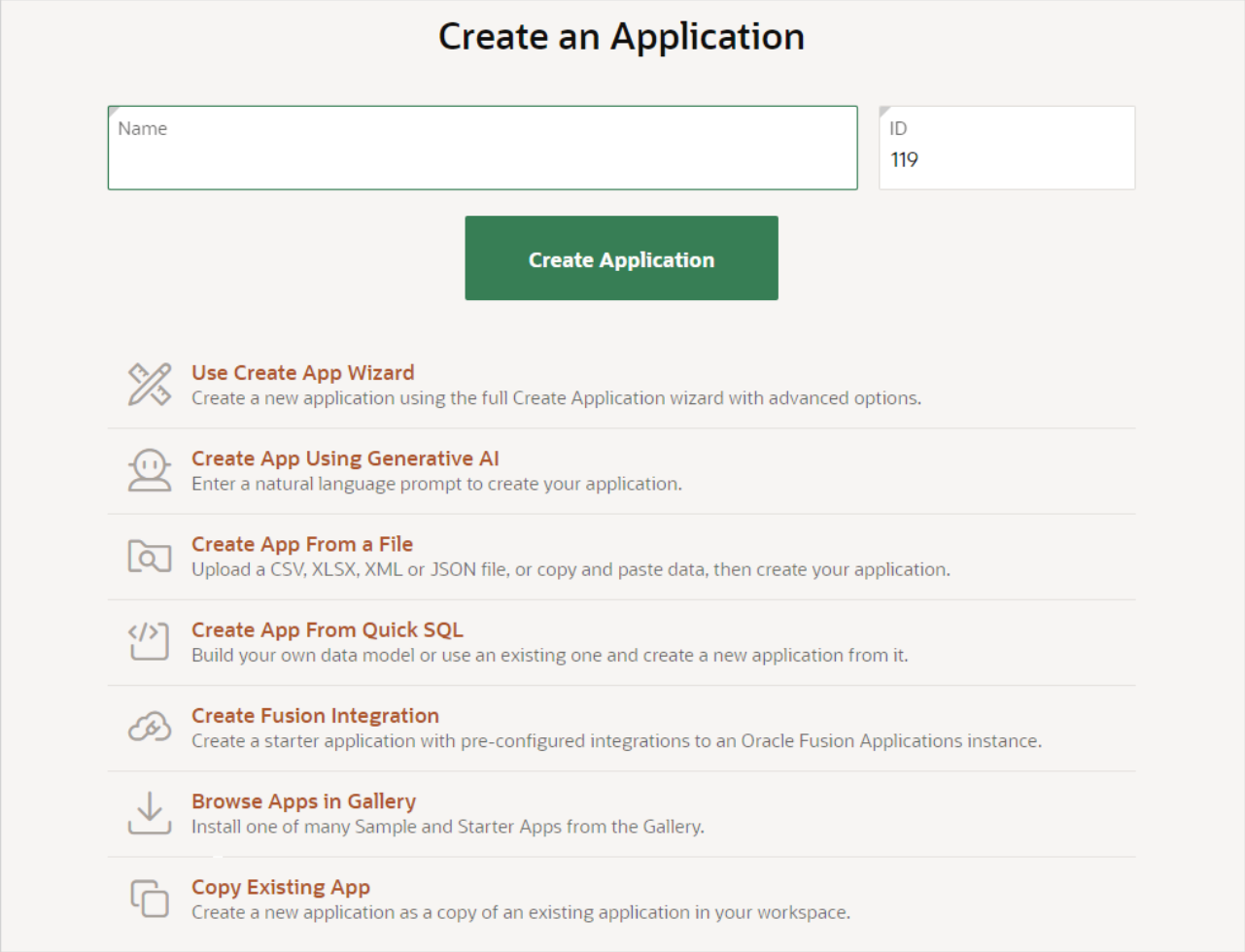
5.5.2 Creating an Application by Uploading File

Upload a file to the database and then create an application based on the new table.

To create an application from a file:

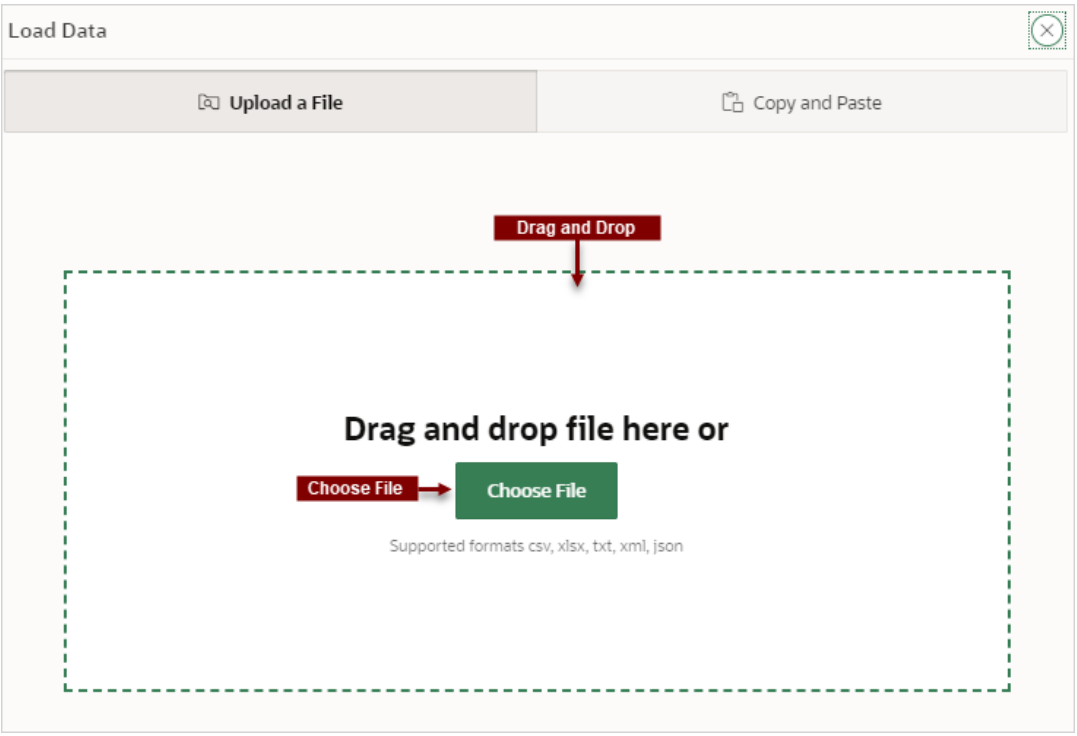
1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.



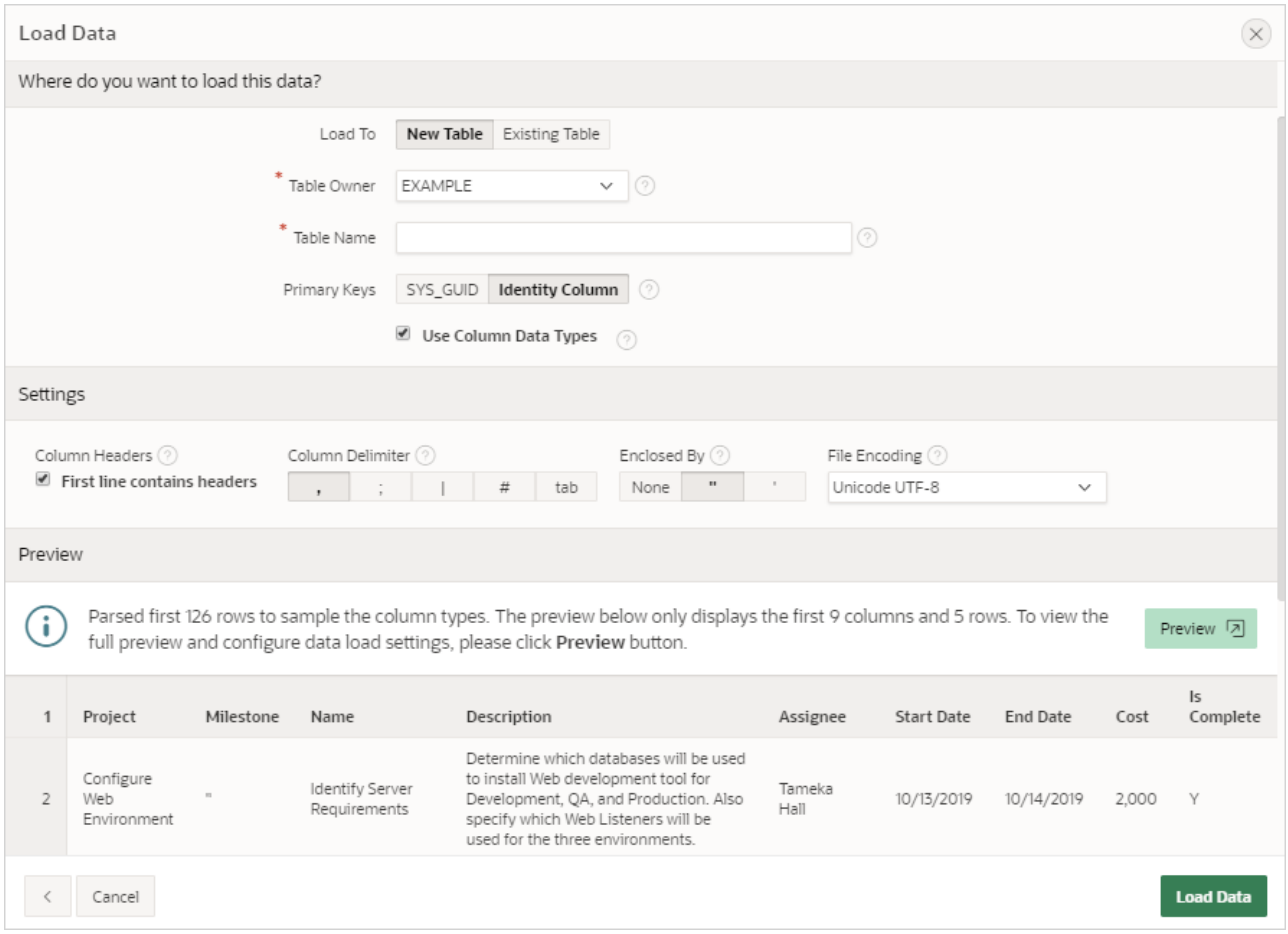
3. Click **Create App From a File**.

   The Load Data page appears with the **Upload a File** tab selected.



4. In the **Upload a File** tab, you have two options:
   - **Drag and Drop** - Drag and drop the file to the region in the center of the page.
   - **Choose File** - Click **Choose File** and then navigate to and select the file.

   The Load Data page appears containing the sections **Where do you want to load this data?**, **Settings**, **Preview**. Note that the Load Data page that appears differs slightly depending upon the selected file type.



5. Under **Where do you want to load this data?**, select where to load the data.

   What appears next depends upon your selection.

6. To load data to a new table:
   a. Load To - Select **New Table**.
   b. Table Owner - Select the database schema.
   c. Table Name - Identify the table name you would like to create. By default, all table names are converted to upper case.
   d. Primary Keys - Select how to set the primary key. Options include **SYS_GUID** or **Identity Column**. To learn more, see field-level Help.
   e. Use Column Data Types - Select to use the column types parsed from the file. If un-checked, the table columns will be created as `VARCHAR2(4000)`.

7. To load data to an existing table:
   a. Load To - Select **Existing Table**.
   b. Table Owner - Select the database schema.
   c. Table Name - Select the database table to load data.

      A message appears indicating the number of columns mapped to the target table.

   d. Click **Configure** to configure any remaining columns. On Configure, under Column Mapping:
      - Rows to Sample - Select the maximum number of rows to be used to sample column types.

- Treat as Null - The entered case insensitive value will be treated as NULL. For example, "tbd", "n/a", "unknown", "-".
- Column Mapping - Configure any additional attributes.
- Click **Save Changes**.

e. Update Method - Options include:

- **Append** - If data already exists in the table, the wizard appends the new rows into the table. If data does not exist, the new rows are simply loaded.
- **Replace** - The entire table is replaced by deleting the table first and then loading the new rows into the table.

8. Under Settings:

   a. Column Headers - Select this option if your data contains column names in the first row.

   b. Column Delimiter - Select a column delimiter character (if CSV or TXT file).

   c. Enclosed By - Select a column enclosed by character (if CSV or TXT file). This character deliminates the starting and ending boundary of a data value. If you specify a delimiter character, the wizard ignores white space occurring before the starting and ending boundary of a data value.

   d. File Encoding (if applicable) - Choose the character set in which the file is encoded.

9. **Preview** displays a sample of columns and rows. The exact number varies based on the type of file. Click one of the following tabs:

   - **Preview** - Displays a sample of columns and rows. The exact number varies based on the type of file.
   - **Column Mapping** - Enables you to configure any remaining columns:
     - Rows to Sample - Select the maximum number of rows to be used to sample column types.
     - Treat as Null - The entered case insensitive value will be treated as NULL. For example, "tbd", "n/a", "unknown", "-".
     - Column Mapping - Configure any additional attributes.
     - Click **Save Changes**.

10. Click **Load Data**.

    A confirmation page appears indicating the table and rows have been created. You have the following options:

    - **View Table** - Click **View Table** to exit from the current wizard and opens Object Browser so you can view the newly created table.
    - **Continue to the Create Application Wizard** - Click **Continue to the Create Application Wizard** to load the Create Application Wizard.

      Note that if all rows fail to load, this button does not display.

11. Click **Continue to Create Application Wizard**.

    The Create an Application page in the Create Application Wizard appears. Report with form and dashboard pages based on the new table display under **Pages**. From here, you can edit page name or create additional pages based on the imported data.

12. Complete the Create Application Wizard.

> **See Also:**
>
> - Using the Full Create Application Wizard
> - Running the Full Create Application Wizard

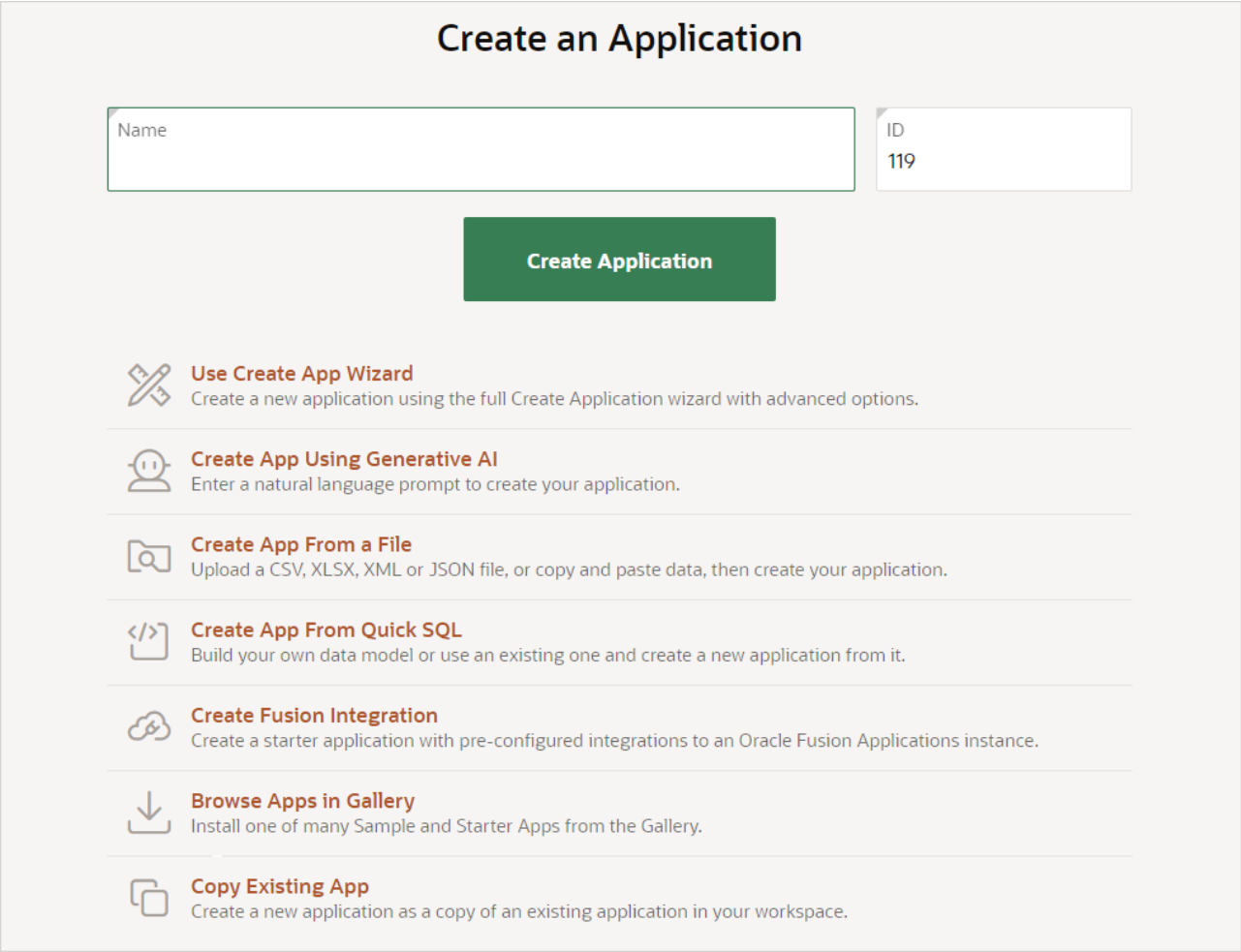
5.5.3 Creating an Application by Copying and Pasting from a File

Copy and paste column delimited data from a CSV file, or selecting a sample and then create an application based on the new table.

To create an application by copying and pasting column delimited data from a CSV file, or selecting a sample:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

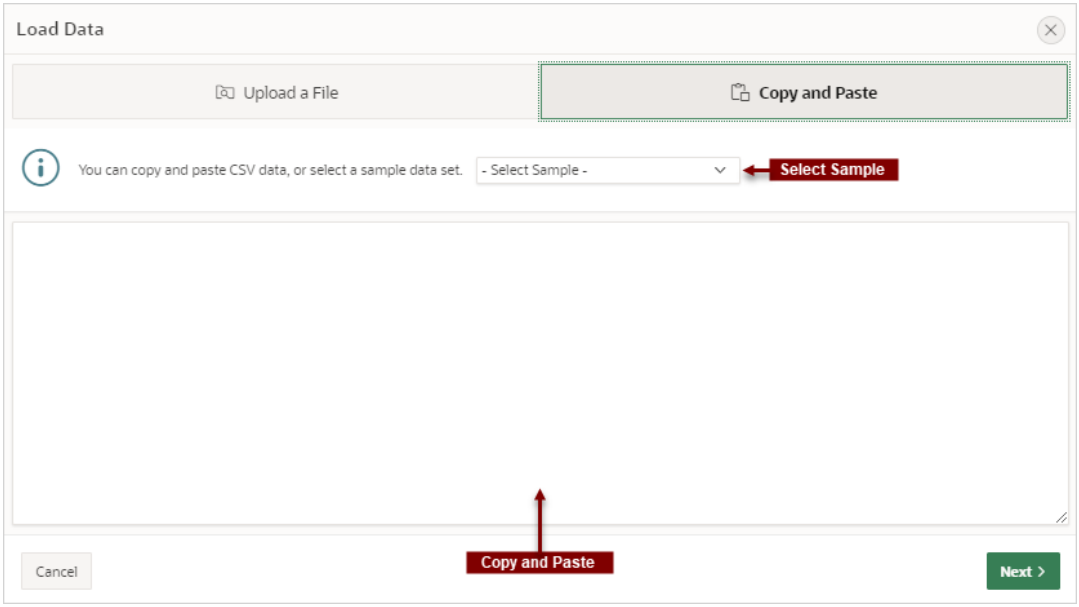
   The wizard prompts you to select a method of creating an application.



3. Click **Create App From a File**.

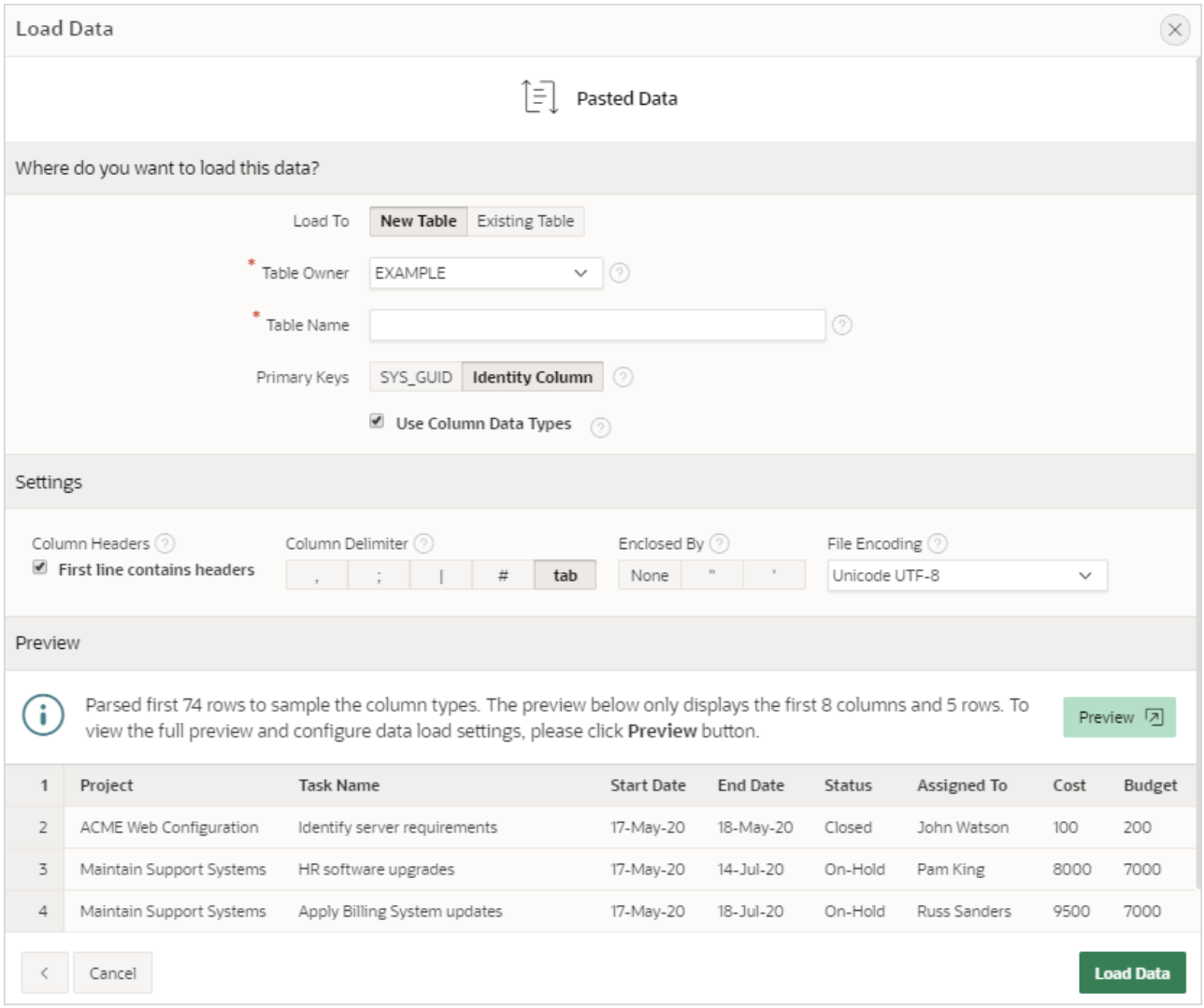
   The Load Data page appears

4. Select the **Copy and Paste** tab.



5. To load data, do one of the following:
   - **Select a Sample** - Select a sample database from the **Select Sample** list.
   - **Copy and Paste** - Copy and paste CSV data.
6. Click **Next**

   The Load Data page appears.



7. Under **Where do you want to load this data?,** select where to load the data.

   What appears next, depends upon your selection.

8. To load data to a new table:

   a. Load To - Select **New Table**.

   b. Table Owner - Select the database schema.

   c. Table Name - Identify the table name you would like to create. By default, all table names are converted to upper case.

   d. Primary Keys - Select how to set the primary key. Options include **SYS_GUID** or **Identity Column**. To learn more, see field-level Help.

   e. Use Column Data Types - Select to use the column types parsed from the file. If un-checked, the table columns will be created as VARCHAR2(4000).

9. To load data to an existing table:

   a. Load To - Select **Existing Table**.

   b. Table Owner - Select the database schema.

   c. Table Name - Select the database table to load data into.

      A message appears indicating the number of columns that are mapped.

d. Click **Configure** to configure any remaining columns. On Configure, under Column Mapping:
   - Rows to Sample - Select the maximum number of rows to be used to sample column types.
   - Treat as Null - The entered case insensitive value will be treated as NULL. For example, "tbd", "n/a", "unknown", "-".
   - Column Mapping - Configure any additional attributes.
   - Click **Save Changes**.

e. Update Method - Options include:
   - **Append** - If data already exists in the table, the wizard appends the new rows into the table. If data does not exist, the new rows are simply loaded.
   - **Replace** The entire table is replaced by deleting the table first and then loading the new rows into the table.

10. Under Settings:
    a. Column Headers - Enable **First Line Contains** if your data contains column names in the first row.
    b. Column Delimiter - Select a column delimiter character.
    c. Enclosed By - Select a column enclosed by character . This character deliminates the starting and ending boundary of a data value. If you specify a delimiter character, the wizard ignores white space occurring before the starting and ending boundary of a data value.
    d. File Encoding - Choose the characterset in which the file is encoded.

11. **Preview** displays a sample of columns and rows. The exact number varies based on the type of file. Click one o the following tabs:
    - **Preview** displays a sample of columns and rows. The exact number varies based on the type of file.
    - **Column Mapping** enables you to configure any remaining columns:
      – Rows to Sample - Select the maximum number of rows to be used to sample column types.
      – Treat as Null - The entered case insensitive value will be treated as NULL. For example, "tbd", "n/a", "unknown", "-".
      – Column Mapping - Configure any additional attributes.
      – Click **Save Changes**.

12. Click **Load Data**.

    A confirmation page appears indicating the table and rows have been created. You have the following options:
    - **View Table** - Exits from the current wizard and opens Object Browser so you can view the newly created table.
    - **Continue to the Create Application Wizard**

13. Click **Continue to Create Application Wizard**.

    The Create Application Wizard appears. The wizard pre-populates pages based on the new table and the pages display under **Pages**. From here you can edit the page name or create additional pages based on the imported data.

14. Complete the Create Application Wizard.

> **See Also:**
>
> - Using the Full Create Application Wizard
> - Running the Full Create Application Wizard

5.6 Integrating APEX with Fusion Applications

Use Oracle APEX to build extensions to Fusion Applications.

- About Oracle APEX and Fusion Applications Integration Requirements
  Learn about requirements for integrating Oracle APEX with Fusion Applications.
- About the Fusion Integration Starter Application
  Learn about the Create Fusion Integration starter application.
- Creating an Application with Integrations to Fusion Applications
  Create a starter application with pre-configured integrations to Fusion Applications.

5.6.1 About Oracle APEX and Fusion Applications Integration Requirements

Learn about requirements for integrating Oracle APEX with Fusion Applications.

Requirements for integrating Oracle APEX with Fusion Applications include:

- A Fusion Application subscription.
- An Autonomous Database with a workload type of Data Warehouse or Transactional Processing.
- SQLNet connections to an Oracle Database in the same tenancy.

> **Note:**
>
> Oracle APEX Application Development (APEX Service) does not support Fusion Application integrations since APEX workloads do not include SQLNet connections.

About the Integration Process

Integrating APEX with Fusion Applications is a three-stage process:

- **Create a Connection** - Run the Database Tools Fusion Integration Wizard to create a connection between the Oracle Fusion Applications instance and the Oracle Database where APEX resides.

  See Integrating Oracle APEX with Fusion Applications in *Oracle Cloud Infrastructure Documentation*.
- **Create a Starter Sample Application** - Run the Create Application Wizard in App Builder to create a starter sample application with pre-configured integration to the Oracle Fusion Applications instance. See Creating an Application with Integrations to Fusion Applications.

5.6.2 About the Fusion Integration Starter Application

Learn about the Create Fusion Integration starter application.

When you run the Create Application Wizard and select the **Create Fusion Integration** option, the wizard creates a starter application and completes the following database setup tasks:

- **Remote Server** - Creates a Remote Server at the workspace-level named according to the convention `APEX_FA_FUSION_INSTANCE_REST_APIS` where *`FUSION_INSTANCE`* is the System Name which can be found in your Fusion Applications Cloud Console. This Remote Server is pre-configured to reference the base URL of your Fusion REST API endpoints.
- **Web Credential** - Creates a Web Credential a name of the format `APEX_FA_FUSION_INSTANCE_DBMS_CRED` which utilizes Database Credentials created by the Database Tools Fusion Integration Wizard. This credential is used both by the pre-configured Authentication Scheme and by the sample REST Data Source.
- **Authentication Scheme** - Creates a pre-configured Authentication Scheme with a name of the format `APEX_FA_FUSION_INSTANCE_OAUTH`. You can copy and subscribe to this Authentication Scheme in your other Fusion integrated applications to easily enable Single Sign-On in those apps too. This Authentication Scheme contains placeholders referencing configuration parameters set at the instance-level by the Database Tools Fusion Integration Wizard.
- **REST Data Source** - Creates a sample REST Data Source named `HCM Self Details` which references the Workspace level Remote Server and Web Credentials to obtain details about the currently signed in user. You can study the setup of this Data Source in order to create your own for other Fusion REST API endpoints.

> **See Also:**
>
> - Establishing User Identity Through Authentication
> - Sample REST Data Sources
> - Managing REST Data Sources
> - Managing Remote Servers

5.6.3 Creating an Application with Integrations to Fusion Applications

Create a starter application with pre-configured integrations to Fusion Applications.

> **Note:**
>
> Before attempting this task, you must run the Database Tools, Integrate APEX with Fusion Applications Wizard. See About Oracle APEX and Fusion Applications Integration Requirements.

To create a starter application with pre-configured integration to an Oracle Fusion Applications instance:

1. On the Workspace home page, click the **App Builder** icon.

2. Click **Create**.

   The wizard prompts you to select a method of creating your application.

3. Click **Create Fusion Integration**.

   > **Note:**
   >
   > If you have not created a connection between the Oracle Fusion Applications instance and the database where APEX resides, a warning message appears.

   The Create Fusion Integration page appears.

4. Configuration Options:

   a. Fusion Instance System Name - Select the Fusion instance with which you wish to integrate the new starter application. Only those Fusion instances that have been configured using the Database Tools integration wizard are available.

   b. Base URL for the REST APIs - The URL of your Fusion instance where the REST API endpoints can be found. Typically, the URL will be the `https://my.host.name/` portion of the URL you use to access your Fusion application in your browser.

      Once you select a **Fusion Instance System Name**, the Base URL is calculated based upon information provided by the Database Tools integration wizard.

      If the information was not provided by the wizard, then the value is calculated based on Oracle OCI standards. As a best practice, Oracle recommends you verify that any provided value matches your environment.

      If that calculation is still incorrect, then enter a value manually.

   c. API Status - Once **Base URL for the REST APIs** is populated, APEX attempts to contact the server at the specified location. The status of that request displays here.

      If the server is contacted, this attribute displays:

      ```
      200-OK
      ```

      If any other status displays, there is a network configuration issue between APEX and the Fusion Applications instance. Although you can proceed and create a starter application, you may not be able to run the application until the issues are resolved.

      If the network configuration or URL is incorrect, then this check may take several minutes to complete.

      > **Tip:**
      >
      > Possible issues include the incorrect configuration of ACLs or ACEs between APEX and the Fusion Applications instance. For more information, see Enabling Network Services in Oracle Database in *Oracle APEX Installation Guide*.

5. Advanced Settings - The wizard creates the following attributes in the starter application and pre-populates with default values.

   a. Application Name - Enter the name used to identify the application to developers.

   b. Application Alias - An alphanumeric identifier, unique within the workspace, for the application. This identifier appears in the URL used to access the application.

c. Application ID - Enter a unique, numeric identifier for your application. This field contains an automatically generated identifier by default. Application IDs between 3000 to 9000 are reserved for internal use by Oracle APEX.

> **Tip:**
>
> You can override the default values to provide your own Application Name and Alias. Be cautious when overriding the Application ID since it must be unique within the entire APEX instance.

6. Click **Create Application**.

   The Create Application Wizard creates a starter application and completes a number of database setup tasks. See About the Fusion Integration Starter Application.

5.7 Using Application Blueprints

Each iteration of the Create Application Wizard saves an application blueprint which you can use to start adding pages to an application, or edit the application definition as a JSON document.

- What Is an Application Blueprint?
  View or load an application blueprint by running the Create Application Wizard.
- Loading an Application Blueprint
  Load an application blueprint by running the Create Application Wizard and clicking the Load Blueprint link at the top of the Create Application Wizard.
- Viewing an Application Blueprint
  View an application blueprint by running the Create Application Wizard and clicking the View Blueprint link at the top of the Create Application Wizard.

> **See Also:**
>
> To learn more about JSON, see https://www.json.org/

5.7.1 What Is an Application Blueprint?

View or load an application blueprint by running the Create Application Wizard.

You can view the application blueprint by running the Create Application Wizard and clicking the **View Blueprint** link at the top of the Create Application Wizard. The Application Blueprint window transforms and displays the current application definition as a JSON document. Edit the document to update application properties and then click **Apply Blueprint** to change the current application details.

If you need to create a large number of similar pages you can copy the relevant JSON code for one page, paste it numerous times, and then make small adjustments to the JSON. This approach can be significantly quicker than manually clicking the **Add Page** button for each page in the wizard.

You can also replace the blueprint with the definition from a previously generated application. Click **Load a Previous Blueprint** to replace the blueprint with the definition from a previously

generated application. This approach is a simple way to iterate the application design by adding additional pages, changing the features, or tweaking the settings, before generating the application again.

5.7.2 Loading an Application Blueprint

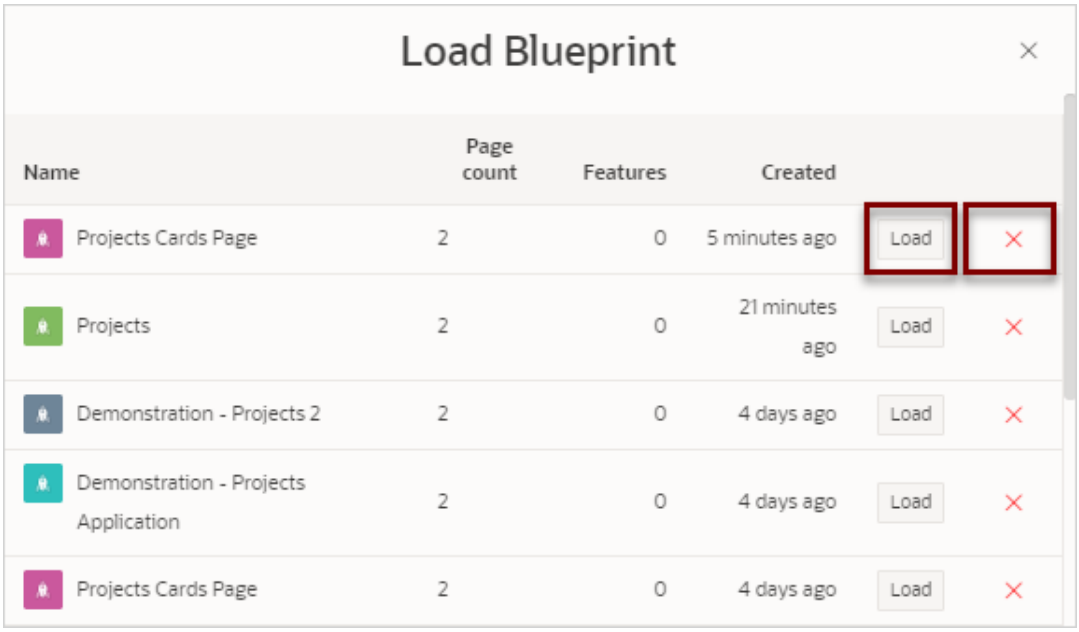
Load an application blueprint by running the Create Application Wizard and clicking the Load Blueprint link at the top of the Create Application Wizard.

To load an application blueprint:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.
3. Click **Use Create App Wizard**.

   The Create an Application page reappears.
4. At the top of the page, click the **Load Blueprint** link.

   The Load Blueprint page appears.



5. On Load Blueprint, you can:
   - **Load a new blueprint**. Locate the application and click **Load**.
   - **Delete a blueprint**. Locate the application and the **Delete Blue** icon.

5.7.3 Viewing an Application Blueprint

View an application blueprint by running the Create Application Wizard and clicking the View Blueprint link at the top of the Create Application Wizard.

To view an application blueprint:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.
3. Click **Use Create App Wizard**.

4. At the top of the page, click the **Load Blueprint** link.

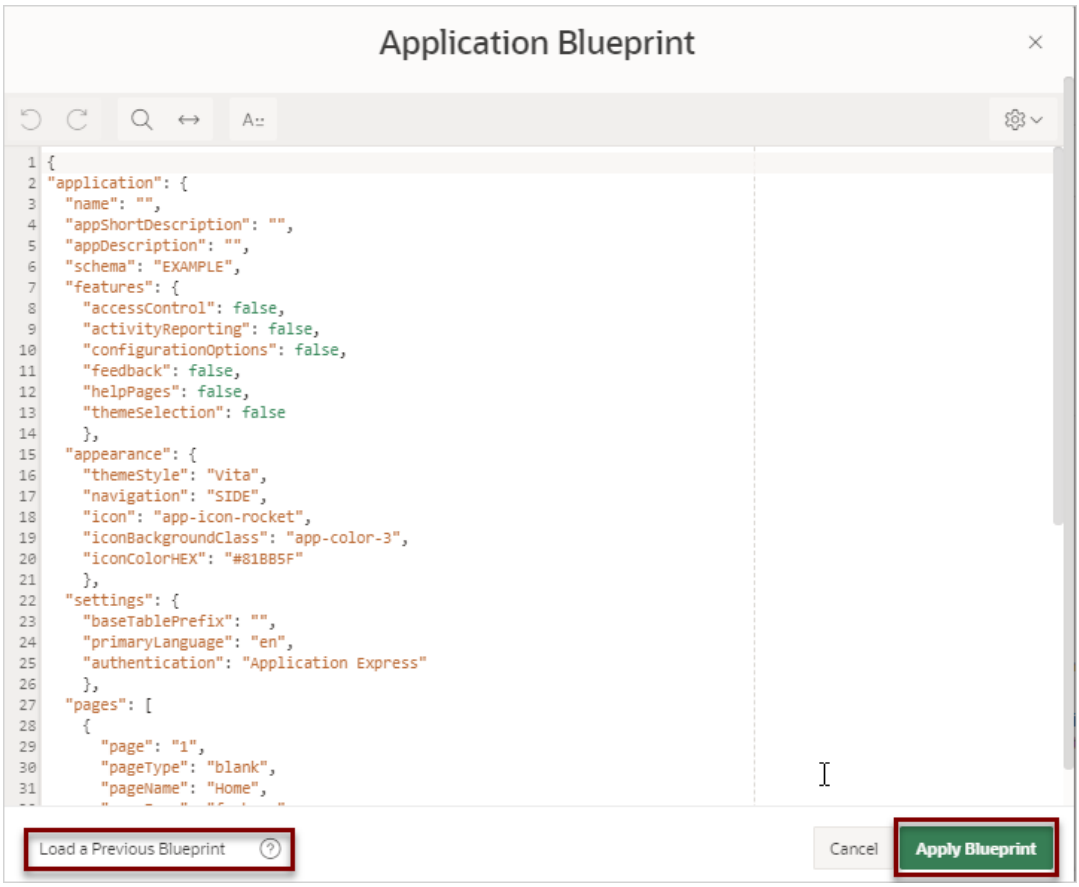
   The Load Blueprint page appears.

5. On Load Blueprint, find the application and click **Load**.

   The Create an Application page reappears.

6. At the top of the page, click **View Blueprint**.

   The Application Blueprint window appears displaying the application definition in JSON format.



7. To edit the document:

   a. Update application definition.

      The following buttons display at the top of the Script Editor:

      - **Undo - Ctrl+Z** — Removes, or undoes, the most recent line edit made in the Script Editor.
      - **Redo - Ctrl+Y** — Repeats the most recent line edit made in the Script Editor.
      - **Find - Ctrl+F** — **Find** resembles a magnifying glass. Click **Find** to perform a basic search.
      - **Replace - Ctrl+Shift+F** — **Replace** resembles a two-sided arrow. Click **Replace** to search and replace existing code.

   b. Click **Apply Blueprint** to change the current application definition.

8. To replace the blueprint with the definition from a previously generated application:

   a. Click **Load Previous Blueprint**.

b. On Load Blueprint, locate the blueprint and click **Select**.

c. Click **Apply Blueprint** to change the current application details.

5.8 Copying an Application

Create a copy of an existing application.

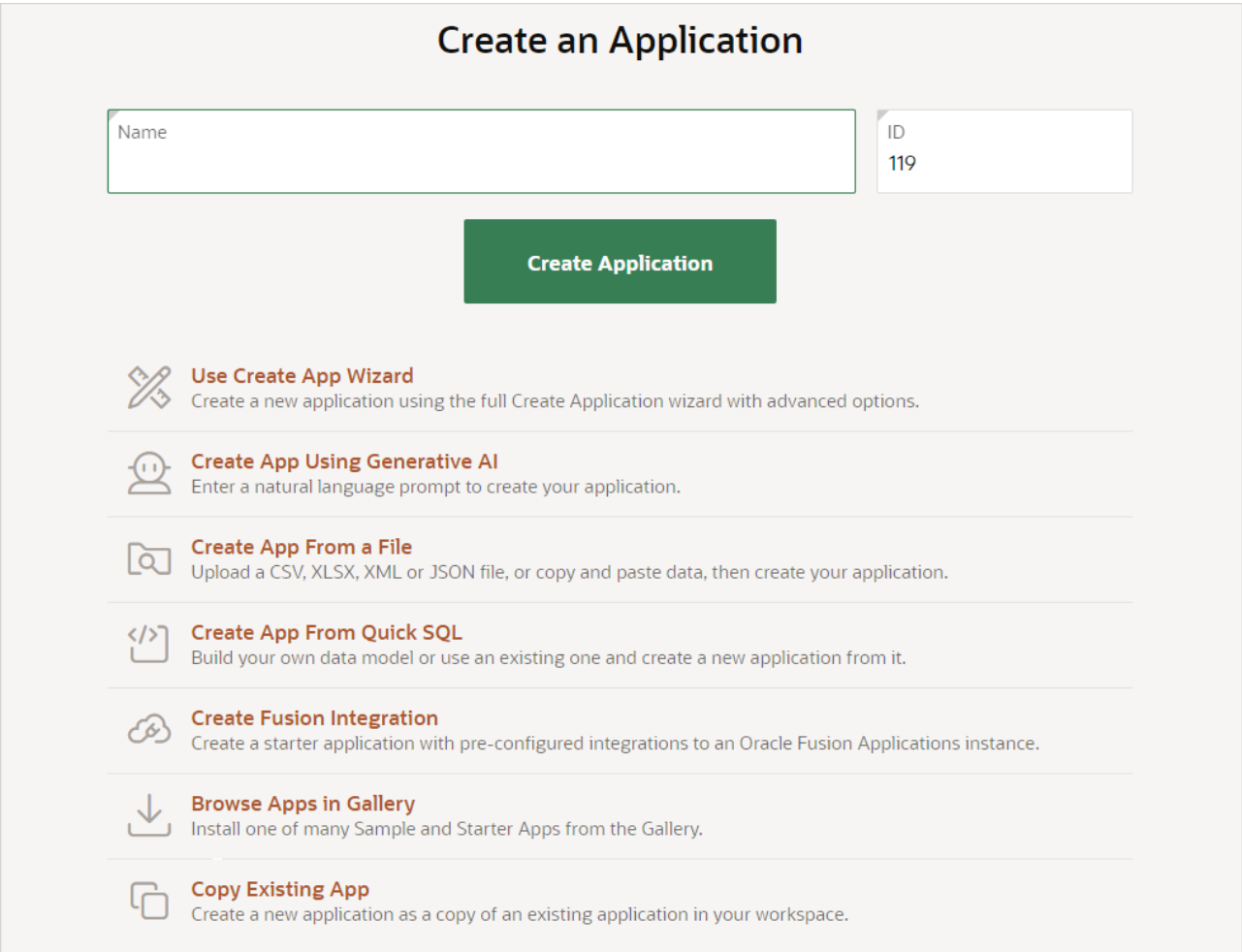
Create a copy of an existing application by running the Create Application Wizard and clicking **Copy Existing App**, or by selecting the application and then **Copy this Application** from the Tasks list on the Application home page.

- Copying an Application by Running the Create Application Wizard
- Copying an Application from the Application Home Page

5.8.1 Copying an Application by Running the Create Application Wizard

To copy application by running the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.



3. Click **Copy Existing App**.

4. For Identify Copy Target:
   a. Copy From - Select the name of the application to copy.
   b. Copy To - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
   c. New Application ID - Specify an integer value that identifies the application. The application ID must be unique within your Oracle APEX instance. Application IDs in the range of 3000 to 9000 are reserved for internal use by APEX
   d. Copy Supporting Object Definitions - You can copy supporting object definitions, configuration options, and scripts with your application. Select **On** to include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file. Select **Off** to not include supporting object definitions.
5. Click **Next**.
6. Confirm your selection and click **Create Application**.

5.8.2 Copying an Application from the Application Home Page

To copy an application from the Application home page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application to copy.

   The Application home page appears.
3. From the Tasks list, click **Copy this Application**.
4. For Identify Copy Target:
   a. New Application ID - Enter a unique integer value to identify the application. Application IDs between 3000 to 9000 are reserved for internal use.
   b. New Application Name - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
   c. Copy Supporting Object Definitions - Copy supporting object definitions, configuration options, and scripts with your application.. Options include:
      - **On** - Include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.
      - **Off** - Do not include supporting object definitions.
   d. Click **Next**.
5. Click **Copy Application**.

5.9 Creating a Working Copy to Merge, Refresh, or Compare

Create a Working Copy of an application to merge, refresh or compare changes.

- About Working Copies
  Learn about Working Copies.
- Creating a Working Copy of an Application
  Create a Working Copy of an application on the Application Home page.
- About Working Copy Menus and Tasks
  Learn about menus and tasks available in a Working Copy.

- Viewing and Editing a Working Copy
  View and edit a Working Copy of an application.
- Comparing Changes between a Working Copy and Main
  Compare changes between a Working Copy with the Main application.
- Comparing Changes Across Working Copies
  View and compare changes between single pages across different Working Copies.
- Merging Changes from a Working Copy into Main
  Merge changes from a Working Copy into the Main application.
- Refreshing a Working Copy with Changes from Main
  Refresh a Working Copy with changes in the Main application.
- Deleting a Working Copy
  Delete a Working Copy copy.
- View or Edit Working Copy Details
  View or edit the Working Copy name or description on Working Copy Details.

5.9.1 About Working Copies

Learn about Working Copies.

A **Working Copy** enables developers to create a special copy of an application, make changes in the copy, and then merge those changes back into the main development application (or **Main application**). Similarly, you can also merge changes from the Main application back into the Working Copy. During the export process, you can view a comparison (or diff) of the Working Copy and the Main application and then select which changes to merge.

A common use case for this feature is when you want to test some changes on a page first and then merge them back into the Main application later.

About the Comparison Process

When you compare a Working Copy to the Main application, a split YAML export is taken of both applications and a checksum is generated on the individual files. The working copy is exported with original IDs to ensure that the IDs are the same between both applications. The checksums are then compared to identify the differences between the two applications.

> **Note:**
>
> The YAML export of both applications is performed in a background job. Therefore, your database must have `JOB_QUEUE_PROCESSES` enabled (that is, `JOB_QUEUE_PROCESSES` must have a value greater than zero). If the job queue is disabled you will see no progress in the progress bar. Please be aware that for multi-tenant databases the CDB `JOB_QUEUE_PROCESSES` setting can override the PDB setting so both should be checked. To learn more about `JOB_QUEUE_PROCESSES`, see Controlling the Number of Concurrent Jobs in *Oracle APEX Installation Guide*.

When comparing a Working Copy to the Main application, use the `MAIN_APP_ID` substitution string to identify the main application ID if the current application is a working copy.

About the Merge Process

The merge process takes a subset of pages and shared components from a Working Copy and imports them back into the Main application. If the page or shared component exists, it is

replaced. Therefore, changes in your Main application that are not in your Working Copy will be overwritten by the Working Copy merge.

Unsupported Component Types

Working Copies do not support the automatic merging of the following component types:

- Translations
- Themes
- Theme Templates
- Supporting Objects
- Application Properties
- Workspace-level components (such as App Groups, Credentials, and Remote Servers)

For unsupported component types, Oracle suggests developers merge the working Working Copy, delete it, and then manually apply the additions or changes for the unsupported component types.

Limitations of Working Copies

Limitations of Working Copies include:

- You cannot create a Working Copy of a Working Copy.
- Deleted working copy pages or Shared Components must be manually deleted in the Main application.
- Components listed as Missing may have been removed in the Working Copy or may only exist in the Main application.
- When a working copy is created all automations are set to disabled. If automations are enabled in the Main application, they will show up in the differences report during a compare or merge.

  If an application contains automations, Oracle recommends deselecting the automations before merging or re-enabling automations manually after merging.

Known Issues for Working Copies

The known issues for Working Copies include:

- **Server Error** message - If you are running APEX on Oracle Database release 19c and using the working copy feature, when you choose the actions **Merge into Main**, **Refresh from Main**, or **Compare Changes**, you may get a "Server Error" message that is caused by an underlying ORA-00600 error message. For example:

  ```
  ORA-00600: internal error code, arguments: [qjsnplsDecDomRefCnt:noValP], [],
  [], [], [], [], [], [], [], [], [], []
  ```

  The fix for this error is to upgrade to Oracle Database release 19.16 or higher.
- **ORA-00001** error during merge - The Working Copy feature raises an ORA-00001 unique constraint error when attempting to merge changes for component names that are not unique. It is important to coordinate the naming of shared components across working copies and the main application to avoid this error occurring.

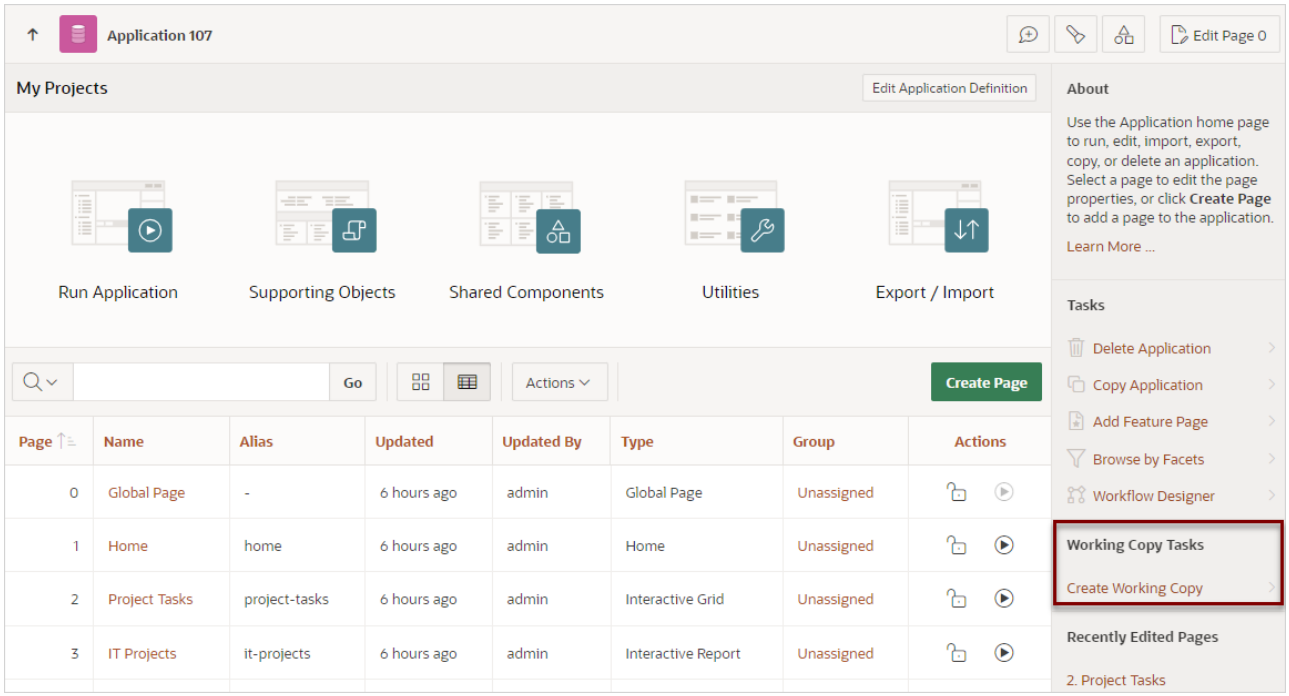
5.9.2 Creating a Working Copy of an Application

Create a Working Copy of an application on the Application Home page.

To create a Working Copy of an application:

1. View the application:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

      The Application home page appears.
2. On the Application home page, find **Working Copy Tasks** on the side bar.



3. Under **Working Copy Tasks**, click **Create Working Copy**.

   The Create Working Copy Wizard appears.
4. On Create Working Copy:
   a. Working Copy Name - Enter a short name or identifier for the Working Copy.
   b. Working Copy Description - Enter a description or purpose for the Working Copy.
   c. Click **Create Working Copy**.

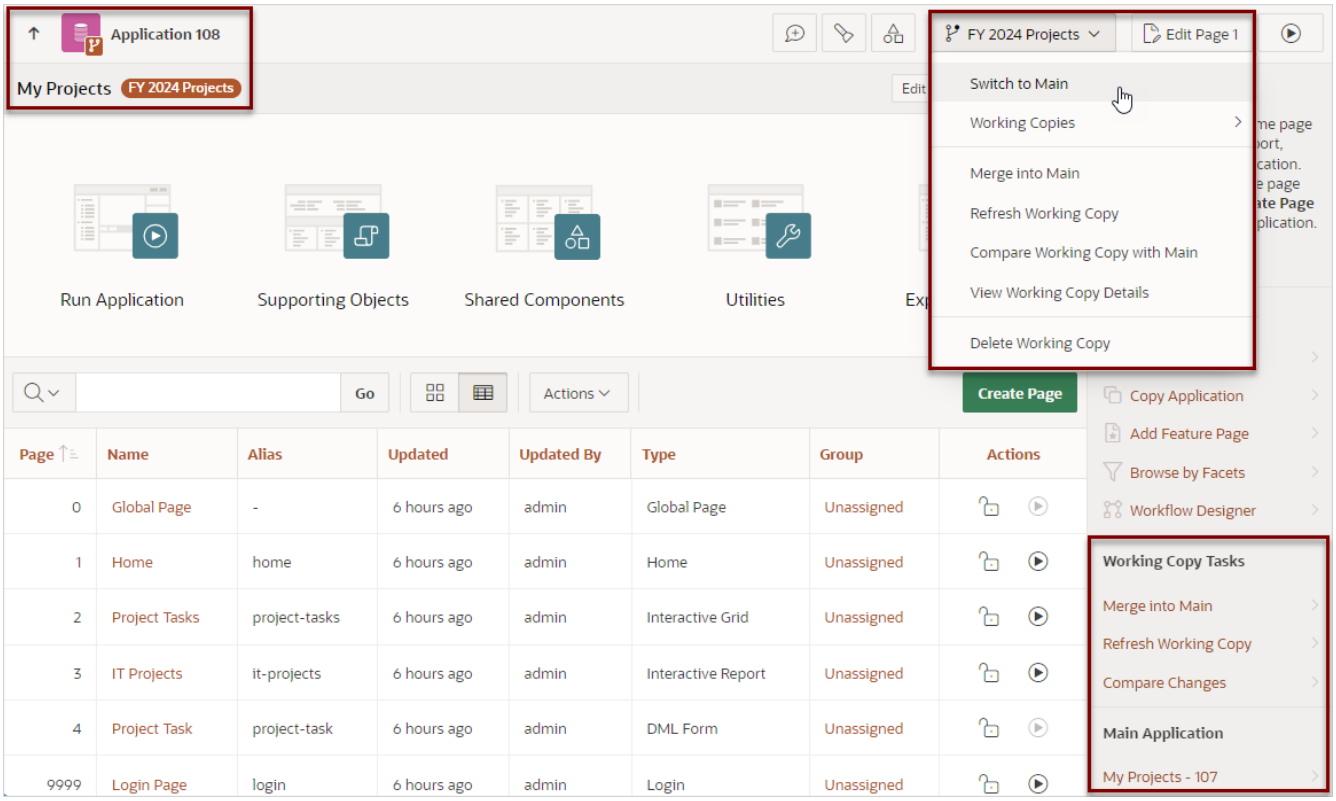
      The Application home page appears. Note that the **Working Copy Name** you defined previously appears next to the application name.

5.9.3 About Working Copy Menus and Tasks

Learn about menus and tasks available in a Working Copy.

Working Copy Menu on the Application Home Page

When viewing a Working Copy on the Application Home page, the Working Copy ***Name*** displays next to the application name and a **Working Copy: *Name*** menu displays on the menu bar between **Shared Components** and **Edit Page**.



Options on the **Working Copy: *Name*** menu include:

- **Switch to Main** - Links to the Main application.
- **Working Copies** - Displays links to existing working copies.
- **Merge into Main** - Merge changes from the Working Copy into the Main application.
- **Refresh Working Copy** - Import changes from the Main application into the Working Copy.
- **Compare Working Copy with Main** - Compare changes between the Working Copy and Main application.
- **View Working Copy Details** - View or edit the Working Copy Name or Description.
- **Delete Working Copy** - Delete the current Working Copy.

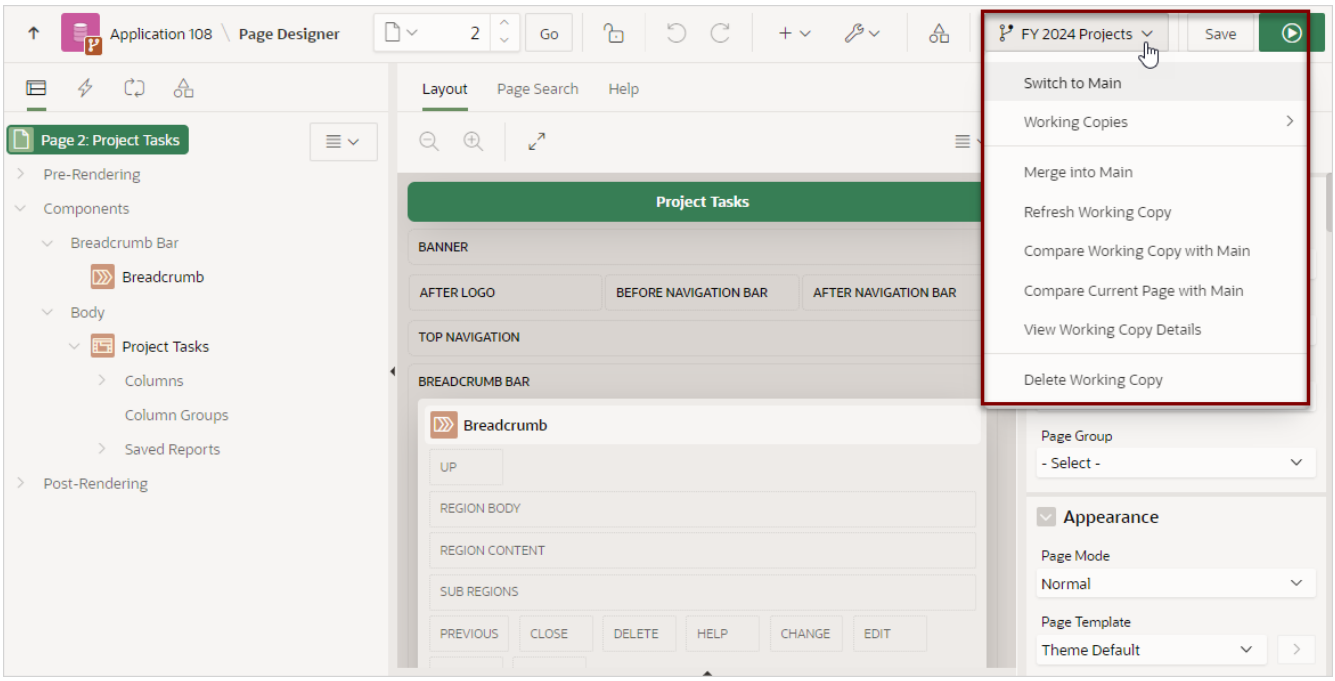
The side bar on Application home page features links to these commonly used options:

- **Working Copy Tasks**:
  - **Merge to Main**
  - **Refresh Working Copy**
  - **Compare Changes**.
- **Main Application** - Include a link to the Main application.

Viewing a Working Copy in Page Designer

A **Working Copy: *Name*** menu also appears between **Shared Components** and the **Save** on the Page Designer toolbar. This menu features the same options available on the Application home page and described previously. One additional option available on this menu:

- **Compare Current Page with Main** - Compare changes between the last saved version of the current page of the Working Copy open in Page Designer with the same page in Main.



> **Tip:**
>
> Detailed discussions of using most **Working Copy: *Name*** menu options can be found later in this topic.

5.9.4 Viewing and Editing a Working Copy

View and edit a Working Copy of an application.

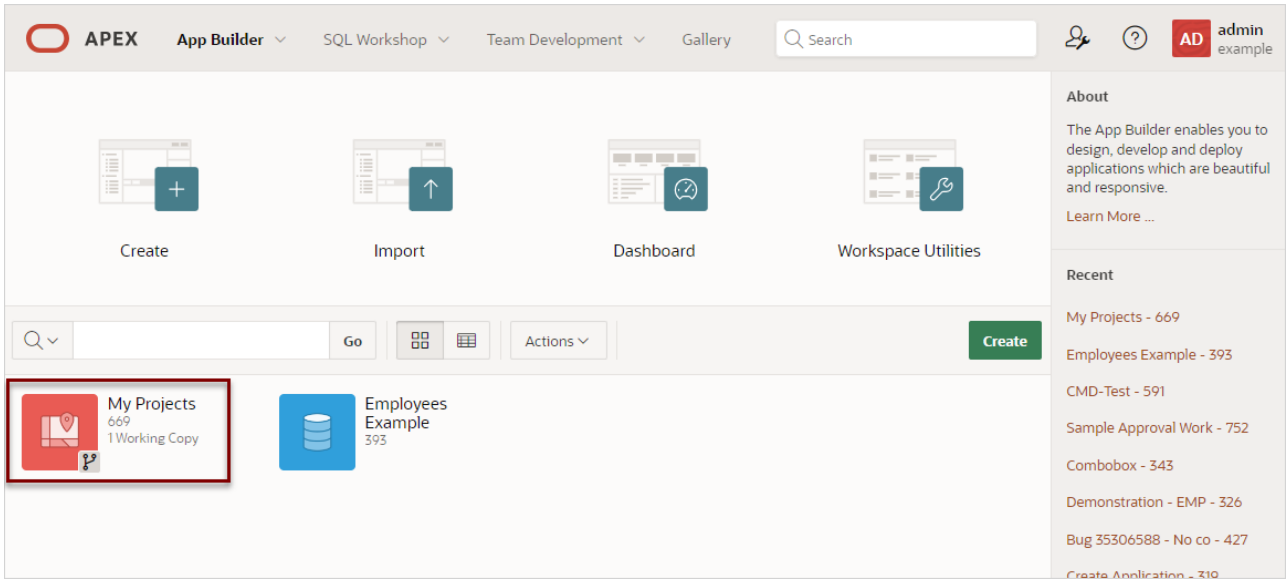
To view or edit a Working Copy:

1. View the Main application:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.

      If an application has a Working Copy, the number of Working Copies display on Application home page. In report view, a Working Copies column appears displaying the number of Working Copies. In Icon view, the number of Working Copies appears under application name to the right of the icon.

   b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.



2. Select the application with the Working Copy you wish to view.

   The Working Copy Application home page appears.

3. Under **Working Copies** on the side bar, select a copy to view.

   The Application home page appears. Note that the Working Copy ***Name*** displays next to the application name.

4. Make changes to the Working Copy.

> **Tip:**
>
> To learn more about all the options available on the Working Copy, see About Working Copy Menus and Tasks

> **See Also:**
>
> - Merging Changes from a Working Copy into Main
> - Refreshing a Working Copy with Changes from Main

5.9.5 Comparing Changes between a Working Copy and Main

Compare changes between a Working Copy with the Main application.

Comparing changes in a Working Copy with the Main application displays a dialog that summarizes what has been changed, added, or is missing.

> **Note:**
>
> To actually merge or refresh the changes, see Merging Changes from a Working Copy into Main and Refreshing a Working Copy with Changes from Main.

To compare changes between a Working Copy and the Main Application:

1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

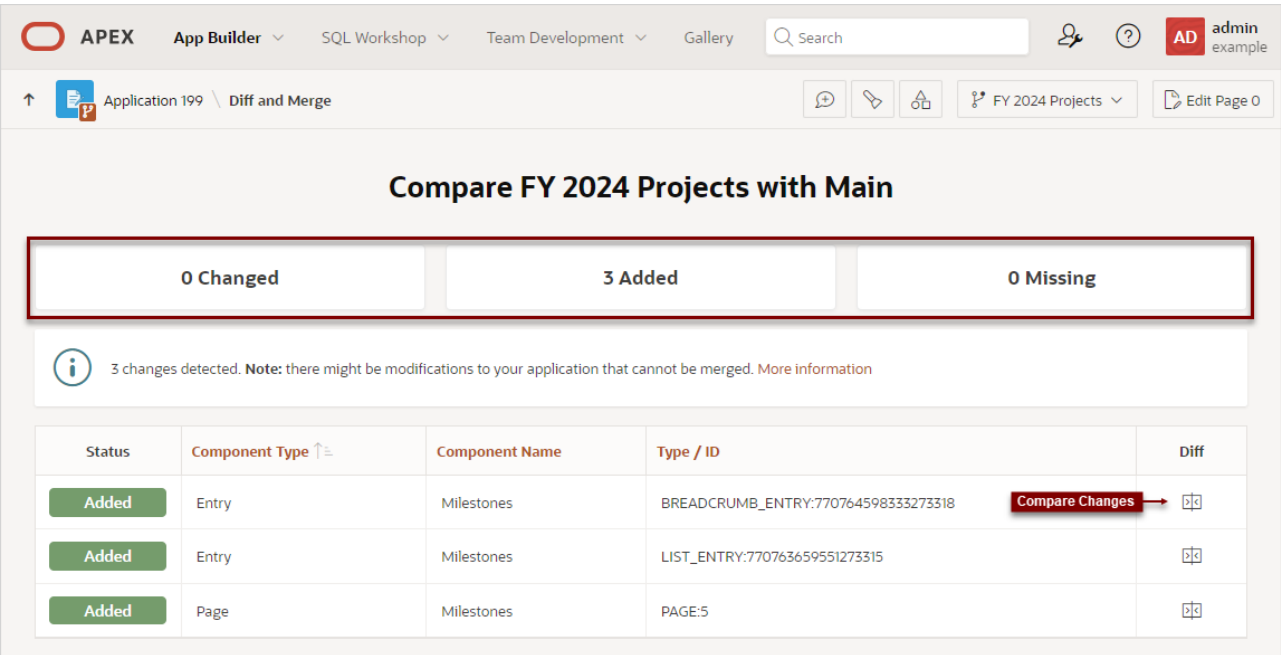
      The App Builder home page appears.
   b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.
   c. Select the Main application.

      The Application home page appears.
   d. Under **Working Copies** on the side bar, select a copy to view.

      The Working Copy appears. Note that the Working Copy ***Name*** displays next to the application name.
2. Click the **Working Copy: *Name*** menu and select **Compare Working Copy with Main**.

> **Note:**
>
> You can also select **Compare Changes** from the side bar. Alternatively, if viewing a page in Page Designer, click the **Working Copy:** *Name* menu and select **Compare Working Copy with Main**.

App Builder exports the Main app and Working copy and compares the two versions. The **Compare *Project Name* with Main** dialog appears summarizing what has been changed, added, or is missing.



3. To view a Diff of the changes, click the **Compare Changes** icon.

   The View Differences window appears. Click **Close** to exit the window.

5.9.6 Comparing Changes Across Working Copies

View and compare changes between single pages across different Working Copies.

When a page has been modified in a Working Copy or in the Main application, the **View changes in other copies** button ( ) appears on the Page Designer toolbar.

Clicking **View changes in other copies** displays a dialog that contains a report of copies of the application where the current page has been modified or locked.

> **Tip:**
>
> The **View changes in other copies** button olny appears on the Page Designer toolbar if a page must have been modified in another Working Copy or in the Main application.

To compare changes across Working Copies:

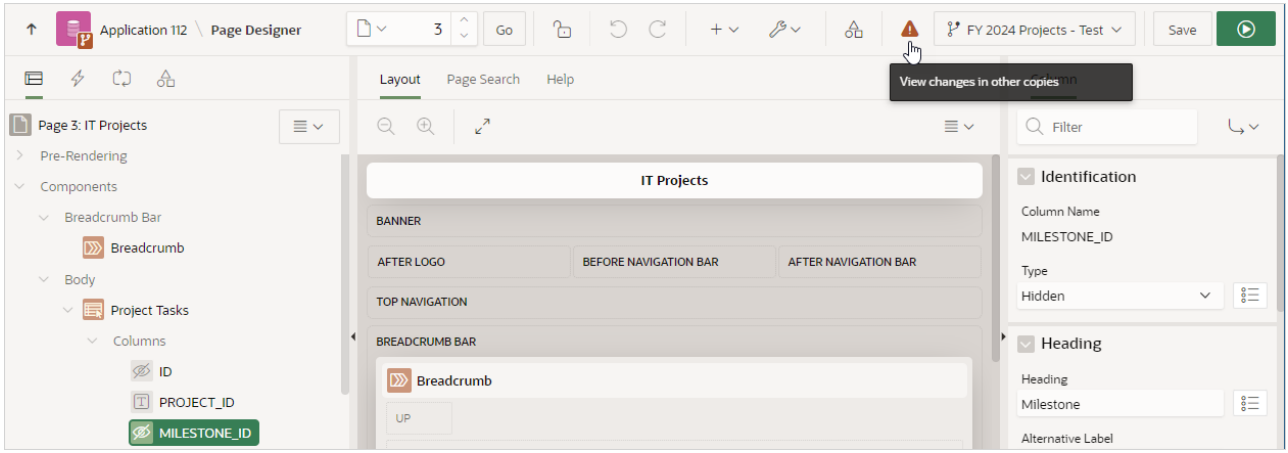
1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.
   b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.
   c. Select the Main application.

      The Application home page appears.
   d. Under **Working Copies** on the side bar, select a copy to view.

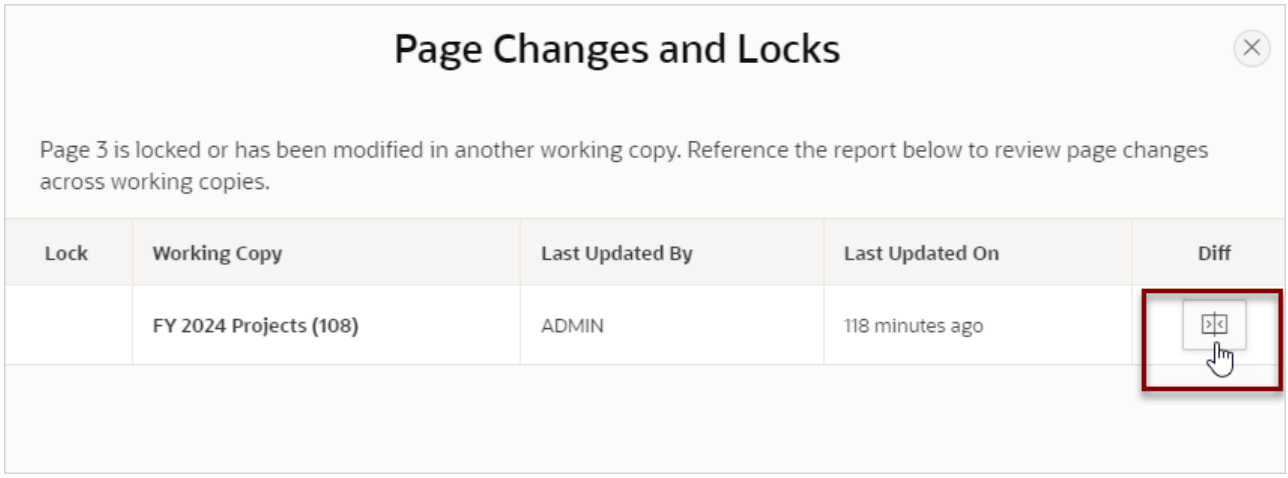
      The Working Copy appears.
2. On the Application home page, select a page that has been modified in another Working Copy or in the Main application.

   Page Designer appears.



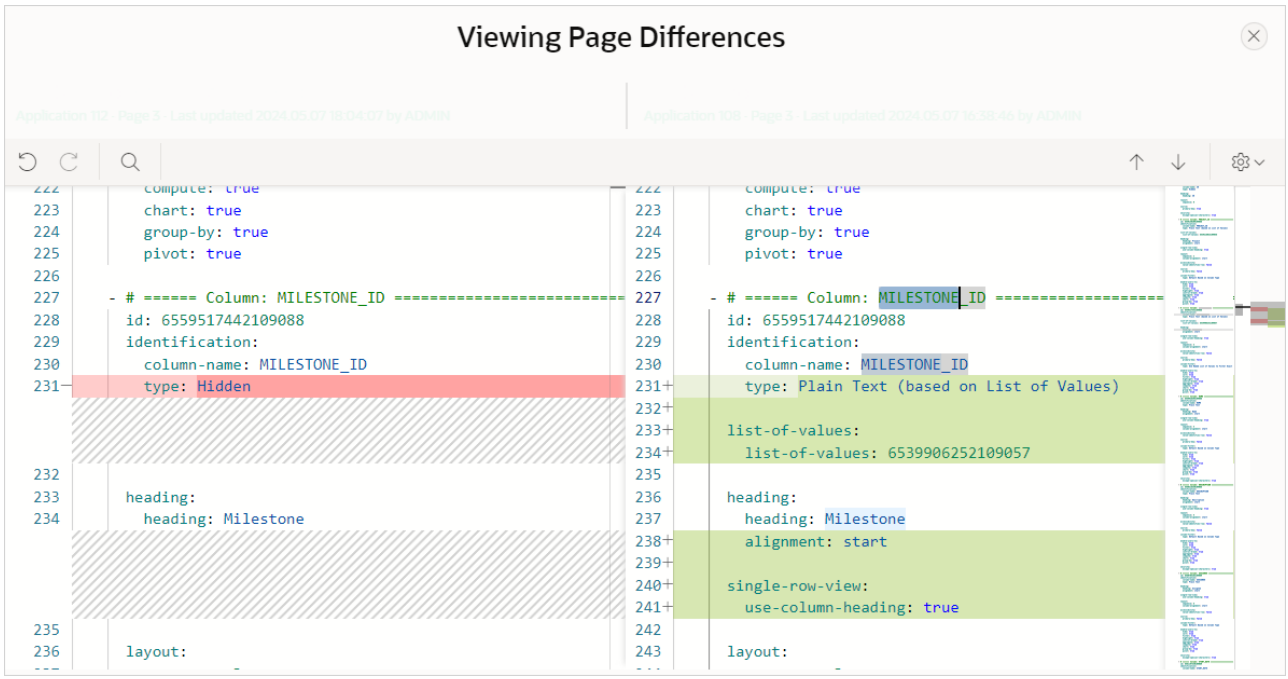
3. On the Page Designer toolbar, click **View changes in other copies**.

   The Page Changes and Locks dialog appears displaying a report of all copies where the current page has been modified or locked, including the Main application.



4. To view a diff of the changes, select the **View Diff** button.

   The Viewing Page Differences dialog appears. Available controls include: **Undo**, **Redo**, **Search**, **Previous Change**, **Next Change**, and the **Settings** menu.



   The previous example shows two Working Copies. In the first Working Copy the `MILESTONE_ID` column is **Hidden**. In the second the Working Copy, the `MILESTONE_ID` column is **Plain Text (based on List of Values)**.

5. Click **Close** to exit the Viewing Page Differences dialog.

5.9.7 Merging Changes from a Working Copy into Main

Merge changes from a Working Copy into the Main application.

To merge changes from a Working Copy into the Main application:

1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

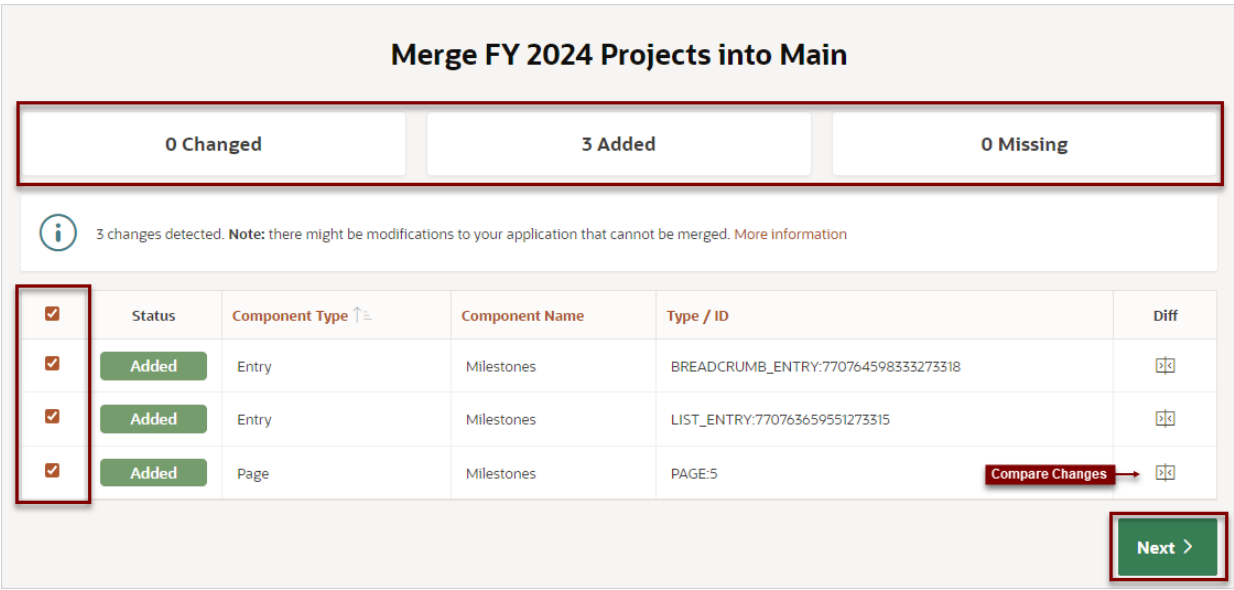
      The App Builder home page appears.
   b. Select the Main Application.

      The Application home page appears.
   c. Under **Working Copies** on the side bar, select a copy to view.

      The Working Copy appears. Note that the Working Copy ***Name*** displays next to the application name.
2. Make changes to the Working Copy.
3. To merge changes into the Main application:
   a. Click the **Working Copy: *Name*** menu and select **Merge into Main**.

      > **Tip:**
      >
      > You can also select **Merge into Main** from the side bar. Alternatively, if viewing a page in Page Designer, click the **Working Copy:*Name*** menu and select **Merge into Main**.

      App Builder exports the Main application and Working Copy and compares the two versions. The **Merge *Project Name* into Main** dialog appears summarizing what has been changed, added, or is missing.

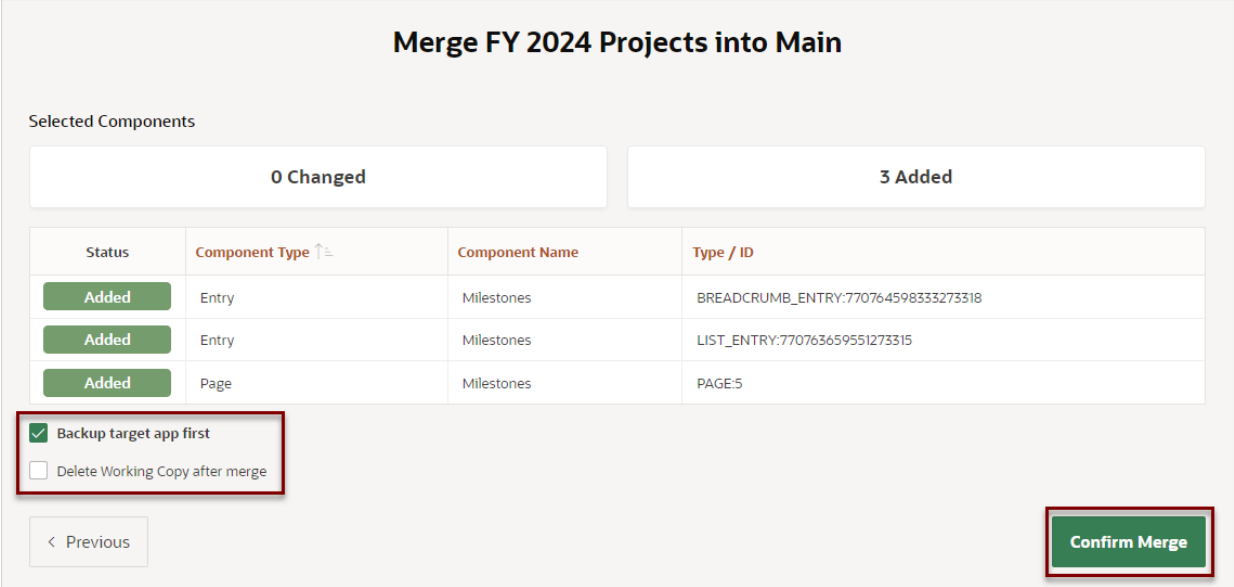


   b. To view a Diff of the changes, click the **Compare Changes** icon.

      The View Differences window appears. Click **Close** to exit the window.
   c. To exclude a change, deselect the row.
   d. To continue, click **Next**.

      The **Merge *Project Name* into Main** dialog now includes two new options appears:

- **Backup target app first**
- **Delete Working Copy after merge**



e. Configure **Backup target app first** and **Delete Working Copy after merge** as appropriate and click **Confirm Merge**.

   The Application home page appears and displays the message:

   `Your changes have been successfully merged!`

5.9.8 Refreshing a Working Copy with Changes from Main

Refresh a Working Copy with changes in the Main application.

Refreshing a Working Copy exports the changes from the Main application into the Working Copy.

To refresh a Working Copy:

1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.

   b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.
   c. Select the Main application.

      The Application home page appears.

   d. Under **Working Copies** on the side bar, select a copy to view.

      The Working Copy appears. Note that the Working Copy ***Name*** displays next to the application name.

2. To refresh a Working Copy with changes from the Main application:
   a. Click the **Working Copy: *Name*** menu on the menu bar and select **Refresh Working Copy**.

> **Tip:**
>
> You can also select **Refresh Working Copy** from the side bar. Alternatively, if viewing a page in Page Designer, click the **Working Copy:*Name*** menu and select **Refresh Working Copy**.

App Builder exports changes from the Main application to the Working Copy and compares the two versions. The **Refresh *Name* from Main** dialog appears summarizing what has been changed, added, or is missing.

b. To view a Diff of the changes, click the **Compare Changes** icon.

   The View Differences window appears. Click **Close** to exit the window.

c. To exclude a change, deselect the row.

d. To continue, click **Next**.

   The following new option appears:

   - **Backup target app first**

e. Configure **Backup target app first** as appropriate and click **Confirm Refresh**.

   The Application home page appears and displays the message:

   ```
   Your changes have been successfully refreshed!
   ```

5.9.9 Deleting a Working Copy

Delete a Working Copy copy.

To delete a Working Copy:

1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.

   b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.

   c. Select the Main Application.

      The Application home page appears.

   d. Under **Working Copies** on the side bar, select a copy to view.

      The Working Copy appears. Note that the Working Copy ***Name*** displays next to the application name.
2. Click the **Working Copy:** *Name* menu on the menu bar and select **Delete Working Copy**.
3. On the Delete Working Copy dialog, click **Delete Working Copy**.

5.9.10 View or Edit Working Copy Details

View or edit the Working Copy name or description on Working Copy Details.

To view or edit Working Copy details:

1. View the Working Copy:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.

b. If not already selected, change to Icon view. On the navigation bar, click **View Icons**.

c. Select the Main Application.

   The Application home page appears.

d. Under **Working Copies** on the side bar, select a copy to view.

   The Working Copy Details dialog appears.

2. Click the **Working Copy:** *Name* menu on the menu bar and select **View Working Copy Details**.

3. On Working Copy Details:

   a. Working Copy Name - Enter a short name or identifier for the working copy application.

   b. Working Copy Description - Enter a description or purpose for the working copy application.

   c. Click **Apply Changes** to save your changes.

5.10 Creating Applications for Mobile Devices

Build applications for mobile devices with mobile-specific design patterns that can provide an optimal user experience for small screens.

- Advantages of the Universal Theme
  Universal Theme - 42 (Universal Theme) features responsive design, a versatile user interface, and easy customization and enables developers to build modern web applications without requiring extensive knowledge of HTML, CSS, or JavaScript.
- Creating a Progressive Web App (PWA)
  Learn about Progressive Web App (PWA) features and requirements.

5.10.1 Advantages of the Universal Theme

Universal Theme - 42 (Universal Theme) features responsive design, a versatile user interface, and easy customization and enables developers to build modern web applications without requiring extensive knowledge of HTML, CSS, or JavaScript.

Key advantages of the Universal Theme include:

- **Responsive Design** - Designed to work just as well on small screen devices (such as smartphones and tablets) as it does on larger screen devices (including laptops and desktops) by responding automatically to the dimensions of the screen on which it is running. The UI components in Universal Theme work across varying screen resolutions while maintaining the same or similar functionality. In addition, Universal Theme takes full advantage of ultra high screen resolutions by utilizing vector graphics where possible, and relying upon CSS3 features for UI styling.
- **Versatile User Interface** - Provides all the components and building blocks necessary to build practically any type of business application user interface.

> **Tip:**
>
> To browse all of the components provided with Universal Theme, run the Universal Theme Reference app. From the **Main Navigation** menu, select **Components**.

- **Easy Customization** - Effortlessly customize and fully control the look and feel of your applications without becoming an expert in UI design, HTML, CSS, or JavaScript. Using Theme Roller and Template Options, you can easily customize your application to fit your company's brand and customize the look and feel of various components using Template Options.

About Migrating Existing jQuery Mobile Apps to the Universal Theme

jQuery Mobile and the jQuery Mobile User Interface used in previous releases has been desupported. If you have an existing mobile application that uses the jQuery Mobile User Interface, you must migrate your existing application to the Universal Theme.

> **Tip:**
>
> To learn more about migrating existing applications to the Universal Theme, run the Universal Theme Reference app. From the **Main Navigation** menu, select **Reference** and then **Migration Guide**.

> **See Also:**
>
> - Accessing the Universal Theme Reference App
> - About Switching the Active Theme
> - Understanding Template Options
> - Using Theme Styles

5.10.2 Creating a Progressive Web App (PWA)

Learn about Progressive Web App (PWA) features and requirements.

- About Progressive Web App Support
  Progressive Web Apps (PWAs) leverage modern web APIs giving users an experience that looks and feels like a native app.
- Creating a Progressive Web App
  Create a Progressive Web App by running the Create Application Wizard and enabling the **Install Progressive Web App** feature.
- Configuring Attributes for a Progressive Web App
  Configure Progressive Web App attributes for both new and existing Progressive Web App applications.

5.10.2.1 About Progressive Web App Support

Progressive Web Apps (PWAs) leverage modern web APIs giving users an experience that looks and feels like a native app.

Developers can create a PWA when creating a new application with the Create Application Wizard, or by editing attributes of an existing application.

Progressive Web App Requirements

Requirements for using this feature include:

- The Oracle APEX application must be served over a secured HTTPS environment, or localhost. If using an unsecured environment, PWA features will not be rendered
- The application Definition, **Properties**, **Friendly URLs** attribute must be **On**. See Accessing the Application Definition Page.

About Progressive Web Apps

Enabling PWA functionality:

- Adds the ability to install the app as a Progressive Web App (PWA).
- Adds a new navigation bar entry named `Install App` to your application.
- Optimizes page loading speed on the mobile device. Improves page load rendering time with a new browser cache architecture for static files (APEX core files and app files).
- Enables users to install the application on devices.
- Provides a customizable offline page when users are offline and cannot request the network. See Progressive Web App (PWA) Messages Requiring Translation.
- Supports hooks into the service worker code for a fully customizable PWA experience.

Creating a New PWA

For a new application, you can create a PWA by enabling the **Install Progressive Web App** feature in the Create Application Wizard and then configuring the Progressive Web App attributes.

Making an Existing App a PWA

To make an existing application a PWA, you must:

- Verify **Friendly URLs** attribute is **On** in the Application Definition, **Properties** section.
- Turn on the **Enable Progressive Web App** attribute and then configure the remaining Progressive Web App attributes.

See Also:

- Accessing the Application Definition Page
- Configuring Attributes for a Progressive Web App

5.10.2.2 Creating a Progressive Web App

Create a Progressive Web App by running the Create Application Wizard and enabling the **Install Progressive Web App** feature.

> **Tip:**
>
> To include mobile friendly reports and forms, create your application and then add pages to it. See Mobile Friendly Legacy Forms.

To create a new application with Progressive Web App functionality enabled:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.

   > **Tip:**
   >
   > To learn more about all the options available, see Choosing an Application Creation Method.

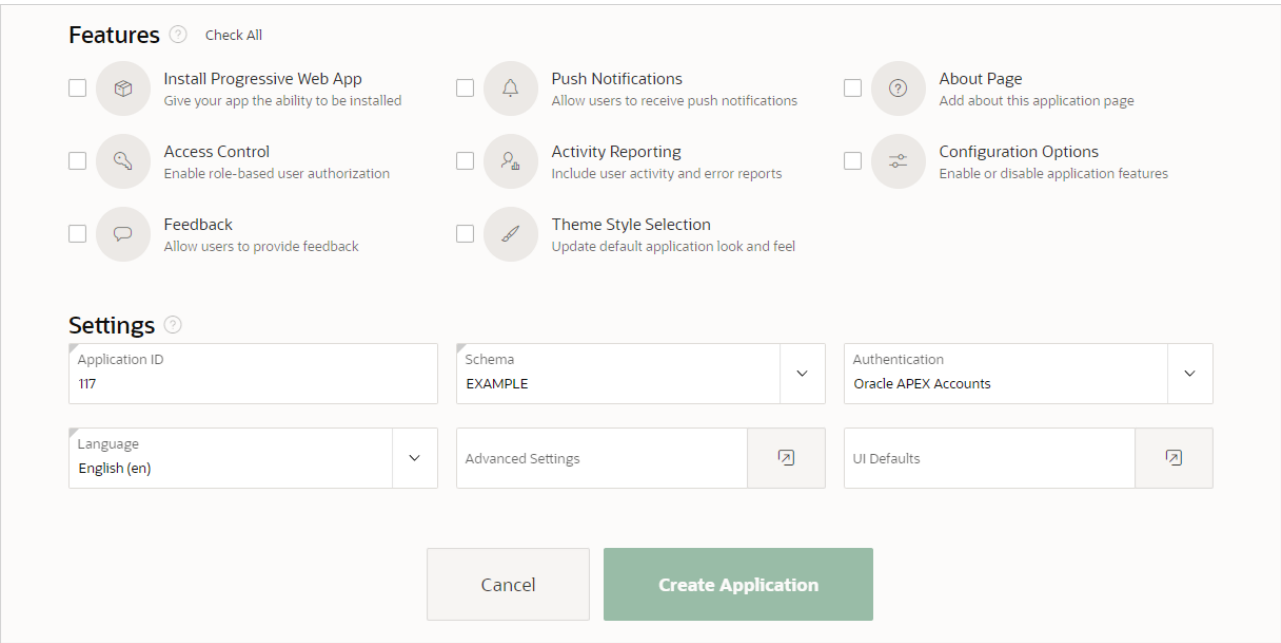
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. For Name, enter the name used to identify the application to developers.
5. To add a page, click **Add Page** and select the desired page type. The user interface changes based on the selected page type.

   > **Tip:**
   >
   > The Create Page Wizard supports the creation of a number of mobile friendly reports and forms. See Mobile Friendly Legacy Forms.

   Pages you create display under **Pages**.
6. For Features, select features to include with your application.

   Features provide application-level functionality and can only be added once per application. To learn more, see Help. To select all features, click **Check All**.



Features specific to a Progressive Web App include:

- **Install Progressive Web App** - Progressive Web Applications (PWA) are designed to be capable, reliable, and installable. A Progressive Web App feels like a platform-specific application. Enabling this option:
  - Adds a new navigation bar entry, `Install App`.
  - Enables users to install the app on devices.
- **Push Notifications** - Enable web apps to send notifications to a user's device even when the user is not currently on the website. Notifications can be sent to a user's desktop or mobile device and can be used to alert the user to new content or events on the website, or to re-engage the user with the website. Enabling this option:
  - Adds a new navigation bar entry, `User Settings`.
  - Enables users to subscribe and unsubscribe to push notifications.
  - When subscribed, users can receive push notifications from the APEX app.

7. Click **Create Application**.
8. Configure Progressive Web App attributes. See Configuring Attributes for a Progressive Web App.

5.10.2.3 Configuring Attributes for a Progressive Web App

Configure Progressive Web App attributes for both new and existing Progressive Web App applications.

To configure Progressive Web App attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. From the Application home page, you can access the Progressive Web App page in two ways:

- **Edit Application Definition** button:
  - Click **Edit Application Definition** to the right of the application name.
  - Click the **Progressive Web App** tab.
- From Shared Components:
  a. Click **Shared Components**.
  b. Under User Interface, click **Progressive Web App**.

4. Under **General**, edit the following:
   a. Enable Progressive Web App - Confirm this option is enabled.
   b. Installable - Confirm this option is enabled.
5. Under **Installability**:
   a. Display - Select the preferred display mode for the application. This option controls how much of the browser UI the user sees. Options include:
      - **Fullscreen** - Opens the PWA without any browser UI. This option takes the entirety of the screen or window.
      - **Standalone**- Opens the PWA to look like a standalone application. The app runs in a distinct window, separate from the browser. Standard browser UI components like the URL bar and buttons are not visible.
      - **Minimal UI** - Similar to standalone, except with browser controlling navigation buttons like back and refresh.
      - **Browser** - Opens the PWA is a regular browser UI.

      Note: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately. For more information, see https://web.dev/add-manifest//.
   b. Screen Orientation - Select the preferred screen orientation when using this Progressive Web App. This option applies mostly for mobile devices.

      Note: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately.
   c. Theme Color - Defines the default theme color for the application.

      This attribute affects how the operating system displays the application (for example, for an Android's task switcher, the theme color surrounds the application).

      Note: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately.
   d. Background Color - Select the background color.

      The background color defines a placeholder background color for the application to display before its stylesheet is loaded. Therefore, the background color should match the theme body background color for a smooth transition between launching the web application and loading the application content.

      Note: Some devices and browsers ignore this attribute. APEX is only relaying this information to the device which handles the Progressive Web App appropriately.
   e. iOS Status Bar Style - The iOS Status Bar Style attribute is rendered as a meta tag on the page to control the appearance of the status bar in a web application when it is added to the home screen on an iOS device. Options include
      - **Default** - This is the default value and it displays the status bar in the usual way.

- **Black** - This value makes the status bar black.
- **Translucent** - This value makes the status bar translucent, enabling the content to be displayed behind it.

  Note: Only iOS devices use this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately.

f. App Description - Explains what the application does. Use this attribute to provide more information to users when they are prompted to install the application.

g. Custom Manifest - Used to store additional JSON properties for the Web App Manifest file. The properties found here will be used to extend the declarative options above, by overriding existing properties or appending new properties.

   For more information on building a Custom Manifest, you can read the Web App Manifest specifications: https://www.w3.org/TR/appmanifest/.

h. Screenshots - Include screenshots for promotional purposes when users are prompted to install the Progressive Web App.

   To add a screenshot:

   - Click **Add Screenshot**.
   - Description - Enter the PWA screenshot label which is used for accessibility purposes when installing the PWA.
   - Sequence - Identifies the display sequence used by this PWA screenshot.
   - Upload a Screenshot:
     - Drag and Drop - Drag and drop the file to the region.
     - Choose a File - Click the **Upload a Screenshot** region and then navigate to and select the file.
   - Build Option - Identifies the build option that this PWA screenshot uses.
   - Comments - Add comments associated with this PWA screenshot.
   - Click **Create**.

i. Shortcuts - Include shortcuts to enable users of installed PWAs to quickly access a specific page. For touch-enabled devices, shortcuts are accessible when doing a long-press on the application icon on the home screen. For other devices, shortcuts are accessible when doing a right click on the application icon on the task bar.

   To add a shortcut:

   - Click **Add Shortcut**.
   - Name - Enter the name of the PWA shortcut.
   - Target URL - Specify the page of the application for the current shortcut.
   - Sequence - Identifies the display sequence used by this PWA screenshot.
   - Upload an icon:
     - Drag and Drop - Drag and drop the file to the region.
     - Choose a File - Click the **Upload an Icon** region and then navigate to and select the file.
   - Description - Enter a description for this PWA shortcut.
   - Build Option - Identifies the build option that this PWA shortcut uses.
   - Comments - Add comments associated with this PWA shortcut.

- Click **Create**.

> **Tip:**
>
> An APEX application can only use PWA Shortcuts if Rejoin Sessions is enabled. Before configuring shortcuts, click either **Enable for Public Sessions** or **Enable for All Sessions**. To learn more, see Session Management.

6. Under **Push Notifications**:

   a. Enable Push Notifications - Enable this option to allow web apps to send notifications to a user's device even when the user is not currently on the website. Notifications can be sent to a user's desktop or mobile device and can be used to alert the user to new content or events on the website, or to re-engage the user with the website.

      Enabling this option:

      - Adds a new navigation bar entry: **User Settings**
      - Enables users to subscribe and unsubscribe to push notifications.
      - When subscribed, users can receive push notifications from the APEX app.

   b. If **Enable Push Notifications** is enabled, the following attributes appear:

      - Credentials - Select the credential to authenticate against. To edit the credential, navigate to Shared Components, Credentials.
        Credentials are required to setup push notification securely. A public and private key pair are used respectively to allow users to subscribe to push notifications on the client side, and for the server to send the notifications to the subscribers.
      - Settings Page - If this application already has the push notification feature page, the page number displays here. If the application does not have the push notification feature page, it will show that the feature page is available to add.
      - Contact Email - Enter the email address to contact in case the push notification service provider (Google, Mozilla, Apple, etc.) wants to contact the owner of the application. We recommend using a group email so it's not dependent on a single individual.

7. Under **Service Worker Configuration**, select a Service Worker.

   Service workers are JavaScript files that can execute code even when the app is not in use. Service workers can listen to events like fetching resources or handling notifications. You can modify a service worker through hooks, or by replacing events completely. Service Worker options include:

   - **Default** - The APEX engine generates it with the following strategy:
     - Install and activate the service worker
     - Serve resources from cache if cache exists
     - Otherwise serve from network, then put resource in cache
     - Serve an offline page if network fails
   - **Configure Hooks** - Fill in the placeholders in the service worker code to add your own. Hooks locations are provided by Oracle APEX. When using this option, you can choose to "Download Configuration" and use it as a File URL.
   - **File URL** - Provide a URL reference to a file that contains the service worker hooks interface. This is useful for sharing service worker hooks across multiple applications.

If using the file reference, make sure to follow the interface architecture appropriately or else the service worker will fail at runtime.

To view the interface structure, see field-level Help for this attribute.

8. Click **Apply Changes**.

5.11 Deleting an Application

You can delete an application from the Application home page, or while editing application attributes.

When you delete an application, you are prompted whether to remove the application definition and/or drop supporting objects.

- Deleting an Application from the Application Home Page
  Delete an application by select Delete this Application from the Tasks list on Application home page.
- Deleting an Application from the Edit Application Definition
  Delete an application by clicking Delete on Edit Application Definition page.

> **See Also:**
>
> Deinstalling Supporting Objects

5.11.1 Deleting an Application from the Application Home Page

Delete an application by select Delete this Application from the Tasks list on Application home page.

To delete an application from the Application home page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. On the Application home appears, verify the application ID and name at the top of the page.
4. From the Tasks list, click **Delete this Application**.
5. When prompted, click **Permanently Delete Now**.

5.11.2 Deleting an Application from the Edit Application Definition

Delete an application by clicking Delete on Edit Application Definition page.

To delete an application from the Edit Application Definition page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Edit Application Definition** button to the right of the application name.

   The Edit Application Definition page appears.

4. Verify the application ID and name.
5. Click **Delete** at the top of the page.
6. When prompted, click **Permanently Delete Now**.

> **See Also:**
>
> Editing the Application Definition

5.12 Managing Application Attributes

Application attributes control the behavior of an entire application and are divided into the categories: Definition, Security, Globalization, User Interface, Progressive Web App, and AI.

Once you create an application, you can review and update all application attributes.

> **Tip:**
>
> How to Create a Custom Application for information on using the Supporting Objects utility to create a custom application.

- Editing the Application Definition
  Use the Application Definition page to change general application attributes such as the application name, application alias, version number and control various properties and behaviors such as friendly URLs, logging, debugging, feedback, compatibility mode, application availability, error handling, and substitution strings.
- Configuring Security Attributes
  Use the Security page to set application-wide security settings.
- Configuring Globalization Attributes
  Edit attributes on the Globalization page to configure application globalization options. In App Builder you can develop applications that can run concurrently in different languages.
- Editing User Interface Attributes
  Use the User Interface page to control the default characteristics of the application and optimize the display.
- Configuring Progressive Web App Attributes
  Use the Progressive Web App page to configure attributes for both new and existing Progressive Web App applications.
- Configuring Generative AI Attributes
  Configure the attributes on the AI page to make Generative AI capabilities available to runtime applications.

5.12.1 Editing the Application Definition

Use the Application Definition page to change general application attributes such as the application name, application alias, version number and control various properties and behaviors such as friendly URLs, logging, debugging, feedback, compatibility mode, application availability, error handling, and substitution strings.

- Accessing the Application Definition Page
  Access the Application Definition page from the Application home page.
- Application Definition Page
  The Application Definition page is divided into the following regions: Name, Properties, Availability, Error Handling, Gloal Notification, Substitutions, Build Options, and Copyright Banner.

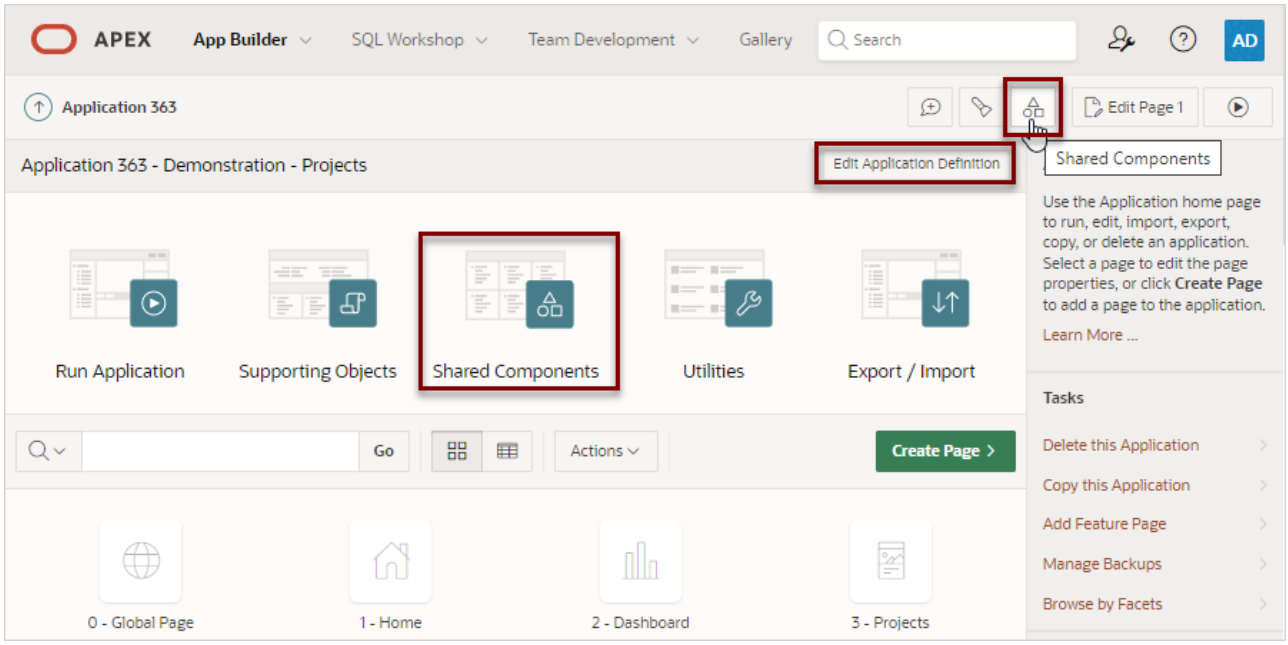
5.12.1.1 Accessing the Application Definition Page

Access the Application Definition page from the Application home page.

To access the Application Definition page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.



3. From Application home page, you can access the Application Definition page in two ways:
   - Click the **Edit Application Definition** button.
   - From Shared Components:
     - Click **Shared Components**.
     - Under Application Logic, click **Application Definition**.

   The Edit Application Definition page appears.
4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

5.12.1.2 Application Definition Page

The Application Definition page is divided into the following regions: Name, Properties, Availability, Error Handling, Gloal Notification, Substitutions, Build Options, and Copyright Banner.

> **Note:**
>
> Required values are marked with a red asterisk (*).

- Name
- Properties
- Availability
- Error Handling
- Global Notification
- Substitutions
- Build Options
- Workflow Settings
- Report Printing
- Application File Storage
- Copyright Banner

5.12.1.2.1 Name

Use **Name** attributes to define basic characteristics of your application, including the application name, an optional alphanumeric alias, and a version number.

Attribute	Description	To Learn More
Name	A short descriptive name for the application to distinguish it from other applications in your development environment.	n/a
Application Alias	Assigns an alternate alphanumeric application identifier. You can use this identifier for the application ID. When the **Friendly URLs** attribute is enabled, the **Application Alias** specified here displays in the URL when running the application. Applications created using Oracle APEX 20.1 or later default the **Application Alias** to the application **Name**.	See: • Understanding Friendly URL Syntax • Customizing a Friendly URL
Version	Enter the application's version number. The value enter displays on the page. You can also automatically tie the version to the date of last modification using the following format masks: • `YYYY.MM.DD` • `MM.DD.YYYY` • `DD.MM.YYYY` If your application version uses YYYY.MM.DD, then App Builder replaces this format mask with the date of last modification of any application attribute.	n/a

Attribute	Description	To Learn More
Application Group	Displays the application group currently associated with this application. To select another application group, make a selection from the list. To remove an application from an existing group, select **Unassigned**.	See Managing Application Groups

5.12.1.2.2 Properties

Use Properties attributes to configure logging, debug behavior, feedback, compatibility, email from address, and the application proxy server.

Attribute	Description	To Learn More
Friendly URLs	Determines whether URLs generated to navigate to other pages in the application should be easier to read. Options include: • **On** - If set to **On**, the application uses Friendly URLs Syntax in which the URL consists of a directory structure based on the Application Alias, Page Alias, and standard web parameter syntax.. If set to **On**, Progressive Web App attributes display on this page. • **Off** - If set to **Off**, the application uses f?p= Syntax. **Note**: The friendly URL will include the workspace **Path Prefix** in the URL. To modify the **Path Prefix**, go to **Workspace Administration**, **Manage Service**, **Set Workspace Preferences**.	See: • Understanding Friendly URL Syntax • Customizing a Friendly URL • Enabling RESTful Services and Defining a Path Prefix for a Workspace in *Oracle APEX Administration Guide*
Allow Feedback	Enables support for end user feedback for this application. Select **Yes** or **No**. If you select **Yes**, you must create a feedback feature page to use this attribute. If you enable this option, a Feedback icon displays. Uses can click the Feedback icon to leave feedback. Disabling this option hides the Feedback icon. This enables you to turn on feedback for testing and turn it off for production	See Managing Feedback
Logging	Determines whether user activity is recorded in the Oracle APEX activity log. Select **Yes** to log every page view and enable an administrator to monitor user activity for each application. Disabling logging may be advisable for high volume applications. This attribute can only be modified if the Application Activity Logging attribute in APEX Administration Services is set to **Use Application Setting**.	See Enabling Application Activity Logging in *Oracle APEX Administration Guide*

Attribute	Description	To Learn More
Debugging	Controls debug mode for the current application. Available options include: • **Yes** - Enables the application to run in a debug mode from a browser at runtime. • **No** - Disables the application from enabling in debug from a browser. Running an application in debug mode is useful when an application is under development. For production applications, Oracle recommends disabling debugging and thus preventing users from viewing application logic. Debug can be enabled programmatically regardless of this debug setting. If the application is run from the APEX development environment, debugging can always be enabled.	n/a
Compatibility Mode	Controls the compatibility mode of the APEX runtime engine. Certain runtime behaviors change from release to release. To not break or change the behavior of existing applications, this attribute is automatically set during upgrade if necessary. Use this attribute to obtain specific application behavior. To keep applications maintainable, set **Compatibility Mode** to the latest release after reviewing the changed behavior in the Release Notes and making the necessary changes. **Note**: In order to refresh any theme in the current release, you must set **Compatibility Mode** to **21.2 to 24.1**.	See: • For more information see **Changed Behavior**, Compatibility Mode in *Oracle APEX Release Notes*. • Updating App Compatibility Mode
Accessible Read-Only Items	Determines whether APEX renders the legacy SPAN-based elements for an item's read-only mode, or the more accessible readonly input or textarea elements. The more accessible input and textarea elements are focusable with the keyboard, and easier to perceive for screen reader users.	N/A

Attribute	Description	To Learn More

Application Email from Address	Determines the email address to use as the from address in the application.
	Enter a valid email address to use as the from address when sending email from an email download or subscription. The value can be a literal string containing a valid email or a static substitution reference defined in the application using substitution syntax `APP_EMAIL`. Examples:
	`john.doe@abc.com`
	`&MY_APP_EMAIL_FROM.`
	
	Oracle does not recommend using an item substitution at the application or page-level since it only works in email download, but not for subscriptions.
	Tip: You can also specify the Email from Address by editing interactive report attributes. See About Emailing from an Interactive Report.
		n/a

Proxy Server	Specify a proxy server. For example, App Builder may require a proxy server when using a region source type of URL. The URL region source embeds the results of the URL (that is, the page returned by navigating to the URL) as the region source. If you use a firewall and the target of a URL is outside the firewall relative to App Builder, you may need to specify a proxy server. You can reference values entered into this field from PL/SQL using the PL/SQL package variable `APEX_APPLICATION.G_PROXY_SERVER`. For example: `www-proxy.company.com`	n/a

Attribute	Description	To Learn More
Text Query Function	Select the Text Query Function type. The Oracle Text query syntax is very complex and powerful and provides a rich set of query operators to deal with wildcards, stemming, fuzzy searches, and so forth. For end users, this syntax is often hard to understand. You can select a function to convert simple end user input to an Oracle Text query with the advanced search features automatically applied.Text Query Function options include: • **None** - No query conversion function is used. The end user's input is passed directly to the Oracle Text search. • **Search Engine** - This function performs a query relaxation search to retrieve relevant results beyond just exact matches. The goal is to get as many related results as possible, allowing users to easily find related information without needing to input precise queries. The search process begins with an exact match search and then progressively relaxes the query using techniques like stemming and fuzzy matching. While maximizing the number of results, the search engine function ensures that the most exact and relevant matches are always ranked at the top of the search results. • **Expert Search** - This function supports more advanced search operators for precise queries. It allows exact matching, Boolean operators (AND, OR, NOT), precedence with parentheses and proximity search. Additional parameters allow customizing search behavior. This is best when requiring highly accurate results from very specific queries. • **Custom** - Provide a custom function. In **Custom Function Name**, you can specify the name of a PL/SQL function which converts end user input to an Oracle Text query. The function takes the end user input as `VARCHAR2` and returns the Oracle Text query as a result **Example - Query Relaxation** This example makes use of Oracle Text Query Relaxation and is transparent to the end user. This example first executes the exact query and then it performs increasing	See: • If **Text Query Function** is **Expert Search**, see APEX_SEARCH, `QUERY_EXPERT_SEARCH` in *Oracle APEX API Reference*. • If **Text Query Function** is **Custom**, see Understanding Oracle Text Application Development in *Oracle Text Application Developer's Guide*

Attribute	Description	To Learn More

	levels of fuzzy search, until the requested amount of rows is returned.
	
	`create or replace function convert_end_user_search ( p_search in varchar2 ) return varchar2 is c_xml constant varchar2(32767) :=`
	
	`'<query><textquery><progres sion>' || '<seq> #SEARCH# </seq>' || '<seq> FUZZY(#SEARCH#,80, 100,W) </seq>' || '<seq> FUZZY(#SEARCH#,50,1000,W) </seq>' || '<seq> FUZZY(#SEARCH#,20,3000,W) </seq>' || '</progression></ textquery></query>';`
	
	`l_search varchar2(32767) := p_search; begin -- -- remove special characters; -- these are irrelevant for full text search -- l_search := regexp_replace( srcstr => l_search, pattern => '[<>{}/ ()*%&!$?.:,;\+#]' );`
	
	`return replace( srcstr => c_xml, oldsub => '#SEARCH#', newsub => '{' || l_search || '}' ); end;`	

Attribute	Description	To Learn More
Tokenize Row Search	Specify how the search terms are applied when a row search is performed within a component. Selecting **On** treats each word of a search term separately and matches a record where the words are contained in any of the columns, together or independently. To search for an exact phrase, use a quotes (") around the search terms. For example, `"ui developer"`. To escape a double quote in a search term, use a double quote (""). Examples: • Search for any record including the words "red" and "shoes" Search term: `red shoes` • Search for any record including the word "developer" and the exact expression "ui designer" Search term: `developer "ui designer"` • Search for any record including the word "author" and the exact expression "the sky is "blue" and nice" Search term: `author "the sky is ""blue"" and nice"` Selecting **Off** treats the whole search term as an exact match.	n/a
Maximum Background Page Process Jobs	Provide the maximum number of scheduler jobs being available for background execution of page processes for this application. Only the configured number of jobs will run at the same time, to execute page processes for this application in the background. If the configured maximum amount of scheduler jobs is active, new submitted executions will wait until a job becomes available. Setting the attribute to zero disables background executions; new submitted executions will stay in enqueued state.	"Configuring Limits for Concurrently Running Scheduler Jobs" in About Background Page Processes

5.12.1.2.3 Availability

Use **Availability** attributes to manage your application by defining an application status and build status. For example, if you select the status **Restricted Access**, you can specify which users have access and can run the application.

Attribute	Description	To Learn More
Status	Specifies whether the application is available or unavailable for use. Options include: • **Available** - Application is available with no restrictions. • **Available with Developer Toolbar** - Application is available for use. For developers, the Developer Toolbar displays on each page. Requires the developer to be logged in to App Builder in the same browser session. • **Available to Developers Only** - Application is available to users having developer privileges. • **Restricted Access** - Application is available to developers named in the **Restrict to comma separated user list**. • **Unavailable** - Application cannot be run or edited. The message in **Message for unavailable application** displays when users attempt to access the application. • **Unavailable (Status Shown with PL/SQL)** - Application cannot be run or edited. • **Unavailable (Redirect to URL)** - Application cannot be run. The user is linked to the URL entered in **Message for unavailable application.**	See: • Changing Build Status for Multiple Applications and Changing Application Build Status in Administration Services in *Oracle APEX Administration Guide* • Controlling Access to Applications, Pages, and Page Components
Build Status	Identifies the build status of the current application. Options include: • **Run and Build Application** - Developers and users can both run and develop the application. • **Run Application Only** - Users can only run the application. This option is intended for applications in a production instance.	See Changing Application Build Status in Administration Services in *Oracle APEX Administration Guide*
Message for unavailable application	Use this attribute with **Status**. If you set Status to **Unavailable**, **Unavailable (Status Shown with PL/SQL)**, or **Unavailable (Redirect to URL)** , the text you enter in this attribute displays. If you set Status to **Available**, the text you enter in this attribute does not display.	n/a

Attribute	Description	To Learn More
Restrict to comma separated user list (status must equal Restricted Access)	Use this attribute with the Status **Restricted Access**. If you set Status to **Restricted Access**, only the users listed in this attribute can run the application. To use this attribute: 1. From the Status list, select a restricted status. 2. Enter a comma-delimited list of users who can run the application in the field provided. 3. Click **Apply Changes**.	n/a

5.12.1.2.4 Error Handling

Use the **Error Handling** attributes described to control or modify how an application logs errors.

> **Tip:**
>
> Error handling functions specified here are overridden by similar page-level attributes.

Attribute	Description	To Learn More
Default Error Display Location	Identifies where the validation error messages display for basic validations performed by Oracle APEX or by plug-ins. Validation error messages can display in a notification area (defined as part of the page template), or within the field label. Options include: • **Inline with Field and in Notification** - Error messages display in a notification area defined as part of the page template. • **Inline with Field** - Error messages display within the field label. • **Inline in Notification** - Displays in the `#NOTIFICATION_MESSAGE#` template substitution string when an error occurs on the page.	n/a

Attribute	Description	To Learn More

Error Handling Function	Enter the name of a PL/SQL error function to be called to modify the existing error message and display a more user-friendly message or log the error if one occurs. This function can reference a package function or standalone function in the database. For example:
	`log_apex_error`
	When referencing a database PL/SQL package or standalone function, use the `#OWNER#` substitution string to reference the parsing schema of the current application. For example:
	`#OWNER#.log_apex_error`
	You must implement error handling functions using the syntax described in the `apex_error` package.
	
	`function <name of function> (`
	`  p_error in`
	`apex_error.t_error )`
	`  return`
	`apex_error.t_error_result`
	
	Note: Error handling specified at the page-level overwrites any error handling function specified here.
		See APEX_ERROR in *Oracle APEX API Reference*

5.12.1.2.5 Global Notification

Use the **Global Notification** attribute to communicate system status to application users. If the page templates used in your application contain the `#GLOBAL_NOTIFICATION#` substitution string, the text entered here displays in that string's place. For example, you can use this attribute to notify users of scheduled downtime, or communicate other messages regarding application availability.

To create a global notification:

1. Include the `#GLOBAL_NOTIFICATION#` substitution string in your page template.
2. Navigate to the Edit Application Definition page and enter a message in the Global Notification attribute.
3. Click **Apply Changes**.

See Also:

- Using Substitution Strings
- Page Templates

5.12.1.2.6 Substitutions

Use **Substitutions** to define static substitution strings for your application. You can use static substitution string for phrases or labels that occur in many places within an application. To create a substitution string, enter the string name in the **Substitution String** column and the string value in the **Substitution Value** column.

Defining static substitution strings centrally enables you to change text strings in multiple places in your application by making a single change to the **Substitution Value** defined on this page.

> **See Also:**
>
> Using Substitution Strings

5.12.1.2.7 Build Options

Build Options displays existing build options in the current application. Most applications have a build option attribute. Build Options have two possible values: `INCLUDE` and `EXCLUDE`. If you specify an attribute to be included, then the Oracle APEX engine includes and enables it at runtime. However, if you specify an attribute to be excluded, then the APEX engine disables it and excludes it at runtime.

Do not specify a build option unless you plan to exclude that object from specific installations.

> **See Also:**
>
> Using Build Options to Control Configuration

5.12.1.2.8 Workflow Settings

Workflow Settings displays the **Task Vacation Rule Procedure**. If implemented, the vacation rule procedure applies to all task definitions in this application. To implement a vacation rule procedure only for a specific task definition, add the procedure on the task definition level instead of the application level.

> **Tip:**
>
> A vacation rule defined at the task definition level overrides a vacation rule defined at the application level.

For more details on implementing a vacation rule procedure, see Example: Adding a Vacation Rule Procedure.

> **See Also:**
>
> Creating a Procedure in *Oracle APEX SQL Workshop Guide*.

5.12.1.2.9 Report Printing

Report Printing controls the print server of the application. These settings override the settings from Instance Administration. Available options include:

- **Native Printing** - Use native printing. No external print server required.
- **Remote Print Server** - Choose a remote server of the type **Print Server** that is configured in the Workspace Utilities.
- **Use Instance Settings** - Use the Report Printing settings defined in Instance Administration.

Click the **Add Remove Print Server** button to open a wizard to create a Remote Print Server, including the credentials. To create a Remote Server, choose the Print Server Type, specify an Endpoint URL and click **Next**. If Authentication is required, specify the credentials. Click **Apply Changes** to add the Remote Print Server and Credentials to your workspace, and update the Report Printing attributes for the application. For detailed steps, see Configuring Report Printing at the Application Level.

> **See Also:**
>
> - Printing Report Regions
> - Configuring Report Printing in *Oracle APEX Administration Guide*

5.12.1.2.10 Application File Storage

Application File Storage determines how Static Application Files are stored.

- File Storage Type - Controls the file storage of the application. Options include:
  - **Database** - Use the database to store Static Application Files.
  - **Oracle Cloud Object Storage** - Use specified server to store files in Object Storage.
- **Add Object Storage** - Click **Add Object Storage** to access the Add Remote File Storage Wizard. Use this wizard to configure remote storage on Oracle Cloud Infrastructure (OCI) Object Storage service.
- **Remote File Server** - Only displays if **File Storage Type** is set to **Oracle Cloud Object Storage**. Select a defined Remote Server already defined in Workspace Utilities. The Remote Server must have the **Server Type**, **File Server**.

> **See Also:**
>
> - Using Workspace Utilities
> - Managing Remote Servers

5.12.1.2.11 Copyright Banner

Use **Copyright Banner** for copyright banner text. This text will be included in the application export file. You can use `#APP_NAME#` substitution string to refer to current application name and `#YEAR#` to refer to the current year, for example:

```
Name: #APP_NAME#

Copyright (c) 1999, #YEAR#, Company Name.
```

> **See Also:**
>
> Exporting Workspaces, Applications, and App Components

5.12.2 Configuring Security Attributes

Use the Security page to set application-wide security settings.

> **Tip:**
>
> Edit application components directly to manage more granular settings. To learn more about security best practices, see Managing Application Security.

- Accessing the Security Page
  Access the Security page page from the Application home page.
- Security Page
  The Security page is divided into the following sections: Authentication, Authorization, Session Management, Session State Protection, Browser Security, Database Session,and Advanced.

5.12.2.1 Accessing the Security Page

Access the Security page page from the Application home page.

To access the Security page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the Security page in two ways:
   - **Edit Application Definition** button:

a. Click **Edit Application Definition** to the right of the application name.
b. Click the **Security** tab.

- From Shared Components:
  a. Click **Shared Components**.
  b. Under Security, click **Security Attributes**.

The Edit Security Attributes page appears.

4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

5.12.2.2 Security Page

The Security page is divided into the following sections: Authentication, Authorization, Session Management, Session State Protection, Browser Security, Database Session,and Advanced.

Use the Security page to set application-wide security settings. Edit application components directly to manage more granular settings.

> **Note:**
>
> Required values are marked with a red asterisk (*).

- Authentication
- Authorization
- Session Management
- Session State Protection
- Browser Security
- Database Session
- Advanced

5.12.2.2.1 Authentication

Authentication is the process of establishing users' identities before they can access an application. Although you can define multiple authentication schemes for your application, only one scheme can be current at a time.

Attribute	Descriptions	To Learn More
Public User	Identifies the Oracle schema (or user) used to connect to the database through the Database Access Descriptor (DAD). Once a user has been identified, the Oracle APEX engine keeps track of each user by setting the value of the built-in substitution string `APP_USER`. When `APP_USER` equals this value, the APEX engine considers the current session to be a "public user" session. The APEX engine supports the following built-in display conditions: • `USER_IS_PUBLIC_USER` • `USER_IS_NOT_PUBLIC_USER` If the current application user (`APP_USER`) equals the value of this attribute, then the user is logged on as a public user. Some applications have public (not logged in) and private (logged in) modes. By determining if the user is the public user, you can conditionally display or hide information. For example, you can show a login button if the user is the public user and a logout link if the user is not a public user. Reference this value using `APEX_APPLICATION.G_PUBLIC_USER`.	See HOME_LINK and Understanding Conditional Rendering and Processing
Authentication Scheme	Identifies the current authentication method used by this application. The purpose of authentication is to determine the application users identity. To create an authentication scheme, click **Define Authentication Schemes**.	See How Authentication Works and Creating an Authentication Scheme
Configuration Procedure	Enter the name of a procedure which configures authentication at runtime.	See Using a Procedure to Configure Authentication at Runtime to view an example.

5.12.2.2.2 Authorization

Application authorization schemes control access to all pages within an application. Unauthorized access to the application, regardless of which page is requested, causes an error page to display.

Attribute	Descriptions
Define Authorization Schemes	Click the **Define Authorization Schemes**button to create an authorization scheme.
Authorization Scheme	Specify an authorization scheme for your application. Application authorization schemes are defined for an application for the purpose of controlling access. Setting a required authorization scheme here at the application level will require all pages of the application to pass the defined authorization check.

Attribute	Descriptions
Run on Public Pages	Controls whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). Options include: • **On** - If you select **On** and the page is public, the application authorization is checked. • **Off** - If you select **Off** and the page is public, the application authorization is ignored. Note: This attribute is ignored if you select **Must Not Be Public User** as Authorization Scheme.
Source for Role or Group Schemes	Define what APEX should use as the source for authorization schemes of type *Is In Role or Group*. The available values are: • **Access Control User Role Assignments** - Enable the user's assigned roles (under Shared Components / Application Access Control). • **Authentication Scheme** - If the application's authentication scheme is APEX Accounts, enable the workspace user's assigned groups. If the application's authentication scheme is Database Accounts, database roles are checked when the authorization scheme is enabled. If the authentication scheme uses Real Application Security, enable RAS Dynamic Roles. • **Custom Code** - Call the procedure apex_authentication.enable_dynamic_groups in a Post-Authentication process, to populate dynamic groups.

See Also:

Attaching an Authorization Scheme to an Application

5.12.2.2.3 Session Management

Use Session Management attributes to reduce exposure at the application-level for abandoned computers with an open web browser.

Attribute	Descriptions
Rejoin Sessions	Control at the application-level whether URLs in this application contain session IDs. When Rejoin Sessions is enabled, APEX attempts to use the session cookie to join an existing session, when a URL does not contain a session ID. To use Rejoin Sessions at the applicaion-level, administrators must enable **Rejoin Sessions** at the instance-level. A more restrictive instance-level setting overrides application and page settings. Rejoin Sessions options include: • **Application Default** - The application-level setting applies. • **Disabled** - If the URL does not contain a session ID APEX creates a new session. • **Enabled for Public Sessions** - If the URL goes to a public page and does not contain a session ID, APEX attempts to utilize the existing session cookie established for that application. APEX only joins using the cookie when the session is not yet authenticated. • **Enabled for All Sessions** - If the URL does not contain a session ID, APEX attempts to utilize the existing session cookie established for that application, providing the following conditions are met: – Session State Protection is enabled for the application and the URL includes a valid checksum. For public bookmarks, the most restrictive item level protection must be either **Unrestricted** or **Checksum Required - Application Level**. – OR, the URL does not contain payload (a request parameter, clear cache or data value pairs). This option requires you set Embed In Frames to **Allow from same origin** or **Deny**. This is not tied to a condition about the URL payload, but also applies to session state protected URLs. **Warning**: Enabling rejoin sessions may expose your application to possible security breaches, as it can enable attackers to take over existing end user sessions. To learn more, see About Rejoin Sessions. **See Also**: • Session Management to configure Rejoin Sessions at the application-level. • Configuring Rejoin Sessions for an Instance in *Oracle APEX Administration Guide*
Deep Linking	Enables or prevents deep linking to an application. Options include: • **Enabled** - The URL to a specific page ultimately redirects there, possibly after the user has logged in. • **Disabled** - If the URL does not contain a valid session ID, APEX starts a new session and redirects to the application's home page. For example, browsers often save the URLs of opened tabs and try to restore the sessions after a restart, causing a deep link. This behavior may be undesirable (for example if a URL points to a page in the middle of a multi-step wizard). Selecting **Disabled**, starts a new session and redirects to the application's home page.

Attribute	Descriptions
Session State Commits	Oracle APEX maintains session state both in PL/SQL global variables for quick access and in database tables. Use this attribute to configure when session state gets written and committed to the database. Options include: • **Immediate** - When state changes during request processing, for example because a PL/SQL process modified a bind variable, immediately write these changes and commit. Note that this commits the main transaction, which may contain changes to your application tables. • **End Of Request** - Oracle APEX maintains a list of session state changes that occur during request processing and applies them at the end of the request. Use this value for better transaction control and Application Continuity in case of unplanned outages. In this context, session state includes item values and attributes of the session itself (for example, the idle timeout). While collections are also part of session state, they are not affected by the Session State Commits attribute.
Maximum Session Length in Seconds	Defines how long (in seconds) sessions can exist and be used by this application. • Enter a positive integer to control how long a session can exist. • Enter 0 to have sessions exist indefinitely. • Leave the value empty in order to use the session duration defined at the workspace-level or the instance level.
Session Timeout URL	Enter an optional URL to redirect to when the maximum session lifetime has been exceeded. The target page in this URL, if implemented in APEX, should be a public page. A common use for this page would be to inform the user of the session expiration and to present a login link or other options. If you do not enter a URL, users will see the message "Your session has timed out" and a link to the application home page. If you enter `#LOGOUT_URL#`, APEX will execute a logout, just like when the user clicked on the application's logout link. Only three substitution items are supported: • `&APP_SESSION.` • `&SESSION.` • `&APP_ID.` Because of the particular purpose of this URL. it is not necessary to include either `&APP_SESSION.` or `&SESSION.` in the link.
Maximum Session Idle Time in Seconds	The Session Idle Time is the time between the last page request and the next page request. Options include: • Enter a positive integer to control the idle time for sessions used by this application. • Leave the value empty in order to use the idle time defined at the workspace level or the instance level.

Attribute	Descriptions
Session Idle Timeout URL	Enter an optional URL to be redirected to when the maximum session idle time has been exceeded. The target page in this URL, if implemented in APEX, should be a public page. A common use for this page would be to inform the user of the session expiration and to present a login link or other options. If you do not enter a URL, users will see the message "Your session has timed out" and a link to the application home page. If you enter `#LOGOUT_URL#`, APEX will execute a logout, just like when the user clicked on the application's logout link.. Only three substitution items are supported in this URL: • `&APP_SESSION.` • `&SESSION.` • `&APP_ID.` Because of the particular purpose of this URL, it is not necessary to include either `&APP_SESSION.` or `&SESSION.` in the link.
Session Timeout Warning in Seconds	The session timeout warning time defines (in seconds) how long before a session times out (either maximum session length, or maximum session idle time), to warn the user. For the maximum session idle time warning, the user will have the opportunity to extend the session. For maximum session length warning, the user will be prompted to save any work, to avoid loss of data when the session maximum time is reached. • Enter a positive integer to control the session timeout warning time for sessions used by this application. • Enter `0` to switch off session timeout warnings for sessions used by this application. • Leave the value empty in order to use the warning time defined at the workspace-level or the instance-level.

See Also:

- About Utilizing Session Timeout
- Configuring Session Timeout for a Workspace and Configuring Session Timeout for an Instance in *Oracle APEX Administration Guide*

5.12.2.2.4 Session State Protection

Enabling Session State Protection can prevent hackers from tampering with URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy. This table describes the attributes available under Session State Protection.

Attribute	Descriptions
Expire Bookmarks	Click **Expire Bookmarks** to reset this hidden application attribute (a checksum salt) salt attribute at any time. Clicking this button causes any bookmarked URLs that contain previously generated checksums to fail when they are subsequently used to access the application. **Tip**: You can also click **Expire Bookmarks** to change the Bookmark Hash Function to switch to a different algorithm for computing checksums.
Manage Session State Protection	Click **Manage Session State Protection** to configure Session State Protection.
Session State Protection	Make a selection from the Session State Protection list, to enable or disable Session State Protection for your application. Selecting **Enabled** turns on session state protection controls defined at the page and item-level. To learn more, see field-level Help.
Allows URLS Created After	Lists the date and time after which bookmarked links are usable to access pages in this application if the bookmarked link contains a checksum and Session State Protection is enabled for the application. Bookmarks created before this date and time are not usable to access this application if the bookmarked link contains a checksum and Session State Protection is enabled for the application. Bookmarks that do not contain checksums or bookmarks that contain checksums that are unnecessary are not affected by this attribute. Their usability is determined using other criteria. A hidden application attribute (a checksum salt) is used during the computation and later verification of checksums included in f?p= URLs generated during page rendering. Checksums are included when Session State Protection is enabled for the application. You can reset this checksum salt attribute at any time by clicking the **Expire Bookmarks** button.
Bookmark Hash Function	Used to create checksums for application-level and user-level checksums in bookmarkable URLs.

See Also:

Preventing URL Tampering

5.12.2.2.5 Browser Security

This table describes the attributes available under Browser Security.

Attribute	Descriptions
Cache	Use **Cache** to enable or disable browser caching of application page contents. If enabled, the browser saves the contents of pages for this application in its cache, both in memory and on disk. Typically when caching is enabled and the browser back button is clicked, the page is loaded from the cache instead of from the server. If disabled, the browser is instructed not to save application page contents and requests the latest page content from the server whenever the URL changes. To avoid the possibility of saving sensitive data, Oracle recommends that this attribute be disabled. Otherwise, it is possible to go back in the browser history after a logout and see cached content from a previous session. Disabling the browser cache also prevents issues with pages that use partial page refreshes, such as is the case with interactive reports. If this attribute is set to **Disabled**, APEX sends the HTTP header `cache-control: no-store` which instructs the browser to not cache the page contents on disk or in memory. Note that this feature requires modern browsers that support the HTTP header response variable cache-control.
Embed in Frames	Controls if a browser may display your application's pages within a frame. Available options include: • **Deny** - The page cannot be displayed in a frame, regardless of the site attempting to do so. • **Allow from same origin** - The page can only be displayed in a frame on the same origin as the page itself. • **Allow** - The page can be displayed in any frame. Displaying pages within frames can be misused with "clickjacking" attacks. In a “clickjacking” attack, the attacker uses multiple layers to trick a user into clicking a button or link on another page when they were intending to click the top level page. Thus, the attacker is hijacking clicks (or keystrokes) meant for their page and routing them to another page. To learn more, see field-level Help.
HTML Escaping Mode	Defines how APEX escapes special characters. Options include: • **Basic**: Escape `&`, `"`, < and >. • **Extended**: Escape `&`, `"`, <, >, `'`, `/` and non-ASCII characters if the database character set is not `AL32UTF8`

HTTP Response Headers	Enter additional application specific HTTP headers that APEX should send on each response and that it does not support in another way (for example, `X-Frame-Options` using the Embed in Frames attribute).
	Example:
	`X-XSS-Protection: 1; mode=block`
	`X-Content-Type-Options: nosniff`

Tip:

Both **Cache** and **Embed in Frames** require modern browsers that support the HTTP header response variable X-Frame-Options.

5.12.2.2.6 Database Session

This table describes the attributes available under Database Session.

Attribute	Descriptions
Parsing Schema	Specifies the schema that all SQL and PL/SQL in the application will be parsed as. You may use `#OWNER#` to reference this value in SQL queries and PL/SQL (for example, in a region or a process).
Initialization PL/SQL Code	Use this attribute to enter a PL/SQL block that sets a context for the database session associated with the current "show page" or "accept page" request. The block you enter here is executed at a very early point during the page request, immediately after the `APP_USER` value is established. The value of `APP_USER` (using `:APP_USER` or `v('APP_USER')`) may be used within the block. Values of other items in session state may be referenced as well, but any such items must have been established in session state before the initiation of the current page request. To view examples, see field-level Help.
Cleanup PL/SQL Code	Use this attribute to enter a PL/SQL block that runs at the end of page processing. It can be used to free or clean up resources that were used, like VPD contexts or database links. To view examples, see field-level Help.
Runtime API Usage	Control how this application can access APEX APIs that modify applications and workspace data while it is running. Options include: • **Modify This Application**: The application can modify itself. • **Modify Other Applications**: The application can change other applications in the workspace. • **Modify Workspace Repository**: The application can change workspace users and groups.

See Also:

Providing Security Through Authorization and *Oracle Label Security Administrator's Guide*

5.12.2.2.7 Advanced

Attribute	Descriptions
Runtime API Usage	Control whether this application can access Oracle APEX PL/SQL APIs that modify applications and workspace data while it is running. Available Runtime API Usage options include: • **Modify This Application** - The application can modify itself. For example, use the `APEX_APPLICATION_ADMIN` API to change the status of a build option. • **Modify Other Applications** - The application can change other applications in the workspace. For example, use the `APEX_APPLICATION_ADMIN` API to change the status of a build option in another application. • **Modify Workspace Repository** - The application can change workspace users and groups. For example, use the `APEX_UTIL.CREATE_USER` API to create a new end user account in the native Oracle APEX user accounts repository.

Attribute	Descriptions
Pass ECID	Enable the **Pass ECID** attribute to pass the Execution Context ID (with a request header named `ECID-Context`) to the external web services for end-to-end tracing. This setting can be overwritten for individual REST calls in a REST Data Source definition or by using `APEX_WEB_SERVICE.SET_REQUEST_ECID_CONTEXT`. When **Pass ECID** is enabled: • The column `ECID` in `APEX_WEBSERVICE_LOG` is populated with the Execution Context ID. • If you are in an APEX session and the application setting for activity logging is enabled, `APEX_WEBSERVICE_LOG.ACTIVITY_LOG_ID` is populated. This enables developers to relate external web service calls to page calls in APEX. • Both `APEX_WEBSERVICE_LOG` and `APEX_WORKSPACE_ACTIVITY_LOG` have a column named `ID`. Because of performance and partition reasons, only `APEX_WEBSERVICE_LOG` has a primary key index on it. **Tip**: You can overwrite the **Pass ECID** by either configuring a specific REST Data Source in an application under Shared Components, REST Data Sources, *YourDataSource*, Advanced, Pass ECID, or by calling the `APEX_WEB_SERVICE.SET_REQUEST_ECID_CONTEXT` API before making a REST call with `APEX_WEB_SERVICE.MAKE_REST_REQUEST`. **See Also**: • Editing a REST Data Source • APEX_WEB_SERVICE in *Oracle APEX API Reference*

5.12.3 Configuring Globalization Attributes

Edit attributes on the Globalization page to configure application globalization options. In App Builder you can develop applications that can run concurrently in different languages.

A single application can be translated to support different languages. Use the attributes on the Edit Globalization Attributes page to specify globalization options such as the primary application language and defaults for date format, time format, timestamp format, time zone format, and CSV encoding.

- Accessing the Globalization Page
  Access the Globalization page page from the Application home page.
- Globalization Page
  The Globalization page has one section: Globalization.

> **See Also:**
>
> Managing Application Globalization

5.12.3.1 Accessing the Globalization Page

Access the Globalization page page from the Application home page.

To access the Globalization page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From Application home page, you can access the application Globalization page in two ways:
   - **Edit Application Definition** button:
     a. Click **Edit Application Definition** to the right of the application name.
     b. Click the **Globalization** tab.
   - From Shared Components:
     a. Click **Shared Components**.
     b. Under Globalization, click **Globalization Attributes**.

   The Globalization page appears.
4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

5.12.3.2 Globalization Page

The Globalization page has one section: Globalization.

- Application Primary Language
- Application Language Derived From
- Document Direction
- Application Date Format
- Application Date Time Format
- Application Timestamp Format
- Application Timestamp Time Zone Format
- Character Value Comparison
- Character Value Comparison Behavior
- Automatic Time Zone
- Automatic CSV Encoding

> **See Also:**
>
> Specifying the Primary Language for an Application

5.12.3.2.1 Application Primary Language

Identifies the language in which an application is developed. This language is the base language from which all translations are made. For example, suppose application 100 was authored in English, translated into French, and published as application 101. English would be the Application Primary Language.

All modifications to the application should be made to the primary language specified here.

5.12.3.2.2 Application Language Derived From

Specifies how Oracle APEX derives the translated application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.

To learn more about the available options, see Field-level Help.

5.12.3.2.3 Document Direction

Sets the document direction. Options include:

- **Language Default** - Uses the direction derived from the language.
- **Left-To-Right**
- **Right-To-Left**

5.12.3.2.4 Application Date Format

Determines the date format to be used in the application.

Use this date format to alter the `NLS_DATE_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If no value is specified, the default date format is derived from the database session at runtime. Consider the following examples:

- `Month DD, YYYY`
- `&MY_DATE_FORMAT.`

5.12.3.2.5 Application Date Time Format

Specify the date time format to be used in the application.

This date time format can be referenced in an application using the substitution reference `&APP_DATE_TIME_FORMAT.`, or in PL/SQL using the function `v('APP_DATE_TIME_FORMAT')`. This attribute does not alter any NLS settings. This value can be a literal string containing a valid Oracle date format mask or an item reference using substitution syntax. If this attribute value is not specified, then a reference to `APP_DATE_TIME_FORMAT` returns the NLS database session date format and the NLS time format. Consider the following examples:

- `Month DD, RRRR HH24:MI`
- `&MY_DATE_TIME_FORMAT.`

5.12.3.2.6 Application Timestamp Format

Determines the timestamp format to be used in the application. Select a timestamp format from the list of values.

Use this timestamp format to alter the `NLS_TIMESTAMP_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle timestamp format mask or an item reference using substitution syntax. If no value is specified, the default timestamp format is derived from the database session at runtime. Consider the following examples:

- `DD-MON-RR HH.MI.SSXFF AM`
- `&MY_TIMESTAMP_FORMAT.`

5.12.3.2.7 Application Timestamp Time Zone Format

Determines the timestamp with time zone format to be used in the application.

Use this date format to alter the `NLS_TIMESTAMP_TZ_FORMAT` database session setting before showing or submitting any page in the application. This value can be a literal string containing a valid Oracle timestamp with time zone format mask or an item reference using substitution syntax. If no value is specified, the default timestamp with time zone format is derived from the database session at runtime. Consider the following examples:

- `DD-MON-RR HH.MI.SSXFF AM TZR`
- `&MY_TIMESTAMP_TZ_FORMAT.`

5.12.3.2.8 Character Value Comparison

Determines the collating sequence for character value comparison in various SQL operations and clauses, for example, `ORDER BY`, `LIKE`, `MIN/MAX`.

Use this value to alter `NLS_SORT` database session parameter for the execution of SQL queries in classic report and interactive report regions. If no value is specified, the default value is derived from the database session at runtime. Consider the following examples:

```
BINARY
GERMAN
CANADIAN_M
```

5.12.3.2.9 Character Value Comparison Behavior

Determines the collation behavior of SQL operations, for example, `LIKE`, `MIN/MAX`.

This value is used to alter `NLS_COMP` database session parameter for the execution of SQL queries in classic report, interactive report, and List view regions, as well as in plug-in API. Options include:

- **Database session NLS setting (default)** - The NLS_COMP value is derived from the database session at runtime.
- **Binary** - Comparisons in `WHERE` clauses and other SQL operations are binary.

- **Linguistic** - Comparisons in `WHERE` clauses and other SQL operations use the linguistic sort specified in the Character Value Comparison attribute (`NLS_SORT`).

5.12.3.2.10 Automatic Time Zone

Controls the setting of the database session time zone. When set to **On**, the client time zone is derived from the client's web browser and set for the duration of the APEX session.

Subsequent page views have the database session time zone set properly per page view. Once set, this setting can be overridden using `APEX_UTIL.SET_SESSION_TIME_ZONE`, or reset using `APEX_UTIL.RESET_SESSION_TIME_ZONE`.

> **See Also:**
>
> *Oracle APEX API Reference*

5.12.3.2.11 Automatic CSV Encoding

Automatic CSV Encoding controls the encoding of all comma-delimited (CSV) report output in an application. The default value for Automatic CSV Encoding is **On**. When Automatic CSV Encoding is set to **On**, CSV report output is converted to a character set compatible with localized desktop applications. The character set for the CSV encoding is determined by the Application Language Derived From setting.

The encoding of pages in App Builder is determined by the character set of the Database Access Descriptor (DAD) used to access Oracle APEX. For example, if the character set of the Database Access Descriptor is AL32UTF8, all pages in all applications in the APEX user interface are encoded in UTF-8.

By default, the CSV output from report regions is encoded in the same character set as the Database Access Descriptor. However, some desktop spreadsheet applications require that the data is encoded in the client desktop operating system character set. In the case of multibyte data, the CSV output from report regions often appears corrupted when opened by a desktop spreadsheet application. This is because the CSV output from report regions is encoded differently than what is required by the desktop application. Enabling Automatic CSV Encoding resolves this issue.

For example, if the user's language preference for an application is `de`, the CSV data is encoded in `Western European Windows 1252`, regardless of the Database Access Descriptor character set setting. If the user's language preference is `zh-cn`, the CSV data is encoded in Chinese GBK.

5.12.4 Editing User Interface Attributes

Use the User Interface page to control the default characteristics of the application and optimize the display.

- Accessing the User Interface Page
  Access the User Interface page from the Application home page.
- User Interface Page
  The User Interface page is divided into the following sections: Icon, Logo, Attributes, Navigation Menu, Navigation Bar, JavaScript, CSS, Concatenated Files, OpenTelemetry, and Advanced.

5.12.4.1 Accessing the User Interface Page

Access the User Interface page from the Application home page.

To access the User Interface page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the User Interface page in two ways:
   - Edit Application Definition button:
     a. Click **Edit Application Definition** to the right of the application name.
     b. Click the **User Interface** tab.
   - From Shared Components:
     a. Click **Shared Components**.
     b. Under User Interface, click **User Interface Attributes**.

   The User Interfaces page appears.
4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

5.12.4.2 User Interface Page

The User Interface page is divided into the following sections: Icon, Logo, Attributes, Navigation Menu, Navigation Bar, JavaScript, CSS, Concatenated Files, OpenTelemetry, and Advanced.

> **Note:**
>
> Required values are marked with a red asterisk (*).

> **Tip:**
>
> To learn more about the attributes on this page, see field-level Help.

- Icon
- Logo
- Attributes
- Navigation Menu
- Navigation Bar
- JavaScript
- CSS

- Concatenated Files
- OpenTelemetry
- Advanced

5.12.4.2.1 Icon

Use the Icon region to manage icons in your APEX applications. When you add an icon to an application it is propagated to the following places: the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.

Application icons are stored in Shared Components, Static Application Files in five different sizes. 32x32, 192x192 and 512x512 are saved as square icons. 144x144 and 256x256 are saved as rounded icons. Any new uploaded icons are automatically resized and propagated to the appropriate places at runtime.

> **Tip:**
>
> For applications created prior to Oracle APEX 23.1, click **Change Icon** on User Interface Attributes page to regenerate the full suite of icons.

To view details or update an existing icon:

1. Click **Edit**.
2. In the Edit Application Icon dialog, you can either:
   - Navigate to file by clicking**Choose File**.
   - Drag and drop a new file to the region provided.

To choose a new icon from library:

1. Click **Change Icon**.
2. In the Edit Application Icon dialog you can:
   - **Upload an icon**. Click the plus (+) sign and navigate to the icon, or drag and drop the icon to the **Upload your own icon** region.
   - **Select an icon from the library**. Select the icon color and then the icon.

To control the upload for all icon sizes, upload your icons to the **Static Application Files** and use the following file paths:

- `#APP_FILES#icons/app-icon-32.png`
- `#APP_FILES#icons/app-icon-144-rounded.png`
- `#APP_FILES#icons/app-icon-192.png`
- `#APP_FILES#icons/app-icon-256-rounded.png`
- `#APP_FILES#icons/app-icon-512.png`

See Also:

- Managing Application Icons
- Managing Static Application Files

5.12.4.2.2 Logo

Use Logo attributes to define an application logo that displays in the top navigation bar. An application logo can be an image, text, image and text, or based on custom markup.

Attribute	Description
Logo	For Logo, select a logo type. Options include: • **None** • **Image** • **Text** • **Image and Text** • **Custom** Additional atttibutes appear based depending upon your selention. To learn more about an attribute see item Help. If you select **Image**, **Text**, **Image and Text**, or **Custom**, additional attributes display and your page template your page template must include the `#LOGO#` substitution string.

See Also:

Managing Application Logos

5.12.4.2.3 Attributes

Attribute	Description	To Learn More
Enable End Users to choose Theme Style	If set to **On**, end users can choose a Theme Style for their sessions within a customization dialog. Only Theme Styles marked as Public are eligible for selection.	See Enabling Users to Select a Theme Style
Add "Built with APEX" to Footer	If set to **On**, Oracle APEX adds the text "Built with ♥ using Oracle APEX" to the footer of every page.	n/a

Attribute	Description	To Learn More
Auto Dismiss Success Messages	Dismiss success message automatically by enabling or disabling the Auto Dismiss Success Messages attribute: • If set to **On** - All success messages in the application are dismissed automatically after 5 seconds. If a user interacts with a message before it is dismissed, the time will be reset and it will be another 5 seconds before the message gets dismissed. • If set to **Off** - Success messages will remain visible until the application user manually dismisses them. **Note**: Enabling this attribute may impact accessibility. Some users may not see or read a success message before it is dismissed and could miss the information in the message. If turning Auto Dismiss Success Messages on, consider providing users with settings that enable them to opt in or out of auto-dismiss, using the apex.message.setDismissPreferences API in *Oracle APEX JavaScript API Reference*	n/a
Home URL	Specify the home page of the application.	n/a
Login URL	Specify the login page of the application.	n/a
Theme	Shows the theme currently associated with the user interface.	See Switching Themes
Theme Style	Select a theme style. This option only displays for newer themes that support theme styles.	See Using Theme Styles
Global Page	If defined, displays the global page for the application.	See Creating a Global Page to Display Components on Every Page

5.12.4.2.4 Navigation Menu

Applications using newer themes, such as *Universal Theme - 42,* provide navigation with navigation menus.

Attribute	Description
Display Navigation	Select **On** to include or **Off** to exclude a navigation men in your application.
Navigation Menu List	Select the list utilized for the navigation menu for the application.

Attribute	Description
Position	Select the position where you would like your navigation menu to be placed on this application. • **Side** renders the navigation menu list template in the `#SIDE_GLOBAL_NAVIGATION_LIST#` position on your page template. • **Top** renders the navigation menu list template in the `#TOP_GLOBAL_NAVIGATION_LIST#` position on your page template. For example, if you choose **Side**, your navigation could be rendered as a tree on the left side of your pages; and choosing **Top** can render your navigation as a menu bar in the header of your pages.
List Template	Select the List Template used to render the navigation menu for this application.
Template Options	Set Template Options for the List Template used for the navigation menu list for the application.

See Also:

Managing Navigation Menus

5.12.4.2.5 Navigation Bar

Applications using newer themes, such as *Universal Theme - 42,* include navigation bar lists. Navigation Bar settings enable you to select list and list templates. Selecting classic implementation uses tabs instead of a list. Navigation Bar attributes only display with newer themes.

Attribute	Description
Implementation	Select how to you wish to implement the navigation bar in this application. • **Classic** renders the navigation bar as a classic navigation bar in the `#NAVIGATION_BAR#` position on your page template. • **List** renders the navigation bar as a list, using the selected list and list template in the `#NAVIGATION_BAR#` position on your page template.
Navigation Bar List	Select the list utilized for the navigation bar on this application.
List Template	Select the List Template used to render the navigation menu for this application.
Template Options	Set Template Options for the List Template used for the navigation menu list for the application.

See Also:

Managing Navigation Bar Lists

5.12.4.2.6 JavaScript

Use these attributes to control or modify how an application handles JavaScript.

Attribute	Description

File URLs	Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode.
	JavaScript file URLs you enter here replaces the `#APPLICATION_JAVASCRIPT#` substitution string in the page template.
	Note: You do not need to include opening or closing script tags. Just write the URL.
	Examples:
	• Standard file reference:
	`/myjs/main.js`
	• Standard file reference which loads the minified file `main.min.js` from `/myjs/minified/` for regular page views and `main.js` from `/myjs/` in debug mode:
	`/myjs/#MIN_DIRECTORY#main#MIN#.js`
	• Use this syntax if the library is also used from a RequireJS require call. The define name must match what is used in the requirejs.config paths and the value must be the global symbol exposed by the library. Only applies when using RequireJS. This avoids RequireJS loading that library a second time.
	`[define hammerjs=Hammer]/myjs/hammer.js`
	• Use this syntax if the file has a dependency on RequireJS. This will automatically load RequireJS before this file.
	`[require requirejs]/myjs/main.js`
	• Use this syntax if the file has a dependency on Oracle JET. This will automatically load RequireJS and the Oracle JET RequireJS configuration before this file.
	`[require jet]/myjs/main.js`

Include Deprecated or Desupported Javascript Functions	Specifies if deprecated or desupported JavaScript functions are included on every page in the application. The functions deprecated or desupported are listed in the Release Notes for every release. If you are confident your application does not contain any references to deprecated or supported functions, do not make a selection and reduce the overall size of the JavaScript files loaded.

Attribute	Description
Include jQuery Migrate	Specifies if the jQuery Migrate plug-in should be included on every page in the application. The jQuery Migrate plug-in restores deprecated features and behaviors of jQuery so that old JavaScript code and jQuery plug-ins will still run properly with the j Query version loaded by APEX. This plug-in also helps find where you are using deprecated or removed jQuery APIs by logging warnings to the browser console. If you are confident your application does not use a jQuery plug-in and does not contain any references to deprecated jQuery features, disable **Include jQuery Migrate** to reduce the overall size of the JavaScript files loaded. **See Also**: Reviewing Old JavaScript

5.12.4.2.7 CSS

In **File URLs**, enter CSS file URLs to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

File URLs you enter here will replace the `#APPLICATION_CSS#` substitution string in the page template.

Note: You do not need to include opening or closing `link` tags. Just include the file URL.

Examples:

- Standard file reference.

  ```
  /mycss/main.css
  ```

- Standard file reference which loads the minified file `main.min.css` from `/mycss/minified/` for regular page views and `main.css` from `/mycss/` in debug mode.

  ```
  /mycss/#MIN_DIRECTORY#main#MIN#.css
  ```

- Conditional media query.

  ```
  [media="only screen and (max-device-width: 480px)"]/mycss/smartphone.css
  ```

- File reference using a query string in the URL referencing the application version, such that when the application is upgraded (and the version is incremented), the browser is instructed to always load the new file and ignore any previously cached files.

  ```
  /mycss/main.css?version=#APP_VERSION#
  ```

5.12.4.2.8 Concatenated Files

Using a concatenated file can increase the performance of loading your page because instead of issuing multiple HTTP requests for each single file, the browser only loads one file. This approach gives you the option to use smaller, more modular files during development and to

use a single concatenated file when running the application outside of the APEX development environment.

To create a concatenated file, click **Concatenated File** and follow the on-screen instructions. To learn more and view examples, see field-level Help.

5.12.4.2.9 OpenTelemetry

OpenTelemetry is a standard for collecting user experience data in web applications, such as page load times and user interactions. Enabling OpenTelemetry makes it easier to monitor and analyze how users engage with an application.

To configure OpenTelemetry, first set the **Client Logging Service URL** and the **Token Relay URL** in **Workspace Utilities**, **OpenTelemetry**.

Then, for a specific application, in **Shared Components**, edit the **User Interface Attributes**. In **OpenTelemetry**, specify the **Product Family**.

5.12.4.2.10 Advanced

Use the attributes under Advanced to define the virtual path the Web server uses to point to the images directory distributed with Oracle APEX.

Attribute	Description
#APP_FILES# Path	Determines the virtual path the Web server uses to point to the static files when using the `#APP_FILES#` substitution string. Do not specify anything to reference files which are stored with your application definition in the database. For performance reasons you can also store your application files on your Web Server. Use any valid URL to reference them. Examples: • `/myFiles/` • `http://contentDeliveryNetwork.com/myFiles/`
#APEX_FILES# Path	Determines the virtual path the web server uses to point to the images directory distributed with App Builder. During installation, the virtual path is configured as `/i/`. If you are unsure, contact your administrator. When embedding an image in static text (for example, in page or region headers or footers), you can reference an image using the substitution string `#APEX_FILES#`. For example, to reference the image `go.gif`, you would use the following syntax: `<img src="#APEX_FILES#go.gif">` **Note:** If the entered image prefix is equal to the instance image prefix, then the application-level attribute will always be null. This easily facilities the movement of an application across different instances that may have different image prefixes. **See Also:** APEX_FILES

Attribute	Description
Media Type	Enter the Internet media type. An Internet media type is two-part identifier for file formats on the internet. A Media Type is composed of at least two parts: a type, a subtype, and one or more optional parameters. This Media Type is used in the Content-Type HTTP header when rendering the page. The page-level Media Type overrides the application-level Media Type. The default value for this attribute is `NULL`. If both the page-level and application-level values for Media Type are `NULL`, the Media Type `text/html` is used.

5.12.5 Configuring Progressive Web App Attributes

Use the Progressive Web App page to configure attributes for both new and existing Progressive Web App applications.

> **Tip:**
>
> Progressive Web App attributes only display if the Application Definition, **Properties**, **Friendly URLs** attribute is **On**.

- Accessing the Progressive Web App Page
  Access the Progressive Web App page from the Application home page.
- Progressive Web App Page
  The Progressive Web App page is divided into the following sections: General, Installability, Push Notifications, and Service Worker Configuration.

> **See Also:**
>
> Creating a Progressive Web App (PWA)

5.12.5.1 Accessing the Progressive Web App Page

Access the Progressive Web App page from the Application home page.

To access the Progressive Web App page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the Progressive Web App page in two ways:
   - **Edit Application Definition** button:
     a. Click **Edit Application Definition** to the right of the application name.
     b. Click the **Progressive Web App** tab.
   - From Shared Components:

a. Click **Shared Components**.

b. Under User Interface, click **Progressive Web App**.

The Progressive Web App page appears.

4. Edit the appropriate attributes.
5. Click **Apply Changes** to save your changes.

5.12.5.2 Progressive Web App Page

The Progressive Web App page is divided into the following sections: General, Installability, Push Notifications, and Service Worker Configuration.

> **Tip:**
>
> To learn more about the attributes on this page, see field-level Help.

- General
- Installability
- Push Notifications
- Service Worker Configuration

5.12.5.2.1 General

> **Tip:**
>
> Progressive Web App attributes only display if the Application Definition, **Properties**, **Friendly URLs** attribute is **On**. See Properties, Friendly URLs.

Use General attributes to enable and disable to configure Progressive Web App applications.

Attribute	Description	To Learn More
General	Enable Progressive Web App functionality. Enabling this option: • Serves static files more efficiently using advanced caching. • Enables the display of declarative Progressive Web App settings.	See Creating a Progressive Web App (PWA)
Installable	Only displays if **Enable Progressive Web App** is **On**. A Progressive Web App can be installed on devices. Installing a Progressive Web App, adds an icon to the device home screen to feel like a native application. Enabling this option: • Add a new navigation bar entry: `Install App`. • Enables users to install the APEX app on devices.	See Creating a Progressive Web App (PWA)

5.12.5.2.2 Installability

> **Tip:**
>
> Installability attributes only display if the Progressive Web App, General attributes **Enable Progressive Web App** and **Installable** attribute are **On**.

Use Installability attributes to configure Progressive Web App (PWA) attributes for display, screen orientation, theme color, background color, IOS status bar style, app description, and custom manifest as well as upload screenshots and shortcuts.

Attribute	Description	To Learn More
Display	Select the preferred display mode for the application. This option controls how much of the browser UI the user sees. Options include: • **Fullscreen** - Opens the PWA without any browser UI. This option takes the entirety of the screen or window. • **Standalone**- Opens the PWA to look like a standalone application. The app runs in a distinct window, separate from the browser. Standard browser UI components like the URL bar and buttons are not visible. • **Minimal UI** - Similar to standalone, except with browser controlling navigation buttons like back and refresh. • **Browser** - Opens the PWA is a regular browser UI. **Note**: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately. For more information, see https://web.dev/add-manifest//	See Creating a Progressive Web App (PWA)
Screen Orientation	Select the preferred screen orientation when using this Progressive Web App. This option applies mostly for mobile devices. **Note**: Some devices and browsers ignore this attribute. APEX is only relaying this information to the device which handles the Progressive Web App appropriately.	See Creating a Progressive Web App (PWA)

Attribute	Description	To Learn More
Theme Color	Defines the default theme color for the application. This attribute affects how the operating system displays the application (for example, for an Android's task switcher, the theme color surrounds the application). **Note**: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately	See Creating a Progressive Web App (PWA)
Background Color	Select the background color. The background color defines a placeholder background color for the application to display before its stylesheet is loaded. Therefore, the background color should match the theme body background color for a smooth transition between launching the web application and loading the application content. This affects how the operating system displays the application (for example, for an Android's task switcher, the theme color surrounds the application). **Note**: Some devices and browsers ignore this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately	See Creating a Progressive Web App (PWA)
iOS Status Bar Style	The iOS Status Bar Style attribute is rendered as a meta tag on the page to control the appearance of the status bar in a web application when it is added to the home screen on an iOS device. Options include: • **Default** - This is the default value and it displays the status bar in the usual way. • **Black** - This value makes the status bar black. • **Translucent** - This value makes the status bar translucent, enabling the content to be displayed behind it. **Note**: Only iOS devices use this attribute. Oracle APEX is only relaying this information to the device which handles the Progressive Web App appropriately.	See Creating a Progressive Web App (PWA)
App Description	Explains what the application does. Use this attribute to provide more information to users when they are prompted to install the application.	See Creating a Progressive Web App (PWA)

Attribute	Description	To Learn More
Custom Manifest	Used to store additional JSON properties for the Web App Manifest file. The properties found here will be used to extend the declarative options above, by overriding existing properties or appending new properties.	See Creating a Progressive Web App (PWA) For more information on building a Custom Manifest, you can read the Web App Manifest specifications: https://www.w3.org/TR/appmanifest/.
Screenshots	Include screenshots for promotional purposes when users are prompted to install the Progressive Web App. Click **Add Screenshot** to upload a promotional screenshot. For best results, Oracle recommends all screenshots have the same aspect ratio.	See Creating a Progressive Web App (PWA)
Shortcuts	**Note**: An APEX application can only use PWA Shortcuts if Rejoin Sessions is enabled. Before configuring shortcuts, click either **Enable for Public Sessions** or **Enable for All Sessions**. To learn more, see Session Management. Use shortcuts enable users of installed PWAs to quickly access a specific page. For touch-enabled devices, shortcuts are accessible when doing a long-press on the application icon on the home screen. For other devices, shortcuts are accessible when doing a right click on the application icon on the task bar. Click **Add Shortcut** to provide users with a shortcut to quickly access a specific page.	See Creating a Progressive Web App (PWA)

5.12.5.2.3 Push Notifications

Web push notifications enable web apps to send notifications to a user's device even when the user is not currently on the website. Notifications can be sent to a user's desktop or mobile device and can be used to alert the user to new content or events on the website, or to re-engage the user with the website.

Enabling this option:

- Adds a new navigation bar entry: **User Settings**
- Enables users to subscribe and unsubscribe to push notifications.
- When subscribed, users can receive push notifications from the APEX app.

If **Enable Push Notifications** is enabled, the following attributes appear:

- Credentials - Select the credential to authenticate against. To edit the credential, navigate to Shared Components, Credentials.
  Credentials are required to setup push notification securely. A public and private key pair are used respectively to allow users to subscribe to push notifications on the client side, and for the server to send the notifications to the subscribers.
- Settings Page - If this application already has the push notification feature page, the page number displays here. If the application does not have the push notification feature page, it will show that the feature page is available to add.

- Contact Email - Enter the email address to contact in case the push notification service provider (Google, Mozilla, Apple, etc.) wants to contact the owner of the application. We recommend using a group email so it's not dependent on a single individual.

5.12.5.2.4 Service Worker Configuration

Use the Service Workers Configuration attributes to configure service workers. Service workers are JavaScript files that can execute code even when the app is not in use.

Service Workers can listen to events such as fetching resources or handling notifications. You can modify a service worker through hooks, or by replacing events completely.

Service Worker options include:

- **Default** - The APEX engine generates it with the following strategy:
  - Install and activate the service worker
  - Serve resources from cache if cache exists
  - Otherwise serve from network, then put resource in cache
  - Serve an offline page if network fails
- **Configure Hooks** - Fill in the placeholders in the service worker code to add your own. Hooks locations are provided by Oracle APEX.

  To learn more, see field-level Help.
- **File URL** - Provide a URL reference to a file that contains the service worker hooks interface. This is useful for sharing service worker hooks across multiple applications. If using the file reference, make sure to follow the interface architecture appropriately or else the service worker will fail at runtime.

  Use **File Reference URL** to provide a URL reference to a file that contains the service worker hooks interface. This is useful for sharing service worker hooks across multiple applications. If using the file reference, make sure to follow the interface architecture appropriately or else the service worker will fail at runtime. To view the interface structure, see field-level Help for this attribute.

5.12.6 Configuring Generative AI Attributes

Configure the attributes on the AI page to make Generative AI capabilities available to runtime applications.

To include Generative AI capabilities (such as an AI Assitant) in a runtime application, you must edit the Application Definition and configure attributes on the AI page. To learn more, see Including Generative AI in Applications.

> **Tip:**
>
> Before you can configure attributes on the AI page, you must define Generative AI Services in Workspace Utilities. If no Generative AI Services are available, the `No AI Services available message` appears. Click **Configure** to access the Generative AI Service page.

- Accessing the AI Page
  Access the AI page from the Application home page.

- AI Page
  The AI page contains one region, Generative AI.

> **See Also:**
>
> Managing Generative AI Services

5.12.6.1 Accessing the AI Page

Access the AI page from the Application home page.

To access the AI page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. On the Application home page, click **Edit Application Definition** to the right of the application name.
4. Click the **AI** tab.
5. Edit the appropriate attributes.
6. Click **Apply Changes** to save your changes.

5.12.6.2 AI Page

The AI page contains one region, Generative AI.

Generative AI

Use Generative AI attributes to enable Generative AI capabilities for an application.

Attribute	Description	To Learn More
Service	Select the default AI Service for this application. If no Generative AI services are available, the `No AI Services available` message appears. Click **Configure** to access the Generative AI Service page and create one.	See Creating a Generative AI Service Object
Consent Message	Enter a consent message for the use of the selected AI Service. The Consent Message you define displays to the user the first time they access the AI Service. The user must agree with the message before they interact with the AI. The user's choice is stored as a preference so they are only prompted once. You can also use the `APEX_AI` package to programmatically set or clear user consent.	See APEX_AI in *Oracle APEX API Reference*

5.13 Managing Application Logos

Add and update an application logo in an Oracle APEX application.

An application logo displays in the top navigation bar of your application at runtime.

> **Tip:**
>
> To include a logo in your application, your page template must include the `#LOGO#` substitution string.

- Managing Logos in the User Interface Attributes
  Edit User Interface attributes to add or update an application logo.
- Editing App Logos at Runtime
  Use Edit Logo on the Runtime Developer toolbar to quickly change an app logo that displays at runtime.

5.13.1 Managing Logos in the User Interface Attributes

Edit User Interface attributes to add or update an application logo.

To manage an application logo by editing User Interface attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the User Interface page in two ways:
   - **Edit Application Definition** button:
     a. Click **Edit Application Definition** to the right of the application name.
     b. Click the **User Interface** tab.
   - From Shared Components:
     a. Click **Shared Components**.
     b. Under User Interface, click **User Interface Attributes**.

   The User Interfaces page appears.
4. Under **Logo**, select a logo type. Additional attributes appear depending upon your selection.
   - **None** - No logo is shown in the app.
   - **Image**- Creates a logo based on a valid image file type.
     – Logo - Select **Image**.
     – Image URL - Enter the URL for the image to be used as the logo, including the filename extension. Note your page template must include the `#LOGO#` substitution string. For example:

       ```
       #APP_FILES#my-logo.png
       ```
   - **Text** - Defines a text only logo.

  - Logo - Select **Text**.
  - Text - Select **Text** - Enter the text that should be display in the application logo.
- **Image and Text** - Defines an image and text logo.
  - Logo - Select **Image and Text**.
  - Image URL - Enter the URL for the image to be used as the logo, including the filename extension. Note your page template must include the `#LOGO#` substitution string. For example:

    ```
    #APP_FILES#my-logo.png
    ```

  - Text - Select **Text** - Enter the text that should be display in the application logo.
- **Custom**
  - Logo - Select **Custom**.
  - **Custom HTML** - Enter markup of your application's logo. The value in this field will be inserted into the `#LOGO#` substitution string of your page template.

5. Click **Apply Changes**.

> **Tip:**
>
> You can also reference an image uploaded to the static file repository using a substitution string. See Referencing Static Application Files and About Referencing Static Workspace Files.

> **See Also:**
>
> - Managing Static Application Files
> - Managing Static Workspace Files
> - Creating Custom Themes
> - Page Templates

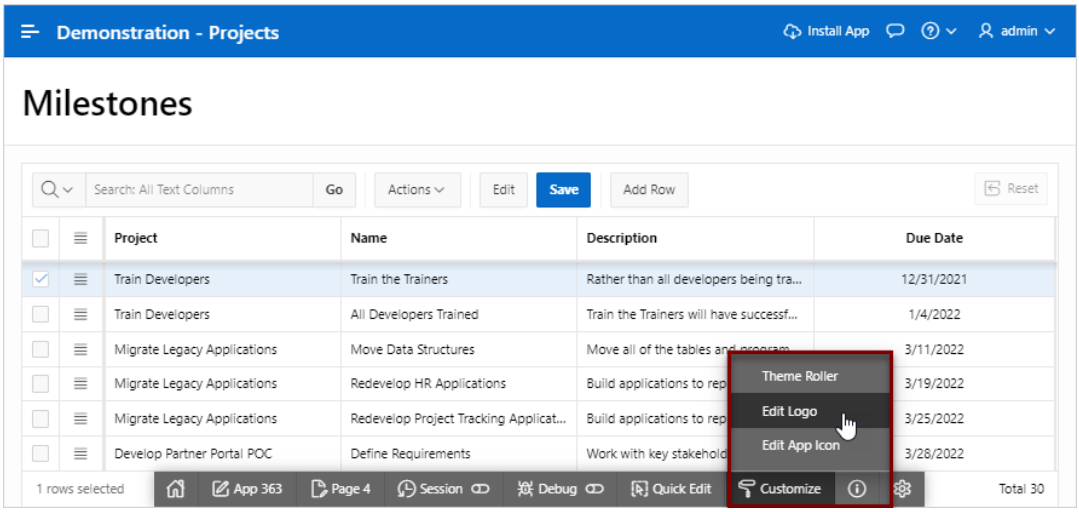
5.13.2 Editing App Logos at Runtime

Use Edit Logo on the Runtime Developer toolbar to quickly change an app logo that displays at runtime.

To add or update an application logo at runtime use Edit Logo:

1. Preview the page by running it.

   The Runtime Developer toolbar displays at the bottom of any editable running page.
2. On the Runtime Developer Toolbar. click **Customize** and then select **Edit Logo**.

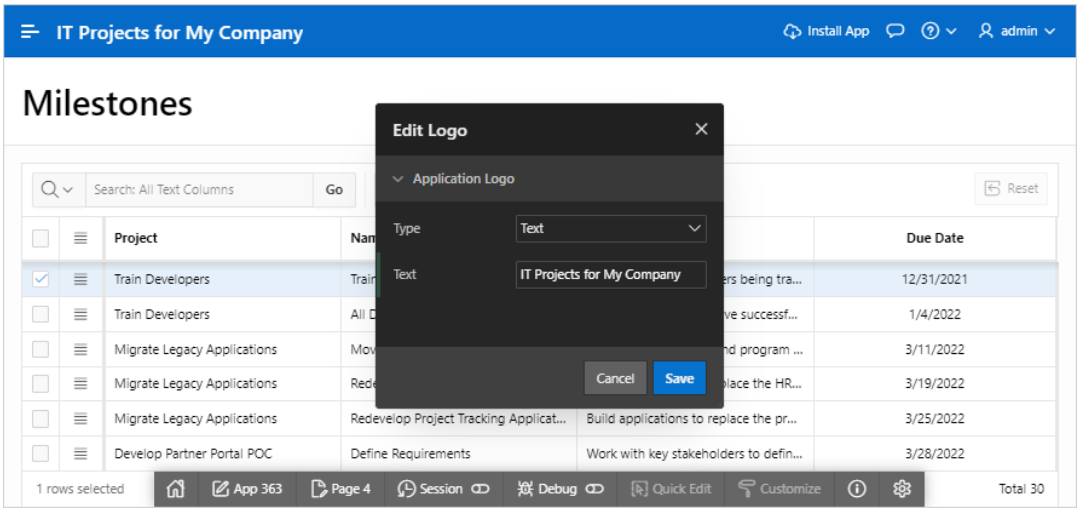


The Edit Logo dialog appears.

3. Type - Select the type of logo. Options include:
   - **None** - No logo is shown in the app.
   - **Text** - Defnes a text only logo.
     - Type - Select **Text**.
     - Text - Enter the text that should be display as the application logo.
   - **Image** - Creates a logo based on a valid image file type and under 500 kb in size.
     - Type - Select **Image**.
     - Specify the image - Drag and drop the file, or click **Choose File** and navigate to the image.
   - **Image and Text**:
     - Type - Select **Image and Text**.
     - Specify the image - Drag and drop the file, or click **Choose File** and navigate to the image.
   - **Custom** define custom HTML to serve as the application's logo:

     Due to security concerns, however, this cannot be configured using Edit Logo. When you select Custom, the **User Interface Attibutes** button appears. Click **User Interface Attibutes** and add the **Custom** logo on the User Interface Attributes page.

In following example, Type is set to **Text** and the Text reads `IT Projects for My Company`. Note that the revised logo, IT Projects for My Company, displays at the top of the page.



4. Click **Save**.

   The confirmation message appears:

   ```
   Logo saved successfully
   ```

5. Click **Cancel** to exit the dialog.

> **Note:**
>
> You can also edit an application logo by editing the application User Interface attributes. See Managing Logos in the User Interface Attributes.

5.14 Managing Application Icons

Add and update the icon that displays as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon in Oracle APEX applications.

- About Application Icons
  Learn about managing application icons in your APEX application.
- Managing Icons in User Interface Attributes
  Edit or upload application icons from the Icon region on the User Interface page.
- Using Edit App Icon on the Runtime Developer Toolbar
  Edit icons in runtime using the Use Edit App Icon option on on the Runtime Developer toolbar, Customize menu.

5.14.1 About Application Icons

Learn about managing application icons in your APEX application.

When you add an icon to an application it is propagated to the following places in your application:

- the Favicon (browser tab icon)
- the PWA icon
- Apple touch icon (when adding a PWA app to iOS)

- App Builder icon (which displays on the application home page)
- Sign In page
- Home page breadcrumb region

Application icons are stored in **Shared Components**, **Static Application Files** in five sizes:

- 32x32 - Mini square icon
- 144x144 - Small rounded icon
- 192x192 - Small square icon
- 256x256 - Large rounded icon
- 512x512 - Large square icon

When editing an individual icon, you take ownership of the icon. Be sure to upload an icon that matches the right size (that is 32x32, 144x144, 192x192, 256x256, or 512x512).

> **Tip:**
>
> The two rounded icons (144x144 and 256x256) are used in places like the macOS dock, where design guidelines encourage the use of rounded icons. The exact specifications for the rounded icons is to have 10% padding on each side and 20% border radius.

You can edit or upload app icons in the following ways:

- **When Creating a New App** - When creating a new application, click **Set Icon** on the Create an Application page in the Create Application Wizard. See Running the Full Create Application Wizard.
- **By Editing User Interface Attributes** - By editing the attributes under **Icon** on the User Interface page. See Managing Icons in User Interface Attributes.
- **By Editing the Icon in Runtime** - In runtime, click **Customize**, **Edit App Icon** on the Runtime Developer Toolbar. See Using Edit App Icon on the Runtime Developer Toolbar
- **By Uploading Icons to Static Application Files** - Application icons are stored in Shared Components, Static Application Files in five sizes: 32x32, 144x144, 192x192, 256x256, or 512x512.

  To control the upload for all icon sizes, upload your icons to the Shared Components, Static Application Files and use the following file paths:

  - `#APP_FILES#icons/app-icon-32.png`
  - `#APP_FILES#icons/app-icon-144-rounded.png`
  - `#APP_FILES#icons/app-icon-192.png`
  - `#APP_FILES#icons/app-icon-256-rounded.png`
  - `#APP_FILES#icons/app-icon-512.png`

> **See Also:**
>
> Managing Static Application Files

5.14.2 Managing Icons in User Interface Attributes

Edit or upload application icons from the Icon region on the User Interface page.

> **Tip:**
>
> If an application uses legacy icons, a note displays at top of the Icon region. Oracle recommends replacing legacy icons by clicking **Change Icon** button.

To manage application icons by editing User Interface attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the User Interface page in two ways:
   - Edit Application Definition button:
     a. Click **Edit Application Definition** to the right of the application name.
     b. Click the **User Interface** tab.
   - From Shared Components:
     a. Click **Shared Components**.
     b. Under User Interface, click **User Interface Attributes**.

   The User Interfaces page appears.
4. Find **Icon**.

   Existing icons display in a table.
5. To edit or update an icon, click **Edit**.
   a. In the Edit Application Icon dialog you can either:
      - Click **Choose File**to navigate to the new icon.
      - Drag and drop a new file to the region provided.
   b. Click **Close**.
6. To choose a new icon from the library, click **Change Icon**.
   a. In the Edit Application Icon dialog, you can either:
      - **Upload an icon**. Click the plus (+) sign and navigate to the icon, or drag and drop the icon to the **Upload your own icon** region.

        If you upload a custom icon, it is resized to a number of sizes and will be propagated to the appropriate places at runtime.
      - **Select an icon from the library**. Select the icon color and then the icon.
   b. Click **Close**.
7. Click **Apply Changes** to save your changes.

5.14.3 Using Edit App Icon on the Runtime Developer Toolbar

Edit icons in runtime using the Use Edit App Icon option on on the Runtime Developer toolbar, Customize menu.

To use Edit App Icon:

1. Preview the page by running it.

   The Runtime Developer toolbar displays at the bottom of any editable running page.
2. On the Runtime Developer toolbar. click **Customize** and then select **Edit App Icon**.

   The Edit Applicatioin Icon dialog appears.
3. In the Edit Application Icon dialog, you can either:
   - Drag and drop the file to the **Upload a new icon** region.
   - Click the **Upload a new icon** region and navigate to the file.
4. When the new icon appears, use the Zoom slider to reposition it within the frame.

   When focused on the icon cropper, the following keyboard shortcuts are also available:
   - Left Arrow - Move the image left
   - Up Arrow - Move the image up
   - Right Arrow - Move the image right
   - Down Arrow - Move the image down
   - I - Zoom in
   - O - Zoom out
   - L - Rotate left
   - R - Rotate right

   Press Shift to move faster.
5. To save your changes, click **Save**.
6. To exit the dialog without saving, click **Cancel**

5.15 Adding Developer Comments

Add comments to an application, a page, or a group of pages using the Developer Comment, Bug, or To Do button. You can use developer comments to communicate application changes, report issues, or record developer suggestions.

- Locating Developer Comment Button
  Locate the Developer Comment button.
- Adding Developer Comments to an Application
  Add developer comments to an application, a page, or a group of pages.
- Viewing and Editing Developer Comments
  Click the Developer Comment button to view or edit existing comments.
- Deleting Developer Comments
  Delete specific developer comments or delete multiple comments at once.

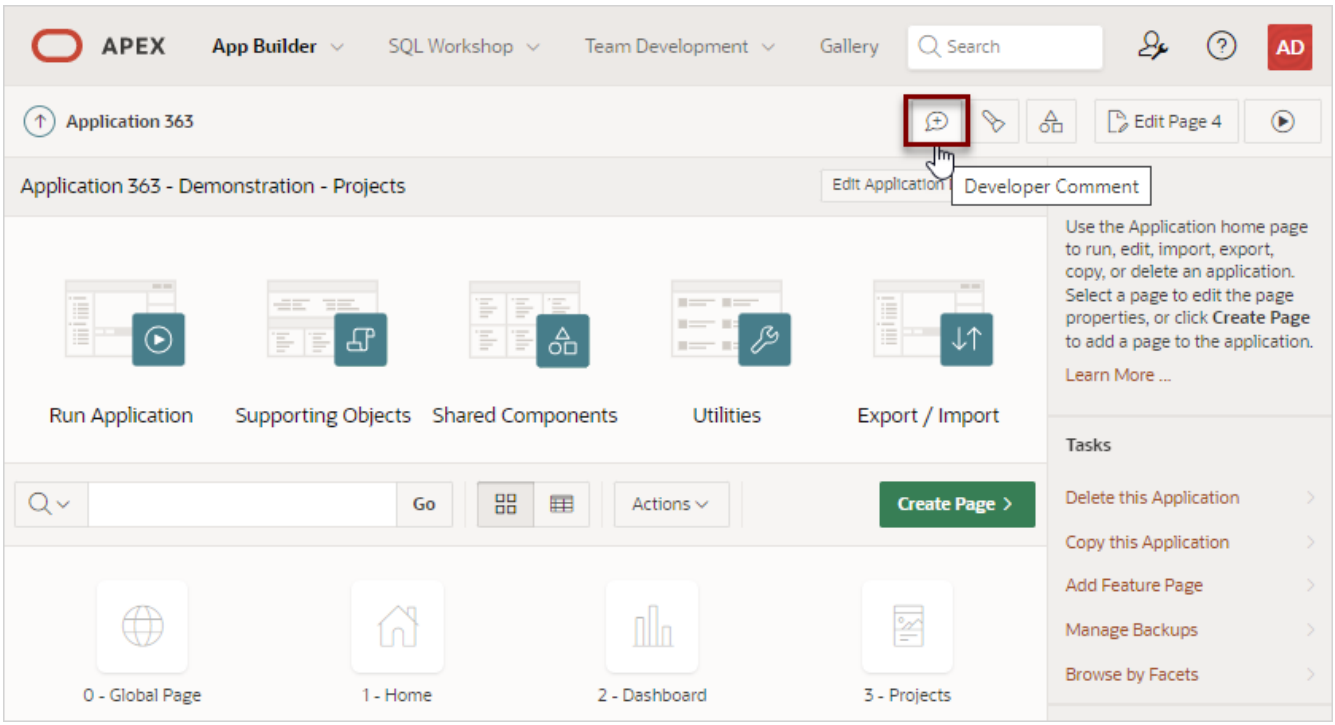
- Accessing the Developer Comments from Shared Components
  Access the Developer Comments page from Shared Components.
- Viewing the Developer Comments Calendar
  View the Developer Comments Calendar page.

> **See Also:**
>
> Adding Application Comments

5.15.1 Locating Developer Comment Button

Locate the Developer Comment button.

The Developer Comment button resembles a small word balloon. This button displays on most pages in App Builder that relate to a specific application or application page.



5.15.2 Adding Developer Comments to an Application

Add developer comments to an application, a page, or a group of pages.

To add a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment** button.
4. In Comment, enter up to 4000 characters of text.

5. Click **Create**.

5.15.3 Viewing and Editing Developer Comments

Click the Developer Comment button to view or edit existing comments.

To edit a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment** button.
4. Click **View Comments**.

   The Developer Comments page appears.
5. Locate the comment to be edited.
6. Click the **Edit** icon.

   The Edit Comment page appears.
7. In Comment, enter up to 4000 characters of text.
8. Click **Apply Changes**.

> **Tip:**
>
> You can also view and edit Developer Comments on the Shared Components page. Navigate to the Shared Components page and click **Developer Comments** in the Tasks region.

5.15.4 Deleting Developer Comments

Delete specific developer comments or delete multiple comments at once.

- Deleting a Specific Developer Comment
  Learn about deleting a developer comment.
- Deleting Multiple Developer Comments
  Learn about deleting multiple developer comments.

5.15.4.1 Deleting a Specific Developer Comment

Learn about deleting a developer comment.

To delete a developer comment:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment** button.
4. Click **View Comments**.

   The Developer Comments page appears.
5. Locate the comment to be deleted.
6. Click the **Edit** icon.

The Edit Comment page appears.

7. Click **Delete**.

5.15.4.2 Deleting Multiple Developer Comments

Learn about deleting multiple developer comments.

To delete multiple developer comments:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Developer Comment** button.
4. Click **View Comments**.
5. Click **Delete Comments**.
6. Select one of the following actions:
   - **Delete all comments**
   - **Delete comments created by a developer**
   - **Delete comments by date**
7. Follow the on-screen instructions.

5.15.5 Accessing the Developer Comments from Shared Components

Access the Developer Comments page from Shared Components.

To view and edit Developer Comments from Shared Components:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under Tasks, click **Developer Comments**.

   The Developer Comments page appears.
5. Locate the comment to be edited.
6. Click the **Edit** icon.

   The Edit Comment page appears.
7. In Comment, enter up to 4000 characters of text.
8. Click **Apply Changes**.

5.15.6 Viewing the Developer Comments Calendar

View the Developer Comments Calendar page.

To view the Developer Comments in a calendar:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.

4. Under Tasks, click **Developer Comments**.

   The Developer Comments page appears.

5. Click **Developer Comments Calendar**.

   The Developer Comments Calendar page appears.

6. Use the Previous, Today and Next buttons to navigate.

5.16 Using the Find Icon

Click the Find icon to search for items, pages, queries, tables, PL/SQL code, view debug reports, view session state, and view errors.

The Find icon resembles a flashlight and displays on many pages in App Builder including the Application home page, Shared Components, and the Edit Application Definition.

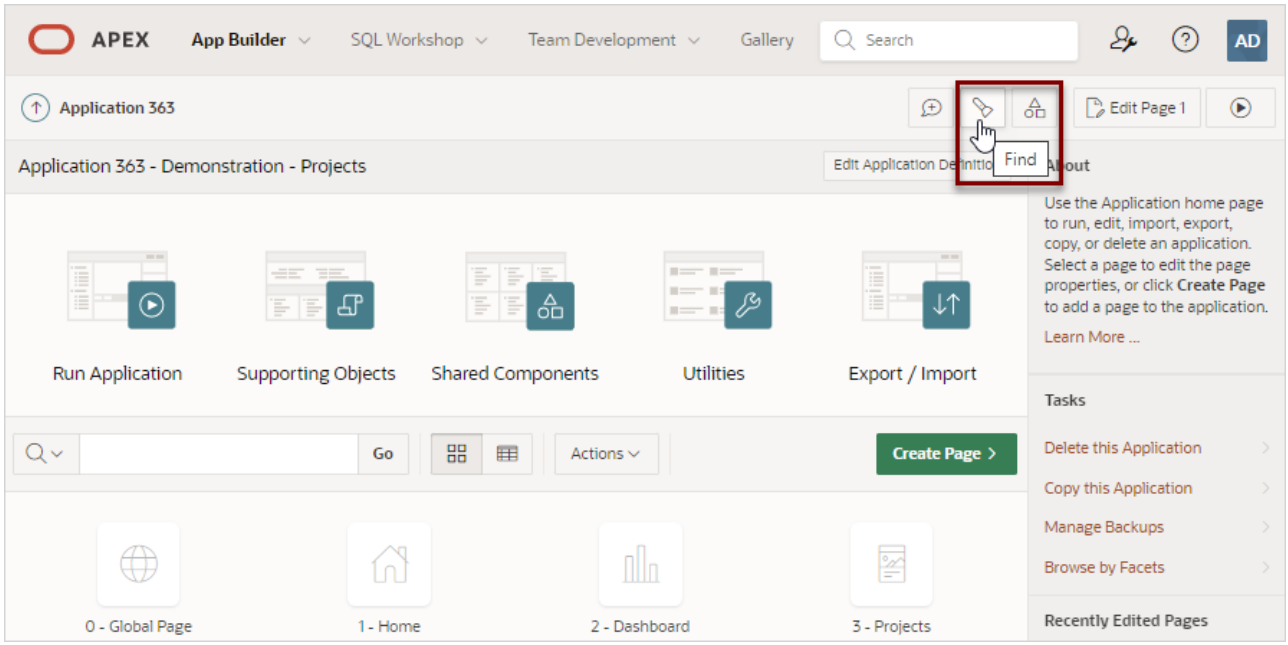
- Locating the Find Icon
  Learn about the Find Icon.
- Searching for Items
  Search for items using the Item Finder.
- Searching for Pages
  Search for pages using the Page Finder.
- Searching for Queries
  Search for queries using the Query Finder.
- Searching for Tables
  Search for tables using the Table Finder.
- Using the PL/SQL Finder
  Locate and view details about stored procedures, functions, and packages using the Procedure Finder.
- Viewing Application Errors
  View a report of application errors.

5.16.1 Locating the Find Icon

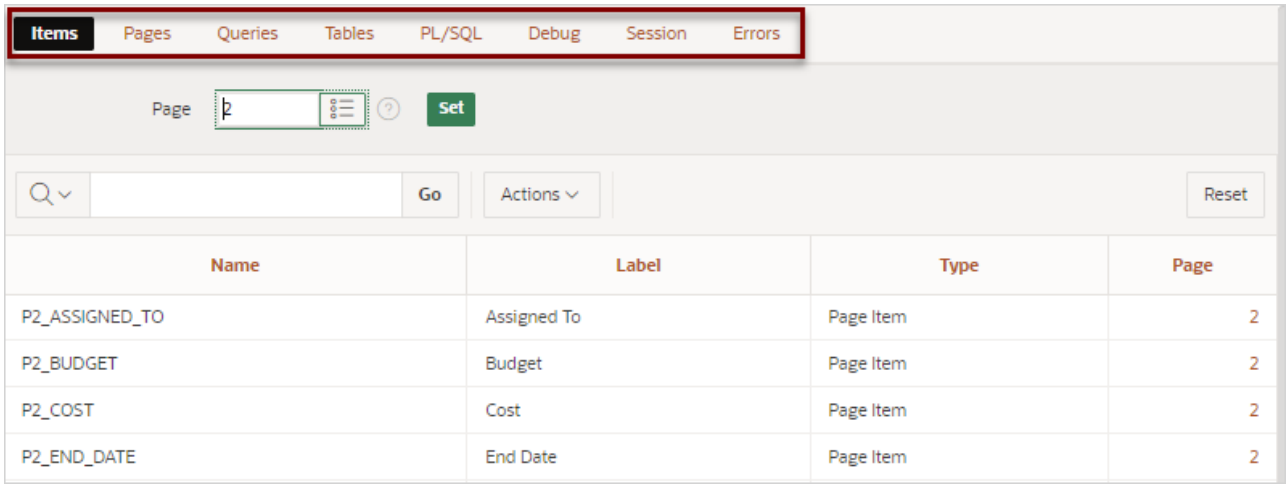
Learn about the Find Icon.

To use the Find icon:

1. Click the **Find** icon.



The Items Finder appears.



2. Click one of the following tabs:
   - **Items** - See Searching for Items.
   - **Pages** - See Searching for Pages.
   - **Queries** - See Searching for Queries.
   - **Tables** - See Searching for Tables.
   - **PL/SQL** - See Using the PL/SQL Finder.
   - **Debug** - See Viewing Debug Messages for an Application.
   - **Session** - See Viewing Session State
   - **Errors** - See Viewing Application Errors.

5.16.2 Searching for Items

Search for items using the Item Finder.

In App Builder, an item can be a text field, text area, password, select list, checkbox, and so on. You can use the Item Finder to search for items within the current application or within the schema associated with the workspace.

To search for items:

1. Click the **Find** icon.

   The Items Finder appears.

   A navigation bar displays in the center the page (see previous illustration) and contains the following controls:

   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
   - **Go button** - Executes a search.
   - **Actions menu** - Clicking the icon displays the Actions menu. Use this menu to customize the display.
2. To edit a specific item, navigate to the appropriate item.

> **See Also:**
>
> Understanding Page Items

5.16.3 Searching for Pages

Search for pages using the Page Finder.

A page is the basic building block of an application. You can use the Pages Finder to search for pages within the current application or within the schema associated with the workspace.

To search for a page:

1. Click the **Find** icon.
2. Click the **Pages** tab.

   The Pages Finder appears. A search bar displays at the top of the page.
3. To link to the Items page, click the page number.

5.16.4 Searching for Queries

Search for queries using the Query Finder.

You can use the Queries Finder to locate a query within your application or within the schema associated with the workspace.

To search for a query using the Queries Finder:

1. Click the **Find** icon.
2. Click the **Queries** tab.

   The Queries Finder appears. A search bar displays at the top of the page.
3. To link to the Items page, click the page number.

5.16.5 Searching for Tables

Search for tables using the Table Finder.

You can use the Tables Finder to view tables within the schema associated with the workspace.

To view tables associated within the current schema:

1. Click the **Find** icon.
2. Click the **Tables** tab.

   The Tables Finder appears. A search bar displays at the top of the page.
3. Select a table name.

   A table definition appears.

   This report displays the column names, data type, length, precision, and scale and the SQL necessary to re-create the table appears at the bottom of the page.

> **See Also:**
>
> Managing Tables in *Oracle APEX SQL Workshop Guide*

5.16.6 Using the PL/SQL Finder

Locate and view details about stored procedures, functions, and packages using the Procedure Finder.

You can use the Procedure Finder to locate and view details about stored procedures, functions, and packages associated with each object within the schema associated with the workspace.

To search for PL/SQL code in the current schema:

1. Click the **Find** icon.
2. Click the **PL/SQL** tab.

   The PL/SQL Finder appears. A search bar displays at the top of the page.
3. To view additional details, select the procedure, function, or package name.

   The procedure, package, or function name appears and additional information including the owner, source name, source type, return type, argument names, data types, and IN/OUT parameters.

> **See Also:**
>
> Managing Database Objects with Object Browser in *Oracle APEX SQL Workshop Guide*

5.16.7 Viewing Application Errors

View a report of application errors.

To view a report of application errors:

1. Click the **Find** icon.
2. Click the **Errors** tab.

   The Application Errors report appears.

5.17 Adding Application Comments

Use Application Comments to describe an application or track developers involved in the application development.

To create an application comment:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
4. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
5. Under Tasks, click **Edit Application Comment**.

   The Application Comments page appears.
6. Enter comments in the Comments field and click **Apply Changes**.

5.18 Managing Application Groups

Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.

- About Using Application Groups
  Application groups are an ideal way to filter applications on the App Builder home page
- Viewing Application Groups
  View existing application groups on the Workspace Utilities, Application Groups page.
- Creating an Application Group
  Create application groups on the Workspace Utilities, Application Groups page.
- Assigning a Application to an Application Group
  Assign an application to application group on the Manage Application Group Assignments page.

- Removing an Application from an Application Group
  Remove an application from an application group on the Manage Application Group Assignments page.
- Deleting an Application Group
  Delete an application from an application group on the Manage Assignments page.

> **See Also:**
>
> Accessing Page Specific Utilities

5.18.1 About Using Application Groups

Application groups are an ideal way to filter applications on the App Builder home page

Adding the Group Column to the App Builder Home Page

Selecting **View Report** on the App Builder home page displays applications as an interactive report. By default, the Group column is hidden.

To add the Group column to the report:

1. Click **View Report** on the App Builder home page navigation bar.
2. Click the **Actions** menu and select **Columns**.
3. In the Select Columns dialog, move **Group** to **Display in Report** and click **Apply**.

Creating a Group Filter

Create filters on App Builder home page in either View Report or View Icon mode.

To create a filter that only displays the members of a particular application group:

1. Click the **Actions** menu and select **Filter**.
2. In the Filter dialog:
   a. From Column, select **Group**. Then use the Operator and Expression lists to create filter criteria and click **Apply**.

5.18.2 Viewing Application Groups

View existing application groups on the Workspace Utilities, Application Groups page.

To view application groups:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.

   The Application Group page appears, displaying the Search bar.

   Available controls include:

   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.

- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a group by name and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **View Icons** (the default) displays each group as a large icon. To edit a group, click the appropriate icon.
- **View Report** displays each group as a line in a report with the columns: Group Name, Description, Applications, Percent of Total, and Graph. To edit a group, click the name.

4. To edit a group or see the applications assigned to a group, click the group name.

   The applications assigned to the group display on the right side of the page.

5.18.3 Creating an Application Group

Create application groups on the Workspace Utilities, Application Groups page.

To create an application group:

1. On the Workspace home page, click the **App Builder** icon
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. On the Application Group page, click **Create**.
5. Enter a name, a description (optional), and click **Create**.

5.18.4 Assigning a Application to an Application Group

Assign an application to application group on the Manage Application Group Assignments page.

> **Tip:**
>
> You can also assign an application to an application group by editing the application definition. Select the group from the Application Group list. See Editing the Application Definition.

To assign an application to a group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. On the Tasks list, click **Manage Unassigned**.

   The Manage Application Group Assignments page appears.
5. Filter the report:
   a. Application - Enter a case insensitive query for the application name, alias, or ID and click **Set**.
   b. Group - Select an application group and clicking and click **Set**.
   c. Show - Filter the display and click **Set**. Available options include **All**, **Assigned**, and **Unassigned**.

   d. Show - Select the number of applications to display and click **Set**.
6. From the New Group list, select a group.
7. Select the applications to be assigned and click **Assigned Checked**.

5.18.5 Removing an Application from an Application Group

Remove an application from an application group on the Manage Application Group Assignments page.

To remove an application from an application group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. On the Tasks list, click **Manage Assignments**.

   The Manage Application Group Assignments page appears.
5. Filter the report:
   a. Application - Enter a case insensitive query for the application name, alias, or ID and click **Set**.
   b. Group - Select an application group and click **Set**.
   c. Show - Filter the display and click **Set**. Available options include **All**, **Assigned**, and **Unassigned**.
   d. Show - Select the number of applications to display and click **Set**.
6. From the New Group list, select **Unassign**.
7. Select the pages to be unassigned and click **Assigned Checked**.

> **Tip:**
>
> You can also remove an application from an application group by editing the application definition. Select **Unassigned** from the Application Group list. See Editing the Application Definition.

5.18.6 Deleting an Application Group

Delete an application from an application group on the Manage Assignments page.

To delete an application group:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Workspace Utilities**.
3. Click **Application Groups**.
4. Assign all applications in the group to Unassign:
   a. On the Tasks list, click **Manage Assignments**.
   b. From the New Group list, select **Unassign**.
   c. Select the pages to be unassigned and click **Assigned Checked**.

5. Return to the Application Groups page.
6. Select the application group to be deleted.
7. On the Application Group page, click **Delete**.

   A confirmation page appears.
8. Confirm your request.

5.19 Using Application Utilities

Learn about the application summaries and reports available from Utilities page.

- Accessing Application Utilities
  Access the Application Utilities page by selecting the application and then clicking the **Utilities** icon.
- Application Utilities Page
  Use the Application Utilities page to access summaries and reports that enable developers to view an application from various perspectives.
- Viewing the Application Dashboard
  Use the Application Dashboard to access reports that summarize application security, templates, page types, application and page components and component utilization.
- Viewing the Recently Updated Pages Report
  Displays a report of pages recently updated by the current user.
- Viewing the Change History Report
  The Change History report displays a summary report of edits to the current applications by developer, component type, and page number
- Viewing Database Object Dependencies
  View the Database Object Dependencies report to review database objects referenced by the currently selected application.
- Accessing APEX Views
  Query various views against APEX metadata.
- Managing Application Backups
  Review backup history, perform manual backups, or restore an application from a backup.
- Using the Embedded Code Utility
  Review all embedded code used in an APEX application on one page. Embedded code is defined as code (such as SQL, PL/SQL and JavaScript) entered in the App Builder but does not include code such as PL/SQL packages or JavaScript libraries that code may reference.
- Viewing Page and Application Checksums
  View page and application checksums to compare to compare different application deployments and determine if the application or pages are different.
- Accessing Font APEX
  View the Font APEX icons library from Application Utilities page or by selecting the Appearance, Icon attibute for select components in Page Designer.
- Accessing Page Specific Utilities
  Use the links in the Page Specific Utilities region to access Cross Page Utilities and Page Groups as well as access handy utilities for regions, buttons, items, computations, validations, process, dynamic actions, and branches.

- Accessing Cross Page Utilities
  Access reports across pages within the selected application.
- Using Interactive Report Utilities
  View and delete saved interactive reports and interactive report subscriptions.

> **See Also:**
>
> Creating Custom Activity Reports Using APEX_ACTIVITY_LOG

5.19.1 Accessing Application Utilities

Access the Application Utilities page by selecting the application and then clicking the **Utilities** icon.

To access Application Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.

   The Application Utilities page appears.

> **Tip:**
>
> To learn more, see Application Utilities Page.

> **See Also:**
>
> - Accessing Cross Page Utilities
> - Accessing Page Specific Utilities

5.19.2 Application Utilities Page

Use the Application Utilities page to access summaries and reports that enable developers to view an application from various perspectives.

The following table describes the links on the Application Utilities page.

> **See Also:**
>
> Accessing Application Utilities

Link	Description	To Learn More
Application Dashboard	View a summary of application components and attributes	See: Viewing the Application Dashboard
Advisor	Performs various checks on your application, including checks for programming errors, security issues, quality assurance, and other best practice	See: Running Advisor to Check Application Integrity
Upgrade Application	Upgrade the current application to a new release. Review components eligible for upgrading to include the latest features.	See: Upgrading Oracle APEX Applications
Recently Updated Pages	View a history of page updates made to this application.	See Viewing the Recently Updated Pages Report
Attribute Dictionary	Manage item and column UI Defaults for a selected page.	See: Using the Attribute Dictionary
Change History	View a report of component updates made in this application..	See: Viewing the Change History Report
Database Object Dependencies	Scan and review database objects referenced by this application.	See: Viewing Database Object Dependencies
Debug Messages	Review debug messages generated by this application.	See: Debugging an Application
Oracle APEX Views	Query various views against APEX metadata.	See: Accessing APEX Views
Export Repository	Manage Export Files in the Export Repository.	See: Managing Export Files from the Export Repository
Manage Backups	View and manage application backups in the current workspace. Backups provide a simple way to recover applications that may have been unexpectedly modified or deleted.	See: Managing Application Backups
Embedded Code	View all embedded code (such as SQL, PL/SQL and JavaScript) from all pages in an APEX application presented in easy-to-use faceted search page.	See: Using the Embedded Code Utility
Page Specific Utilities	**Page Specific Utilities** display in a region on the right-side of the Utilities page. Use the links in the **Page Specific Utilities** region to access Cross Page Utilities and Page Groups as well as access handy utilities for managing regions, buttons, items, computations, validations, process, dynamic actions, and branches.	See: • Accessing Page Specific Utilities • Accessing Cross Page Utilities
Interactive Report Utilities	**Interactive Report Utilities** display below **Page Specific Utilites** on the right-side of the Utilities page. Click **Saved Reports** to delete saved interactive reports in the current application. Click **Subscriptions** to manage interactive report subscriptions.	See: Using Interactive Report Utilities
Application Checksum	View the application checksum to determine if the same application is deployed in different workspaces.	See Viewing the Application Checksum

Link	Description	To Learn More
Font APEX Icons	Search Font APEX icons and HTML markup.	See Accessing Font APEX
Page Checksum	View the page checksum to compare two application deployments and determine which pages are the same and which are different.	See Viewing the Page Checksum

5.19.3 Viewing the Application Dashboard

Use the Application Dashboard to access reports that summarize application security, templates, page types, application and page components and component utilization.

To view the Application Dashboard:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Application Dashboard**.

   The Application Dashboard appears and contains the following regions:

   - **Application Overview** - Lists detailed information and statistics about the current application, including the ID, name, alias, parsing schema, associated group, number of pages, and associated theme.
   - **Security** - Lists the current authentication scheme, number of public and non public pages, and the number of authorization schemes used within the current application.
   - **Templates** - Contains links to reports of templates used within the current application.
   - **Pages by Type** - Lists counts of page types included in the application, including DML Form, Dynaic Form, Global, Cards, Faceted Search, Home, Login, Navigation Page, Interactive Grid, Interactive Report, Report, Static HTML, and so on.
   - **Application Components** - Contains links to reports of application-level controls, including Application Computations, Application Items, Application Processes, Lists, Lists of Values, Plug-ins, Shortcuts, and Tabs.
   - **Page Components** - Contains links to reports of page-level controls and logic including Buttons, Branches, Dynamic Actions, Classic Report Columns, Computations, Interactive Report Columns, Items, Processes, and Regions.
6. To view a specific report, click the number link adjacent to the template, page type, application component, or page component.
7. The Application Dashboard also displays the following charts:

   - **Native Region Component Utilization** - Displays a chart of the total number of all types of region components used in the app. This is a visualization of the information provided in the Pages by Type section, the Page Components, and others.
   - **Plugin Region Utilization** - Displays a chart of the total number of plug-ins used in the app.

> **See Also:**
>
> Identifying At Risk Password Items

5.19.4 Viewing the Recently Updated Pages Report

Displays a report of pages recently updated by the current user.

To view the Recently Updated Pages report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Recently Updated Pages**.

   The Recently Updated Pages report appears.

5.19.5 Viewing the Change History Report

The Change History report displays a summary report of edits to the current applications by developer, component type, and page number

To view the Change History report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Change History**.

   The Change History report appears. Use the Since list to control the amount of time to include in your report.
6. From **Since** at the top of the page, select the amount of time to include in your report and click **Set**.

5.19.6 Viewing Database Object Dependencies

View the Database Object Dependencies report to review database objects referenced by the currently selected application.

- About the Database Object Dependencies Page
  Learn about the Database Object Dependencies page.
- Generating the Database Object Dependencies Report
  Generate the Database Object Dependencies Report from the application Utilities page.

- Understanding the Database Object Dependencies Report
  Learn about the Database Object Dependencies report.

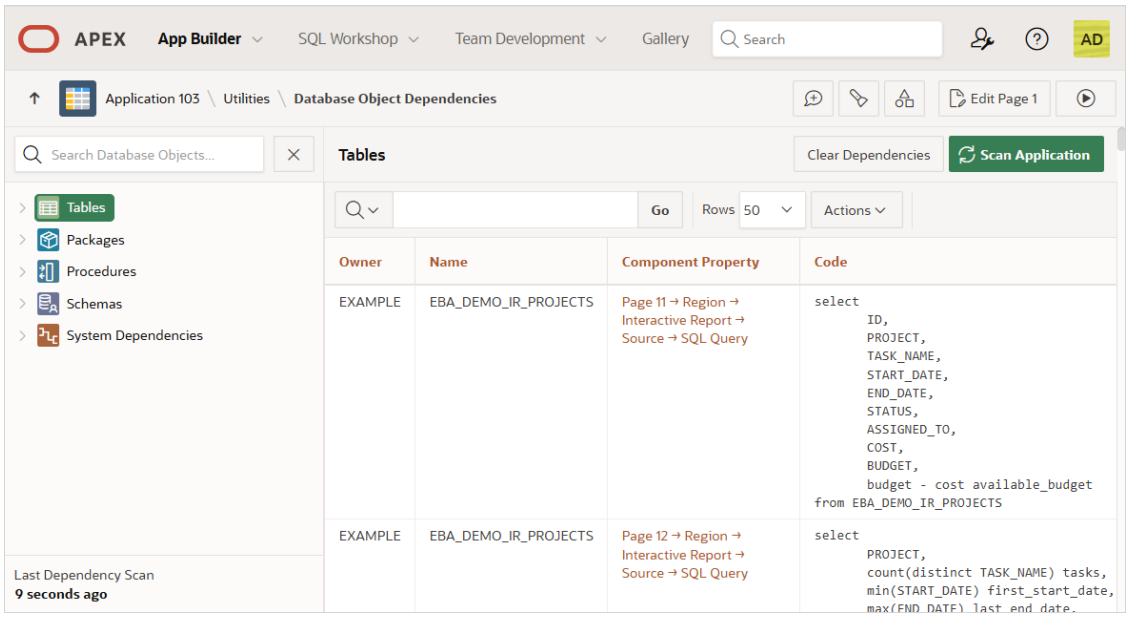
5.19.6.1 About the Database Object Dependencies Page

Learn about the Database Object Dependencies page.

Applications must to be scanned to generate the Database Object Dependencies report. Scanning the application identifies database object dependencies including the following:

- **Tables, Views and Columns** - Identify the tables, views and columns referenced by components including interactive reports, classic reports, and lists of values.
- **Packages, Procedures and Functions** Identify packages, procedures and functions referenced by processes, computations and validations.
- **System Dependencies** - Identify dependencies on Database Objects owned by system users include `SYS`, `SYSTEM`, `PUBLIC` and the APEX Schema Owner.
- **Errors** - Identify references to database objects that result in errors.

The left pane on the Database Object Dependencies page features a hierarchical tree control that enables you to visualize all the tables, views, packages, and procedures the application references as well as their components such as table columns, package procedures, and functions. To view specific objects, expand the object type. In the following example, **Tables** is selected.



5.19.6.2 Generating the Database Object Dependencies Report

Generate the Database Object Dependencies Report from the application Utilities page.

> **Tip:**
>
> To ensure up-to-date results, Oracle recommends refreshing the Database Object Dependencies report after an application or its database schema have been modified. See Rescanning the Application.

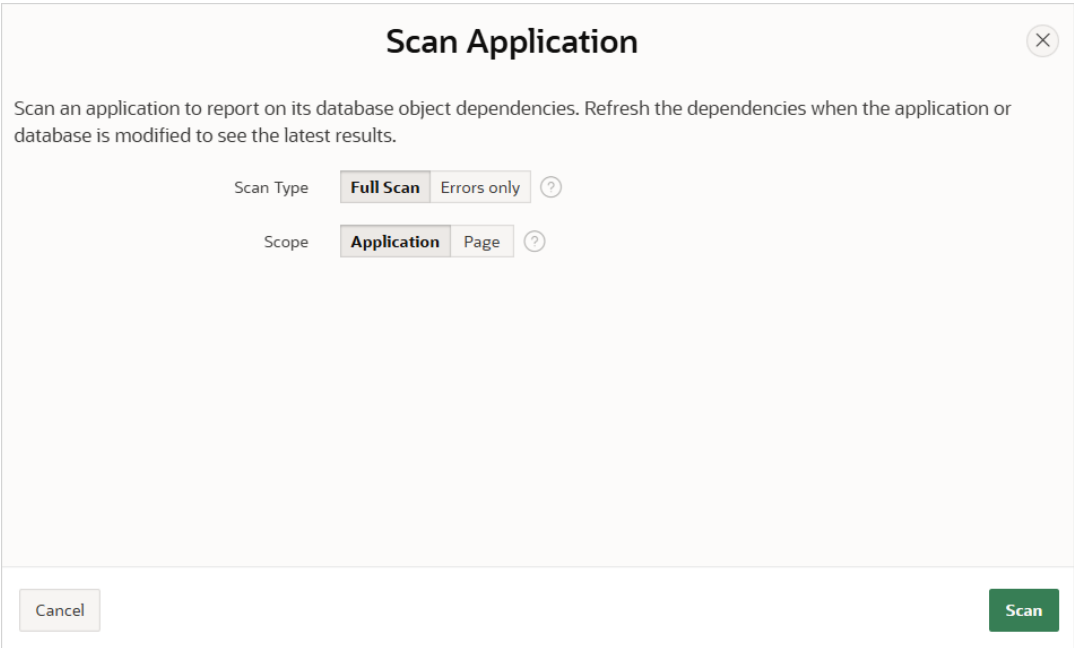
To generate the Database Object Dependencies report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. On the Utilities page, click **Database Object Dependencies**.

   The Database Object Dependencies page appears.
6. Click **Scan Application**.

   The Scan Application dialog appears.



7. On Scan Application, configure the options:
   a. Scan Type - Select an option:
      - **Full Scan** - Scan all sources for dependencies including errors.
      - **Errors Only** - Create a report on code errors only.

b. Scope - Select an option:

- **Application** - Scan the entire application including all pages and shared components.
- **Page** - Restrict the scan to a single page only.

  If you select a page, select a page from the **Page** list.

c. Click **Scan**.

The Database Object Dependencies page appears.

5.19.6.3 Understanding the Database Object Dependencies Report

Learn about the Database Object Dependencies report.

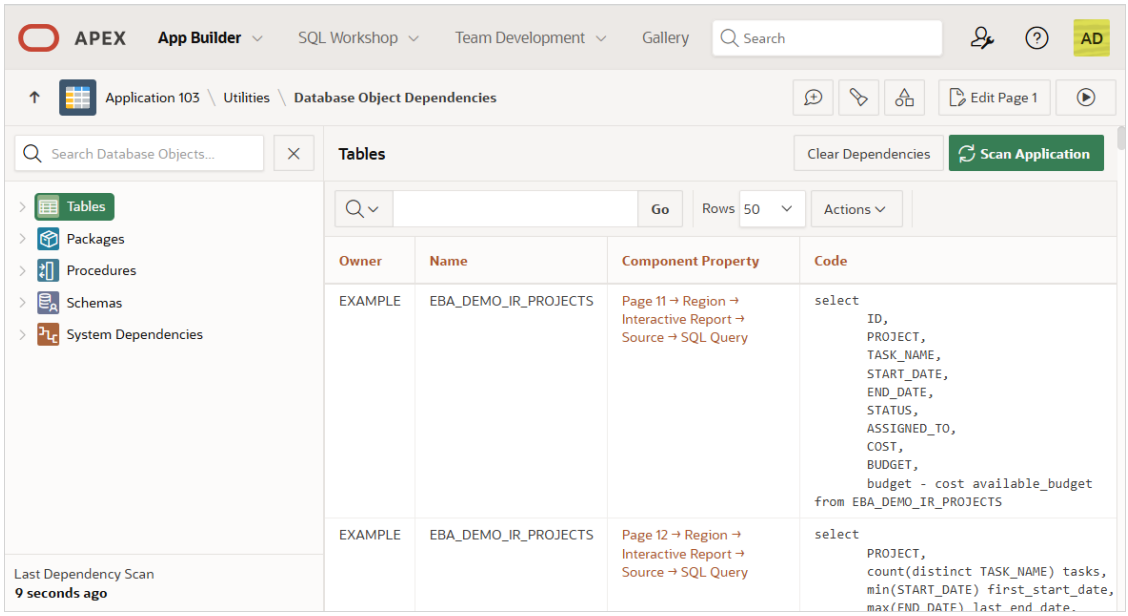
- Using the Database Object Dependencies Page
  Learn about using Database Object Dependencies page.
- Viewing Errors
  Learn about viewing errors on Database Object Dependencies page.
- Rescanning the Application
  Refresh the dependencies on the Database Object Dependencies page by rescanning the application.
- Clearing Object Dependencies
  Remove dependency records from the Database Object Dependencies page by clicking Clear Dependencies.

5.19.6.3.1 Using the Database Object Dependencies Page

Learn about using Database Object Dependencies page.

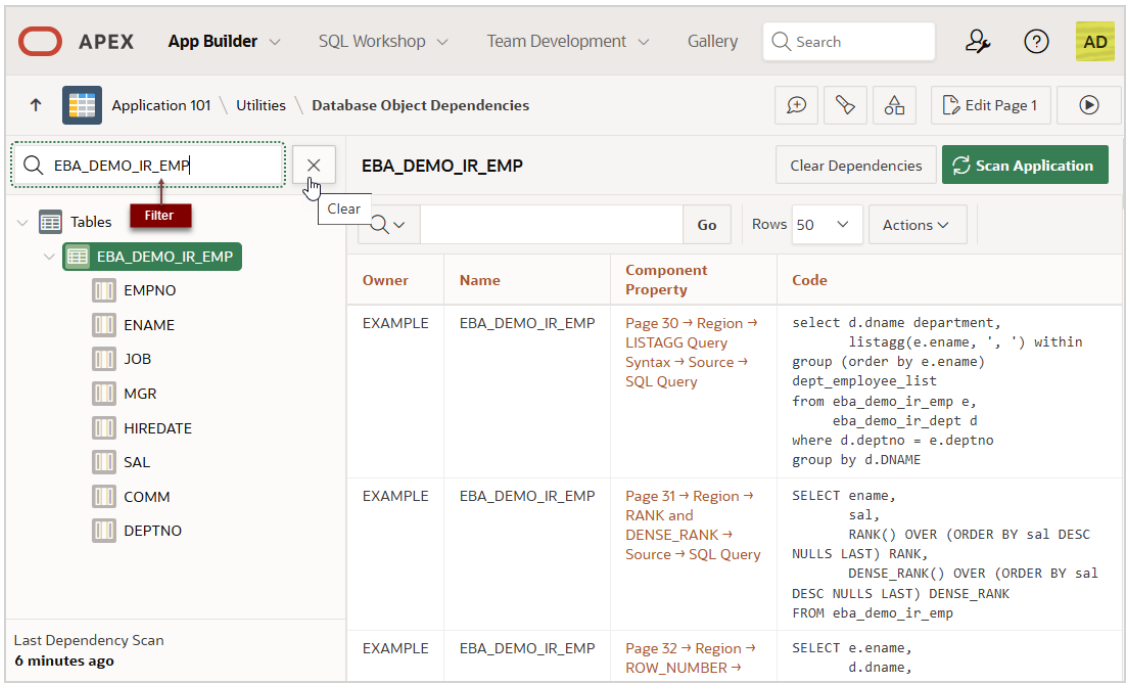
The Database Object Dependencies page features two panes:

- **Tree** - The left pane displays as a tree of database object types, such as Tables, Packages, Procedures, Schemas, and System Dependencies. To view specific objects, expand the object type. For example, expanding the Tables node displays a list of tables. Expanding a table displays specific columns.
- **Report** - The right pane displays a report based on the selected node in the tree. In the following example **Tables** is selected. The report displays four columns: **Owner**, **Name**, **Component Property**, and **Code**. The **Component Property** column displays as link. Click the link to navigate to the component.



Filtering Objects

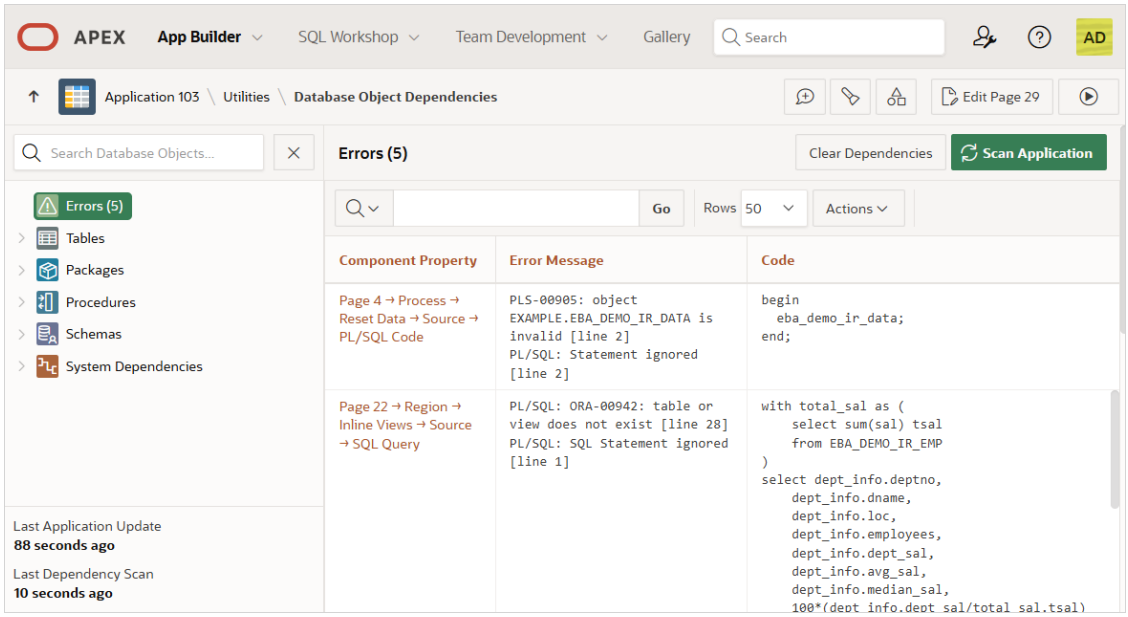
Filter objects by entering a search term in the filter that displays above the tree. In the following example, the search term `EBA_DEMO_IR_EMP` displays in the filter and the report displays the `EBA_DEMO_IR_EMP` table. Filters you create are persistent and are preserved within the APEX session.



5.19.6.3.2 Viewing Errors

Learn about viewing errors on Database Object Dependencies page.

If an application contains code or object references that cause errors, an **Errors** node displays in the tree. The following example shows an application with five errors. The report in the right pane displays three columns: **Component Property**, **Error Message**, and **Code**. **Component Property** displays as link. Click the link to navigate to the component.



5.19.6.3.3 Rescanning the Application

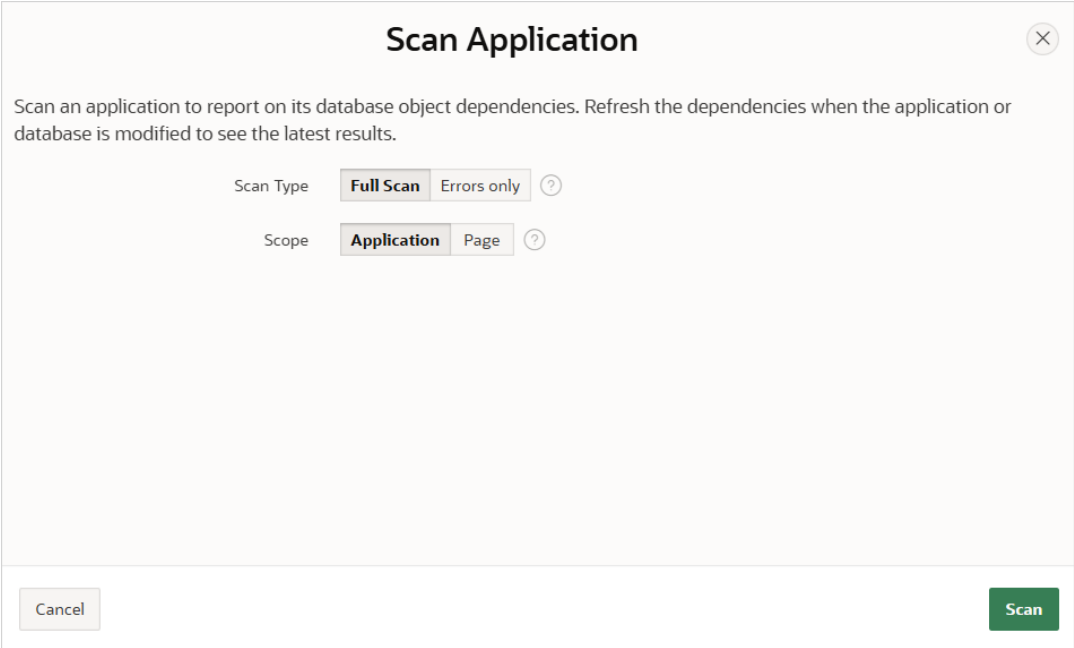
Refresh the dependencies on the Database Object Dependencies page by rescanning the application.

If the application or database has been modified, click **Scan Application** on the Data Object Dependencies page to scan the application again and refresh the dependencies.

To rescan an application:

1. View an existing Data Object Dependencies page:
   a. Navigate to the Workspace home page.
   b. Select the application.
   c. Click **Utilities**.
   d. On the Utilities page, click **Database Object Dependencies**.
2. On the Database Object Dependencies page, click **Scan Application**.

   The Scan Application dialog appears.



3. On Scan Application, configure the options:
   a. Scan Type:
      - **Full Scan** - Scan all sources for dependencies including errors.
      - **Errors Only** - Create a report on code errors only.
   b. Scope:
      - **Application** - Scan the entire application including all pages and shared components.
      - **Page** - Restrict the scan to a single page only.

        If you select a page, select a page from the **Page** list.
   c. Click **Scan**.

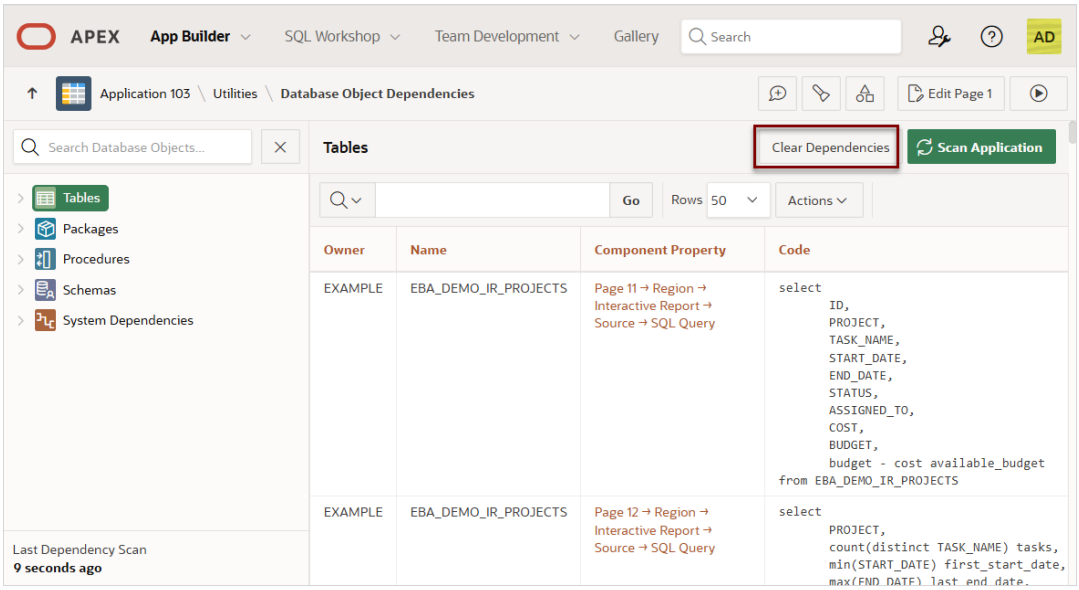
   The Database Object Dependencies page is refreshed.

5.19.6.3.4 Clearing Object Dependencies

Remove dependency records from the Database Object Dependencies page by clicking Clear Dependencies.

To clear dependency records:

1. View an existing Data Object Dependencies page:
   a. Navigate to the Workspace home page.
   b. Select the application.
   c. Click **Utilities**.
   d. On the Utilities page, click **Database Object Dependencies**
2. On the Database Object Dependencies page, click **Clear Dependencies**.



A dialog appears confirming your request.

3. Click **Clear Dependences** again..

   The Application home page appears.

4. On Scan Application, configure the options:

   a. Scan Type - Select an option:

      - **Full Scan** - Scan all sources for dependencies including errors.
      - **Errors Only** - Create a report on code errors only.

   b. Scope - Select an option:

      - **Application** - Scan the entire application including all pages and shared components.
      - **Page** - Restrict the scan to a single page only.

        If you select a page, select a page from the **Page** list.

   c. Click **Scan**.

   The Database Object Dependencies page appears.

5.19.7 Accessing APEX Views

Query various views against APEX metadata.

APEX Views are data dictionary views that expose the metadata for applications. In APEX Views, you cannot only see the view, but you can also see the data they contain.

To access the APEX Views page:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.

4. Click **Utilities**.

5. Click **Oracle APEX Views**.

   The APEX Views report appears. Use the Search bar at the top of the page to customize the display.

   > **Tip:**
   >
   > Click **Tree View** to switch to a relational view of this information with expandable parent and children entries. The list of APEX dictionary views is also available by querying the `APEX_DICTIONARY` view.

6. Select a view.
7. On Selected Columns, configure the columns.
   a. From **Select Columns** use the controls to add and remove columns or change the column order.
   b. Click the **Results** button.

   The Results report appears.
8. To view the executed query, expand the **Query** region.

   You can also copy this query for reuse with SQL Commands or SQL Developer.
9. To download the Results, click the **Download** link.

5.19.8 Managing Application Backups

Review backup history, perform manual backups, or restore an application from a backup.

- About Application Backups
  Backups provide a simple way for developers to automatically recover applications that may have been unexpectedly modified or deleted.
- Viewing All Application Backups
  View a summary of all application backups on the Manage Backups page.
- Managing Backups for a Specific Application
  Manage Backups for a specific application on the Manage Backup page in application Utilities.

5.19.8.1 About Application Backups

Backups provide a simple way for developers to automatically recover applications that may have been unexpectedly modified or deleted.

Oracle APEX backs up modified applications as a part of daily maintenance. Backups are automatic and do not require an administrator or developer involvement. However, applications are not backed up if they meet the following criteria:

- If no changes have been made to the application since the last backup, then the application is not backed up again.
- Run-time only applications are not be backed up.
- Applications that are installed into runtime-only instances are not backed up.

About Configuring the Number of Backups

By default, APEX only retains the 25 most recent backups. Older backups are deleted to make room for new ones.

Instance administrators can configure the number of backups per application, or disable automatic backups all together. To configure the number of backups for each application, sign in to the Administration Services application and navigate to **Manage Instance**, **Feature Configuration**, **Application Development**, and update the **Number of backups per application** attribute. Valid values are between 0 and 30. A value of zero (0) disables the backup feature.

> **See Also:**
>
> Configuring the Number of Application Backups in *Oracle APEX Administration Guide*

Viewing the Backup Log

Instance administrators monitor the performance of the backup job in the Backup Log. To view the Backup Log, sign in to the Administration Services application and navigate to **Monitor Activity**, **Backup Log**. The Backup Log reports when a backup job ran, the number of applications backed up by the job, and the elapsed time required to backup the applications.

> **See Also:**
>
> Viewing the Backup Log in *Oracle APEX Administration Guide*

Managing Backups

Developers can view all backups in App Builder in **Workspace Utilities**, **Manage Backups**. To view backups for a specific application, click the application name.

5.19.8.2 Viewing All Application Backups

View a summary of all application backups on the Manage Backups page.

App Builder automatically backs up modified applications on a daily basis.

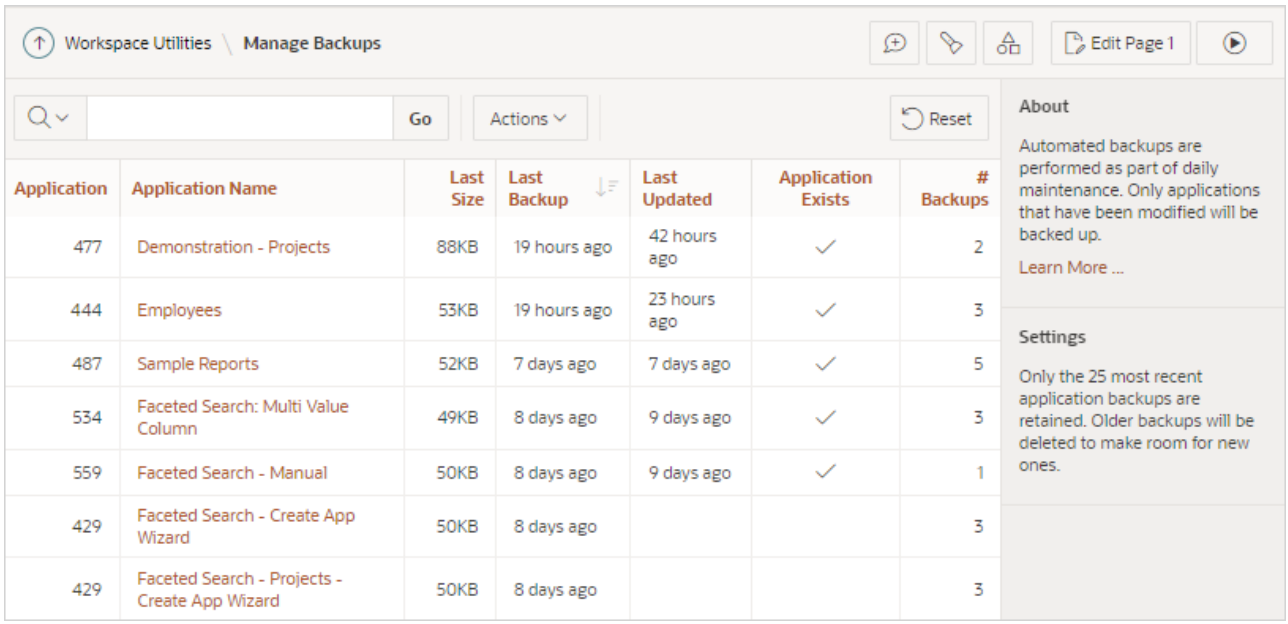
To view the Manage Backups page:

1. Navigate to the Workspace Utilities page:
   a. Sign in to Oracle APEX.
   b. On the Workspace home page, click the **App Builder** icon.
   c. Click the **Workspace Utilities** icon.
2. Click **Manage Backups**.

   The Manage Backups page appears displaying the following columns:

   - Application - Displays the Application ID
   - Application Name - Displays the application name when the backup occurred. If the name has changed, both names display in the report.

- Last Size
- Last Updated
- Application Exists
- # Backup - Displays the number of backups.



3. To view backup details for a specific application, click the Application Name.

   The Manage Backups page for the application appears.

> **Tip:**
>
> You can also access the Manage Backups page from the App Builder home page. Select **Manage Backups** from the Tasks region.

5.19.8.3 Managing Backups for a Specific Application

Manage Backups for a specific application on the Manage Backup page in application Utilities.

> **Note:**
>
> You can also access the backups for a specific application from the Application home page. Navigate to the application and select **Manage Backups** from the Tasks list.

To manage backups for a specific application:

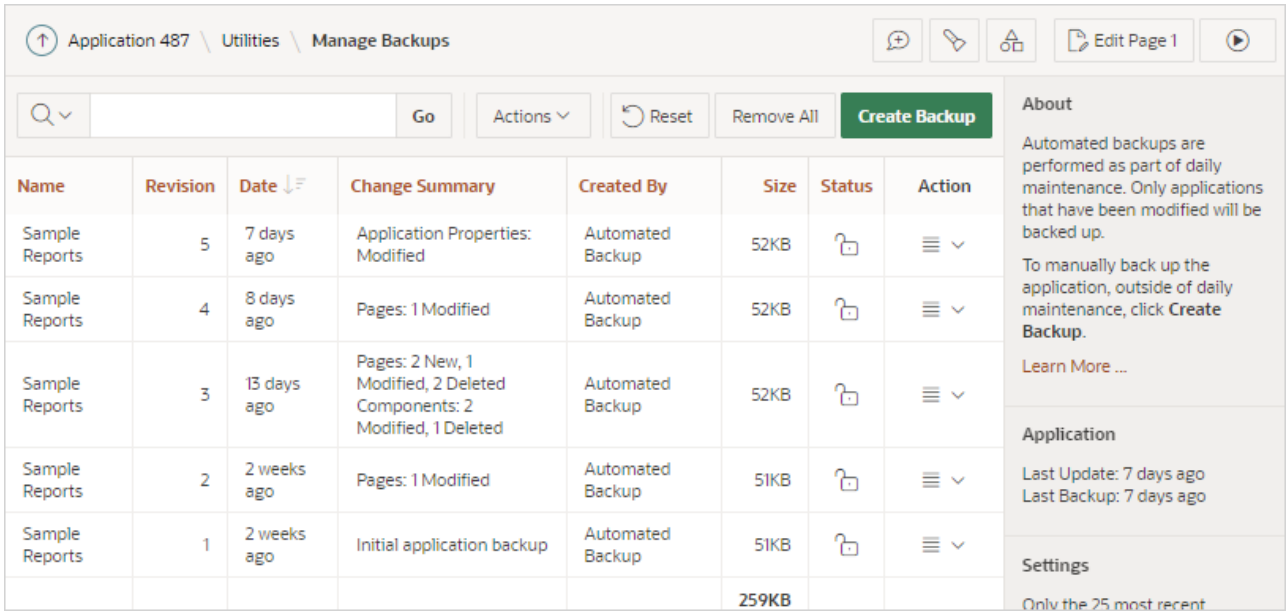
1. Navigate to the Manage Backups page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Click the **Workspace Utilities** icon.
   c. Click **Manage Backups**.

The Manage Backups page appears.

2. To view backups for a specific application, click the Application Name.

   The Manage Backups page appears displaying the following columns:

   - Name - Application name at the time of backup.
   - Revision - Revision number.
   - Date - Backup date.
   - Change Summary - Indicates the number of new, modified, or deleted pages and components since the last backup.
   - Created By - Indicates whether the back up was automated or manual. If manual, the user who created the backup displays.
   - Size - Backup size.
   - Status - Generally displays a Lock or Unlock status icon. Locking a backup prevents it from being removed if the backup limit is reached. If a developer restores an application, then a Restore icon appears.
   - Action - Menu options include: Restore, Download, View Details, Manage Lock, or Delete. These options are discussed in detail in the steps that follow.



3. To restore a backup:

   a. Locate the application and click the **Actions** menu.

   b. From the Actions menu, select **Restore**.

   c. In the Restore Application dialog:

      i. Install As Application - Select one of the following:

         - **Restore Application ID** - The existing application ID will be used. This option overwrites the existing application without changing the ID.
         - **Auto Assign New Application ID** - A new application ID will be created for you.
         - **Change Application ID** - You specify a new application ID.

   ii. Click **Restore Application**.

4. To download a backup:
   a. Locate the application and click the **Actions** menu.
   b. From the Actions menu, select **Download**.
   c. Import the Export Files.
5. To view backup details:
   a. Locate the application and click the **Actions** menu.
   b. From the Actions menu, select **View Details**.
6. To manage backup locks:
   a. Locate the application and click the **Actions** menu.
   b. From the Actions menu, select **Manage Lock**.

      Locking a backup prevents it from being deleted during daily maintenance.
   c. In the Manage Backup Lock dialog, select **Unlock Backup** or **Lock Backup**.
7. To delete a backup:
   a. Locate the application and click the **Actions** menu.
   b. From the Actions menu, select **Delete**.
   c. In the Delete Backup dialog, select **Delete Backup**.
8. To backup an application manually:
   a. Click **Create Backup**.
   b. In the Create a Backup dialog, enter comments and click **Backup**.
9. To remove all backups for an application:
   a. Click **Remove All**.
   b. In the Remove All Backups dialog, confirm your selection by clicking **Remove All Backups**.

> **See Also:**
>
> Importing an Application Export

5.19.9 Using the Embedded Code Utility

Review all embedded code used in an APEX application on one page. Embedded code is defined as code (such as SQL, PL/SQL and JavaScript) entered in the App Builder but does not include code such as PL/SQL packages or JavaScript libraries that code may reference.

The Embedded Code Utility displays all embedded code from all pages in an application in an easy-to-use faceted search page. Using the Embedded Code Utility is ideal for tasks such as code reviews, security evaluations, and application turning where historically developers would need to click on every property in a page or Shared Component.

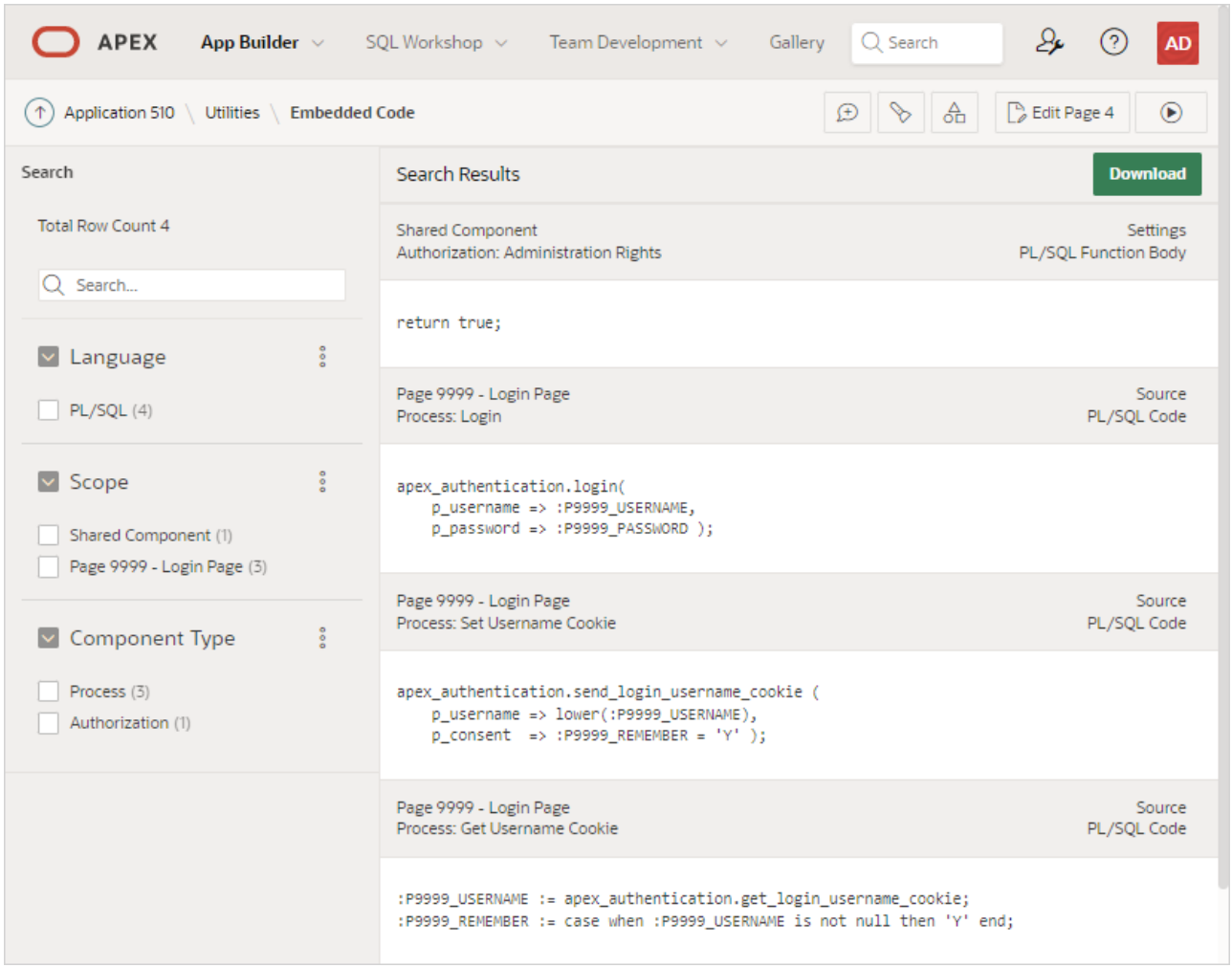
To access the Embedded Code Utility:

1. Navigate to the Workspace home page.

2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Embedded Code**.

   The Embedded Code page appears. Note that large applications may take longer to display.



The left region contains facets that enable you to filter the Search Results report. The values of the facets depend upon the embedded code in the application. The left region contains the following facet filters:

- **Language** - Identifies the programming language of the embedded code. Currently, SQL, PL/SQL and JavaScript are supported.

  Only those languages that are included in the application appear. For example, if an application uses SQL and PL/SQL, but not JavaScript, then only the first two options will appear as filters. Selecting a filter refreshes the report and only displays the code that matches the languages selected.

- **Scope** - Identifies where in the application the embedded code is used.

  Shared Component refers to any application component that is not part of a page such as application processes, authentication schemes, lists of values and so on. All other values refer to a specific page of an application. Selecting a page limits the results to only the embedded code for that page.

- **Component Type** - Identifies the type of component that contains the embedded code.

  Only those component types that contain embedded code display in the filter. For example, a classic report that uses a SQL query results in a **Region** component type to be included. Similarly, if an application does not have any buttons, then a **Button** filter does not appear. Note that not all conditions require embedded code.

A Search Results report displays in the right region and includes identifying information about the component as well as where the embedded code is used by the component. For example, a page item may have a Source query used to populate the item's value, and it can also have a Server-side Condition to determine if the item should be included on the page.

6. Click **Download** to create a ZIP archive containing files for each of the components by language type.

   The directory hierarchy of the files is similar to what is created when you split export files into separate scripts. The difference is that all files will be located below the `embedded_code` directory as opposed to the `application` directory.

5.19.10 Viewing Page and Application Checksums

View page and application checksums to compare to compare different application deployments and determine if the application or pages are different.

- Viewing the Application Checksum
  View the application checksum to compare if the same application is deployed in different workspaces.
- Viewing the Page Checksum
  View the page checksum to compare two application deployments and determine which pages are the same and which are different.

5.19.10.1 Viewing the Application Checksum

View the application checksum to compare if the same application is deployed in different workspaces.

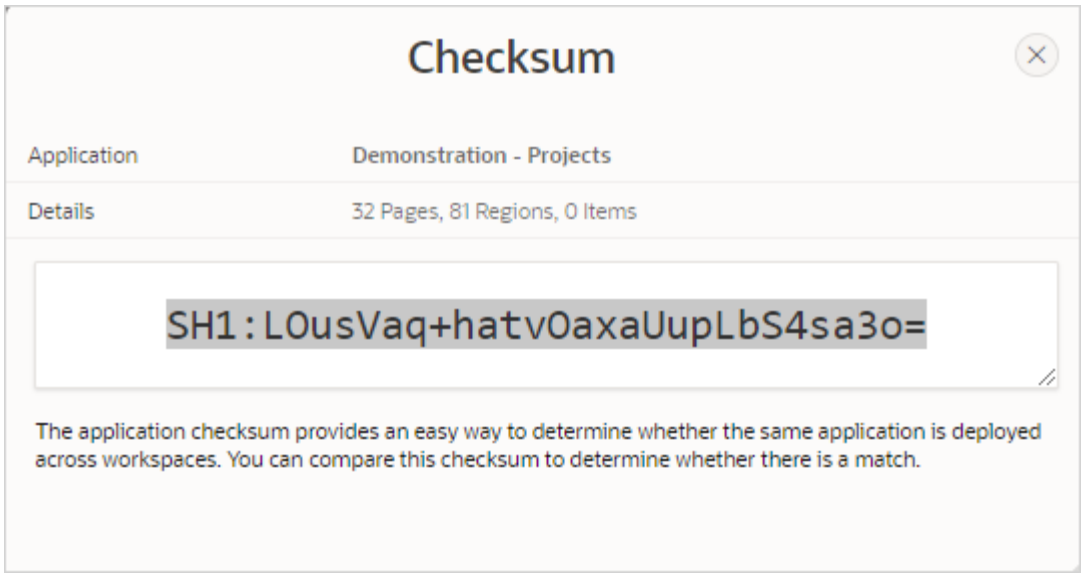
The application checksum is a unique identifier consisting of number and character strings. View the application checksum to verify if the same application is deployed across workspaces.

To view the application checksum:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Application Checksum**.

The Checksum window appears displaying the application name and details such as the number of pages, regions, or items. The application checksum displays in the center of the window.



6. Copy the checksum.

5.19.10.2 Viewing the Page Checksum

View the page checksum to compare two application deployments and determine which pages are the same and which are different.

A page checksum is a unique identifier consisting of number and character strings. Use the page checksum to verify if the same page is deployed across workspaces.

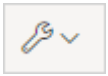
To view the page checksum:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Page Checksum**.

   The Page Checksum Report appears and includes the following columns: Name, Page (number), Alias, Checksum, Updated By, Group, and Updated.
6. To view the page checksum in Page Designer:
   a. From the Page Checksum Report, click the page name.

      Page Designer appears.
   b. From the Page Designer toolbar, click the **Utilities** menu (

) and select **Checksum**.

The Checksum window appears displaying the page name and details such as the number of regions, items, or buttons. The checksum displays in the center of the window.

c. Copy the checksum.

5.19.11 Accessing Font APEX

View the Font APEX icons library from Application Utilities page or by selecting the Appearance, Icon attibute for select components in Page Designer.

- About Font APEX Icons
  Font APEX is the icon library for Oracle APEX and the Universal Theme.
- Browsing Icons from Application Utilities
  You can search and filter icons on the Font APEX page in Utilites.
- Browsing Font Icons from Page Designer
  For select page components, browse Font APEX icons in Page Designer by configuring the Appearance, Icon attribute.

5.19.11.1 About Font APEX Icons

Font APEX is the icon library for Oracle APEX and the Universal Theme.

Small icons are based on a 16x16 grid and are ideally suited for buttons and menus. Large icons are based on a 32x32 grid and are well suited for places where you need to provide a larger graphic, such as cards, media lists, or hero regions.

Many APEX components automatically use the large or small icons based on the context (such as Template Options). For example, when using Cards, the Block and Featured template option automatically shows the larger versions of the icons. In contrast, the Basic and Compact template options use icons from the smaller set. Font APEX contains over 1150 icons. You can choose from 25 modifiers that you can place on top of any existing icons so you can customize any icon to fit your needs.

5.19.11.2 Browsing Icons from Application Utilities

You can search and filter icons on the Font APEX page in Utilites.

Once you find the appropriate icon, you can customize the icon by configuring attributes. As you make selections, the icon image changes and HTML and Icon attributes are updated. You can then copy the updated HTML Markup and icon property for use in your applications.

To view Font APEX icons:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

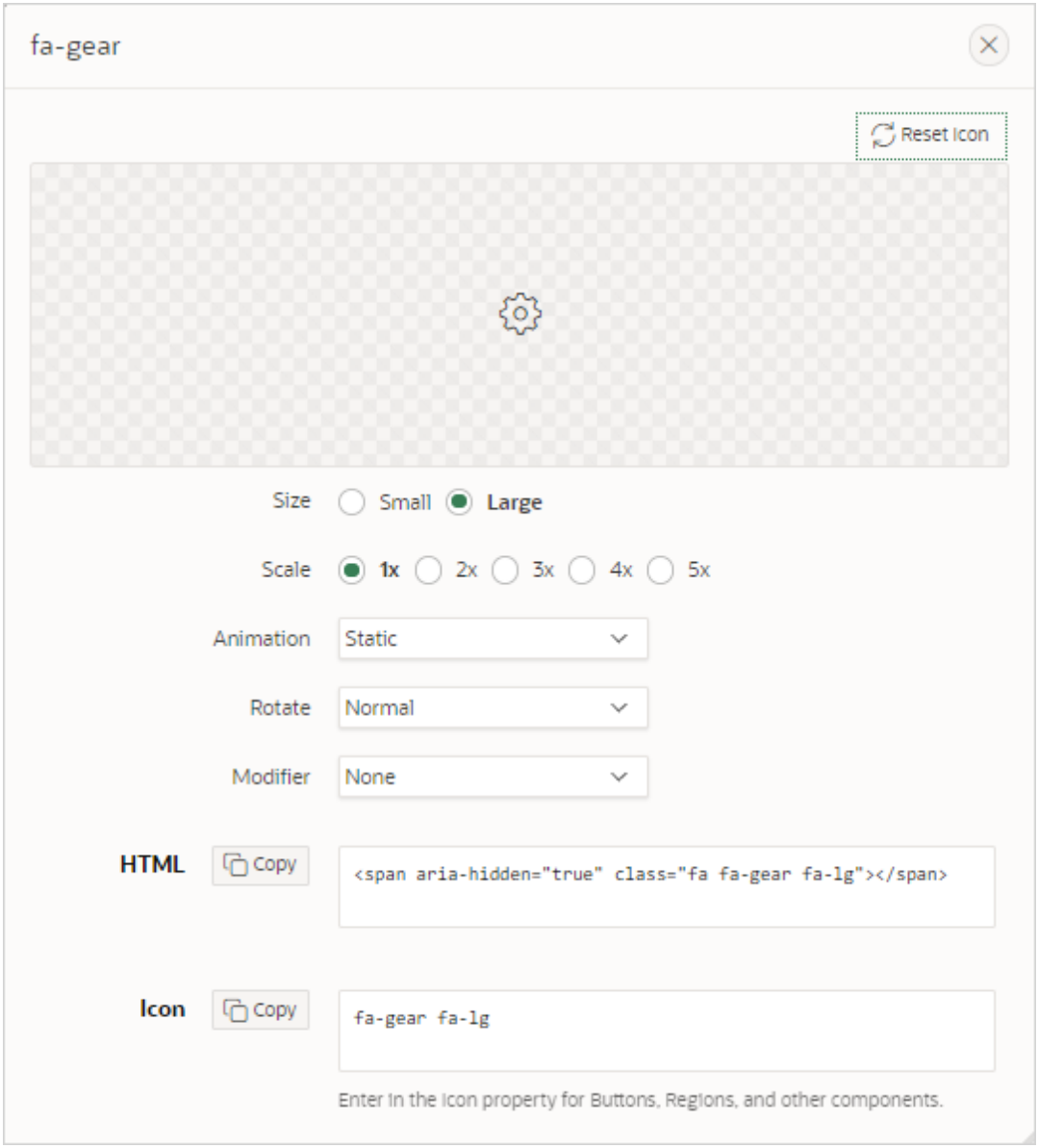
   The Application home page appears.
4. Click **Utilities**.
5. Click **Font APEX Icons**.

   The Font APEX page appear.
6. Use the Search region on the right side of the page to narrow the display.
   - Choose **Small** or **Large**

- In the Search field, enter keywords and press **Enter**.
- Select a Category.

7. Under Icons, select an icon.

   The following example shows **fa-gear**.



8. Configure the attributes:
   - Size
   - Scale
   - Animation
   - Rotate
   - Modifier

   As you configure attributes, the icon image changes and the HTML and Icon attributes are updated to reflect your selections.

9. HTML - Displays the HTML Markup. Click **Copy** to copy the HTML Markup.

10. Icon - Displays the icon property for buttons, regions, and other components. Click **Copy** to copy the icon..
11. Click **Reset Icon** in the upper right corner to revert the attributes to the default settings.

5.19.11.3 Browsing Font Icons from Page Designer

For select page components, browse Font APEX icons in Page Designer by configuring the Appearance, Icon attribute.

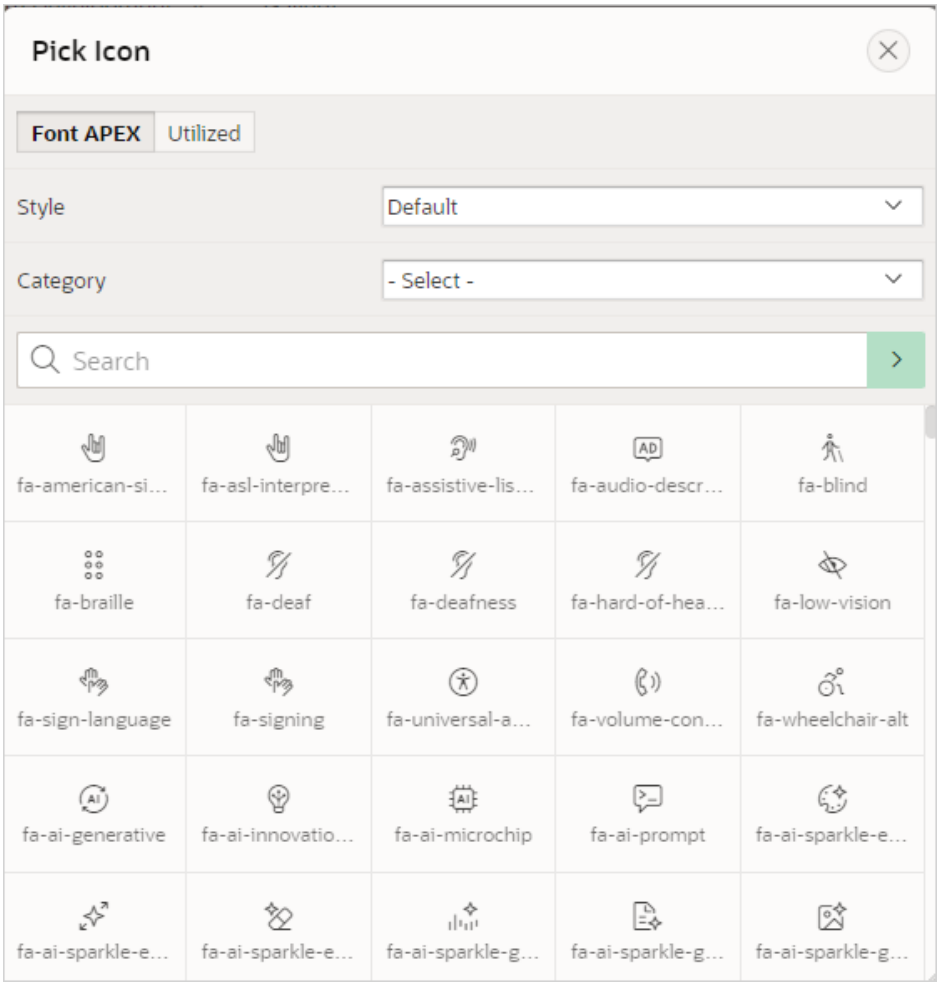
The example that follows demonstrates how to select a Font APEX icon for a Static Content region.

To browse Font APEX icons for a Static Content region:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create a Static Content region:
   a. Right-click an existing container to display the context menu.
   b. From the context menu, select **Create Region**.
3. In the Property Editor, edit the appropriate attributes:
   a. Identification, Name - Enter a region name. The region name is required, it is used by Page Designer to identify the region on the tree view and the grid layout view. If the Title attribute is empty, the region Name is used as the runtime region title.
   b. Identification, Title - Enter a region title. The region title is optional and only displays when it is defined in the region template.
   c. Identification, Type - Specify how the region is rendered. Select **Static Content**.
   d. Select an icon from the Pick Icon dialog:
      - Appearance, Icon - Click the **List of Values** icon.

        The Pick Icon dialog appears.



- Font APEX - Select the Style, Category, and icon.
- Utilized - View currently selected icons.

e. Edit any remaining attributes. The attributes that display depend upon the region Type.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Click **Save** or **Save and Run Page**.

5.19.12 Accessing Page Specific Utilities

Use the links in the Page Specific Utilities region to access Cross Page Utilities and Page Groups as well as access handy utilities for regions, buttons, items, computations, validations, process, dynamic actions, and branches.

To access Page Specific Utilities page:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.

   The App Builder home page appears
3. Select an application.

   The Application home page appears.

4. Click **Utilities**.
5. Under **Page Specific Utilities**, select one of the following:
   - Cross Page Utilities
   - Page Groups
   - Region Utilities
   - Button Utilities
   - Item Utilities
   - Computations Utilities
   - Validation Utilities
   - Process Utilities
   - Dynamic Action Utilities
   - Branch Utilities

> **See Also:**
>
> Accessing Cross Page Utilities

5.19.13 Accessing Cross Page Utilities

Access reports across pages within the selected application.

To access the Cross Page Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.

   The Cross Page Utilities page provides access to the following reports:

   - **Delete Multiple Pages**. Delete multiple pages at once.
   - **Page Attributes**. View a report of details for all pages in the current application including the page name, page title, defined groups, type and number of components and controls, and application logic. To customize the report, use the search bar at the top of the page. To link to a specific page, click the page number.
   - **Page Locks**. Use Page Locks to prevent conflicts during application development. By locking a page, you prevent other developers from editing it.

> **See Also:**
>
> - Deleting Multiple Pages
> - Locking and Unlocking a Page

5.19.14 Using Interactive Report Utilities

View and delete saved interactive reports and interactive report subscriptions.

- Deleting Saved Interactive Reports
  View and delete saved interactive reports in the current application.
- Viewing Subscriptions
  View and delete interactive report subscriptions.

5.19.14.1 Deleting Saved Interactive Reports

View and delete saved interactive reports in the current application.

To delete saved interactive reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. Click **Utilities**.
4. Under Interactive Report Utilities, select **Saved Reports**.

   The report displays all of the saved report for the application which are not primary defaults. Each row includes the page number, region name, report type, report name, report alias, and owner.
5. To view private saved reports, deselect the checkbox for the **Primary Default** filter located below the search bar.
6. To delete a report, select the checkbox adjacent to the page number and click **Delete Checked**.
7. To view the page containing the report, click the page number.

> **See Also:**
>
> - Linking to Shared Interactive Reports
> - Saving Interactive Reports
> - Managing Saved Interactive Reports in *Oracle APEX Administration Guide*

5.19.14.2 Viewing Subscriptions

View and delete interactive report subscriptions.

End users can receive updated versions of a report by subscribing to it. Workspace administrators can view and manage these subscriptions on the Subscriptions page.

To manage interactive report subscriptions:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

The Application home page appears.

3. Click **Utilities**.
4. Under Interactive Report Utilities, select **Subscriptions**.

   The Subscriptions report appears.
5. To delete a subscriptions, select the checkbox adjacent to the page number and click **Delete Checked**.

5.20 Using Workspace Utilities

Use the Workspace Utilities page to acccess REST Enabled SQL Services, Generative AI Services, Remote Servers, Application Groups, Web Credentials, Workspace Themes, Export, Oracle APEX Views, Manage Backups, and REST Source Catalogs.

- Accessing Workspace Utilities
  Access Workspace Utilities by clicking the Workspace Utilities icon in the center of the {Conref}App Builder home page.
- Workspace Utilities Page
  The Workspace Utilities page contains links to common workspace utilities including REST Enabled SQL Services, Remote Servers, Application Groups, Web Credentials, Workspace Themes, Export, Oracle APEX Views, Manage Backups, and REST Source Catalogs.
- Managing Vector Providers
  Utilize Vector Providers to create vector embeddings for a text string.
- Viewing Cross Application Reports
  Use the links under Cross Application Reports page to view and edit information across multiple applications in the current workspace.

5.20.1 Accessing Workspace Utilities

Access Workspace Utilities by clicking the Workspace Utilities icon in the center of the {Conref}App Builder home page.

To access Workspace Utilities:

1. Sign in to Oracle APEX.
2. On the Workspace home page, click the **App Builder** icon.

   The App Builder home page appears.
3. On the App Builder home pack, click the **Workspace Utilities** icon.

   The Workspace Utilities page appears.

> **See Also:**
>
> Workspace Utilities Page

5.20.2 Workspace Utilities Page

The Workspace Utilities page contains links to common workspace utilities including REST Enabled SQL Services, Remote Servers, Application Groups, Web Credentials, Workspace Themes, Export, Oracle APEX Views, Manage Backups, and REST Source Catalogs.

The following table describes the links on the Workspace Utilities page.

> **See Also:**
>
> Accessing Workspace Utilities

Link	Description	To Learn More
REST Enabled SQL Services	Manage references to external REST Enabled SQL services.	See: Managing REST Enabled SQL References
Generative AI	Manage Generative AI Services for APEX Applications and App Builder.	See: Managing Generative AI Services
Remote Servers	Manage Remote Server objects used in REST Data Sources and Authentications.	See: Managing Remote Servers
Vector Providers	Vector providers are configured to convert text into an embedding.	See: Managing Vector Providers
Application Groups	Organize applications by assigning them to application groups. To use application groups, first create a group and then assign applications to it.	See: Managing Application Groups
Web Credentials	Manage secure credentials to connect to REST Enabled SQL or other REST services.	See: Managing Web Credentials
Workspace Themes	Manage workspace themes that can be utilized by any application within the workspace.	See: Managing Workspace Themes
Export	Export workspace components.	See: Exporting Workspaces, Applications, and App Components
Oracle APEX Views	Query the various views against APEX meta data.	See: Accessing APEX Views
Manage Backups	View and manage application backups in the current workspace. Backups provide a simple way to recover applications that may have been unexpectedly modified or deleted.	See: Managing Application Backups
REST Source Catalogs	Manage catalogs of REST Sources for integration into applications.	See: Managing REST Source Catalogs
OpenTelemetry	Configure OpenTelemetry, a standard for collecting user experience data including page load times and user interactions in web applications.	See: OpenTelemetry
Cross Application Reports	Cross Application Reports display on the side bar on the Workspace Utilities page. Access these reports to view and edit information across multiple applications in the current workspace.	See: Viewing Cross Application Reports

Link	Description	To Learn More
Tasks, Data Dictionary Cache	View statistics and data analysis of schema tables and caches in a workspace	See: Viewing and Refreshing the Data Dictionary Cache for a Workspace in *Oracle APEX Administration Guide*

5.20.3 Managing Vector Providers

Utilize Vector Providers to create vector embeddings for a text string.

> **Note:**
>
> Vector Providers requires Oracle Database 23ai or later.

- About Vector Providers
  Learn about Vector Providers.
- Creating a Vector Provider
  Create a Vector Provider on the Vector Providers page in Workspace Utilities.
- Editing and Deleting a Vector Provider
  Edit or delete Vector Provides on the Vector Providers Detail page in Workspace Utilities.
- Viewing the Vector Providers History Report
  View a report of recent modifications made to Vector Provider in this workspace.

5.20.3.1 About Vector Providers

Learn about Vector Providers.

Starting with Oracle Database 23ai, you can use the `VECTOR` data type to store vector embeddings alongside business data in the database. Using embedding models, you can transform unstructured data into vector embeddings that can then be used for semantic queries on business data.

About Vector Providers

A Vector provider enables you to convert text into an embedding. The functionality can be covered by an AI Service, but also locally by an ONNX model in the database or a user-defined PL/SQL function.

Vector Providers are stored at the workspace-level and available in all applications. When you export an application, used Vector Providers are added to the export file. When you import the application into another workspace, Oracle APEX checks whether the target workspace already contains Vector Providers with the same static ID. If a Vector Provider already exists, the application uses it. Otherwise the Vector Provider from the import file are created in the target workspace.

Retrieving and Using Embeddings

You can retrieve the embedding from a Vector Provider using the `GET_VECTOR_EMBEDDINGS` functions in `APEX_AI` package. To learn more, see APEX_AI in *Oracle APEX API Reference*.

Once your retrieve the embedding, you can create a Oracle Vector Search configuration based on your Vector Provider. To learn more, see Creating an Oracle Vector Search.

5.20.3.2 Creating a Vector Provider

Create a Vector Provider on the Vector Providers page in Workspace Utilities.

> **Note:**
>
> Vector Providers requires Oracle Database 23ai or later.

To create a Vector Provider:

1. Navigate to the Vector Provider page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **Vector Providers**.

      The Vector Providers page appears.
2. Click **Create**.

   The Create/Edit page appears.
3. Under Identification:
   a. Identification, Provider Type - Select a vector provider type. Options include:
      - **Database ONNX Model**
      - **Generative AI Service**
      - **Custom PL/SQL**

      Note that UI pages based on the **Provider Type** you select.
   b. Identification, Name - Enter the name of the Vector Provider. The name must be unique and will be displayed in the Vector Provider overview page.
   c. Identification, Static ID - Enter a Static ID for the Vector Provider. The Static ID must be unique and will be referenced when using the service with the `APEX_AI` package, `APEX_AI.GET_VECTOR_EMBEDDINGS`.

      Example: `VP_OPENAI_EMBEDDING`
4. If Provider Type is **Database ONNX Model**:
   a. Identification, Name - Enter the name of the Vector Provider. The name must be unique and will display in the Vector Provider overview page.
   b. Identification, Static ID - Enter a Static ID for the Vector Provider. The Static ID must be unique and will be referenced when using the service with the `APEX_AI` package, `APEX_AI.GET_VECTOR_EMBEDDINGS`.
   c. Local Embedding, ONNX Model Owner - Select the owner of the ONNX model which should be used for the vector provider. All schemas assigned to your workspace are listed.
   d. Local Embedding, ONNX Model Name - Select the ONNX model that will be used to get an embedding for a given term.
5. If Provider Type is **Generative AI Service**:
   a. Identification, Name - Enter the name of the Vector Provider. The name must be unique and will display in the Vector Provider overview page.

b. Identification, Static ID - Enter a Static ID for the Vector Provider. The Static ID must be unique and will be referenced when using the service with the `APEX_AI` package, `APEX_AI.GET_VECTOR_EMBEDDINGS`.

c. Identification, AI Provider - Select the AI Provider to use for this Vector Provider. Options include:

- **OCI Generative AI Service**
- **Open AI**
- **Cohere**

The UI changes based on the AI Provider you select. See item Help, to learn more about an attribute.

6. If Provider Type is **Custom PL/SQL**:

a. Identification, Name - Enter the name of the Vector Provider. The name must be unique and will display in the Vector Provider overview page.

b. Identification, Static ID - Enter a Static ID for the Vector Provider. The Static ID must be unique and will be referenced when using the service with the `APEX_AI` package, `APEX_AI.GET_VECTOR_EMBEDDINGS`.

c. Local Embedding, Custom Function Name - Specify the name of a custom PL/SQL function which converts an end user input to an embedding. The function needs to take the end user input as `VARCHAR2` and returns a `VECTOR` data type as a result.

7. Click **Create**.

5.20.3.3 Editing and Deleting a Vector Provider

Edit or delete Vector Provides on the Vector Providers Detail page in Workspace Utilities.

To edit or delete a Vector Provider:

1. Navigate to the Vector Provider page:

a. On the Workspace home page, click the **App Builder** icon.

b. On the App Builder home page, click the **Workspace Utilities** icon.

c. On the Workspace Utilities page, click **Vector Providers**.

The Vector Providers page appears.

2. To edit a Vector Provider:

a. On the Vector Providers page, click the Vector Provider Name.

The Vector Provider Details appears.

b. Edit the appropriate attributes. To learn more about a specific attribute, see item Help.

c. Click **Apply Changes**.

3. To delete a Vector Provider:

a. On the Vector Providers page, click the Vector Provider Name.

The Vector Provider Details appears.

b. Click **Delete**.

c. Confirm your selection and click **Delete** again.

5.20.3.4 Viewing the Vector Providers History Report

View a report of recent modifications made to Vector Provider in this workspace.

To edit or delete a Vector Provider:

1. Navigate to the Vector Provider page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **Vector Providers**.

      The Vector Providers page appears.
2. On the Vector Providers page, click **History**.

   The Vector Providers History report appears.

5.20.4 Viewing Cross Application Reports

Use the links under Cross Application Reports page to view and edit information across multiple applications in the current workspace.

To access cross application reports:

1. Sign in to Oracle APEX.

   The Workspace home page appears.
2. Click the **App Builder** icon.

   The App Builder home page appears.
3. Click the **Workspace Utilities** icon.
4. Locate the **Cross Application Reports** region on the right side of the window. Available reports include:
   - All Reports
   - Application Attributes
   - Build Options
   - Developer Comments
   - Comments Calendar
   - Application Comparison
   - Component Comparison
   - Security Profiles
   - Authentication Schemes
   - Password Items
   - Theme Summary
5. To view all reports:
   a. Click **All Reports**.

      The Cross Application Reports page appears. Available reports include:
      - **Application Attributes**:

  - Application Attributes
  - Build Status and Application Status
  - Build Status
  - Developer Comments
  - Developer Comments Calendar
- **Application Comparison Reports**:
  - Application Comparison
  - Component Summary Counts
- **Security**:
  - Security Profiles
  - Authentication Schemes
  - Pasword Items
- **User Interface**:
  - Theme Summary

b. On the Cross Application Reports page, select a report to review.

6
Managing Pages in an Application

A page is the basic building block of an application. Developers add pages to an application by running the Create Page Wizard.

- Adding a New Page to an Application
  Add a new page by running the Create Page Wizard from the Application home page.
- About Copying Application Pages
  Copy an application page using the Copy Page Wizard.
- Creating a New Page by Copying
  Create a new page by copying a page from the current application or copying a page in another application.
- Understanding Page Types in the Create Page Wizard
  Run the Create Page Wizard to add a new page to existing application. How the wizard works depends upon the type of page you are creating.
- Managing Feature Pages
  Learn about adding and removing Feature pages.
- Creating Dialog Pages
  Oracle APEX includes support for modal and non-modal dialog pages. Modal and non-modal dialog pages are only supported if the current application theme contains at least one page template of template type Dialog Page.
- About Deleting Application Pages
  Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. You can only delete unlocked pages or pages that you have locked.
- Managing Page Groups
  Organize pages into page groups to make them to access easier. To use page groups, you create a group and then assign pages to the group.
- Locking and Unlocking a Page
  Prevent conflicts during application development by locking pages in your application. By locking a page, you prevent other developers from editing it.

> **See Also:**
>
> Editing Pages in Page Designer and Using Application Utilities

6.1 Adding a New Page to an Application

Add a new page by running the Create Page Wizard from the Application home page.

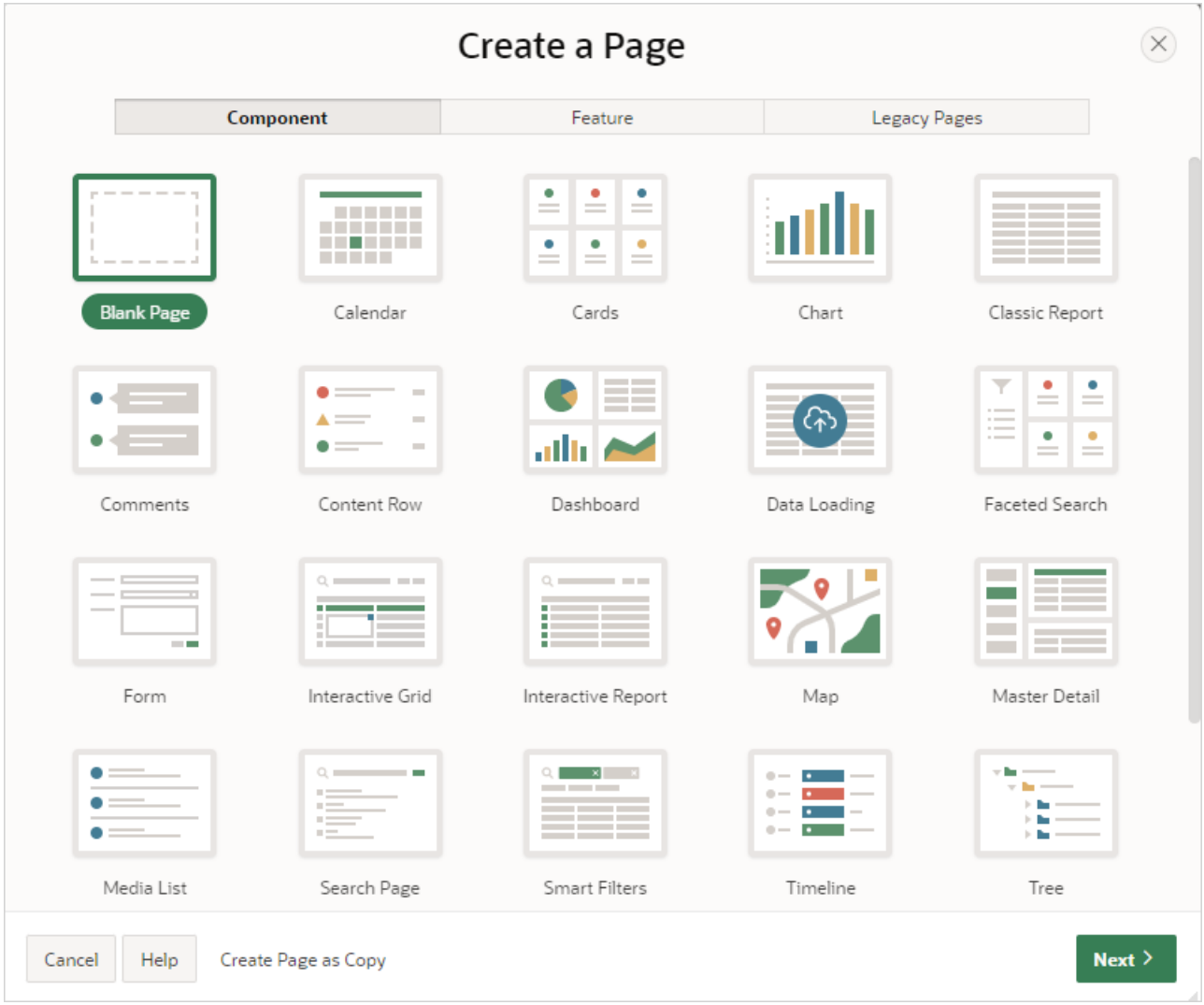
To create a new page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.

The Application home page appears.

3. Click the **Create Page** button.

   The Create Page Wizard appears.



4. Select a page type:

   - **Component** - **Component** pages provide page-level functionality. Examples of Component pages include Blank Page, Calendar, Cards, Chart, Classic Report, Comments, Content Row, Dashboard, Data Loading, Faceted Search, Form, Interactive Grid, Interactive Report, Map, Master Detail, Media List, Search Page, Smart Filters, Timeline, Tree, Unified Task List.Report, Wizard, Unified Task List, Wizard, and Plug-in Page.

     For details about available **Component** pages, see Component Page Types.

   - **Feature** - **Feature** pages provide application-level functionality. Examples of Feature pages include Push Notifications, About Page, Access Control, Activity Reporting, Configuration Options, Email Reporting, Feedback, Login Page, and Theme Style Selection.

     For more details about Feature pages, see Feature Page Types.

- **Legacy** - **Legacy** pages are pages that Oracle still supports but does not recommend using.

  For more details about Legacy pages, see Legacy Page Types and Managing Application Legacy Components.

> **Tip:**
>
> To create a new page by creating a copy, click **Create Page as Copy**. See Creating a New Page by Copying.

What appears next depends upon the option you select.

5. Click **Next** and follow the on-screen instructions. To learn more about an attribute, see field-level Help.
6. To access the Copy Page Wizard, click **Create Page as Copy**.

> **Tip:**
>
> To learn more, see the Create Page Wizard discussion for each component.

6.2 About Copying Application Pages

Copy an application page using the Copy Page Wizard.

You can access the Copy Page Wizard from either the Create Page Wizard or from the Create menu on the Page Designer toolbar.

During the copy process, you can also copy shared components or change mappings to shared components in the target application.

About Copying Pages Between User Interfaces in Older Themes

The Copy Page Wizard enables you to copy a page to a different user interface. During the copying process, the user can select the user interface for the new page. If the application contains a Desktop and a Mobile theme, both user interface options display.

Accessing the Copy Page Wizard

Access the Copy Page Wizard as follows:

- From the Create Page Wizard - Click **Create Page as Copy** to copy a page from the current application or copy a page in another application.
  1. Access the Create Page Wizard.
  2. Click **Create Page as Copy**.
  3. Follow the on-screen instructions.

     For more details, see Creating a New Page by Copying.
- From Page Designer - Copy a page in the existing application from Page Designer.
  1. View the page in Page Designer.
  2. From the Page Designer toolbar, click the **Create** menu and then select **Copy Page**.

For more details, see Copying a Page in Page Designer.

6.3 Creating a New Page by Copying

Create a new page by copying a page from the current application or copying a page in another application.

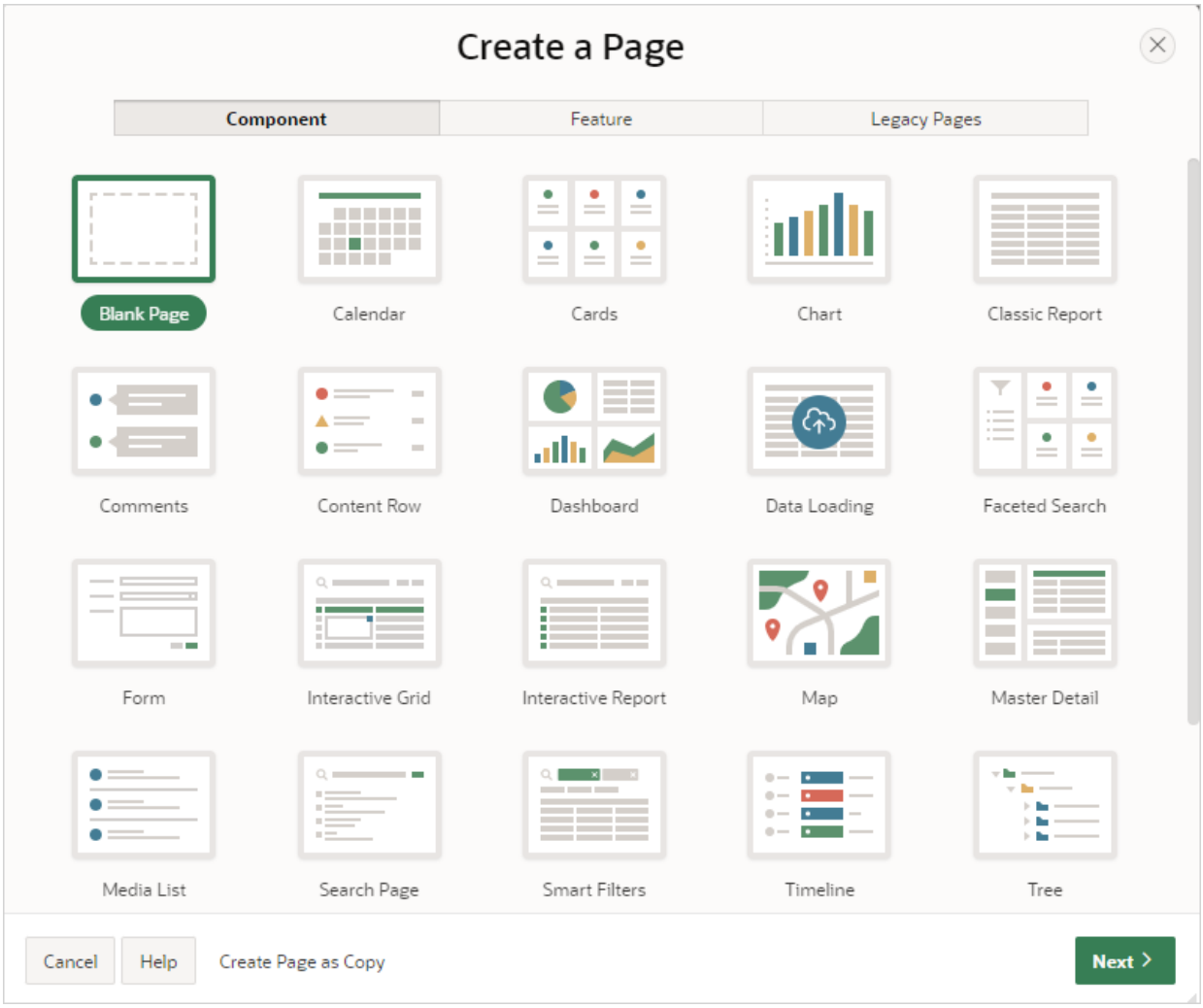
You can access the Copy Page Wizard from the Create Page Wizard.

> **Tip:**
>
> You can also copy a page from Page Designer. See Copying a Page in Page Designer.

To create a new page by copying an existing page:

1. Access the Copy Page Wizard:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. Click the **Create Page** button.

      The Create Page Wizard appears.



d. Click **Create Page as Copy**.

   The Copy Page Wizard appears.

2. Copy Page Option, Create a page as a copy of - Specify how to copy a page.

   Options include:

   - **Page in this application**
   - **Page in another application**

3. To copy a page in the current application:

   a. Create a page as a copy of - Select **Page in this application** and click **Next**.

   b. Page to Copy:

      - Application - Displays the app ID.
      - Copy From Page - Select the page you want to copy.
      - New Page Number - Enter the new page number.
      - Copy Public Reports - All default reports are copied automatically without specifying whether they are public or private.
        Available options include:
        – **On** - Copy public reports with the page.

  - **Off** - Do not copy public reports with the page.

    Public reports are shared report settings saved by users from an interactive report or interactive grid.

- Breadcrumb - Select whether you want to use a breadcrumb navigation control on the new page, and which breadcrumb navigation control you want to use.
- Click **Next**.

c. Navigation Menu:

- Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
- Click **Next**.

d. New Names:

- Review options under **Old Value** and **New Value**.
- Enter changes as appropriate and click **Next**.

e. Click **Copy**.

4. To copy a page in another application:

a. Create a page as a copy of - Select **Page in another application** and click **Next**.

b. Page to Copy:

- Copy From Application - Select the application that contains the page you wish to copy.
- Copy From Page - Select the page you want to copy.
- New Page Number - Enter the new page number.
- New Page Name - Enter the page name for your new page.
- Breadcrumb - Select whether you want to use a breadcrumb navigation control on the new page and which breadcrumb navigation control you want to use.
- Click **Next**.

c. Navigation Menu:

- Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
- Click **Next**.

d. Templates - When you copy a page from another application, all the templates used by that page need to be mapped to templates in the current application. This report displays the templates in the page you are copying from. Review the report and map to the correct template and click **Next**.

e. Click **Finish**.

6.4 Understanding Page Types in the Create Page Wizard

Run the Create Page Wizard to add a new page to existing application. How the wizard works depends upon the type of page you are creating.

> **Tip:**
>
> Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface was based on jQuery Mobile. Because jQuery Mobile is desupported, you must migrate existing mobile applications to the Universal Theme.

- Component Page Types
  Learn about available Component page types when running the Create Page Wizard and using the Universal Theme.
- Feature Page Types
  Learn about available Feature page types when running the Create Page Wizard and using the Universal Theme.
- Legacy Page Types
  Learn about Legacy page types available when running the Create Page Wizard. Oracle still supports legacy pages, but does not recommend using them.

6.4.1 Component Page Types

Learn about available Component page types when running the Create Page Wizard and using the Universal Theme.

Page Type	Description	To Learn More
Blank Page	Creates a page with no built-in functionality. Once you create the application, you can create regions on the page in Page Designer.	See online Help when creating this page.
Calendar	Generates a calendar with monthly, weekly, and daily views.	See: Creating Calendars
Cards	A **cards** page features an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row). Features: • Customize every aspect of a cards region's UI (including layout, appearance, icon, badge, and media). • Declaratively create links from a cards page by adding actions. Action types include Button, Full Card, Title, Subtitle, or Media. • Supports advanced HTML expressions including template directives for client-side conditional formatting.	See: Managing Cards
Chart	Creates a new page containing a chart. Supported chart types include area, bar, box plot, bubble, combination, status meter gauge, donut, funnel, gantt, line, line with area, pie, polar, pyramid, radar, scatter, and stock charts.	See: Creating Charts

Page Type	Description	To Learn More
Classic Report	A **classic report** features a simple report page based on the formatted result of a SQL query. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Enable column sorting in Page Designer by selecting a column and configuring Column, Sorting attributes. • The most versatile report as it is template-driven. Choose from the predefined report templates, including Media List, Comments, Timeline, Badge List, Value Attributes Pairs, and so on. Or, create your own template to customize and visualize data. • Download the report as a CSV, HTML, Excel, or PDF. • Supports Report Layouts to format PDF exports.	See: Managing Classic Reports
Comments	Creates a page with a Comments region that displays user comments and status updates. Only available in applications with the latest version of Universal Theme.	See online Help when creating this page.
Content Row	Creates a page with a Content Row region that can be used to display many types of content. Content Row regions support an icon, title and description, miscellaneous attributes, and actions. Only available in applications with the latest version of Universal Theme.	See online Help when creating this page.
Dashboard	Creates a dashboard page. Choose from four different chart layouts which you can easily customize in Page Designer. Available chart layouts include Primary and 2 secondary charts, Grid of 4 charts, Primary and 3 secondary charts, and Grid of 6 charts.	See: Creating a Dashboard Page Using the Create Application Wizard
Data Loading	If a Data Load Definition exists, you can create a Data Loading page. The Create Page Wizard creates a single page with the Native Data Loading page process to provide the ability to upload data from a file or by copy and paste, and then upload the data. This Data Loading page enables end users to manage the loading of data.	See: Creating a Data Loading Page Using the Create Page Wizard
Faceted Search	A **faceted search** page displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). Displays search results as cards or a classic report. Features: • Select options from facets in the left Search region to narrow down search results. • Set filters using facets that show possible values together with the occurrence count within the result set. • Built-in search capability. • Optionally include a chart for any facet. • Toggle between a bar chart (which is the default) and pie chart type. • Create facet groups for data models having multiple flag columns.	See: Managing Faceted Search

Page Type	Description	To Learn More
Form	Forms enable users to update a single row or multiple rows in a table. When a local table which has foreign keys is selected as the data source, then for those columns, the item type is set as **Select List**.	See: Creating a Form
Interactive Grid	An **interactive grid** presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. Features: • Extensive customization capabilities using menus or inline editing. • Turn editing on or off using the region Attribute, Edit, Enabled. • Built-in search capability. • For both editable and non-editable reports, users can change the report layout and create private reports: – Column Heading menu - Change the report layout with Control Break, Aggregations, Freeze, Hide, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Refresh, Flashback. * Format - Control Break, Highlight, Stretch Column Widths. * Selection - Cell Selection, Copy to Clipboard, Refresh Rows. * Chart - Create chart. * Report - Create, edit, and manage saved Private or Alternative reports. * Download - Download or email the report as a CSV, HTML, PDF, or Excel file • If editable, users can edit the data directly: – Row Actions menu - Single Row View, Add Row, Duplicate Row, Delete Row, Refresh Row, Revert Changes. – Add Rows - Add new rows by clicking the **Add Row** button. – Edit Rows - Edit a row by selecting or double-clicking a cell with the mouse, editing the content, and clicking the **Save** button.	See: Managing Interactive Grids

Page Type	Description	To Learn More
Interactive Report	An **interactive report** page features a searchable, easily customizable report. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Built-in search capability. • Change the report layout and create private reports using menus: – Column Heading menu: Change the report layout with Sort Ascending, Sort Descending, Hide Column, Control Break, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Compute, Flashback. * Format - Control Break, Highlight, Rows Per Page. * Chart - Create a chart. * Group By - Group sets of results by one or more columns with Group By. * Pivot - Pivot reports transpose rows into columns to generate results in a cross tab format. * Report - Create, edit, and manage saved Named reports. * Download - Download or email the report as a CSV, HTML, Excel, or PDF. * Subscription - Subscribe to report updates in CSV, HTML, Excel, or PDF format.	See: Managing Interactive Reports
Map	The map component consists of a background map and spatial geometry objects (or layers) sourced from data in a local table. Supported spatial geometry objects (or layers) include: Points, Lines, Polygons, Heat Map, or Extruded Polygons. Spatial geometry objects can be either: • **Geometry Column** - Supported datatypes include `SDO_GEOMETRY`, `VARCHAR2`, or `CLOB`. `VARCHAR2` and `CLOB` columns must contain geometry information in GeoJSON format. • **Two Numeric Columns** - These columns must contain longitude and latitude values. This option only applies to Point and Heat Map layers.	See: Creating Maps

Page Type	Description	To Learn More
Master Detail	A master detail reflects a one-to-many relationship between two tables in a database. Master detail forms enable users to insert, update, and delete values from two tables or views. Master Detail options include: • **Stacked** - Creates a single page master detail with editable interactive grids. Users select a row in the master grid to update the detail grids. • **Side by Side** - Create a single page master detail utilizing side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report. • **Drill Down** - Creates a two page (or Drill Down) master detail. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grids for the detail.	See: Creating Master Detail Forms
Media List	Creates a page with a Media List region which can be used to display content in a list with an icon, heading, description, and badge. Only available in applications with the latest version of Universal Theme.	See online Help when creating this page.
Search Page	Create a decalarative search for an application by creating a search configuration that contain information about a searchable data source and then creating a search page that displays the results.	See Adding Search to an Application
Smart Filters	A **smart filters** page features single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a Smart Filter behaves similarly to faceted search, it features a more space efficient layout. Features: • Includes a search field at the top of the page with filters that display as suggestion chip. • The Search Results can display as a classic report, cards, map, or calendar. • Clicking a filter name displays a list of values. • Clicking to right of a suggestion chip changes it to an applied filter chip and moves to the Applied Filters Area. Clicking the applied filter chip again displays a list. • Built-in search capability.	See: Managing Smart Filters
Timeline	Creates a page with a Timeline region for displaying a series of events. The Timeline region can be used to showcase the history of a given widget, recent updates, or new interactions within an application. Only available in applications with the latest version of Universal Theme.	See online Help when creating this page.
Tree	Creates a tree to graphically communicate hierarchical or multiple level data and optionally navigating to a page or URL specific to each tree node.	See: Managing Trees
Unified Task List	Create a page that allows users to view, action, or edit their approval tasks.	See: Managing Unified Task Lists

Page Type	Description	To Learn More
Wizard	Create a collection of pages to be used as a wizard. Generally wizards are used for entering data across multiple steps.	See online Help when creating this page.
Workflow Console	Displays all workflows for a given report context. Creates two pages in the application: an overview page with all the workflows for the given report context and a workflow details page that opens when a user selects a workflow from the overview page.	See: Managing Workflows and Tasks and About the Workflow Console, Workflow Details, and Workflow Dashboard
Plug-in Page	Creates a new page based on a region type plug-in. Plug-ins enable developers to declaratively extend, share, and reuse the built-in types available with Oracle APEX.	See: Implementing Plug-ins

6.4.2 Feature Page Types

Learn about available Feature page types when running the Create Page Wizard and using the Universal Theme.

Page Type	Description	To Learn More
Push Notifications	Enable web apps to send notifications to a user's device even when the user is not currently on the website. Notifications can be sent to a user's desktop or mobile device and can be used to alert the user to new content or events on the website, or to re-engage the user with the website. Enabling this option:. • Adds a new navigation bar entry, `User Settings`. • Enables users to subscribe and unsubscribe to push notifications. • When subscribed, users can receive push notifications from the APEX app.	See Creating a Progressive Web App (PWA)
About Page	Include an **About this Application** page which features a description field that describes the application, includes the application version, and a count of the number of pages.	See online Help when creating feature.
Access Control	Creates pages to manage an access control list. Use the Application Access Control shared component to associate application roles with application users. This wizard also adds a **reader**, **contributor** and **administrator** role and corresponding authorization scheme to your application. Apply these authorization schemes to pages and page components to manage access by user and role.	See Controlling Access to Applications, Pages, and Page Components
Activity Reporting	Includes numerous reports on end user activity for your application. Determine the most active users, the most used pages, the performance of pages, and errors raised, to better understand how your application is being utilized and areas for improvement.	See online Help when creating this feature.

Page Type	Description	To Learn More
Configuration Options	Enables application administrators to enable or disable specific functionality within the application. This feature is useful if you select features that need additional development effort before they can be used by end users. You can expand this feature to application-specific features. If developers define additional build options and associate them with specific functionality throughout the application, then they can be added to the configuration settings for administrators.	See online Help when creating this feature.
Email Reporting	Include numerous reports on emails queued from this application, emails sent, and errors sending emails.	See online Help when creating this feature.
Feedback	Provides a mechanism for end users to post general comments for application administrators and developers. The posts include useful session state information to help developers determine where the end user sent the feedback from. Creating Feedback: • Creates Navigation bar icon which users can click to leave feedback. • Creates an report for viewing and updating feedback. • Captures the application and page ID, feedback comments, date and time, and user information.	See Managing Feedback.
Login Page	Creates a Login Page which enables the user to enter login credentials for accessing your application. Developers can select a template and labels for the Username and Password fields as well as determine if the Username value is stored a system-generated cookies.	See online Help when creating this feature.
Theme Style Selection	Enables administrators to select a default color scheme (theme style) for the application. Administrators can also choose to allow end users to choose their own theme style. End users can simply click on the Customize button at the bottom of the home page, and select from the available theme styles. For example, users with visual impairment may prefer to utilize the Vista theme style which has a much higher color contrast.	See online Help when creating this feature.

See Also:

Managing Feature Pages

6.4.3 Legacy Page Types

Learn about Legacy page types available when running the Create Page Wizard. Oracle still supports legacy pages, but does not recommend using them.

Legacy page available when running the Create Page Wizard include:

- Form on Local Procedure

- Summary Page
- Legacy Data Loading

> **See Also:**
>
> Managing Application Legacy Components

6.5 Managing Feature Pages

Learn about adding and removing Feature pages.

- About Feature Pages
  Feature pages provide application-level functionality. Examples of Feature pages include About Page, Access Control, Activity Reporting, Configuration Options, Email Reporting, Feedback, Login Page, and Theme Style Selection.
- Adding Features to an Existing App
  Add a Features page to an existing application by selecting **Add Feature Page** from the Tasks list on the Application home page.
- About Removing Features Created with the Create Page Wizard
  Remove features created with a wizard using build options.

6.5.1 About Feature Pages

Feature pages provide application-level functionality. Examples of Feature pages include About Page, Access Control, Activity Reporting, Configuration Options, Email Reporting, Feedback, Login Page, and Theme Style Selection.

> **Tip:**
>
> See Feature Page Types for details about available Feature pages.

Add Feature pages in the following ways:

- **Create a new application**. Run the Create Application Wizard and select options under **Features**. See Running the Full Create Application Wizard and Available Features in the Create Application Wizard.
- **Create a new page**. Run the Create Page Wizard. On the Features tab, select a feature. See Adding a New Page to an Application and Feature Page Types.
- **Add to an existing application**. Add a Features page to an existing application. Navigate to the application. On the Application home page, select **Add Feature Page** from the Tasks list. On the Features tab, select a feature. See Feature Page Types.

6.5.2 Adding Features to an Existing App

Add a Features page to an existing application by selecting **Add Feature Page** from the Tasks list on the Application home page.

To add a Features page to an existing application:

1. On the Workspace home page, click **App Builder**.
2. Select an application.

   The Application home page appears.
3. From the Tasks list, select **Add Feature Page** button.

   The Create Page Wizard, Feature tab appears.
4. Select a feature.

   For more details about Feature pages, see Feature Page Types.

   What appears next depends upon the Feature type you select.
5. Click **Next** and follow the on-screen instructions. To learn more about an attribute, see field-level Help.

6.5.3 About Removing Features Created with the Create Page Wizard

Remove features created with a wizard using build options.

When you add a feature using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You can use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Oracle APEX engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the APEX engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features created with a wizard by first removing the components associated with the build option and then deleting the build option.

> **See Also:**
>
> - Including or Excluding Build Options
> - Deleting Build Options and Associated Components

6.6 Creating Dialog Pages

Oracle APEX includes support for modal and non-modal dialog pages. Modal and non-modal dialog pages are only supported if the current application theme contains at least one page template of template type Dialog Page.

- About Supported Dialogs
  When you create a new page, the wizard prompts you to select a Page Mode. The Page Mode determines if the page is a normal application page or a dialog page.

- Creating a Dialog Page Template
  Create a dialog template if your theme does not include one and how to edit dialog template attributes to change the display.
- Editing a Dialog Page in Page Designer
  Change how a dialog page displays by editing attributes in Page Designer.
- Generating a Cancel Dialog Dynamic Action on the Dialog Page
  For dialog pages that contain a Cancel button, create a dynamic action that controls what happens when the button is clicked.
- Generating a Close Dialog Page Process on the Dialog Page
  Close a dialog page by creating a Close Dialog page process.
- About Branching with Dialog Pages
  Open dialog pages using buttons or links.
- Handling Refresh Actions on Parent Page
  Learn about handling refresh actions on parent pages.

6.6.1 About Supported Dialogs

When you create a new page, the wizard prompts you to select a Page Mode. The Page Mode determines if the page is a normal application page or a dialog page.

Oracle APEX supports two types of dialog pages:

- **Modal Dialog** - A modal dialog is an overlay window positioned within the same browser window. A modal dialog remains active and focused until the user has finished with it and closes it. While a modal dialog is active, the user is unable to interact with the rest of the page until the dialog is closed.
- **Non-modal Dialog** - A non-modal dialog displays a separate popup browser window. A user can interact with a non-modal dialog and content on the page. Use a non-modal dialog when the requested information is not essential to continue. This type of window can be left open while work continues elsewhere.

Oracle APEX includes dialog page templates in the *Universal Theme - 42*. When you set the Page Mode to either **Modal** or **Non-Modal**, APEX automatically uses the default dialog page template.

You can further define the appearance and layout of a dialog by specifying the Dialog Template.

6.6.2 Creating a Dialog Page Template

Create a dialog template if your theme does not include one and how to edit dialog template attributes to change the display.

The dialog page template contains JavaScript function calls that handle the initialization, closure, and cancellation of a dialog page. When you create a dialog page template, you must define these and other Dialog Page Templates attributes.

> **See Also:**
>
> Required Dialog Page Template Attributes

If you are using an existing application and the theme does not contain a dialog page template, then you must create one. While you can create one from scratch, the easiest option is to copy an existing dialog template.

To create a dialog page template:

1. Navigate to the Shared Components, Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.
2. Click **Create**.
3. In the Create Template Wizard:.
   a. Template Type - Select **Page** and click **Next**.
   b. Creation Method - Select **As a Copy of an Existing Template** and click **Next**.
   c. Copy From - Select the application to copy from and click **Next**.
   d. Identify Theme - Select the theme you are copying from and the theme you are copying to and click **Next**.
   e. New Templates - Locate the Modal Dialog templates to copy (for example, **Drawer**, **Modal Dialog**, or **Wizard Modal Dialog**) and select **Yes**.
   f. Select **Copy Page Templates**.
4. Verify the dialog page template is active in the current theme

> **See Also:**
>
> Selecting a Default Page Template

6.6.3 Editing a Dialog Page in Page Designer

Change how a dialog page displays by editing attributes in Page Designer.

To edit a dialog page in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears. Attributes are organized in groups.
2. In Page Designer locate the **Appearance** group.
3. Appearance, Page Mode - Select one of the following:
   - **Modal Dialog** - The page is presented as a modal dialog. A modal dialog is an overlay window positioned within the viewport, which remains active and focused until the end user dismisses (or closes) it. The underlying page is grayed out and the end user is prevented from interacting with the rest of the page until the dialog is closed.

- **Non-Modal Dialog** - The page is presented as a non-modal dialog. A non-modal dialog is an overlay window positioned within the viewport, where the end user can interact with the dialog, and content on the page that launched the dialog. A non-modal dialog can be used when the requested information is not essential to continue; work can continue elsewhere while the window is still open.

Note that options Modal and Non-Modal Dialog will only be available for selection when the current application theme contains a default Dialog Page Template.

4. Appearance, Dialog Template - Select a dialog template to define the appearance and layout of this page.

   Dialog templates are defined in the application theme. When a dialog page is created, the template is automatically set to **Theme Default** which renders the page using the default page template defined in the current theme. Making a selection here overrides the default dialog template.

   Universal Theme provides the following dialog templates:

   - **Modal Dialog** - Useful for most dialog use cases.
   - **Wizard Modal Dialog** - Provides a streamlined user interface and is well suited for modal dialog wizards, alerts, notifications, or small forms.

   Dialog templates are defined in the application theme. When a dialog page is created, the template is automatically set to **Theme Default** which renders the page using the default page template defined in the current theme.

5. Edit the remaining attributes in **Appearance** group as needed.

   To learn more about an attribute, see field-level Help.

6. Under Dialog:

   a. Dialog, Width - Enter the width of the dialog, in pixels or a percentage. The substitution string `#DIALOG_WIDTH#` in the dialog initialization code of the page template is replaced with the value entered in this item.

   b. Dialog, Height - Enter the height of the dialog, in pixels, for example, `500`. The substitution string `#DIALOG_HEIGHT#` in the dialog initialization code of the page template is replaced with the value entered.

   c. Dialog, Maximum Width - Enter the maximum width of the dialog in pixels. The substitution string `#DIALOG_MAX_WIDTH#` in the dialog initialization code of the page template is replaced with the value entered in this item.

   d. Dialog, Attributes - Dialog support varies depending on the user interface of the page. A subset of dialog attributes are declaratively supported: height, width, and maximum width.

      To define any additional attributes for the dialog, enter the name/value pairings in a comma separated list. The substitution string `#DIALOG_ATTRIBUTES#` in the dialog initialization code defined at page template level, is replaced by any value(s) entered in this item.

   e. Dialog, CSS Classes - Enter classes to add to this component. You may add multiple classes by separating them with spaces.

   f. Dialog, Chained - Enable **Chained** if the dialog is opened by another dialog and should reuse the same dialog window (for example, if this dialog is part of a multiple page dialog wizard and you wish to reuse the dialog window for each step of the wizard).

Disable **Chained** if this dialog is opened by another dialog and should be opened in a new dialog window (for example, when this dialog is modal and you wish to open it within a parent modal dialog window).

g. Dialog, Resizable - Enable **Resizable** to enable end users to resize this modal dialog.

7. Edit additional attributes as needed.

   To learn more about an attribute, see field-level Help.

8. To save your changes, click **Save**.

6.6.4 Generating a Cancel Dialog Dynamic Action on the Dialog Page

For dialog pages that contain a Cancel button, create a dynamic action that controls what happens when the button is clicked.

The dynamic action must have True Action event which is set to **Cancel Dialog**. This dynamic action will use the cancellation JavaScript call contained in the page template (`apex.navigation.dialog.cancel`) to handle the canceling the dialog.

> **See Also:**
>
> Managing Dynamic Actions

6.6.5 Generating a Close Dialog Page Process on the Dialog Page

Close a dialog page by creating a Close Dialog page process.

In previous releases, this Close Dialog page processes were referred to as a `Close Popup Window`. This dialog will use the closure JavaScript call contained in the page template (`apex.navigation.dialog.close)`, to handle the closing of the dialog. The page process also supports the return of dialog page items, which you may wish to reference on the parent page.

> **Tip:**
>
> You can also close a dialog page with the Close Dialog dynamic action. Both the dynamic action and the process support returning dialog page items. Use the page process when the page needs to be submitted and use the dynamic action when the page does not need to be submitted. Both end up calling the closure JavaScript call. Alternatively, you can create a page branch, which will close a dialog page and redirect to the page specified in the branch.

> **See Also:**
>
> - Understanding Page Processes
> - Controlling Navigation Using Branches

6.6.6 About Branching with Dialog Pages

Open dialog pages using buttons or links.

Dialog pages support the following branching options:

- Dialog page to dialog page.

  You can branch from one dialog page to another dialog page if:

  - Both pages have the same Page Mode. For example, a Modal Dialog page and a page where the dialog Chained attribute is set to **On**. Setting the Chained attribute to **On** enables support for chained or nested dialogs such as the dialogs that appear in many internal APEX wizards. Chained or nested dialogs enable you to re-use a dialog window to display each page.
- Dialog page to Normal page.
  - When branching from a dialog page to a normal page, the dialog page is closed and the user navigates to a new page in the browser. The new page can be a parent (or launching) page or another Normal page in an application.

6.6.7 Handling Refresh Actions on Parent Page

Learn about handling refresh actions on parent pages.

If you have generated a Close Dialog page process on the dialog page, you must implement refresh actions on the parent page using the new `Dialog Closed` dynamic action event type. You can use the set type Dialog Return Item when creating a Set Value True Action, to set a single value, based on the return item of a dialog.

> **Tip:**
>
> For a partial page refresh of a parent page, implement the close dialog with dynamic action (for example, for report and form pages where you want to edit data in a modal dialog form and only want the report refreshed on close). In contrast, branching is the best approach if you have a multi-step modal wizard or if you have several possible target pages when closing the dialog.

> **See Also:**
>
> Managing Dynamic Actions

6.7 About Deleting Application Pages

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries. You can only delete unlocked pages or pages that you have locked.

> **See Also:**
>
> Deleting Pages

6.8 Managing Page Groups

Organize pages into page groups to make them to access easier. To use page groups, you create a group and then assign pages to the group.

> **Tip:**
>
> Page groups do not have any function other than to aid developers in organizing their application pages.

- Viewing Page Groups
  View page groups on by navigating to Utilities, Page Specific Utilities.
- Creating a Page Group
  Create a page group by navigating to Utilities, Page Specific Utilities.
- Editing a Page Group Definition
  Create a page group by defining a group name and description.
- Assigning a Page to a Page Group
  Assign a page to a page group in Page Designer.
- Viewing Pages by Page Group
  View pages by page group.
- Deleting a Page Group
  Remove all pages from a group before deleting it.

6.8.1 Viewing Page Groups

View page groups on by navigating to Utilities, Page Specific Utilities.

To view page groups:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.

   The Page Groups page appears.

   A Search bar displays at the top of the page. Available controls include:

   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a page group by name and click **Go**.
   - **Go button** - Executes a search or applies a filter.

- **View Icons** - Displays each group as a large icon. To edit a group, click the appropriate icon.
- **View Report** - Displays each group as a line in a report. To edit a group, click the name.
- **Actions menu** - Use the Actions menu to customize the report view.

5. To view the pages associated with a group, click **Pages by Page Group**.

6.8.2 Creating a Page Group

Create a page group by navigating to Utilities, Page Specific Utilities.

To create a page group:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. On the Page Groups page, click **Create**.
6. Enter a name, a description (optional), and click **Create**.

6.8.3 Editing a Page Group Definition

Create a page group by defining a group name and description.

To edit the Page Group definition:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Select a page group.
6. On the Page Group page, edit the name or description and click **Apply Changes**.
7. To move to the next Page Group definition, click the **Previous** (<) and **Next** (>) buttons and repeat the previous step.

6.8.4 Assigning a Page to a Page Group

Assign a page to a page group in Page Designer.

To assign pages to a page group:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Click the **Page Assignments** tab.

   The Page Assignments page appears.

6. Click the page number.

   The page displays in Page Designer appears.
7. Identification, Name - Select a page group.
8. Click **Apply Changes**.

6.8.5 Viewing Pages by Page Group

View pages by page group.

To access the Pages by Page Group page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click the **Utilities** icon.
4. From Page Specific Utilities, click **Page Groups**.
5. Click the **Pages by Page Group** tab.

   The Pages by Page Group page appears. Click the Page Name to view the page.
6. To remove a page from a group:
   a. Click the page number.

      The page displays in Page Designer appears.
   b. Identification, Name, - Select `– Select –`.
   c. Click **Apply Changes**.

6.8.6 Deleting a Page Group

Remove all pages from a group before deleting it.

To delete a page group:

1. Remove all pages from the group to be deleted:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. From Page Specific Utilities, click **Page Groups**.
   d. Click the **Pages by Page Group** tab.
   e. Click the page number.

      The Page Attributes page appears.
   f. Under Name, Group, select `– No Group Assigned –`.
   g. Click **Apply Changes**.
2. When you have deleted all pages in the group, click the **Page Groups** tab.
3. Select the group to be deleted.
4. Click **Delete**.

   A confirmation page appears.
5. Confirm your request.

6.9 Locking and Unlocking a Page

Prevent conflicts during application development by locking pages in your application. By locking a page, you prevent other developers from editing it.

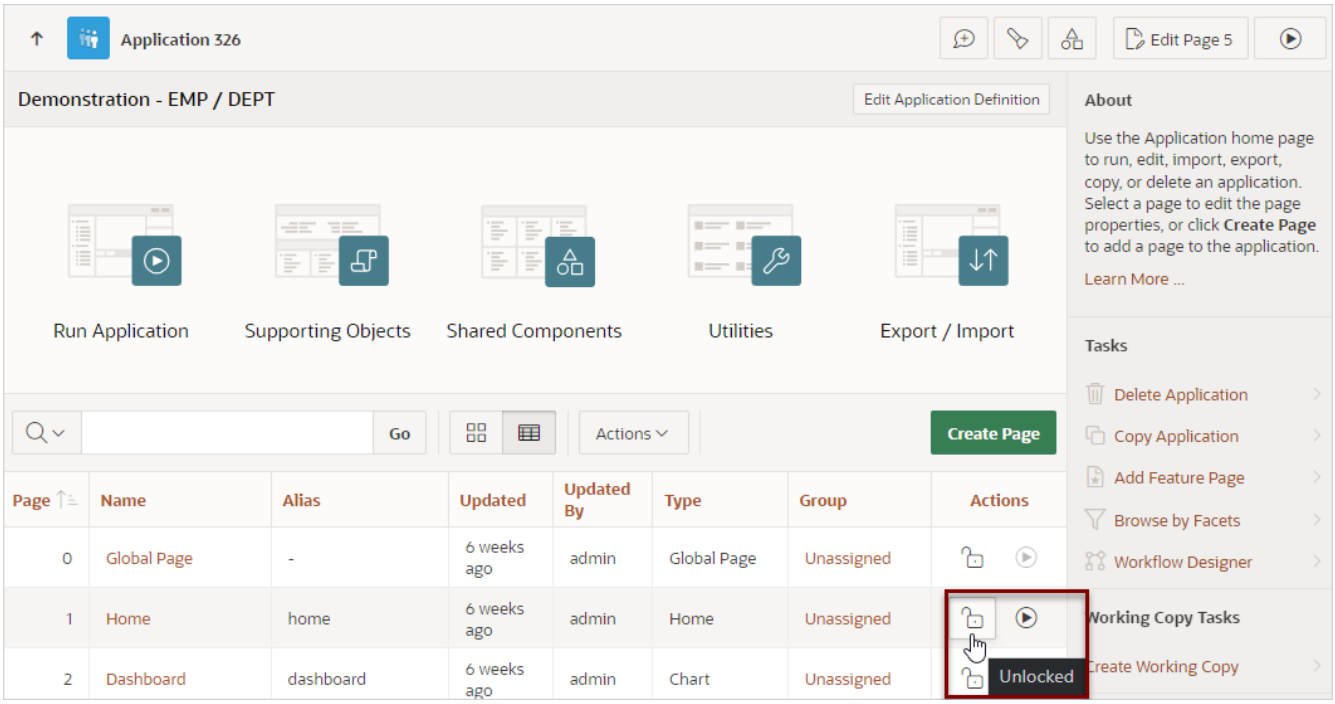
- Determining If a Page Is Locked
  Learn how to determine if a page is locked.
- Locking a Page
  Learn how to lock a page.
- Unlocking a Page
  Learn how to unlock a page.

6.9.1 Determining If a Page Is Locked

Learn how to determine if a page is locked.

A lock icon indicates whether a page is currently locked. If a page is unlocked, the icon appears as an open padlock. If the page is locked, the icon appears as a locked padlock. A lock icon appears on the Application home page and on the page.

To view the lock icon on the Application home page, click the **View Report** icon. A report appears. The lock icon appears in the report under Actions.



6.9.2 Locking a Page

Learn how to lock a page.

Developers can lock a page on the Application home page, Page Locks page, and in Page Designer.

- Locking a Page from the Application Home Page
  Lock a page from the Application Home page.
- Locking a Page in Page Designer
  Lock a page in Page Designer.
- Locking a Page from Utilities
  Lock a page from the Cross Page Utilities, Page Locks page.

6.9.2.1 Locking a Page from the Application Home Page

Lock a page from the Application Home page.

To lock a page from the Application Home Page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the search bar, click the **View Report** icon.
4. Under Actions, click the **Lock** icon.
5. Enter a comment in the Comment field.
6. Click **Lock Page(s)**.

6.9.2.2 Locking a Page in Page Designer

Lock a page in Page Designer.

The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is unlocked, the Page Lock button displays as an unlocked padlock.

To lock a page in Page Designer:

1. Navigate to the appropriate page in Page Designer.
2. On the Page Designer toolbar, click the **Page Unlocked** button.
3. Enter a comment in the Comment field.
4. Click **Lock**.

   When a page is locked, the Page Lock button displays as a locked padlock.

> **See Also:**
>
> Viewing a Page in Page Designer

6.9.2.3 Locking a Page from Utilities

Lock a page from the Cross Page Utilities, Page Locks page.

To lock a page from the Page Locks page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.

4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Select the appropriate pages and click **Lock Checked**.
7. Enter a comment in the Comment field.
8. Click **Lock Page(s)**.

6.9.3 Unlocking a Page

Learn how to unlock a page.

Only the developer who locks a page can unlock it. However, a developer with administrative privileges can unlock pages locked by other developers.

> **Tip:**
>
> To view information about an existing page lock, navigate to the Lock Details page and expand the Lock History region.

- Unlocking Pages from the Application Home Page
  Unlock a page from the Application Home page.
- Unlocking a Page in Page Designer
  Unlock a page in Page Designer.
- Unlocking Pages from Utilities
  Unlock a page from the Cross Page Utilities, Page Locks page.
- Unlocking Pages As a Workspace Administrator
  Unlock pages as a Workspace Administrator.

6.9.3.1 Unlocking Pages from the Application Home Page

Unlock a page from the Application Home page.

To unlock a page from the Application Home page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the search bar, click the **View Report** icon.
4. Under Actions, click the **Locked** icon.

   The Lock Details page appears.
5. Click **Unlock**.

6.9.3.2 Unlocking a Page in Page Designer

Unlock a page in Page Designer.

The Page Lock button on the Page Designer toolbar indicates whether a page is locked. If the page is locked, the Page Lock button displays as a locked padlock. Passing your cursor over a locked Page Lock button displays a tooltip that indicates who locked it.

To unlock a page in Page Designer:

1. Navigate to the appropriate page in Page Designer.

   If a page is locked, the Page Lock button displays as a locked padlock.

2. On the Page Designer toolbar, click the **Page locked by you** button.

   If the page has been locked by another user the Lock button appears in red. You cannot unlock a page locked by another user.

3. Click **Unlock**.

   When a page is unlocked, the Page Lock button displays as an unlocked padlock.

Viewing a Page in Page Designer

6.9.3.3 Unlocking Pages from Utilities

Unlock a page from the Cross Page Utilities, Page Locks page.

To unlock a page from the Page Locks page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Utilities** icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Select the appropriate pages.
7. Click **Unlock Checked**.

6.9.3.4 Unlocking Pages As a Workspace Administrator

Unlock pages as a Workspace Administrator.

To unlock pages as a Workspace Administrator:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the Utilities icon.
4. On the Page Specific Utilities list, click **Cross Page Utilities**.
5. Click **Page Locks**.
6. Click **Lock Administration**.
7. Select the appropriate pages.
8. Click **Unlock Page(s)**.

7
Editing Pages in Page Designer

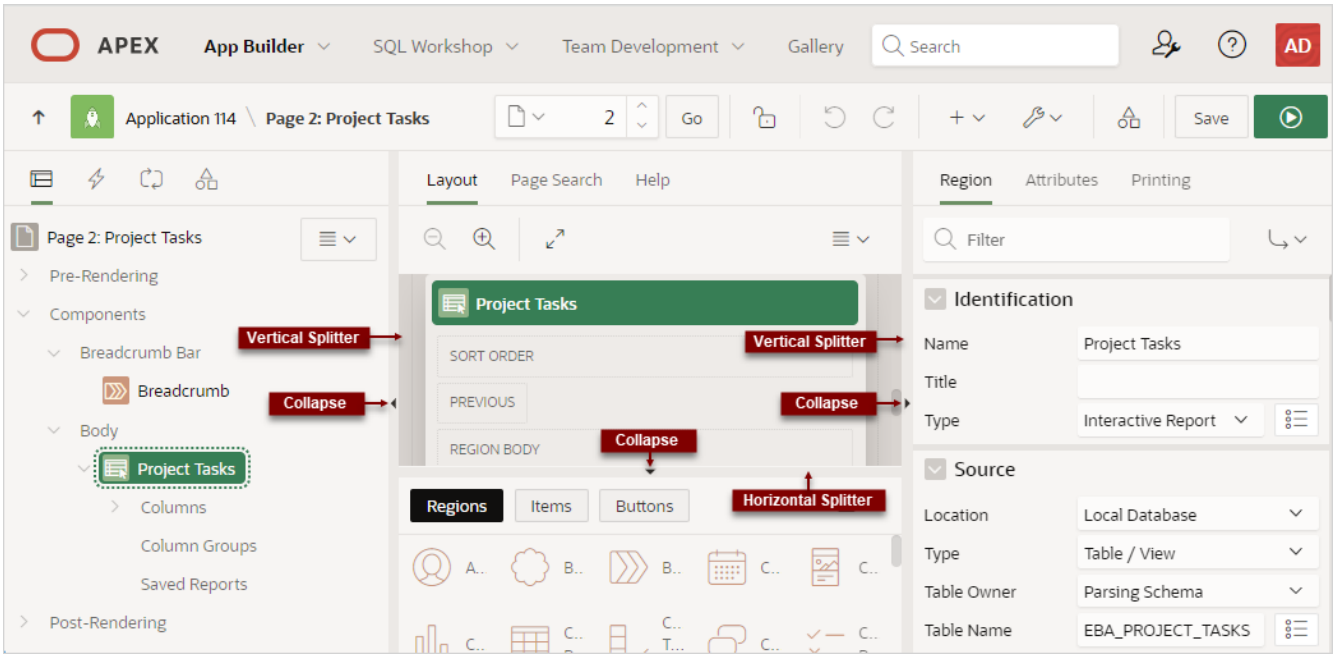
Developers view and edit application pages in Page Designer. Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.

- About Page Designer
  Use Page Designer to maintain and enhance pages within an Oracle APEX application.
- Understanding Page Designer UI Elements
  The Page Designer window features three panes: a left pane, a central pane, and a right pane. You select a component in the left or central pane and then edit the component's attributes in the Property Editor located in the right pane.
- Using Page Designer
  Use Page Designer to view and edit application pages.
- Managing Page Attributes
  Edit page attributes to control basic information and functionality for a given page, including the page name or title, header and footer text, navigation menus, inclusion of page-level JavaScript or CSS files, error handling, and the selected authorization scheme.
- Running a Page from Page Designer
  The Oracle APEX engine dynamically renders and processes pages based on data stored in database tables. To view a rendered version of an application, you run or submit it to the APEX engine.
- Runtime Developer Toolbar
  Use the Runtime Developer toolbar to quickly edit the current application or currently running page, or view session state.
- Deleting Pages
  Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

7.1 About Page Designer

Use Page Designer to maintain and enhance pages within an Oracle APEX application.

Page Designer is a full featured Integrated Development Environment (IDE) that includes a toolbar and multiple panes.



The Page Designer window is divided into three main panes:

- **Left Pane** - Includes four tabs that display as a tree: Rendering, Dynamic Actions, Processing, and Shared Components.
- **Central Pane** - Includes the following tabs: Layout, Component View, Page Search, and Help.
- **Right Pane** - Displays the Property Editor. Use the Property Editor to update attributes for the selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

You can adjust the size of each pane by selecting and dragging the horizontal and vertical splitters. To expand or collapse each pane, click the small triangle labeled Collapse in the center of each splitter.

Tip:

You can view an online summary that describes the Page Designer window. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

See Also:

- Understanding Page Designer UI Elements
- Viewing Help in Page Designer
- Editing Pages in Page Designer

7.2 Understanding Page Designer UI Elements

The Page Designer window features three panes: a left pane, a central pane, and a right pane. You select a component in the left or central pane and then edit the component's attributes in the Property Editor located in the right pane.

The **left** pane includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components. The **central** pane contains four tabs: Layout, Component View, Page Search, and Help and a Gallery pane at the bottom. The **right** pane contains the Property Editor.

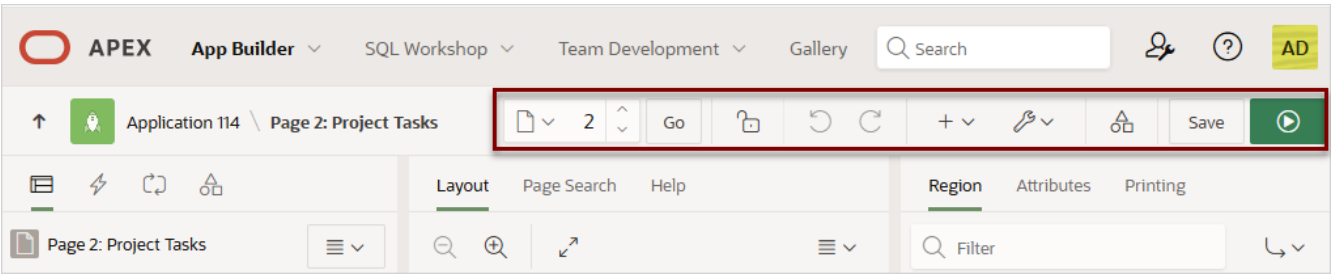
> **Tip:**
>
> You can view an online summary of this information. View a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

- Page Designer Toolbar
  The Page Designer toolbar displays at the top of the page and contains both buttons and menu options. Passing your cursor over an active button and menu displays a tooltip.
- Left Pane of Page Designer
  Use the left pane in Page Designer to select or add containers and components on a page.
- Central Pane of Page Designer
  The top of the central pane in Page Designer contains the **Layout**, **Page Search**, and **Help** tabs. The lower part of the central pane contains the **Gallery**.
- Right Pane of Page Designer (Property Editor)
  Learn about the Property Editor in the right pane of Page Designer.
- Page Designer Code Editor
  Learn about using the Code Editor in Page Designer.

> **See Also:**
>
> Using Page Designer

7.2.1 Page Designer Toolbar

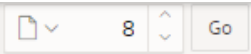
The Page Designer toolbar displays at the top of the page and contains both buttons and menu options. Passing your cursor over an active button and menu displays a tooltip.



> **Tip:**
>
> You can view an online summary of how to use Page Designer. To view a page in Page Designer, click the **Help** menu at the top of the page, and select **Getting Started with Page Designer**.

The Page Designer toolbar features the following buttons and menu options.

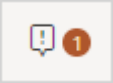
Page Selector



The **Page Selector** displays the current page. Click the down arrow (labeled **Page Finder**) to search for pages. Alternatively, enter a page number in the field and click **Go**. To navigate to the previous or next page, click **Navigate to Next Page** (up arrow) and **Navigate to Previous Page** (down arrow).

> **See Also:**
>
> Searching for Pages

Show Messages



When you add a new component to a page, the **Show Messages** icon displays current errors and warnings. Clicking a message changes the focus within Property Editor to the corresponding attribute associated with the error or warning.

Page Unlocked and Page Locked

Indicates the lock status of a page. **Page Unlocked** indicates the current page is unlocked and editable. **Page Locked** indicates the page is unavailable for editing. The appearance of the padlock icon changes depending upon the lock status:

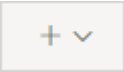
- Clear unlocked padlock — Indicates the page is unlocked.
- Red locked padlock — Indicates the page is locked by you.
- Red locked padlock with person — Indicates the page is locked by another user. To view the lock owner or update the lock comment, click the red padlock.

> **See Also:**
>
> Locking and Unlocking a Page

Undo and Redo

Applies to actions that result in a change to the page data. **Undo** reverts the previous update you made within Page Designer. **Redo** reapplies the last update that was undone using **Undo**.

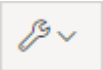
Create



Features a graphical plus sign (+). Create menu options include:

- **Page**. Access to the Create Page Wizard. See Adding a New Page to an Application.
- **Copy Page**. Access the Copy Page Wizard. See Copying a Page in Page Designer.
- **Breadcrumb Region**. Access to the Create Breadcrumb Wizard. See Creating Breadcrumbs and Adding a Breadcrumb Region with a Wizard.
- **Shared Component**. Access to the Create Application Component Wizard. Shared components are common elements that can display or be applied on any page within an application. See Managing Shared Components.
- **Page Group**. Links to the Page Group page. Use page groups to organize pages. See Managing Page Groups.
- **Developer Comment**. Access the Developer Comments dialog. Developers can add comments to an application, a page, or a group of pages. See Adding Developer Comments.
- **Issue**. Create a new issue in Team Development. See Creating an Issue.

Utilities



Features a graphic of a wrench. Utilities menu options include:

- **Delete Page**. Delete the current page. See Deleting Pages.
- **Advisor**. Access Oracle APEX Advisor (Advisor). Use Advisor to check the integrity and quality of your APEX application. See Running Advisor to Check Application Integrity.
- **Caching**. Links to the Caching page which displays a report of Cached Pages and Cached Regions. Enabling caching is an effective way to improve the performance. See Enabling Region Caching.
- **History**. Displays a report of changes to the current page.
- **Checksum**. View the page checksum. You can also access the page checksum from the Application Utilities page. See Viewing the Page Checksum and Using Application Utilities.
- **Export**. Export the current page. See Exporting Workspaces, Applications, and App Components.
- **Show**. Includes the following options:
  - **Tooltips**. Disable and enable tooltips within the UI.
  - **Text Messages Picker**. Enable and disable Messages Picker dialog within the UI. See Selecting Translation Messages in Page Designer.

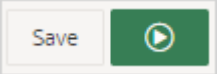
  - **Layout View**. Disable or enable the Layout tab in the central pane of Page Designer. See Layout Tab.

- **Layout**. Controls the number of panes that display in Page Designer. See Switching Between Three Pane and Two Pane Mode.

Shared Components

Links to the Shared Components page. Shared components can display or be applied on any page within an application. See Managing Shared Components.

Save and Save and Run Page



Click **Save** to save the current page. Click **Save and Run Page** to save and then run the current page. See Running a Page from Page Designer.

7.2.2 Left Pane of Page Designer

Use the left pane in Page Designer to select or add containers and components on a page.

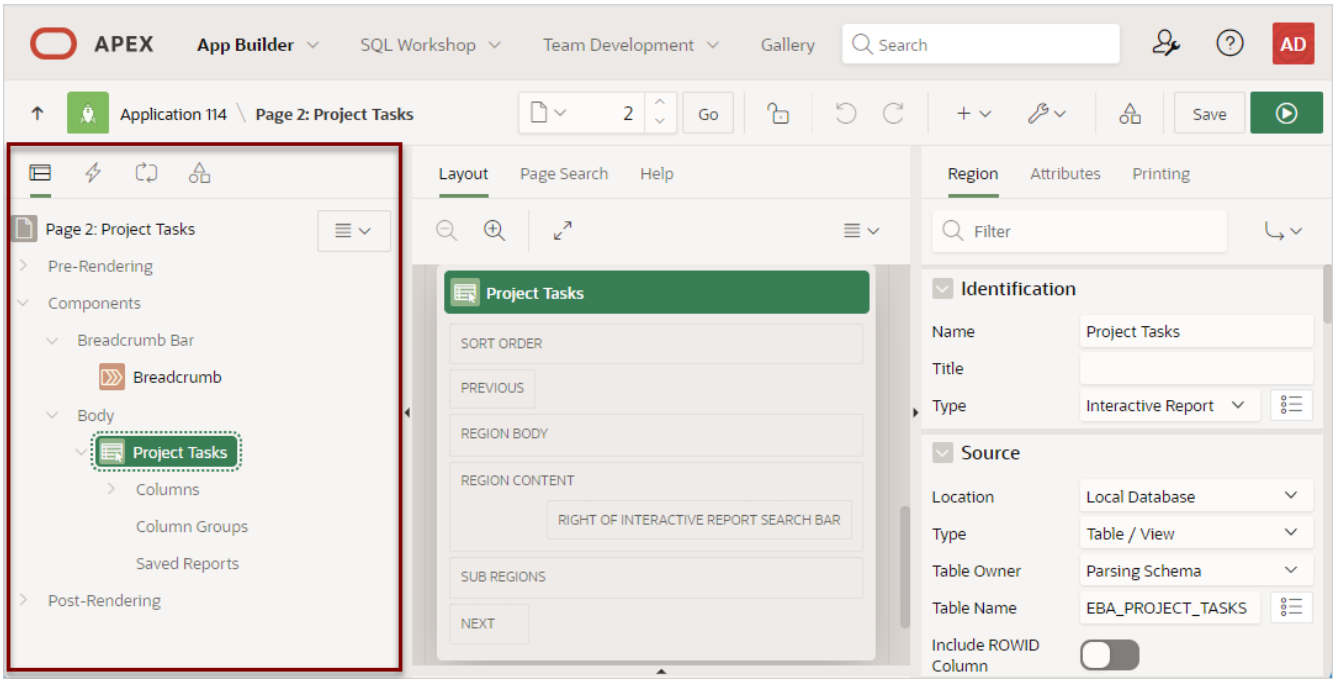
Once selected, edit the container or component attributes in the Property Editor located in the right pane.

- About the Left Pane of Page Designer
  The left pane in Page Designer includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components.
- Rendering Tab
  Learn about the Rendering tab in the left pane of Page Designer.
- Dynamic Actions Tab
  Learn about the Dynamic Actions tab in the left pane of Page Designer.
- Processing Tab
  Learn about the Processing tab in the left pane of Page Designer.
- Page Shared Components Tab
  Learn about the Page Shared Components tab in the left pane of Page Designer.

7.2.2.1 About the Left Pane of Page Designer

The left pane in Page Designer includes four tabs: Rendering, Dynamic Actions, Processing, and Page Shared Components.

Each tab displays a list of the corresponding component types and components created on the current page.



Key features of the Left pane include:

- **Context menus**. Right-click a component or control to display a context menu. You can also click the Rendering menu to select options for the selected component.
- **Quick Access to the Property Editor**. Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop.** Drag components up and down within the tree to change the position or sequence of the selected component.
- **Tooltips**. Position the mouse over a component or control to view a tooltip of basic information, such as the region type, item type, and so on.

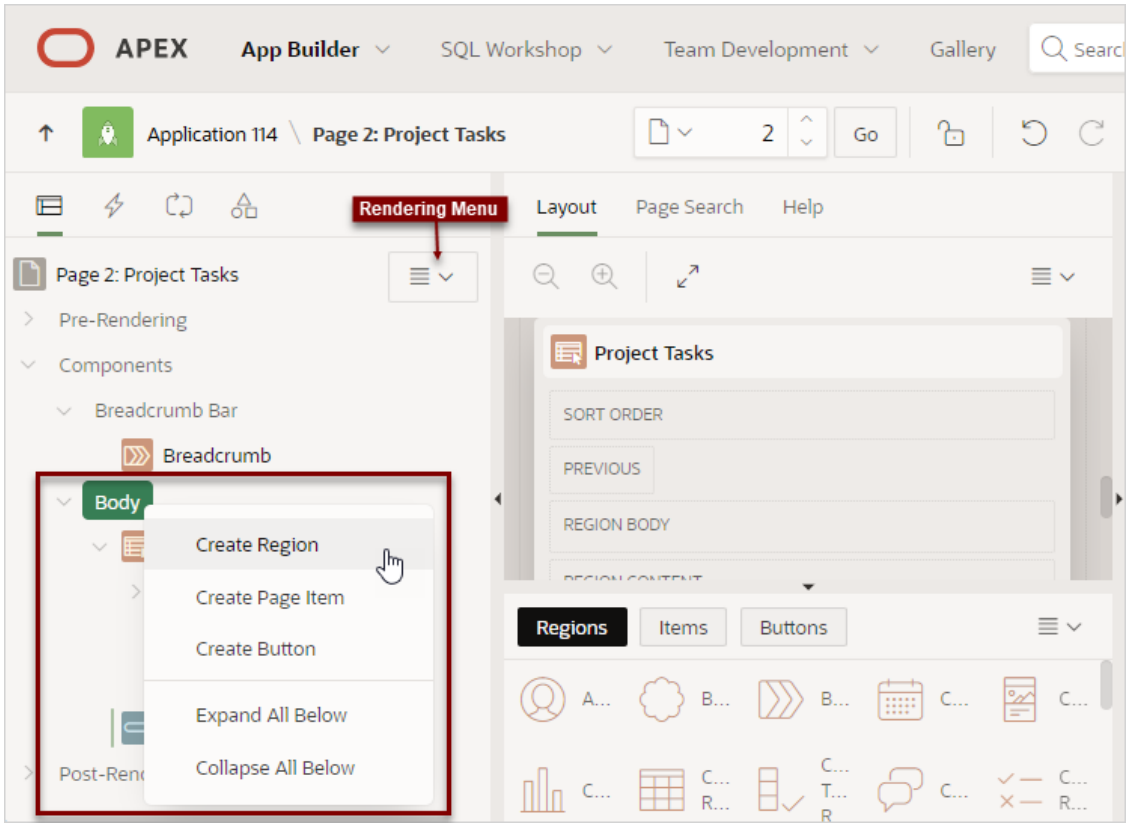
7.2.2.2 Rendering Tab

Learn about the Rendering tab in the left pane of Page Designer.

The Rendering tab displays containers and components that are rendered on a page. Components include regions, page items, and buttons. Application logic such as Dynamic Actions and Validations display inside of regions, page items, and buttons when applicable.

Components can placed at the page-level at a specific page position, or they can be nested inside regions at a specific region position. Additionally, the Rendering tab displays Pre-Rendering and Post-Rendering application logic such as Processes, Computations, and Branches. The Rendering tab groups and orders components based on how Oracle APEX processes them at runtime

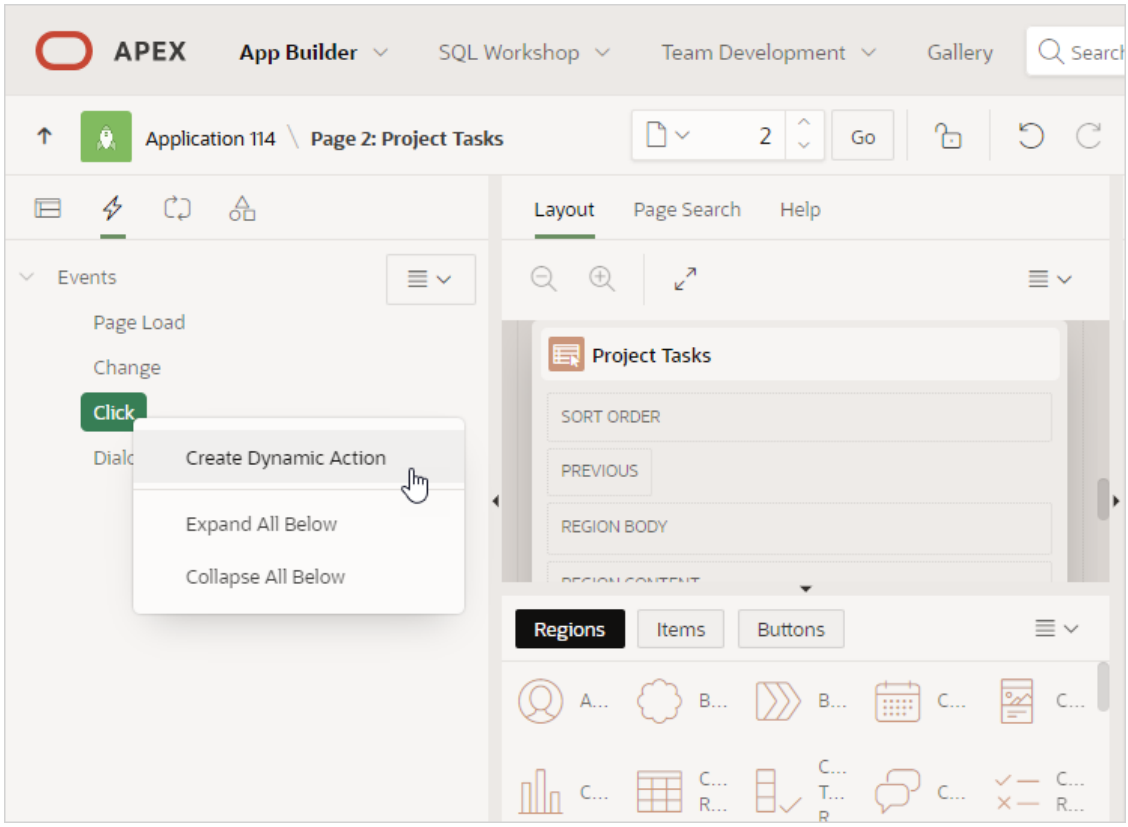
To create a new component, right-click and select an option from the context menu. Alternatively, select an option from the Rendering menu.



7.2.2.3 Dynamic Actions Tab

Learn about the Dynamic Actions tab in the left pane of Page Designer.

The Dynamic Actions tab displays dynamic actions defined on the current page. By creating a dynamic action, you can define complex client-side behavior declaratively without the need for JavaScript. To create a new dynamic action, right-click and select an option from the context menu.



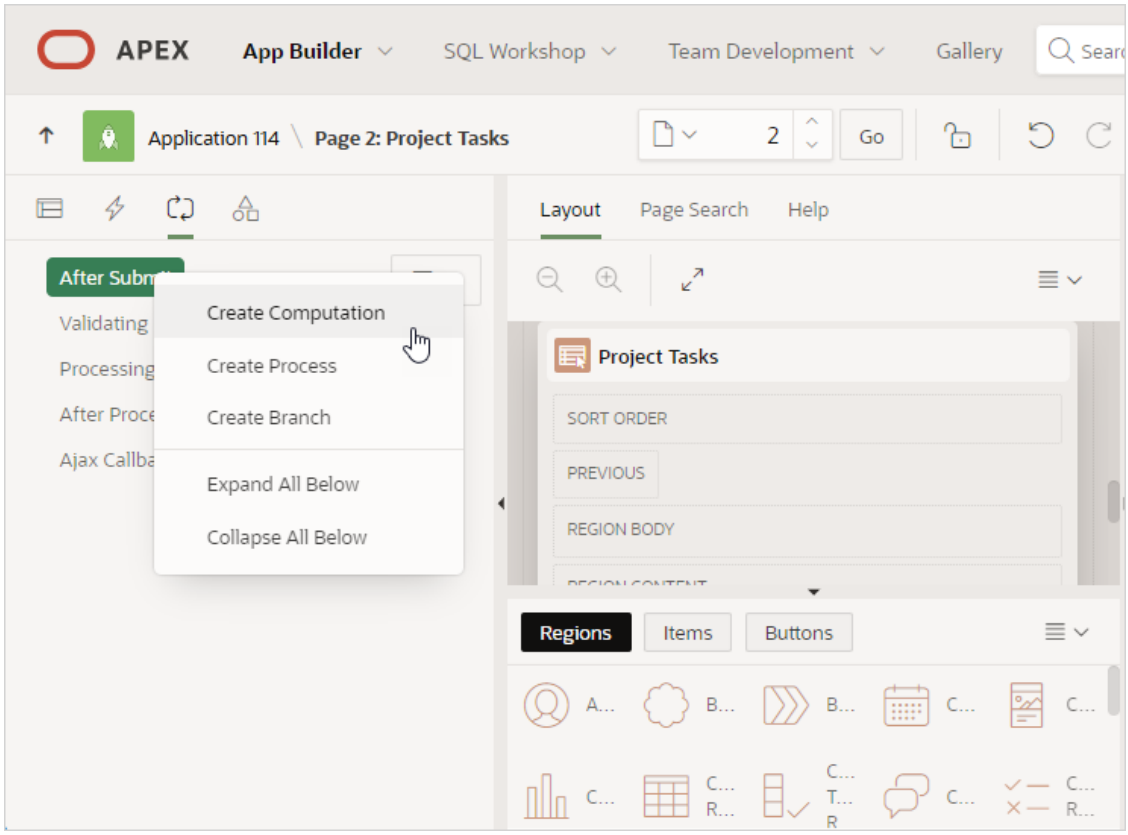
See Also:

Managing Dynamic Actions

7.2.2.4 Processing Tab

Learn about the Processing tab in the left pane of Page Designer.

The Processing tab displays application logic defined on the page and groups and orders components based on how APEX processes them. To create a new process, right-click and select an option from the context menu.



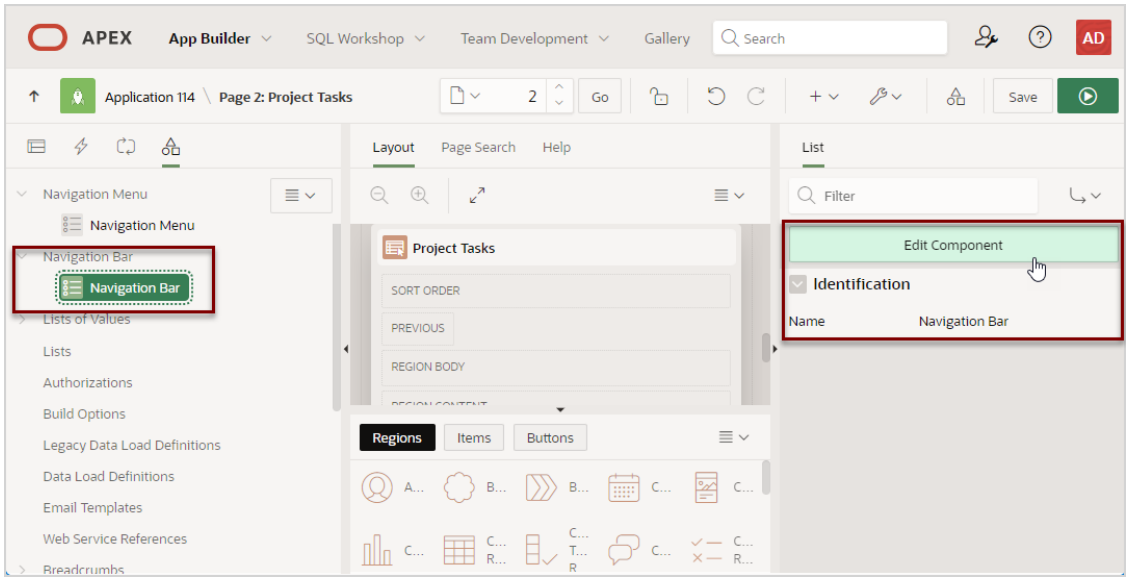
> **See Also:**
>
> Managing Computations, Validations, and Processes

7.2.2.5 Page Shared Components Tab

Learn about the Page Shared Components tab in the left pane of Page Designer.

The Page Shared Components tab displays shared components associated with the current page. To edit a Shared Component, select it in the Page Shared Components tab and then click **Edit Component** in right pane.

In the following example, **Navigation Bar** is selected in the left pane and the cursor is positioned on **Edit Component** button in the right pane.



> **See Also:**
>
> Managing Shared Components

7.2.3 Central Pane of Page Designer

The top of the central pane in Page Designer contains the **Layout**, **Page Search**, and **Help** tabs. The lower part of the central pane contains the **Gallery**.

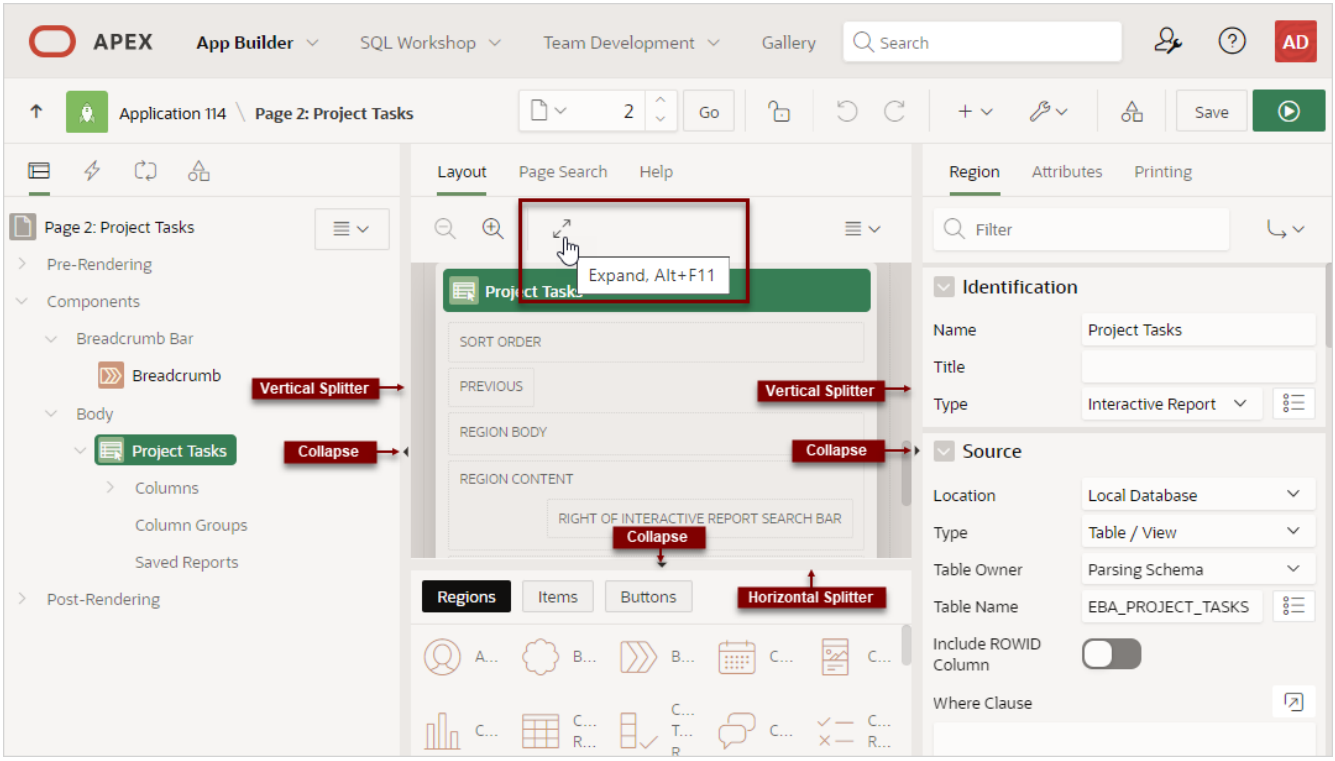
- Layout Tab
  Learn about the Layout tab in the central pane of Page Designer.
- Page Search Tab
  Learn about the Page Search tab in the central pane of Page Designer.
- Help Tab
  Learn about the Help tab in the central pane of Page Designer.
- Gallery
  Learn about the Gallery in the central pane of Page Designer.

7.2.3.1 Layout Tab

Learn about the Layout tab in the central pane of Page Designer.

The Layout tab is a visual representation of how the components are positioned on the page. The Layout tab features context menus. By selecting a component and right-clicking, you can delete, move, or copy the component to other regions, or new regions on the page. You can also move existing regions, items, and buttons relative to other components by simply clicking on the component and dragging it to the new location. For example, you can place items next to one another by dragging the second item to the end of the first item and dropping it in the dark yellow box that appears when you hover in the desired location. You can quickly add new components to an existing page by dragging the component from the Gallery up to the desired position within the Layout tab.

You can adjust the size of each pane by selecting and dragging the horizontal and vertical splitters. To expand or collapse each pane, click the small triangle labeled **Collapse** in the center of each splitter. Alternatively, you can click the **Expand** button to make the Layout tab larger and then **Restore** to return it to the previous size.



> **Tip:**
>
> Hidden items do not display in the Layout tab, but do display in the Rendering tree.

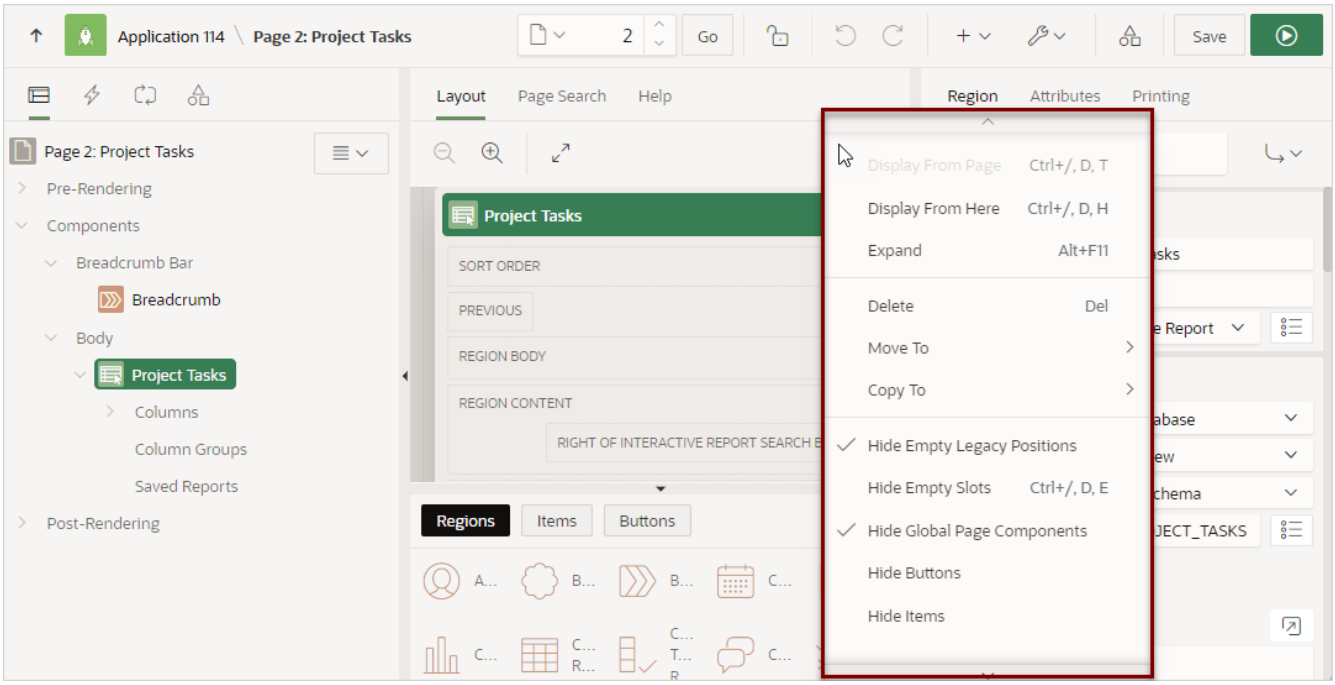
Key features of the Layout tab include:

- **Context menus**. Right-click a component or control to display a context menu.
- **Quick Access to the Property Editor**. Select a component or multiple components to display the corresponding attributes in the Property Editor in the right pane.
- **Drag and Drop** . Move, copy, and reorder regions, items, and buttons by dragging and dropping. You can also add new regions, items, and button by dragging them from the Gallery and dropping them to the desired position on the page.
- **Tooltips**. Position the mouse over a component or control to view a tooltip of basic information, such as the region type, item type, and so on.

> **Tip:**
>
> When dragging components to a new position, such as dragging an item onto the same line as an exiting item, you need to drag the component into the desired position and wait until the grid changes and the new drop position displays. See Adding a Region, Item, or Button by Dragging and Dropping.

Layout Menu

The Layout menu displays on the right side of the pane. Use this menu to customize the type and amount of information that displays and to delete, move, and copy selected components. To have a specific region fill the the Layout tab, select the region and select **Display from Here** from the Layout menu. The selected region fills the Layout tab. To restore the view, select **Display from Page**.

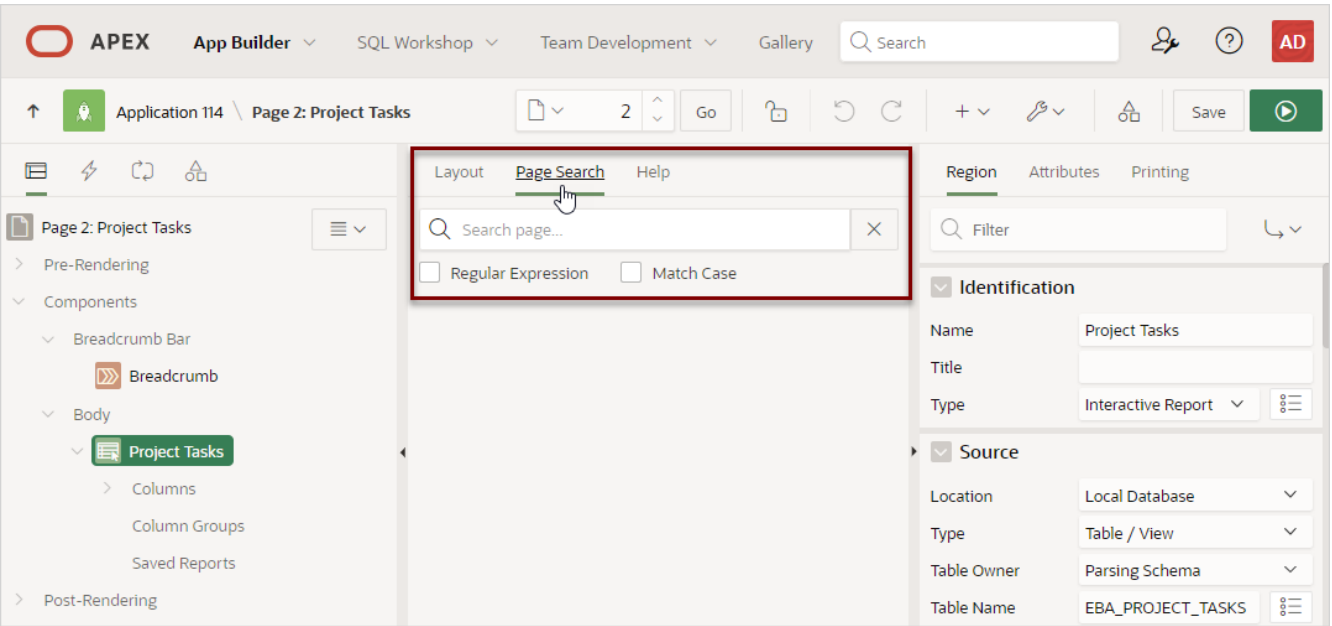


> **See Also:**
>
> Adding a Region, Item, or Button by Dragging and Dropping

7.2.3.2 Page Search Tab

Learn about the Page Search tab in the central pane of Page Designer.

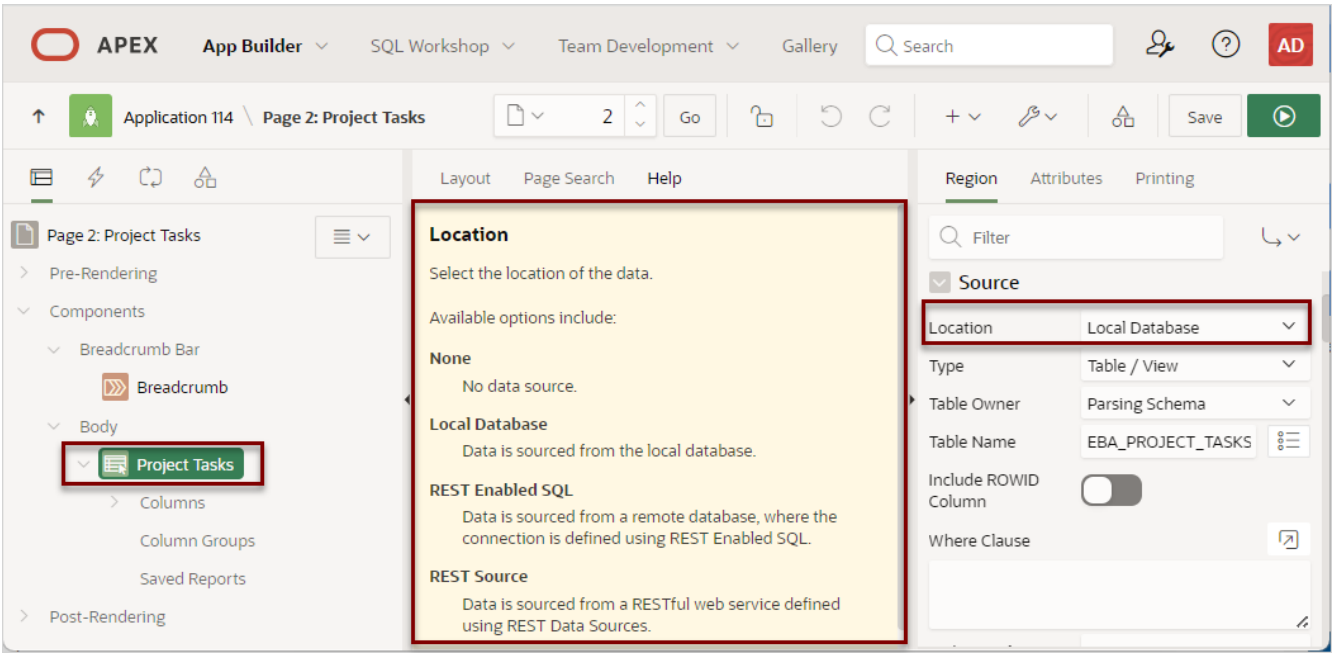
The Page Search tab displays to the right of Layout in the central pane. Use Page Search to search all page metadata including regions, items, buttons, dynamic actions, columns, and so on. To search a page, enter a search terms in the field provided. To match the case, select **Match Case**. To search for a regular expression, select **Regular Expression**.



7.2.3.3 Help Tab

Learn about the Help tab in the central pane of Page Designer.

Page Designer includes Help for every Property Editor attribute. To view Help, select the attribute and click the **Help** tab. Once you activate the Help pane, the content that displays changes every time you select another attribute. In this example, the **Source**, **Type** attribute is selected and the Help tab describes what the attribute is for and the available options.

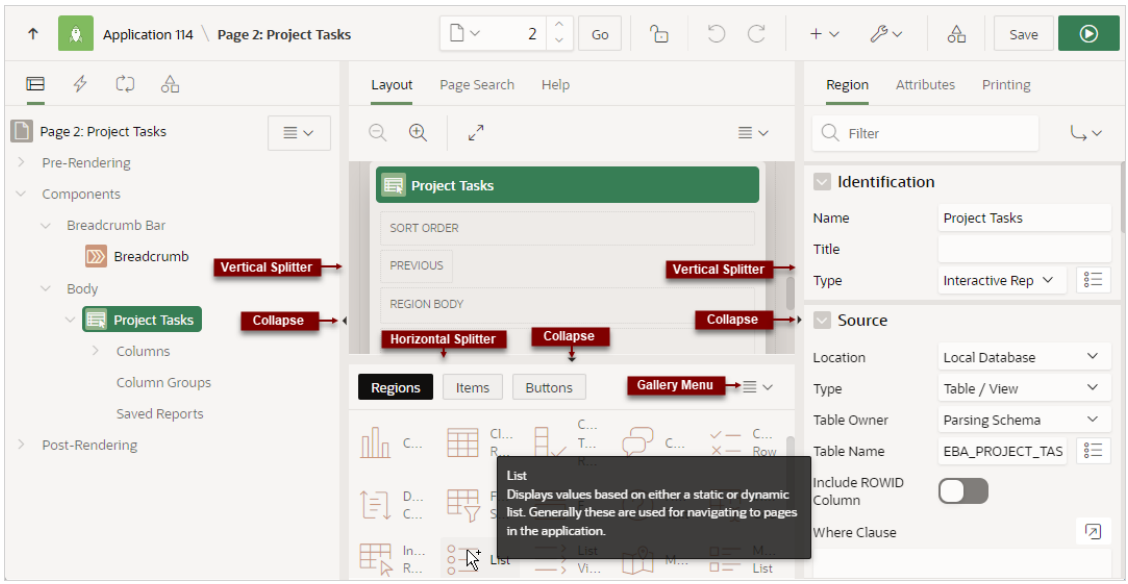


> **See Also:**
>
> Viewing Help in Page Designer

7.2.3.4 Gallery

Learn about the Gallery in the central pane of Page Designer.

The Gallery displays beneath the Layout tab in the central pane of Page Designer. The Gallery contains three tabs: Regions, Items, and Buttons. When you pass the cursor over a control or component, a tooltip displays that describes it. You can adjust the size of the pane by dragging the horizontal and vertical splitters. To collapse or restore the Gallery pane, click the small triangle in the center of each splitter.



You can add new controls and components to a page by selecting the control or component from the Gallery and dragging and dropping them into the Layout tab. Alternatively, you can select a control or component in the Gallery and right-click to view a context menu.

Gallery Menu

The Gallery menu displays on the right side of the Gallery pane. By default, the Gallery only displays supported controls and components.

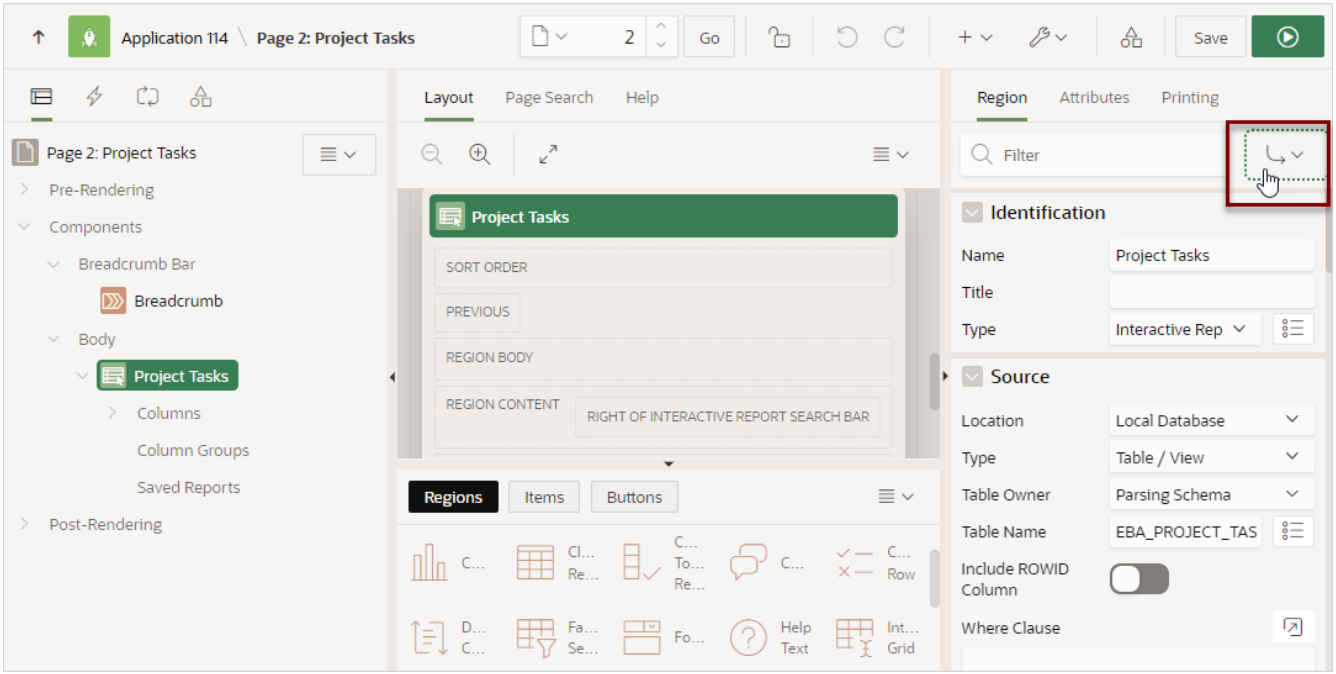
- Adding a Region, Item, or Button by Dragging and Dropping
- About Adding Components Using Gallery Context Menus
- Managing Application Legacy Components

7.2.4 Right Pane of Page Designer (Property Editor)

Learn about the Property Editor in the right pane of Page Designer.

Property Editor

Use the Property Editor to edit the attributes of the currently selected component. Property Editor displays all attributes for the currently selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

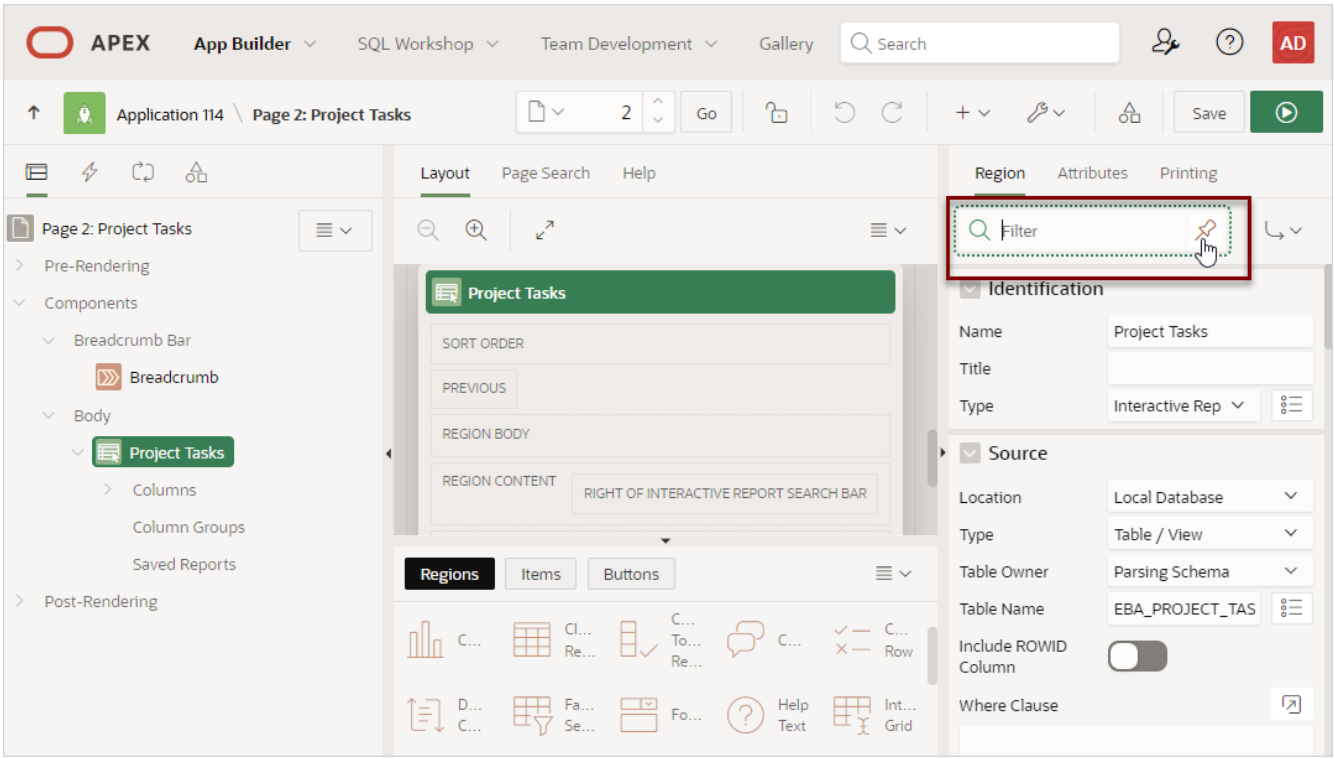


Property Editor organizes attributes into functional groups. To quickly access a group, click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

Searching for Attributes

To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the keyword appear. To return to the default display, delete the keyword in the Filter Properties field.

Click the **Pin Filter** button on the right side of the search field to search across different items, regions, and processes. You can “pin” keywords to have them persist as you click around Page Designer. To “unpin” the search, click the **Pin Filter** button again.



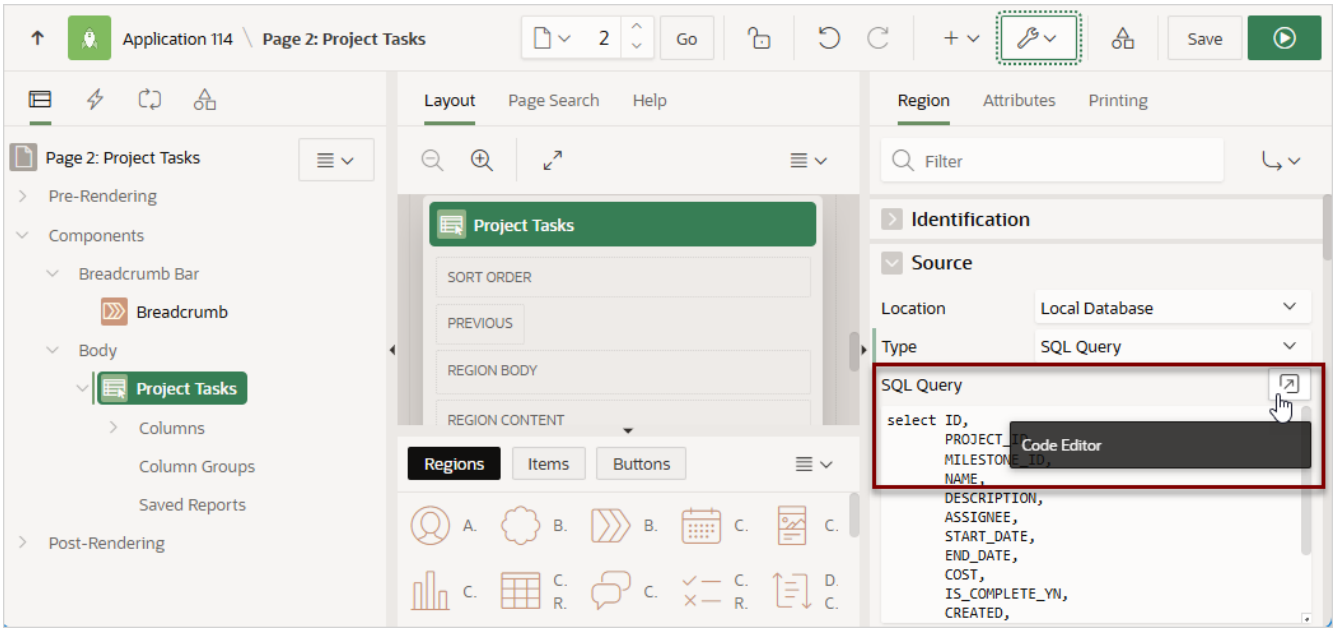
See Also:

- Using Page Designer
- Managing Page Attributes

7.2.5 Page Designer Code Editor

Learn about using the Code Editor in Page Designer.

For attributes requiring large amounts of code (for example, a region Source), you can enter text directly into the field or access a full-featured Code Editor.



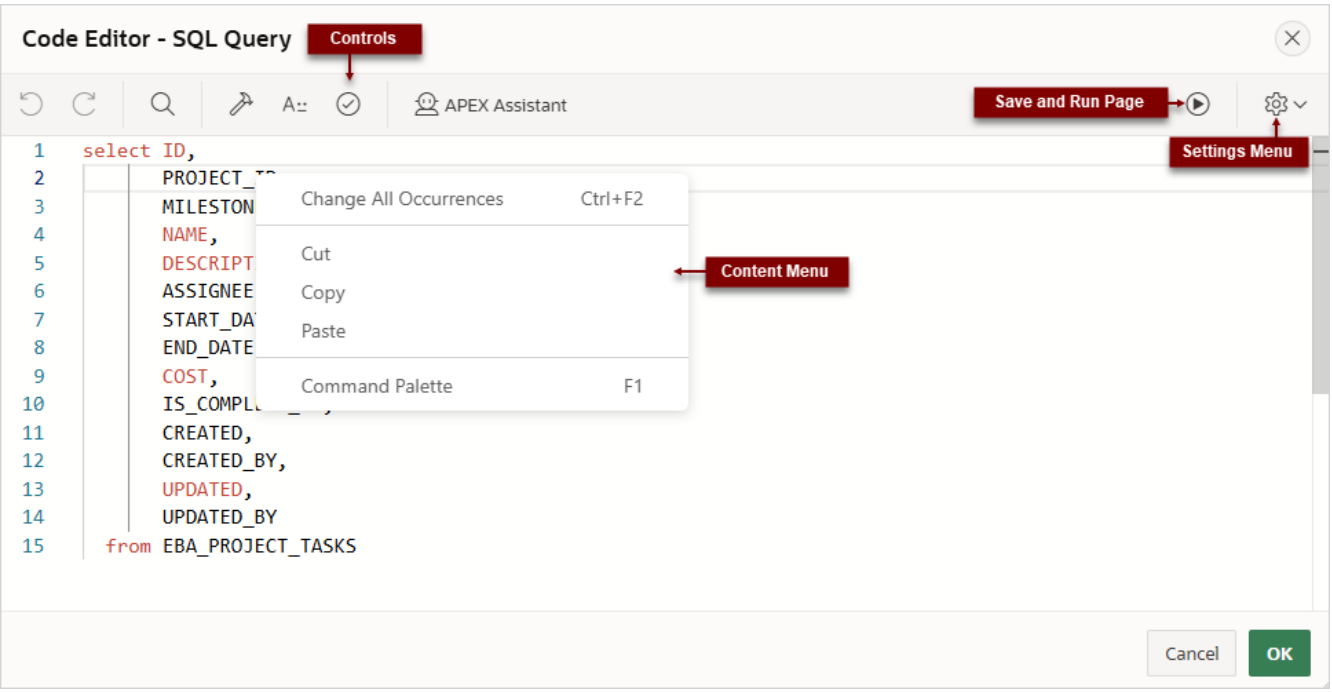
> **See Also:**
>
> Accessing Property Editor

You access Code Editor by clicking the **Code Editor:SQL Query** button, which resembles an arrow within a square.

Code Editor Key Features

> **Note:**
>
> If you have configured Generative AI for your workspace, APEX includes integrated AI Chat capability (called APEX Assistant) in all all Code Editors. To learn more, see Managing Generative AI in APEX.

The Code Editor displays as a modal dialog and provides an enhanced code editor to edit PL/SQL, SQL, HTML, CSS, and JavaScript component properties. The code highlighting is determined by the input required for the specific property.



Code Editor controls include:

- Undo - Ctrl+Z
- Redo - Ctrl+Shift+Z
- Find - Ctrl+F
- Query Builder
- Auto Complete - Ctrl+Space
- Validate - Ctrl+Alt+V
- Context Menu - Right-click and select one of the following:
  - Change All Occurrences (Ctrl+F2) - Right click and select **Change All Occurrences** from the context menu.
  - Cut

- Copy
- Command Palette (F1) - Displays the Command Palette.

> **See Also:**
>
> Accessing Oracle APEX Using a Keyboard in *Oracle APEX Accessibility Guide*

Command Palette

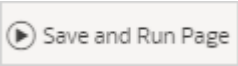
Right-click to display a context menu and select **Command Palette**. The Command Palette includes over 80 commands including:

- Replace (Ctrl+H)
- Replace with Next Value (Ctrl+Shift+.)
- Replace with Previous Value (Ctrl+Shift+,)

Multi Cursor Selection

- Add cursors at arbitrary positions - Select a position with your mouse and press Shift+Alt+click (Shift+Option+click on macOS).
- Set cursors above the current position - Select a position with your mouse and press Ctrl+Alt+Up (Option+Command+Up on macOS).
- Set cursors below the current position - Select a position with your mouse and press Ctrl+Alt+Down (Option+Command+Down on macOS).

Save and Run Page



Click **Save and Run Page** to save and view a rendered version of the current page without having to close the Code Editor dialog.

> **Tip:**
>
> For **global** and **dialog** pages, the **Save and Run Page** button only saves the page. To view a rendered version of a dialog page, you must run it from the page that is responsible for opening it.

Settings Menu

The **Settings** menu displays on the right side and includes the following options:

- Indentation - Options include: Tab Inserts Spaces, Tab Size, and Indent Size
- Themes - Options include: Automatic, Light, Dark, and High Contrast Dark
- Show Ruler

7.3 Using Page Designer

Use Page Designer to view and edit application pages.

> **Tip:**
>
> To view a list of keyboard shortcuts, press **Alt+Shift+F1**. Alternately, click the **Help** menu at the top of the page and select **Shortcuts**.

- Working with Page Designer
  Learn how to view pages in Page Desinger and manage the left, central, and right window panes.
- Managing Regions, Items, and Buttons
  Learn about creating and copying regions, page items and buttons in Page Designer.
- Creating Branches, Validations, Computations, and Processes
  Learn about managing branches, validations, computations, and processes in Page Designer.
- Copying a Page in Page Designer
  Copy a page in Page Designer by selecting Copy Page from the Create menu on the Page Designer toolbar.
- Commenting Out Page Components in Page Designer
  Exclude or include page components at runtime using the Comment Out and Uncomment context menu option in Page Designer.

> **See Also:**
>
> - Understanding Page Designer UI Elements
> - Viewing Help in Page Designer

7.3.1 Working with Page Designer

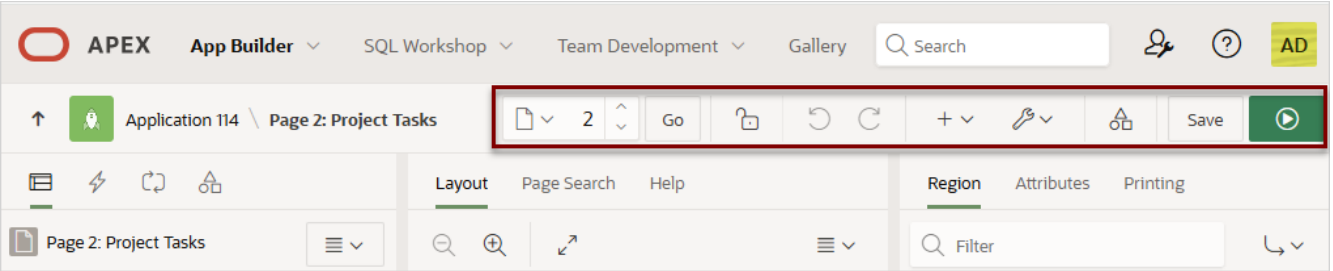
Learn how to view pages in Page Desinger and manage the left, central, and right window panes.

- Searching for Pages
  Use the Page Selector to search for pages within the current application.
- Viewing a Page in Page Designer
  To view a page in Page Designer, select an application and then a page.
- Accessing Property Editor
  The Property Editor displays in the right pane of Page Designer and displays all attributes for currently selected components.
- Dragging and Dropping Tabs Between Panes
  Drag and drop tabs within a Page Designer pane.

- Switching Between Three Pane and Two Pane Mode
  Use the Layout options on the Utilities menu on the Page Designer toolbar to switch between pane modes.
- About Adding Components Using Gallery Context Menus
  Add a component to a page by right-clicking a component in the Gallery to view a context menu.

7.3.1.1 Searching for Pages

Use the Page Selector to search for pages within the current application.



The Page Selector is the first control in the Page Designer toolbar and displays the current page number.

To search for pages within an application:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. On the Page Designer toolbar, click the **Page Selector**.

   The Page Finder appears.
3. Click the tabs at the top of the window to change the view:
   - **All Pages** - Displays all pages in the current application.
   - **Recently Edited** - Displays recently edited pages.
4. Enter search terms in the Search field and press **Enter** or click **Go**.

   Search for page numbers, page names, user interface, or group names. The search is not case sensitive and supports regular expressions.

See Also:

- Running a Page from Page Designer
- Managing Page Groups

7.3.1.2 Viewing a Page in Page Designer

To view a page in Page Designer, select an application and then a page.

To view a page in Page Designer:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
4. Select a page.

   The default view, Page Designer, appears.

   You preview a page by running it. Running an application page displays the rendered version as end users would see it.

> **See Also:**
>
> Running a Page from Page Designer

7.3.1.3 Accessing Property Editor

The Property Editor displays in the right pane of Page Designer and displays all attributes for currently selected components.

To access Property Editor:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the left pane or the Layout tab, select components or controls to edit. Passing the cursor over a control or component, displays a tooltip that describes it.

   > **Tip:**
   >
   > Once you select a component, press ALT+6 to navigate directly to the Property Editor. To learn more, see *Oracle APEX Accessibility Guide*

   The Property Editor displays attributes for the selected component or control. As you select different components in either the left pane or the Layout tab, the Property Editor automatically updates to reflect the selected component. When you select multiple components, the Property Editor only displays common attributes. Updating a common attribute updates that attribute for all of the selected components.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

7.3.1.4 Dragging and Dropping Tabs Between Panes

Drag and drop tabs within a Page Designer pane.

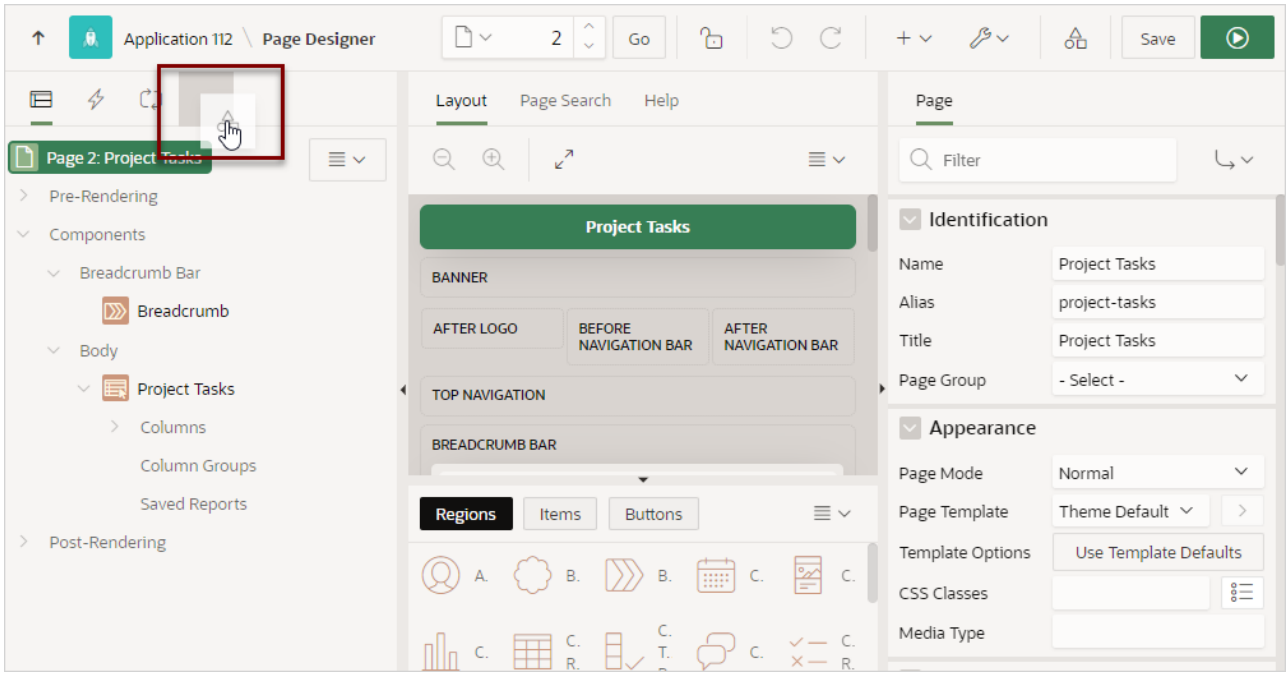
By default, Page Designer displays three panes which contain one or more tabs. You can drag and drop tabs within a pane or between panes by selecting the pane heading (or icon) and then dragging and dropping it to a new location. To return to the default display, click **Utilities** on the Page Designer toolbar and select **Layout** and then **Reset Layout**.

To drag and drop tabs in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Select a tab heading (or icon) and then drag and drop it to a new location.

   In the following example, the **Page Shared Components** tab is selected and being moved within the left pane.



3. To return to the default display:
   a. On the Page Designer toolbar, click **Utilities**.
   b. Select **Layout** and then **Reset Layout**.

7.3.1.5 Switching Between Three Pane and Two Pane Mode

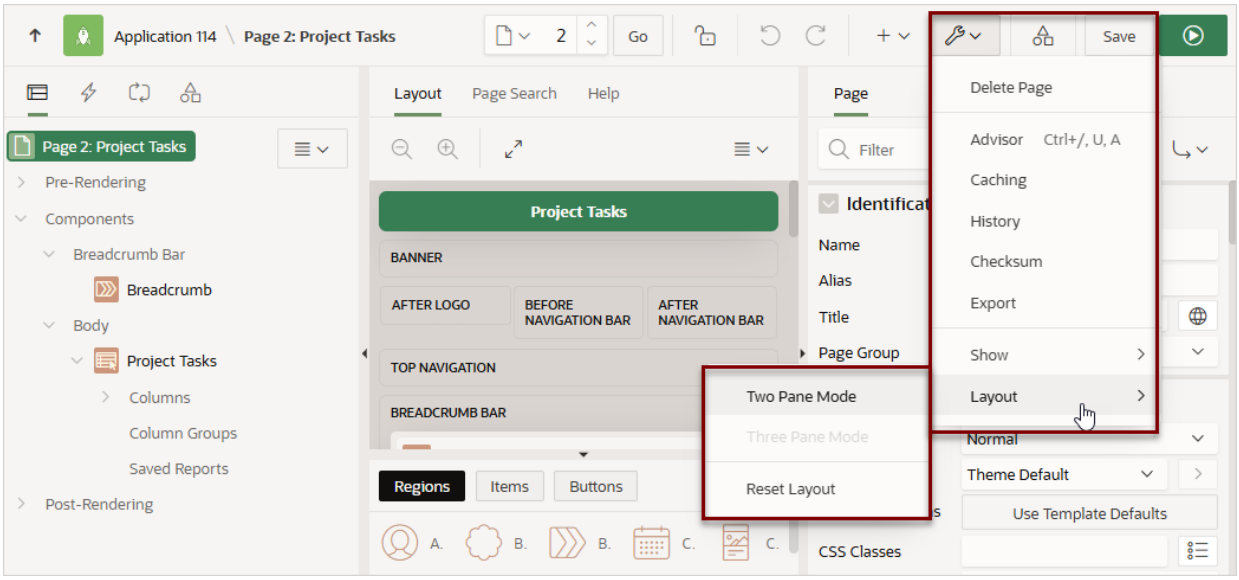
Use the Layout options on the Utilities menu on the Page Designer toolbar to switch between pane modes.

Three Pane Mode (the default) displays the left pane, central pane, and the right pane. **Two Pane Mode** displays just the central pane and the right pane. A typical use case for accessing Two Pane Mode, is when you want to view just the Rendering tree and the Property Editor. To return to the default display, click **Utilities** on the Page Designer toolbar and then select **Layout** and then **Reset Layout**.

To hide and show Page Designer panes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. To select a page mode:
   a. On the Page Designer toolbar, click **Utilities** and then **Layout**.



The Layout menu has three options: **Two Pane Mode**, **Three Pane Mode**, and **Reset Layout**.

b. From Layout, select one of the following:

- **Two Pane Mode** - Displays just the central pane and the right pane.
- **Three Pane Mode** (the default) - Displays the left pane, central pane, and the right pane.

3. To return to the default display:

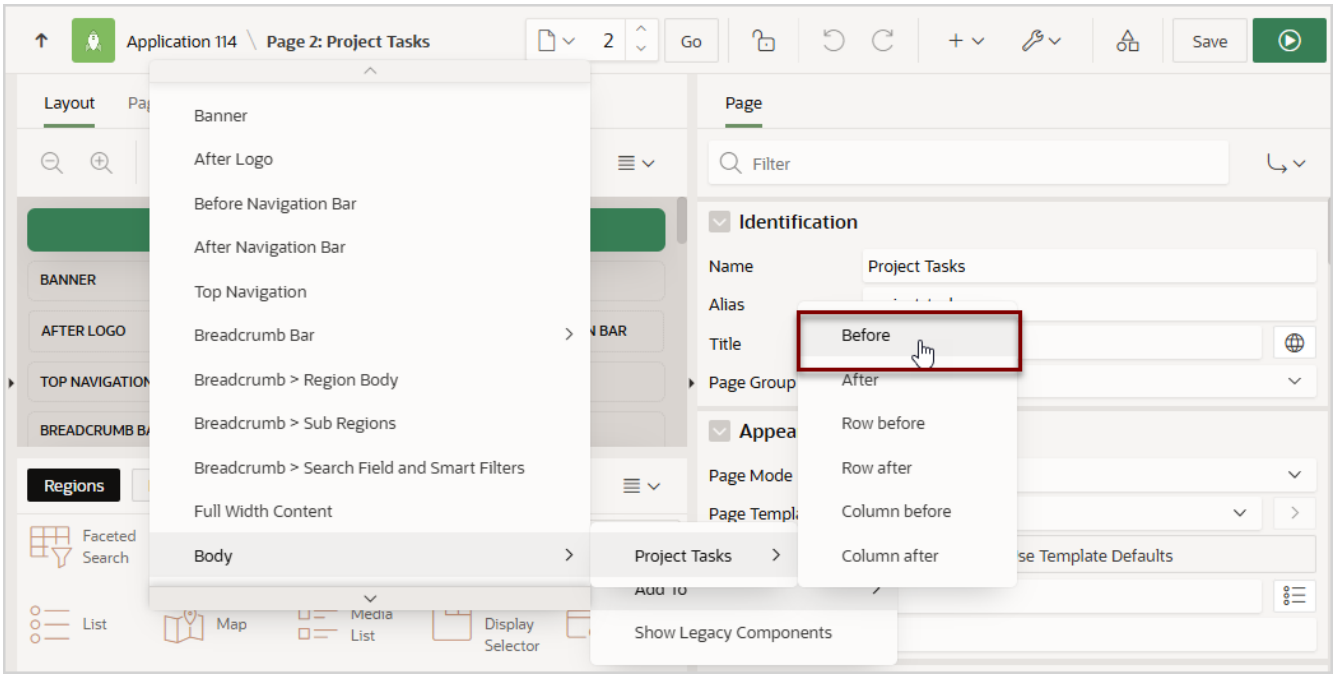
a. On the Page Designer toolbar, click **Utilities**.

b. Select **Layout** and then **Reset Layout**.

7.3.1.6 About Adding Components Using Gallery Context Menus

Add a component to a page by right-clicking a component in the Gallery to view a context menu.

In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Each context menu features an Add To option which displays the actual locations where you can add the component. In this example, the developer is adding a new report to before the Content Body region.



> **See Also:**
>
> Adding a Region, Item, or Button by Dragging and Dropping

7.3.2 Managing Regions, Items, and Buttons

Learn about creating and copying regions, page items and buttons in Page Designer.

- Adding a Region, Item, or Button by Dragging and Dropping
  Add regions, page items and buttons by dragging and dropping from the Gallery.
- Adding a Region, Item, or Button from the Rendering Tab
  Add a component or container from the Rendering tab by right-clicking an existing component or container to display a context menu.
- Copying a Region, Item, or Button
  Copy regions, items, and buttons by using context menus or pressing **CTRL** and dragging and dropping with the mouse.

> **See Also:**
>
> - Managing Buttons
> - Managing Page Items
> - Managing Regions

7.3.2.1 Adding a Region, Item, or Button by Dragging and Dropping

Add regions, page items and buttons by dragging and dropping from the Gallery.

Each application page consists of regions, page items and buttons. Those components can be added to page positions or region positions, under the overarching container called **Components**.

To add a region, item, or button by dragging and dropping from the Gallery:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. If needed, select the Layout tab in the central pane.
3. In the Gallery at the bottom of the central pane, locate the component or control you want to create.

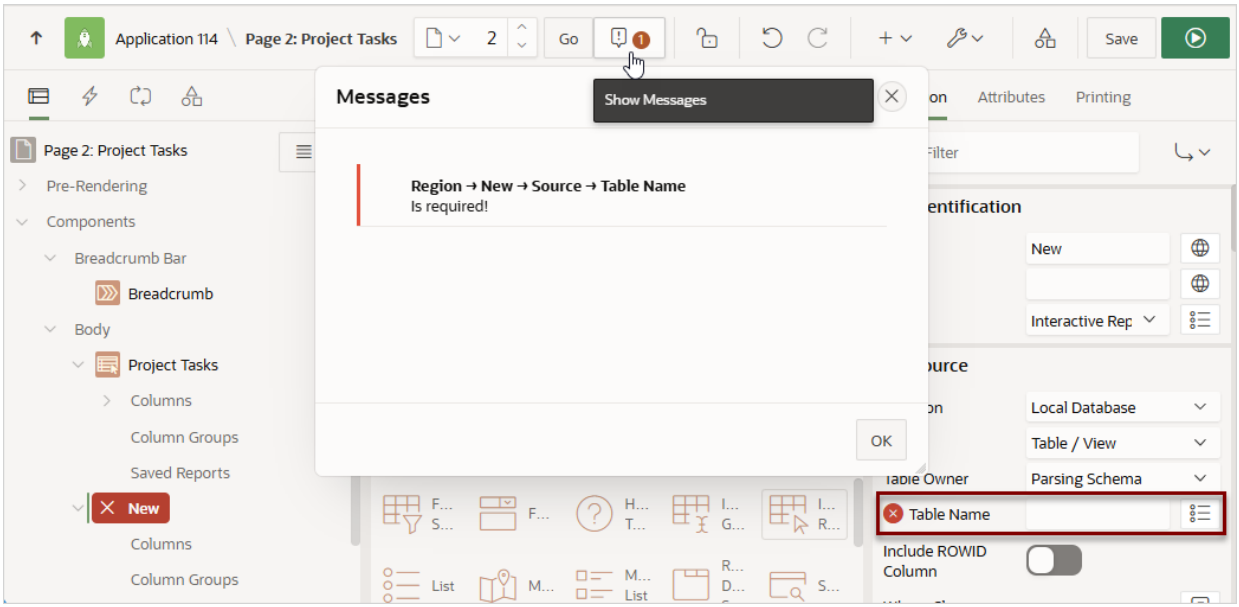
   The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.

   > **Tip:**
   >
   > In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.

4. Click and hold the mouse on the component to be created and drag it the desired location in the Layout tab.

   When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.
5. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.



b. Select an error to highlight the associated attribute in the Property Editor.

6. Edit the appropriate attributes in the Property Editor.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

7.3.2.2 Adding a Region, Item, or Button from the Rendering Tab

Add a component or container from the Rendering tab by right-clicking an existing component or container to display a context menu.

To create new component or container in the Rendering tab:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In the left pane, select the **Rendering** tab.
3. Locate and select the appropriate region, item, or region button.
4. Right-click an existing component or container to display the context menu for that component type.
5. Select a command (for example, Create Region, Create Sub Region, Create Page Item, Create Button and so on). The options available depend upon the node you select.
6. Edit the appropriate attributes in the Property Editor.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

> **Tip:**
>
> To add a legacy component, from the Type attribute in the Property Editor, select **Show Legacy...**. Legacy components are added to the list. Although Oracle still supports legacy application components, Oracle does not recommend using them.

7. Click **Save** or **Save and Run Page**.

> **See Also:**
>
> - Right Pane of Page Designer (Property Editor)
> - Managing Regions
> - Viewing Page Items
> - Creating a Button

7.3.2.3 Copying a Region, Item, or Button

Copy regions, items, and buttons by using context menus or pressing **CTRL** and dragging and dropping with the mouse.

Each application page consists of regions, page items and buttons. Those components can be added to page positions or region positions, under the overarching container called **Components**. Developers can copy regions, items, and buttons by using context menus or pressing **CTRL** and dragging and dropping with the mouse.

Copying a Region, Item, or Button Using Context Menus

To copy a region, item, or button using context menus:

1. View the page in Page Designer.
2. In the central pane, select the **Layout** tab.
3. Locate and select the appropriate region, item, or region button.
4. Right-click to display the context menu.
5. From the context menu, select **Copy To** and then select the location.
6. Edit the component attributes.

Copying a Region, Item, or Button by Dragging and Dropping

To copy a region, item, or button by dragging and dropping:

1. View the page in Page Designer.
2. In the left pane, select the **Rendering** tab.
3. Locate and select the appropriate region, item, or region button.
4. Press **CTRL** and then use the mouse to drag and drop the component to the new location.

   A copy of the component appears in the new location.

5. Edit the component attributes.

Copying a Region, Item, or Button to Another Page

To copy a region, item, or button to another page:

1. View the page in Page Designer.
2. In the left pane, select the **Rendering** tab.
3. Locate and select the appropriate region, item, or region button.
4. Right-click the component, and select **Copy to other Page**.

   The appropriate wizard appears.
5. Follow the on-screen instructions.

7.3.3 Creating Branches, Validations, Computations, and Processes

Learn about managing branches, validations, computations, and processes in Page Designer.

- Creating Pre-Rendering Branches, Computations, and Processes
  Create branches, computations, and processes that execute before rendering the page:
- Creating Page Submission Branches, Validations, Computations, and Processes
  Create page submission branches, validations, computations, and processes in the Processing tab.

> **See Also:**
>
> - Managing Computations, Validations, and Processes
> - Controlling Navigation Using Branches

7.3.3.1 Creating Pre-Rendering Branches, Computations, and Processes

Create branches, computations, and processes that execute before rendering the page:

To create branches, computations, and processes that execute before rendering the page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the left pane, select the Rendering tab.
3. Expand the Pre-Rending node at the top of the Rendering tab.
4. Determine the processing point. Right-click **Before Header**, **After Header**, or **Before Regions** to display the context menu for that processing point.
5. Select the appropriate option: **Create Branch**, **Create Computation**, or **Create Process**.
6. Edit the appropriate attributes in the Property Editor.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

> **Tip:**
>
> If you have changed the default display to group components by component type, then you must to navigate to the Processing tab to maintain pre-rendering components.

> **See Also:**
>
> Managing Computations, Validations, and Processes

7.3.3.2 Creating Page Submission Branches, Validations, Computations, and Processes

Create page submission branches, validations, computations, and processes in the Processing tab.

To create branches, validations, computations, and processes that execute when the page is submitted:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the left pane, select the Processing tab.
3. Navigate to After Submit, Validating, Processing, or After Processing as appropriate within the processing tree.
4. Right-click the identifier or an existing component to display the context menu for that processing point.
5. Select **Create Branch**, **Create Validation, Create Computation**, or **Create Process**.
6. Edit the appropriate attributes in the Property Editor.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Click **Save** or **Save and Run Page**.

> **Tip:**
>
> You can use context menus to duplicate, delete, and copy a component to other pages.

> **See Also:**
>
> Managing Computations, Validations, and Processes

7.3.4 Copying a Page in Page Designer

Copy a page in Page Designer by selecting Copy Page from the Create menu on the Page Designer toolbar.

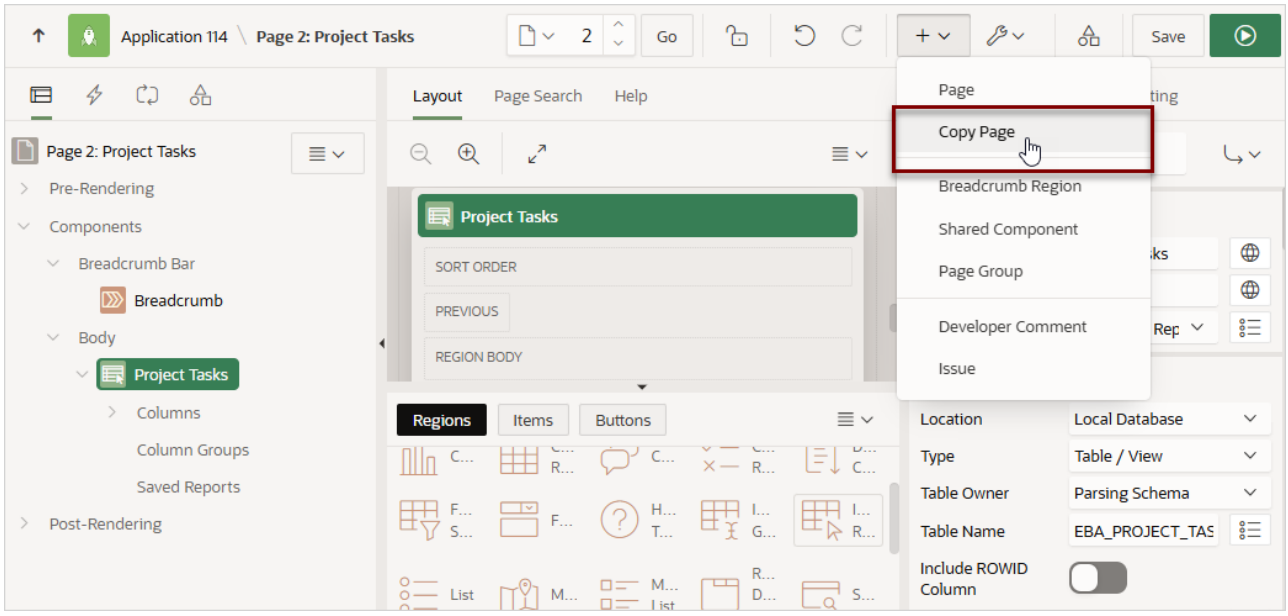
In Page Designer you can copy the currently selected page or another page in the current application. During the copy process, you can also copy shared components or change mappings to shared components in the target application. The example that follows demonstrates how to copy a page containing an interactive grid.

> **Tip:**
>
> You can also copy a page from the Create Page Wizard by clicking **Create Page as Copy**. See Creating a New Page by Copying.

To copy a page in Page Designer:

1. View the page you want to copy in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. On the Page Designer toolbar, click the **Create** menu and then select **Copy Page**.



The Copy Page Wizard appears.

3. Create a page as a copy of - Specify how to copy a page and click **Next**. Options include:
   - **Page in this application**
   - **Page in another application**

   The UI changes slightly depending upon your selection.
4. If you select **Page in this application**:
   a. Page to Copy:
      - Application - Displays the current application ID.
      - Copy From Page - Select the page you want to copy.
      - New Page Number - Specify a new page number.
      - New Page Name - Enter a new page name.
      - Copy Public Reports - All **default** reports are copied automatically. Specify whether to copy **public** reports. Select **On** or **Off**.
      - Breadcrumb - Select whether to use a breadcrumb navigation on the new page and the breadcrumb navigation control you want to use.
      - Click **Next**.

   b. Navigation Menu:
      - Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
      - Click **Next**.

   c. New Names - Review and update names as needed.

   d. Click **Copy**.
5. If you select **Page in another application**:
   a. Page to Copy:

- Copy from Application - Select the application that contains the page you wish to copy.
- Copy From Page - Select the page you want to copy.
- Copy to New Page Number - Enter the page number for your new page.
- New Page Name - Enter a new page name.
- Copy Public Reports - All **default** reports are copied automatically. Specify whether to copy **public** reports. Select **On** or **Off**.
- Breadcrumb - Select whether you to use a breadcrumb navigation on the new page and the breadcrumb navigation control you want to use.
- Click **Next**.

b. Navigation Menu:

- Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
- Click **Next**.

c. Templates - When you copy a page from another application, all the templates used by that page must be mapped to templates in the current application. This report displays the templates in the page you are copying from.

- Review the report and map to the correct template.
- Click **Next**.

d. Click **Finish**.

7.3.5 Commenting Out Page Components in Page Designer

Exclude or include page components at runtime using the Comment Out and Uncomment context menu option in Page Designer.

> **Note:**
>
> The functionality described in this topic is only available by default in new applications created in Oracle APEX, release 23.2 or later.

You can exclude application components at runtime by right-clicking the component in the Rendering tab and selecting **Comment Out** from the context menu. To include the component at runtime, select the page component again but this time select **Uncomment** from the context menu. The **Comment Out** and **Uncomment** context menu option appears in the Rendering tab, the Dynamic Actions tab, and the Processing Tab.

> **Tip:**
>
> The **Comment Out** and **Uncomment** context menu options are created from a build option named **Commented Out**. The **Commented Out** build option displays as a toggle in the tree context menu in the left pane of Page Designer. To learn more, see Using Build Options to Control Configuration.

To exclude or include components in Page Designer using the **Comment Out** and **Uncomment** context menu options:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. To exclude a page component at runtime:
   a. In the Rendering tab, right-click the page component.
   b. From the context menu, select **Comment Out**.

      The Rendering tab displays the component with a line drawn through it, indicating the page component will be excluded at runtime.
3. To include (or Uncomment) a page component:
   a. In the Rendering tab, right-click the component.
   b. From the context menu, select **Uncomment**.

      The line through component indicating it is excluded disappears.

7.4 Managing Page Attributes

Edit page attributes to control basic information and functionality for a given page, including the page name or title, header and footer text, navigation menus, inclusion of page-level JavaScript or CSS files, error handling, and the selected authorization scheme.

- Editing Page Attributes
  Edit page attributes by viewing the page in Page Designer and selecting the page name.
- Incorporating JavaScript Using Page Designer
  Include JavaScript on a page by editing the page attributes in the JavaScript group.
- Configuring Rejoin Sessions for a Page
  Configure the Rejoin Sessions attribute to determine whether URLs to this application page contain session IDs.
- Configuring Dialog Attributes
  Configure Dialog attributes in Page Designer.
- Configuring Pages to Warn Users of Unsaved Changes
  Enable the Warn on Unsaved Changes page attribute to warn users if they navigate away from a page containing unsaved changes.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

7.4.1 Editing Page Attributes

Edit page attributes by viewing the page in Page Designer and selecting the page name.

To edit page attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the page name.
3. Edit the appropriate attributes in the Property Editor.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To save your changes, click **Save** or **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor) and Viewing Field-Level and Page Designer Help

7.4.2 Incorporating JavaScript Using Page Designer

Include JavaScript on a page by editing the page attributes in the JavaScript group.

Including JavaScript on a page is a good approach for functions that are very specific to a page and a convenient way to test a function before you include it in a `.js` file.

To add JavaScript code to a page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the page name.

   The Property Editor displays the Page attributes.
3. In the Property Editor, find the **JavaScript** group.
4. Under JavaScript, edit following attributes:
   a. **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `dignified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

b. **Function and Global Variable Declaration** - Enter JavaScript code such as functions or global variable declarations referenced on this page. If the same JavaScript is required on multiple pages, consider putting the code into an external JavaScript file to avoid duplication and to leverage browser caching of static files.

   Code you enter here replaces the `#PAGE_JAVASCRIPT#` substitution string in the page template.

   > **Tip:**
   >
   > Do not to include opening or closing script tags. Just include the JavaScript code.

   To reference a shortcut, use:

   `"SHORTCUTNAME"`

c. **Execute When Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle APEX.

> **Tip:**
>
> To learn more about an attribute and view examples, select the attribute and click the **Help** tab in the central pane.

For example, adding the following to the Function and Global Variable Declaration attribute would create a function called `test`, which you could then use from anywhere on the current page.

```
function test(){
    alert('This is a test.');
  }
```

5. Click **Save**.

7.4.3 Configuring Rejoin Sessions for a Page

Configure the Rejoin Sessions attribute to determine whether URLs to this application page contain session IDs.

The Rejoin Sessions attribute controls support for joining existing sessions (similar to deep linking).

> **Warning:**
>
> Enabling rejoin sessions may expose your application to possible security breaches, as it can enable attackers to take over existing end user sessions. To learn more, see About Rejoin Sessions.

To use Rejoin Sessions at the page level, administrators must enable **Rejoin Sessions** at the instance level. A more restrictive instance-level setting overrides application and page settings.

To configure Rejoin Sessions for a page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the page name.

   The Property Editor displays the Page attributes.
3. In the Property Editor, find the **Security** group.
4. Under Security, Rejoin Sessions, select an option:
   - **Application Default** - Inherits the setting defined at application-level.
   - **Disabled** - If the URL does not contain a session ID Oracle APEX creates a new session.
   - **Enabled for Public Sessions** - If the URL goes to a public page and does not contain a session ID, APEX attempts to utilize the existing session cookie established for that application. APEX only joins using the cookie when the session is not yet authenticated.

     Enabled for Public Sessions requires that Embed In Frames be set to **Allow from same origin** or **Deny**. This is not tied to a condition about the URL payload, but also applies to session state protected URLs.
   - **Enabled for All Sessions** - If the URL does not contain a session ID, APEX attempts to utilize the existing session cookie established for that application, providing the following conditions are met:
     – Session State Protection is enabled for the application and the URL includes a valid checksum. For public bookmarks, the most restrictive item level protection must be either **Unrestricted** or **Checksum Required - Application Level**.
     – The URL does not contain payload (a request parameter, clear cache or data value pairs).

       This setting also requires that Embed In Frames is set to Allow from same origin or to **Deny** for the application.

   > **Tip:**
   >
   > The option **Application Default** inherits the setting defined at application-level. See Browser Security .
5. Click **Save**.

See Also:

- About Rejoin Sessions
- Session Management to configure Rejoin Sessions at the application-level
- Configuring Rejoin Sessions for an Instance in *Oracle APEX Administration Guide*

7.4.4 Configuring Dialog Attributes

Configure Dialog attributes in Page Designer.

Tip:

Creating a dialog page is a multiple step process. To learn more, see Creating Dialog Pages.

To configure Dialog attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the page name.

   The Property Editor displays the Page attributes.
3. In the Property Editor, find **Appearance** and verify that the Page Mode is set to either **Modal Dialog** or **Non-Modal Dialog**.
4. Find **Dialog** and configure these attributes:
   a. Width
   b. Height
   c. Maximum Width
   d. Attributes
   e. CSS Classes
   f. Chained

   Tip:

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. To save your changes, click **Save** or **Save and Run Page**.

> **See Also:**
>
> Creating Dialog Pages and Right Pane of Page Designer (Property Editor)

7.4.5 Configuring Pages to Warn Users of Unsaved Changes

Enable the Warn on Unsaved Changes page attribute to warn users if they navigate away from a page containing unsaved changes.

To enable the Warn on Unsaved Changes page attribute:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the page name.

   The Property Editor displays the Page attributes.
3. In the Property Editor, find the **Navigation** group.
4. For **Warn on Unsaved Changes**, select **On** to warn users if they navigate away from a page that contains unsaved changes.
5. Click **Save**.

> **Tip:**
>
> The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.

> **See Also:**
>
> - Configuring Button Attributes to Warn Users of Unsaved Changes
> - Configuring Item Attributes to Warn Users of Unsaved Changes
> - warnOnUnsavedChanges in *Oracle APEX JavaScript API Reference*

7.5 Running a Page from Page Designer

The Oracle APEX engine dynamically renders and processes pages based on data stored in database tables. To view a rendered version of an application, you run or submit it to the APEX engine.

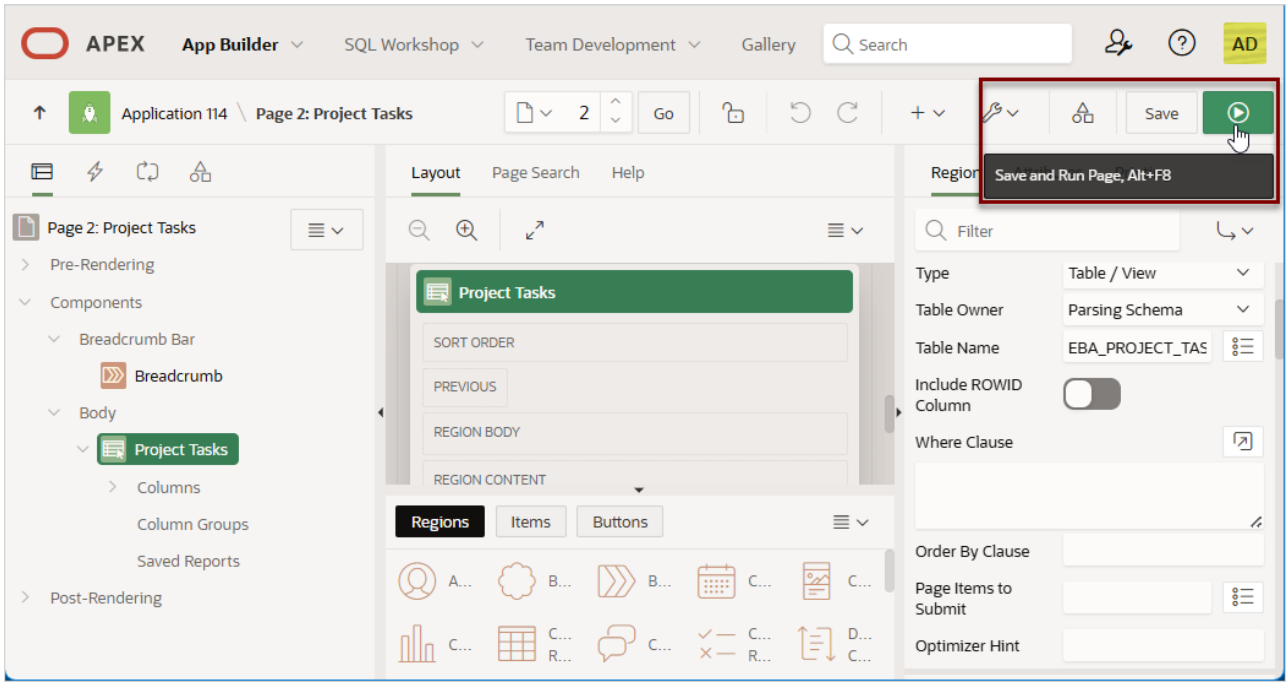
> **Tip:**
>
> You cannot run Modal and Non-Modal dialog pages directly using the Save and Run Page button. To view a Modal and Non-Modal dialog page, run the normal page that is responsible for opening the dialog page and access it from there.

To run a page in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

   Page Designer appears. You preview a page by running it. Running an application page displays the rendered version as end users would see it.
2. Click the **Save and Run Page** button in the upper right corner. The Save and Run Page button resembles a green right-facing arrow.



A rendered version of the application page appears.

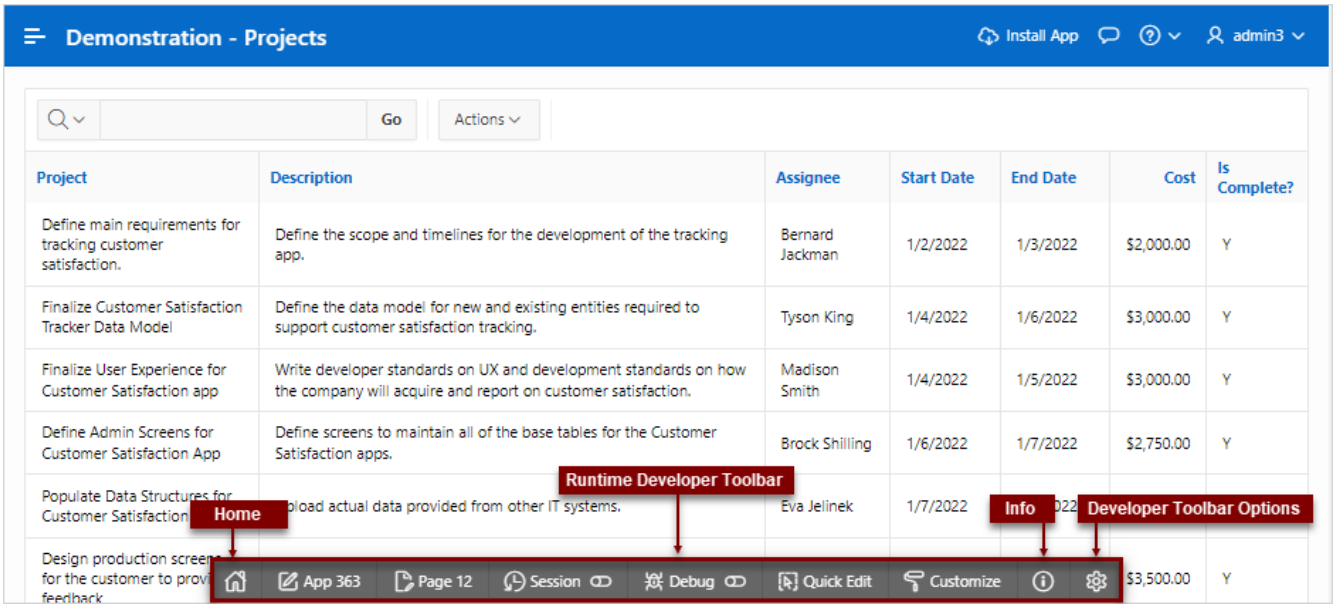
> **See Also:**
>
> - Running an Application or Page
> - Runtime Developer Toolbar

7.6 Runtime Developer Toolbar

Use the Runtime Developer toolbar to quickly edit the current application or currently running page, or view session state.

When you run an application from App Builder and view a page, the Runtime Developer toolbar displays at the bottom of any editable page.

> **Tip:**
>
> The Runtime Developer toolbar only displays if the Availability, Status attribute on the Application Definition page is set to **Available with Developer Toolbar**. See Editing the Application Definition.



The Runtime Developer toolbar contains of the following controls:

- **Oracle APEX Home**
  (

) - Links to the Workspace home page. See Understanding the Workspace Home Page.
- **App XX** (where *XX* is the app ID) - Links to the Application home page. See Understanding the Application Home Page.

- **Page XX** (where *XX* is the page number) - Displays the currently running page ID. Click the page ID to view the page in Page Designer and then edit the page.
- **Session** - Displays two menus:
  - **View Session State** - Displays session state information for the current page. See Viewing Session State.
  - **Session Overrides** - Displays the Session Overrides dialog which lists session values that can be overidden in the current application. See Managing Session State Overrides.
- **Debug** - Controls the display of debug information. Available options include:
  - **Enable Debug** - Specifies the level of detail included when you click **View Debug** .
    - **No Debug** - Enables and disables Debug mode
    - **Info (default)** - The debug log contains the default or normal amount of information. This corresponds with a log level of 4 and a Debug URL argument of `YES`.
    - **App Trace** - The debug log contains messages up to level 6. This corresponds with a Debug URL argument of `LEVEL6`.
    - **Full Trace** - The log will contain the maximum amount of information. This corresponds with a Debug URL argument of `LEVEL9`. **Full Trace** can slow request processing but provides the most amount of detail.

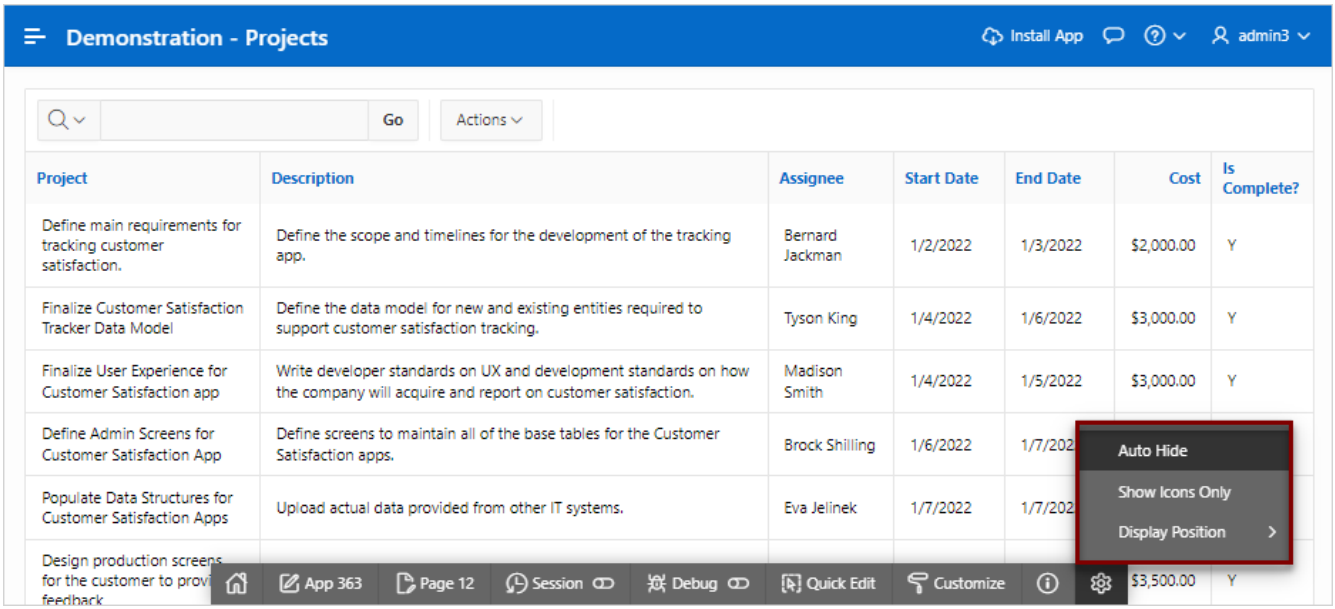
    See Utilizing Debug Mode.
  - **View Debug** - Displays debug information for the current page.
- **Quick Edit** - Enables developers to enter either Quick Edit mode or access Live Template Options.
  - **Quick Edit Mode** - Click **Quick Edit** and then select the desired component to instantly access the component in Page Designer. Press `ESCAPE` or click outside a component to exit quick edit mode.
  - **Edit Live Template Options** - Click **Quick Edit** and then move the mouse over the component for which you want to modify template options and click the **Wrench** icon in the upper right corner. See Using Quick Edit to Modify Live Template Options.
- **Customize** - Customize menu options include:
  - **Theme Roller** - Access a live CSS editor that enables you to quickly change the theme style, colors, rounded corners and other application attributes. This option only displays for themes supporting Theme Roller. See Using Theme Roller.
  - **Edit Logo** - Change the logo users see at runtime. See Managing Application Logos.
  - **Edit App Icon** - Update or upload a new application icon. See Managing Application Icons.
- **Info**
  (

) displays two menus:
  - **Show Layout Columns** - Toggles between **Show Layout Columns** and **Hide Layout Columns**. This option only applies if a layout is used and that layout supports showing layout columns and the page has more than one column.

  - **Show Page Timing** - Displays the Page Performance Timing dialog. Click **Copy** to copy the data in table form and then paste it into another application. Click **Clear** to remove the current timing events.
  - **Show Landmarks** - Landmarks provide a powerful way to identify the structure of an APEX application and should be used to complement, but not replace Heading structure. See Landmarks in *Oracle APEX Accessibility Guide*.
  - **Show Headings** - View headings on the current page. See Headings in *Oracle APEX Accessibility Guide*.
- **Developer Toolbar Options**
  (

) - Displays on the far right and resembles a mechanical gear. Click **Developer Toolbar Options** to control how the Runtime Developer Toolbar displays.

About Developer Toolbar Options



Clicking **Developer Toolbar Options** ( ) displays a submenu containing the following controls:

- **Auto Hide** - Hides the toolbar, making more of the page visible.
- **Show Icons Only** - Removes the labels from the buttons to make the toolbar smaller.
- **Display Position** - Controls the placement of the Developer toolbar. Options include: **Top**, **Left**, **Bottom**, or **Right**.

About Runtime Toolbar and the Environment Banner

If a workspace or instance-level environment banner is enabled, the text and color will be applied to Runtime Developer toolbar.

An environment banner displays a colored banner with text at either the left or top of page. You can define an environment banner for a given workspace or an entire development instance. Workspace-level environment banners override those defined at the instance-level.

> **See Also:**
>
> Configuring a Workspace Environment Banner and Configuring an Environment Banner in Administration Services in *Oracle APEX Administration Guide*

About JavaScript Error Detection

If a page has one or more JavaScript errors, a red error button displays on the left side of the Runtime Developer toolbar. If you have enabled **Auto Hide**, the Runtime Developer toolbar displays indicating the error.

Click the red error button to view an alert dialog which explains you should view the Browser console. The red error button persists until error has been resolved, the Browser console is cleared, and you refresh the page.

7.7 Deleting Pages

Deleting a page deletes the corresponding tabs, breadcrumbs, and list entries.

- Deleting a Page
  Delete a page in Page Designer.
- Deleting Multiple Pages
  Delete multiple pages in Cross Page Utilities.
- Deleting a Range of Pages
  Delete a range of pages in Cross Page Utilities.

> **See Also:**
>
> Locking and Unlocking a Page

7.7.1 Deleting a Page

Delete a page in Page Designer.

To delete a page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Verify the page number.
3. On the Page Designer toolbar, select the **Utilities** menu and click **Delete Page**.

   The Confirm Page Delete page appears.
4. Click **Permanently Delete Page**.

> **See Also:**
>
> Page Designer Toolbar

7.7.2 Deleting Multiple Pages

Delete multiple pages in Cross Page Utilities.

To delete multiple pages:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.
5. Click **Delete Multiple Pages**.
6. Select the pages to be deleted and click **Delete Pages**.

7.7.3 Deleting a Range of Pages

Delete a range of pages in Cross Page Utilities.

To delete a range of pages:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities, click **Cross Page Utilities**.
5. Click **Delete Multiple Pages**.
6. Click the **Delete Pages by Range** tab.
7. Enter the pages to be deleted and click **Delete Pages**.

8
Developing Reports

In Oracle APEX, a report is the formatted result of a SQL query. You can generate the SQL query by selecting a table or view in a wizard or by defining the SQL query manually.

> **Tip:**
>
> To view examples, install the sample app, *Sample Reporting*. To learn more, see Sample Apps.

- Understanding Report Page Types
  Learn about the available report page types when running the Create Application Wizard and Create Page Wizard.
- Creating a Report Using the Create Application Wizard
  Run the Create Application Wizard to create a new application containing one or more pages. Supported report pages include cards, faceted search, interactive grid, interactive report, and classic report.
- Creating a Report Using the Create Page Wizard
  Run the Create Page Wizard from an existing application to create a variety of different reports.
- Creating a Report by Dragging and Dropping from the Gallery
  Create a report by dragging and dropping a report type from the Gallery.
- Managing Cards
  A cards page features an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row).
- Managing Smart Filters
  A smart filters page features a single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a smart filter behaves similarly to faceted search, it features a more space efficient layout.
- Managing Faceted Search
  A faceted search page displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). The report region displays search results as cards or a classic report.
- Managing Interactive Grids
  An interactive grid presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard.
- Managing Interactive Reports
  An interactive report page features a searchable, easily customizable report.
- Managing Classic Reports
  A classic report features a simple report page based on the formatted result of a SQL query.

- Managing Sorting with Order By Item
  Create an inline static LOV by configuring region-level Order By Item attributes.
- Printing Report Regions
  Configure a report region to print by exporting it to several different formats.
- Understanding BLOB Support in Forms and Reports
  Oracle APEX includes declarative `BLOB` support to enable developers to declaratively upload files in forms, and download or display files in reports.

> **See Also:**
>
> Developing Forms and Managing Application Components

8.1 Understanding Report Page Types

Learn about the available report page types when running the Create Application Wizard and Create Page Wizard.

The Create Application Wizard and Create Page Wizard include a number of pages for displaying data. Some of these page types, such as classic reports, interactive reports, and interactive grids, are straightforward reports. However, others include a report region and an associated search region. For example, a faceted search page contains a search region and a report region which can display as either a cards region or a classic report. Similarly, a smart filters page includes a search region and a search results report which can be a classic report, cards, map, or calendar.

> **Note:**
>
> To learn more about reports and forms optimized for mobile devices, see Creating Applications for Mobile Devices and Legacy Forms Optimized for Mobile Environments.

- Report Page Type Use Cases
  Review common use cases for available report page types.
- Comparison of Report Page Types
  Compare report pages types available when running the Create Application Wizard and Create Page Wizard.

8.1.1 Report Page Type Use Cases

Review common use cases for available report page types.

The following table describes uses cases for the report page types available in the Create Page Wizard and Create Application Wizard.

> **Note:**
>
> For a more detailed comparison of report page types, see Comparison of Report Page Types.

Report Page Type	Use Case	Learn More...
Cards	• Display data at a glance using information tiles that can easily include an image. • Choose between three cards page layouts: Grid, Float, or Horizontal (Row). • Provides summary information from a modern layout with one-click access to more details on another report or form page.	See: • Managing Cards • Example Cards Page • Install the *Sample Cards* sample app. See Installing Apps from the Gallery.
Classic Report	• Display data in a simple report region. • Update data by including a report and update form combination during creation. • Column sorting configurable.	See: • Managing Classic Reports • Example Classic Report • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.
Faceted Search	• Display and filter data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). • Show search results as cards or a classic report. • Includes built-in search capability. • Users select facets in the Search region to narrow down the search results.	See: • Managing Faceted Search • Example Faceted Search • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Use Case	Learn More...
Interactive Grid	• Display data in a searchable, customizable report that supports inline editing at runtime using the mouse or keyboard. • Enable end users to create highly customized reports. Users can alter the report layout by using either the Column Heading menu or Actions menu. • Move columns by using the Actions menu, Column dialog, or by dragging and dropping with the mouse. • Save customizations and report layout by saving a private or alternative report just for yourself, a public report, or as a primary report that can be viewed by others. • Save a report layout by creating an Alternative or Private report. • Download or email the report as a CSV, HTML, PDF, or Excel file. • Enable and disable editing capability at by editing attributes in Page Designer. • If editable: – Select options from the Row Action menu. – Add and remove rows or edit data by directly editing a cell.	See: • Understanding Report Page Types • Understanding Report Page Types • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.
Interactive Report	• Display data in a searchable, customizable report. • Enable end users to create highly customized reports. Users can alter the report layout by using either the Column Heading menu or Actions menu. • Update data by including an update form during creation. • Save customizations and report layout by saving a private report just for yourself, a public report, or as a default report that can be viewed by others. • Download, email, or subscribe to report update subscriptions in CSV, HTML, Excel, or PDF format.	See: • Managing Interactive Reports • Example Interactive Report • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Use Case	Learn More...
Smart Filters	• Display data using a single search field region at the top of the page and results report. • Offers a more compact UI when compared to faceted search. • Search results report can display as a classic report, cards, map, or calendar. • End users select filters that display as suggestion chips. • Includes built-in search capability. • Configurable search suggestions	See: • Managing Smart Filters • Example Smart Filters • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

8.1.2 Comparison of Report Page Types

Compare report pages types available when running the Create Application Wizard and Create Page Wizard.

The following table describes the commonly used report page types available in the Create Page Wizard and Create Application Wizard.

Report Page Type	Description	Learn More...
Cards	A cards page features an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row). Features: • Customize every aspect of a cards region's UI (including layout, appearance, icon, badge, and media). • Declaratively create links from a cards page by adding actions. Action types include Button, Full Card, Title, Subtitle, or Media. • Supports advanced HTML expressions including template directives for client-side conditional formatting.	See: • Managing Cards • Example Cards Page • Install the *Sample Cards* sample app. See Installing Apps from the Gallery.

Report Page Type	Description	Learn More...
Classic Report	A classic report features a simple report page based on the formatted result of a SQL query. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Enable column sorting in Page Designer by selecting a column and configuring Column, Sorting attributes. • The most versatile report as it is template-driven. Choose from the predefined report templates, including Media List, Comments, Timeline, Badge List, Value Attributes Pairs, and so on. Or, create your own template to customize and visualize data. • Download the report as a CSV, HTML, Excel, or PDF. • Supports Report Layouts to format PDF exports. Limitations compared to other report types: • Using templates requires changing the SQL query column names to match the chosen template (for row templates) • No support for scroll pagination or virtual scrolling. Other pagination options available. • No column header groups, frozen columns, and template directives.	See: • Managing Classic Reports • Example Classic Report • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Description	Learn More...
Faceted Search	A faceted search page displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). The report region displays search results as cards or a classic report. Features: • Select options from facets in the left Search region to narrow down search results. • Set filters using facets that show possible values together with the occurrence count within the result set. • Built-in search capability. • Optionally include a chart for any facet. • Toggle between a bar chart (which is the default) and pie chart type. • Create facet groups for data models having multiple flag columns.	See: • Managing Faceted Search • Example Faceted Search • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Description	Learn More...
Interactive Grid	An interactive grid presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. Features: • Extensive customization capabilities using menus or inline editing. • Turn editing on or off using the region Attribute, Edit, Enabled. • Built-in search capability. • For both editable and non-editable reports, users can change the report layout and create private reports: – Column Heading menu - Change the report layout with Control Break, Aggregations, Freeze, Hide, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Refresh, Flashback. * Format - Control Break, Highlight, Stretch Column Widths. * Selection - Cell Selection, Copy to Clipboard, Refresh Rows. * Chart - Create chart. * Report - Create, edit, and manage saved Private or Alternative reports. * Download - Download or email the report as a CSV, HTML, PDF, or Excel file • If editable, users can edit the data directly: – Row Actions menu - Single Row View, Add Row, Duplicate Row, Delete Row, Refresh Row, Revert Changes. – Add Rows - Add new rows by clicking the **Add Row** button. – Edit Rows - Edit a row by selecting or double-clicking a cell with the mouse, editing the content, and clicking the **Save** button.	See: • Understanding Report Page Types • Understanding Report Page Types • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Description	Learn More...
	Limitations compared to other report page types: • Rows are fixed height. No variable height option. • Does not remember current page. • Tabular form limitations: submit selected rows, always edit for small number of records • Interactive report features not supported: Pivot view, Group By view, Computed columns, and complex filters.	

Report Page Type	Description	Learn More...
Interactive Report	An interactive report page features a searchable, easily customizable report. Features: • Report can display as a standard application page or as a modal dialog. • Enable the **Include Form** option to create a report and update form combination. • Built-in search capability. • Change the report layout and create private reports using menus: – Column Heading menu: Change the report layout with Sort Ascending, Sort Descending, Hide Column, Control Break, and Filter. – Actions menu: * Columns - Configure the columns to show or hide. * Filters - Create filters to alter the display. * Data - Sort, Aggregate, Compute, Flashback. * Format - Control Break, Highlight, Rows Per Page. * Chart - Create a chart. * Group By - Group sets of results by one or more columns with Group By. * Pivot - Pivot reports transpose rows into columns to generate results in a cross tab format. * Report - Create, edit, and manage saved Named reports. * Download - Download or email the report as a CSV, HTML, Excel, or PDF. * Subscription - Subscribe to report updates in CSV, HTML, Excel, or PDF format. Limitations compared to other report page types: • No scroll pagination. • Freeze columns not supported. • Column groups not supported. • Charts not as powerful as interactive grid.	See: • Managing Interactive Reports • Example Interactive Report • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

Report Page Type	Description	Learn More...
Smart Filters	A smart filters page features a single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a smart filter behaves similarly to faceted search, it features a more space efficient layout. Features: • Includes a search field at the top of the page with filters that display as suggestion chip. • The Search Results can display as a classic report, cards, map, or calendar. • Clicking a filter name displays a list of values. • Clicking to right of a suggestion chip changes it to an applied filter chip and moves to the Applied Filters Area. Clicking the applied filter chip again displays a list. • Built-in search capability.	See: • Managing Smart Filters • Example Smart Filters • Install the *Sample Reporting* sample app. See Installing Apps from the Gallery.

8.2 Creating a Report Using the Create Application Wizard

Run the Create Application Wizard to create a new application containing one or more pages. Supported report pages include cards, faceted search, interactive grid, interactive report, and classic report.

> **Tip:**
>
> To learn more, see the Create Application Wizard discussion for each component.

- Report Options When Running the Create Application Wizard
  Learn about supported report options available when running the Create Application Wizard.

8.2.1 Report Options When Running the Create Application Wizard

Learn about supported report options available when running the Create Application Wizard.

Available Reports When Creating Applications

The Create Application Wizard supports the following report types.

Report Type	Description
Cards	A cards page consists of individual boxes, which resemble index cards, laid out on a page. Each card displays three pieces of information: Card Title, Description Column, and Additional Text Column. First, you select a table or view on which to build the page. Second, you select the Card Title, Description Column, and Additional Text Column. **See Also**: Managing Cards
Faceted Search	Creates a faceted search region and report. First, you select a report type (**Report** or **Cards**. Second, you select the Table or View on which to build the faceted search region and report. If you select a **Table**, facets are auto discovered using APEX Data Dictionary Cache. If you select a **View**, facets are not auto-discovered. Developers can create the faceted report page based on a View, but the only Search facets included are searchable `VARCHAR2` columns. **See Also**: Managing Faceted Search
Smart Filters	A smart filters page features a single search field at the top of the page with filters and a report. Each filter displays as a suggestion chip with a single count of how often the specific suggestion value occurs. Users can search, or click a smart chip to quickly filter the report. While a smart filters page behaves similarly to faceted search, it features a more space efficient layout. Smart filter pages feature a single search field at the top of the page with filters which display as smart chips with counts that show how often a specific value occurs. Users can search, or click a smart chip to quickly filter the report. **See Also**: Managing Smart Filters
Interactive Grid	An interactive grid presents users a set of data in a searchable, customizable report. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse. First, you select a page source (that is, **Table or View** or **SQL Query**). Second, you determine if the interactive grid is editable by selecting **Allow Editing** or **Read Only**. If you select **Allow Editing**, users can add to, modify, and refresh the data set directly on the page. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. You choose a table on which to build the interactive grid. **See Also**: Managing Interactive Grids
Interactive Report	Creates a page that contains the formatted result of a SQL query. First, you select a page source (that is, **Table or View** or **SQL Query**). Second, you select a report type,**Interactive Report**. To include a form page for creating or updating records, select **Include Form**. Select **Include Form** to include a form page for creating or updating records. If the report is based on table which has a foreign key constraints to another table, the developer can also define Lookup Columns. Use **Lookup Columns** to replace identifiers with a display column, such as showing the department name instead of the department number. **See Also**: Managing Interactive Reports

Report Type	Description
Additional Pages, Classic Report	Creates a page that contains the formatted result of a SQL query. First, you select a page source (that is, **Table or View** or **SQL Query**). Second, you select the report type **Classic Report**). To include a form page for creating or updating records, select **Include Form**. If the report is based on table which has a foreign key constraints to another table, the developer can also define Lookup Columns. Use **Lookup Columns** to replace identifiers with a display column, such as showing the department name instead of the department number. **See Also**: Managing Classic Reports
Additional Pages, Multiple Reports	Creates multiple report pages containing an interactive report with form. For each page, you choose a table on which to build the interactive report with form. Click **Edit** to change the default options (such as providing a custom **SQL Query**), specifying another report type (such as, selecting**Classic Report** instead of **Interactive Report**). Select **Include Form** to include a form page for creating or updating records. **See Also**: Managing Classic Reports

See Also:

Creating an Interactive Report Using the Create Application Wizard and Available Page Types in the Create Application Wizard

8.3 Creating a Report Using the Create Page Wizard

Run the Create Page Wizard from an existing application to create a variety of different reports.

- Report Options When Running the Create Page Wizard
  Learn about available report options when running the Create Page Wizard and using the Universal Theme.
- Creating a Report on New Page
  Use the Create Page Wizard to create a wide variety of reports on a local database or on an external data source.

8.3.1 Report Options When Running the Create Page Wizard

Learn about available report options when running the Create Page Wizard and using the Universal Theme.

Tip:

Universal Theme - 42 is optimized to work equally well in either a mobile or desktop environment. In previous releases and in earlier themes, the Mobile user interface is based on jQuery Mobile. Oracle recommends migrating existing mobile applications to the Universal Theme as soon as possible.

Available Reports in the Create Page Wizard

The Create Page Wizard supports the following reports when running the Create Page Wizard with the Universal Theme.

Report Type	Description	To Learn More
Cards	A cards page consists of individual boxes, which resemble index cards, laid out on a page. Each card displays three pieces of information: Card Title, Description Column, and Additional Text Column. First, you select a table or view on which to build the page. Second, you select the Card Title, Description Column, and Additional Text Column.	See: Managing Cards
Faceted Search	Creates a faceted search page which contains a left Search region and a report. You select a table or view on which to build the page. The wizard shows facet selections with options, enabling you to select or deselect the facet columns. When a page is based on local table, then facet recommendations display based on the statistics of the data in the selected table. In all other cases (that is, when the page is not based on local table), only `VARCHAR2` columns are displayed for facet selections. You choose if the report is a classic report or a cards report.	See: Managing Faceted Search
Smart Filters	A smart filters page features a single search field at the top of the page with filters that display suggestion chips. While a smart filters page behaves similarly to faceted search, a smart filters page features a more space efficient layout. You select a table or view on which to build the page. When a page is based on local table, then filter recommendations are displayed based on the statistics of the data in the selected table. In all other cases (that is, when the page is not based on local table), only `VARCHAR2` columns are displayed for filter selection. Each suggestion chip displays a single count for the specific suggestion values. The suggestion chip count only displays for filters that support counts, such as LOV-based filter types. Users can search or click a suggestion chip to quickly filter the report. You choose if the report is a classic report or a cards report.	See: Managing Smart Filters

Report Type	Description	To Learn More
Interactive Grid	An interactive grid presents users a set of data in a searchable, customizable report. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. Enabling **Include Form Page** includes a modal form page. Then, you select the Data Source and the Source Type (that is, **Table** or **SQL Query**). If the Data Source contains `BLOB` columns, then those columns are automatically excluded from the region source since interactive grid regions do not support BLOB columns. If the Data Source is a local table which has foreign keys selected, then for those columns, the column type is set as **Select List**. If a if matching LOV already exists, a Shared LOV is automatically created or re-used. You determine if the interactive grid is editable by enabling or disabling **Editing Enabled**. When this option is enabled, then the grid is rendered read-only and separate form page is created to create and edit the data.	See: Managing Interactive Grids
Interactive Report	Creates a page that contains the formatted result of a SQL query. Enabling **Include Form Page** includes a modal form page. Then, you select the Data Source and the Source Type (that is, **Table** or **SQL Query**). If a Data Source contains `BLOB` columns and if a local table with primary key constraints is selected as the Data Source, then `BLOB` columns are set as **Download BLOB** type, with proper BLOB attributes. If the Data Source is a local table, which has foreign keys selected, then for those columns, the column type is set as **Plain Text (based on List of Values)**. If a if matching LOV already exists, a Shared LOV is automatically created or re-used.	See: Managing Interactive Reports
Classic Report	Creates a page that contains the formatted result of a SQL query. You choose a table on which to build a report or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. When a Data Source contains `BLOB` columns and if a local table with primary key constraints is selected as the Data Source, then `BLOB` columns are set as **Download BLOB** type, with proper BLOB attributes. When a local table, which has foreign keys is selected as the Data Source, then for those columns, column type is set as **Plain Text (based on List of Values)**. If a if matching LOV already exists, a Shared LOV is automatically created or re-used. Enable **Include Form Page** to include a form page for creating or updating records. With the exception of sorting and simple filtering, end users cannot customize a classic report.	See: Managing Classic Reports

8.3.2 Creating a Report on New Page

Use the Create Page Wizard to create a wide variety of reports on a local database or on an external data source.

The component(s) described in this topic include built-in support for read and write access to remote data sources. In order to build a component on a remote data source, you must first create the application and configure the external data source.

> **Tip:**
>
> To learn more about external data sources, see Managing Application Data.

To add a report on a new page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select a report type. Options include:
   - **Cards**
   - **Faceted Search**
   - **Smart Filter**
   - **Interactive Grid**
   - **Interactive Report**
   - **Classic Report**

   > **Tip:**
   >
   > To learn more about each report type, see Report Options When Running the Create Page Wizard.
5. What displays next changes based on your selection. Follow the on-screen instruction. To learn more about an attribute, see field-level Help.

   > **Tip:**
   >
   > When prompted to select a Data Source, select **Local Database**, **REST Enabled SQL**, **REST Data Source**, **JSON Source**, or **Duality View**.
6. To view the report, click **Save and Run Page**.

   The rendered page appears.

> **Tip:**
>
> You can also run the Create Page Wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

8.4 Creating a Report by Dragging and Dropping from the Gallery

Create a report by dragging and dropping a report type from the Gallery.

> **Tip:**
>
> When you drag and drop a report region from the Gallery, you must provide the underlying SQL query.

To add a report by dragging and dropping from the Gallery:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

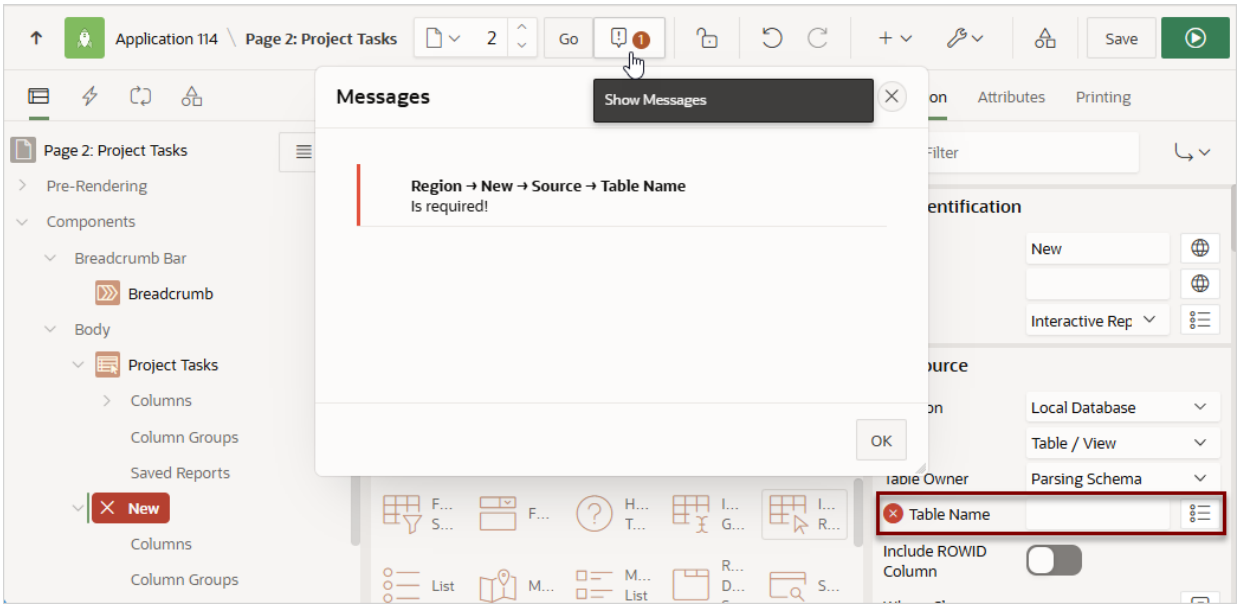
   Page Designer appears.
2. In the Gallery at the bottom of the central pane, locate the report type you want to create.

   The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.

   > **Tip:**
   >
   > In addition to dragging and dropping components from the Gallery, you can also right-click a component in the Gallery to view a context menu. Select **Add To** and then the location where you want to add the component.
3. Click and hold the mouse on the component to be created and drag it the desired location in the Layout tab.

   When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.
4. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.



b. Select an error to highlight the associated attribute in the Property Editor.

5. Edit the appropriate attributes in the Property Editor.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.5 Managing Cards

A cards page features an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row).

A cards report region declaratively supports customizations of layout and appearance and the inclusion of icons, badges, media, and actions. Use cards to embed and share media sourced from a BLOB column, URL or video in iFrame.

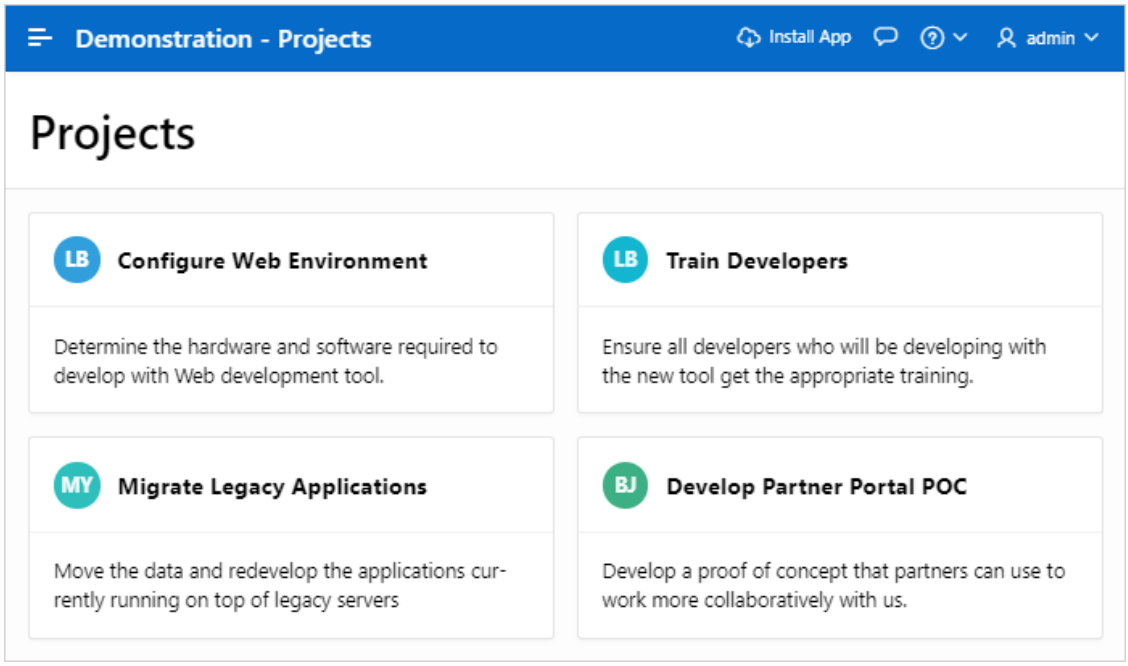
- Example Cards Page
  View an example cards page.
- Creating a Cards Page Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains a cards page.
- Creating a Cards Page Using the Create Page Wizard
  Run the Create Page Wizard to add a new cards page to an existing application.
- Editing Cards Region Attributes
  Edit region attributes to alter the SQL source, change the region layout and appearance, or create region conditions.
- Editing Card Attributes
  Edit card specific attributes by clicking the Attributes tab in the Property Editor.
- Using Actions to Link from a Cards Page
  Create actions to link from a cards page.

8.5.1 Example Cards Page

View an example cards page.

Both the Create Application Wizard and Create Page Wizard support the creation of cards pages. First, you select a table or view or provide a SQL query on which to build the page. Then, you specify the general page layout, for example, Grid, Float, or Horizontal (Row). Finally, you specify the Title Column, Body Column, and optionally Icon Initials Column.

The following is an example of a cards page built on the `EBA_PROJECTS` table.



In this card example, each card displays the following attributes:

- **Title** - Maps to the `NAME` column.
- **Icon** - Maps to the `PROJECT_LEAD` column.
- **Body** - Maps to the `DESCRIPTION` column.

Key features of cards pages include:

- Customize every aspect of a cards region's UI (including layout, appearance, icon, badge, and media).
- Declaratively create links from a cards page by adding actions. Action types include Button, Full Card, Title, Subtitle, or Media.
- Supports advanced HTML expressions including template directives for client-side conditional formatting.

> **Tip:**
>
> To view an example, install the sample app, *Sample Cards*. To learn more, see Installing Apps from the Gallery.

8.5.2 Creating a Cards Page Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains a cards page.

A cards page consists of individual boxes, which resemble index cards, laid out on a page. Each card displays three pieces of information.

To create an application that contains a cards page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Cards**.
8. On Add Cards Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Table or View - Select a table or view on which to build the page.
   d. Select a layout. Options include: **Grid**, **Float**, or **Horizontal (Row)**
   e. Select the information to display on each card. Options include:
      - Title Column
      - Body Column
      - Icon Initials Column
      - Badge Column

      The Title Column and Body Column are pre-selected from the Data Dictionary Cache. Developers can change these defaults.
   f. Advanced - Expand **Advanced** to configure these options:
      - Set as Home Page - Enable this to make this page the home page for the application.
      - Set as Administration Page - Enable this to have this page display under Administration.
      - Page Help - Enter text to be displayed when the user selects Page Help.
   g. Click **Add Page**.
9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.

11. Click **Create Application**.

> **See Also:**
>
> Accessing the Data Dictionary Cache Report from Workspace Administration or Accessing the Data Dictionary Cache Report from Administration Services in *Oracle APEX Administration Guide*

8.5.3 Creating a Cards Page Using the Create Page Wizard

Run the Create Page Wizard to add a new cards page to an existing application.

To create a cards page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Cards**.
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table /View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.
8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Cards Layout - Select a layout. Options include: **Grid**, **Float**, or **Horizontal (Row)**
11. Card Attributes - Select the information to display on each card. Options include:
    a. **Title Column**
    b. **Body Column**
    c. **Icon Initials Column**
    d. **Badge Column**
12. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> - Accessing the Data Dictionary Cache Report from Workspace Administration or Accessing the Data Dictionary Cache Report from Administration Services in *Oracle APEX Administration Guide*
> - Understanding Page Types in the Create Page Wizard

8.5.4 Editing Cards Region Attributes

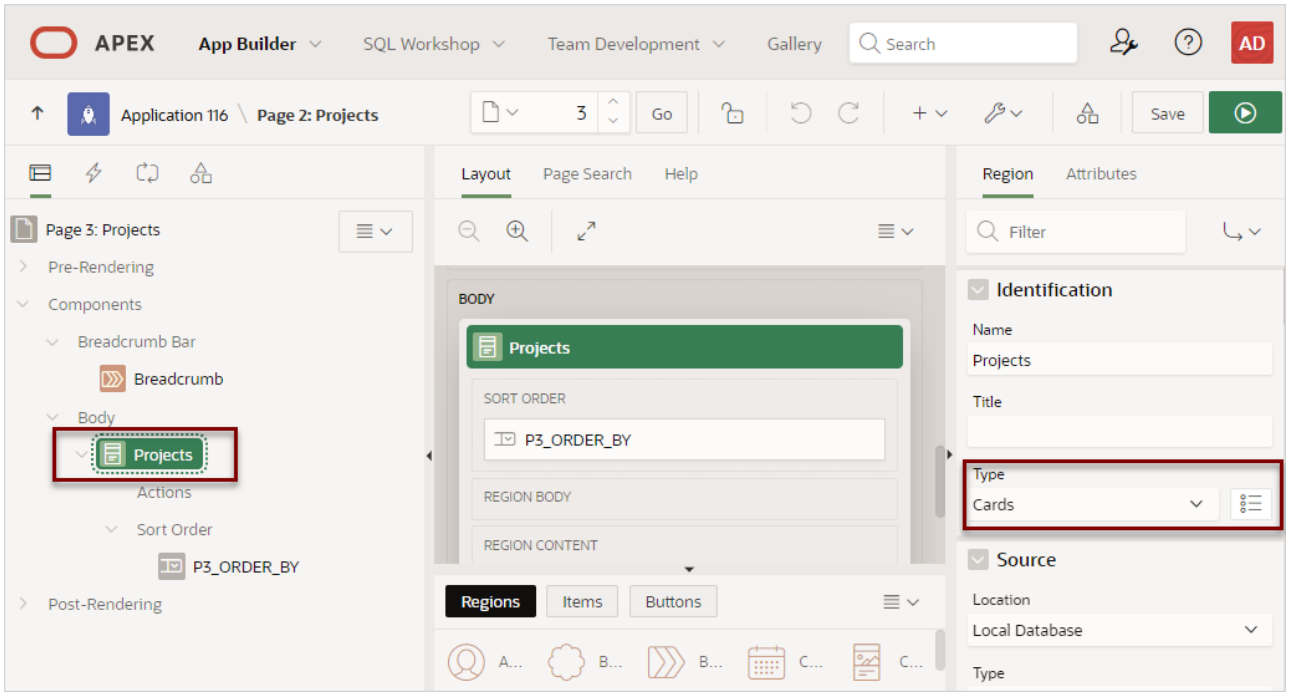
Edit region attributes to alter the SQL source, change the region layout and appearance, or create region conditions.

To edit a region attributes:

1. View the cards page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab under Content Body, locate and select the cards region (for example, **Projects**).

   In the Property Editor, the **Region** tab displays the attributes. Attributes are organized in groups.



   In the previous example, note that the **Type** attribute indicates that **Projects** is a **Cards** region.
3. Edit the appropriate attributes.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

5. Click **Save** or **Save and Run Page**.

8.5.5 Editing Card Attributes

Edit card specific attributes by clicking the Attributes tab in the Property Editor.

Customize a cards page by editing the following attribute groups: **Appearance**, **Title**, **Subtitle**, **Body**, **Secondary Body**, **Icon and Badage**, or **Media**.

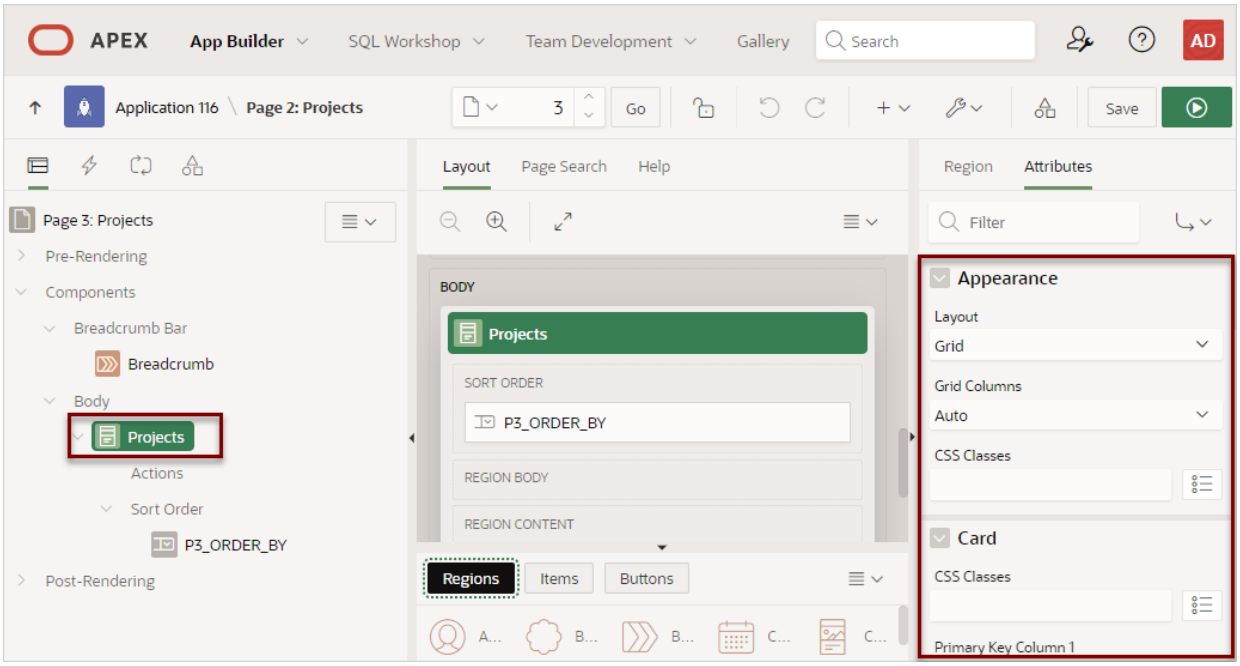
To edit a cards Attributes:

1. View the cards page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the Cards region.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. Select the cards region:
   a. In the Rendering tab under Content Body, locate and select the cards region (for example, **Projects**).
   b. In the Property Editor, select the **Attributes** tab.

      In the Property Editor, the **Atributes** tab displays the card page attributes. Attributes are organized in groups.



5. Edit the appropriate attributes.

Title, **Subtitle**, **Body**, **Secondary Body**, and **Media** include an **Advanced Formatting** option which enables support for HTML expressions and supported substitution strings. All free form text attributes, such as CSS Classes, Icon or Image Description, or Badge Label, support column substitutions through standard substitution string syntax (for example, `&ENAME.` or `&DNAME.`.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
7. Click **Save** or **Save and Run Page**.

> **See Also:**
>
> Using Substitution Strings

8.5.6 Using Actions to Link from a Cards Page

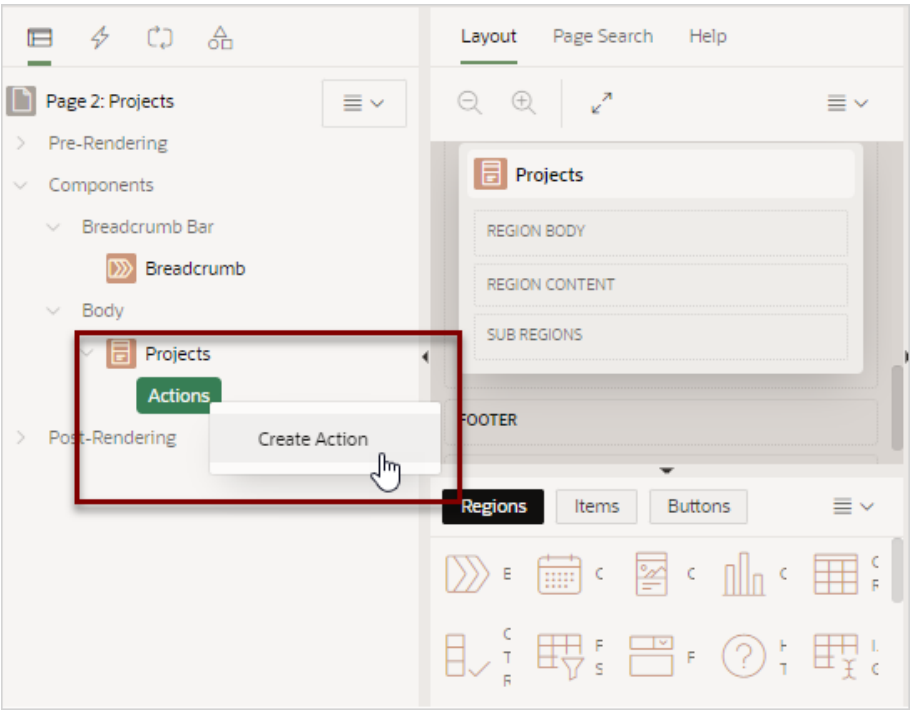
Create actions to link from a cards page.

A cards page functions as a colorful way to display a subset of information and then enable the user to link to more detail. To create a link from a cards page, you create an action and then select an action type. You can create an action to link from a full card, a title, a subtitle, media, or a button.

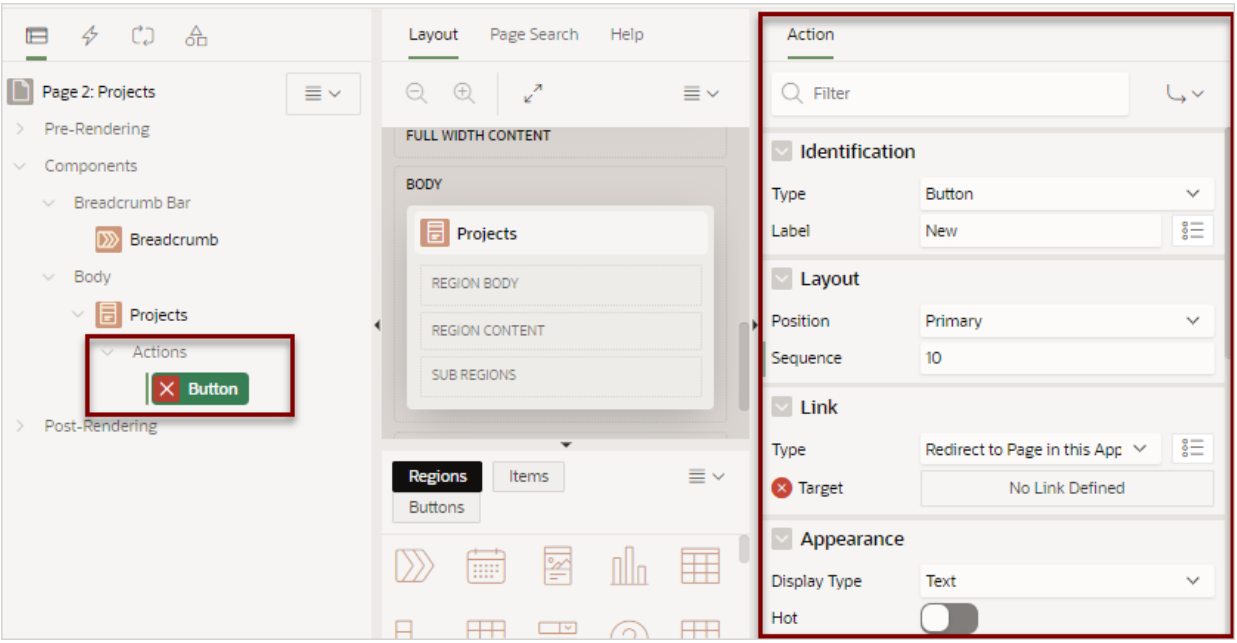
To create an action:

1. View the Cards page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select the Cards page.

   Page Designer appears.
2. Add an action:
   a. In the Rendering tab under Content Body, locate the cards region.
   b. Under the cards region, right-click **Actions** and select **Create Action**.



A new Action appears and the **Action** tab displays in Property Editor.



Edit the action in the Property Editor (right pane).

3. Identification, Type - Select the type of action to add to the card:
   - **Button** - Adds button to the card.
   - **Full Card** - Adds link to the full card.

- **Title** - Adds link to the card title.
- **Subtitle** - Adds link to the card subtitle.
- **Media** - Adds link to the card media.

You can define multiple actions of the same type. However, only button actions are rendered multiple times. For all other action types, the first TRUE display condition gets rendered, and the rest are ignored. If defining multiple action, add a display condition based on the column value and set it to check for each row.

4. Identification, Label - (Button only) Enter the text to display on the button. If button display type is icon, this text becomes the ALT text for the icon image.
5. Define the Link:
   a. Link, Type - Select the action to be performed when the specified action type is clicked. Options include:
      - **Redirect to Page in this Application** - Redirect to a page in the current application, with optional additional attributes for resetting pagination, setting the request value, clearing cache and setting item values, on the target page.
      - **Redirect to Page in a different Application** - Redirect to a page in a different application, with optional additional attributes for resetting pagination, setting the request value, clearing cache and setting item values, on the target page.

        > **Tip:**
        >
        > Redirecting to a page in different application relies on deep linking being enabled in the target application. To enable deep linking, go to the application's Security attributes and enable **Deep Linking**. See Configuring Security Attributes.

      - **Redirect to URL** - Redirect to the target URL you specify.
   b. Link, Target - Click **No Link Defined**.

      The Link Builder Target dialog appears. The attributes that display differs depending upon the Link Type. When redirecting to a page, the following attributes appear:
      - Target, Page - Enter a page number or page alias, or select the page number from the select list.
      - Set Items, Name - Enter or select page items (for example, `P3_ID`), application items (for example, `LAST_PAGE`) or interactive report filters (for example, `IR_EMPNO`) to be set into session state.
      - Set Items, Value - Enter or select a report column (for example,`&ID.`), enter an application or page item (for example, `&P2_EMPNO.`), or enter a static value (for example, `2`).
      - Clears/Reset, Clear Cache - Enter a comma-delimited list of page numbers (for example, `2,3`), items (for example, `P3_EMPNO,LAST_PAGE`), and report settings to be cleared. To reset pagination for a report enter `RP`. To clear or reset interactive report settings enter `CR` or `RR`.
      - Action - Select **None**, **Clear Regions**, **Reset Regions**, or **Reset Pagination**.

        Reset Pagination resets the pagination back to the first set of records, for reports on the target page. This has the same affect as passing **RP** in Clear Cache.
      - Advanced: Request - Enter the request to be used.

- Click **OK** to save your changes and exit the dialog.

6. Server Side Condition, Type - Select a condition type that must be met in order for this component to be rendered or processed.

   If you refer to Columns in your condition, you must specify For Each Row in order for the condition to be evaluated for each card. Otherwise, set to Once to evaluate the validations one time for the region.

7. Click **Save** or **Save and Run Page**.

   The Cards page loads.

8. Click a project to view the Maintain Project form.

8.6 Managing Smart Filters

A smart filters page features a single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a smart filter behaves similarly to faceted search, it features a more space efficient layout.

You can create a smart filters page with either the Create Application Wizard or Create Page Wizard, or add a smart filters region to an existing report in Page Designer.

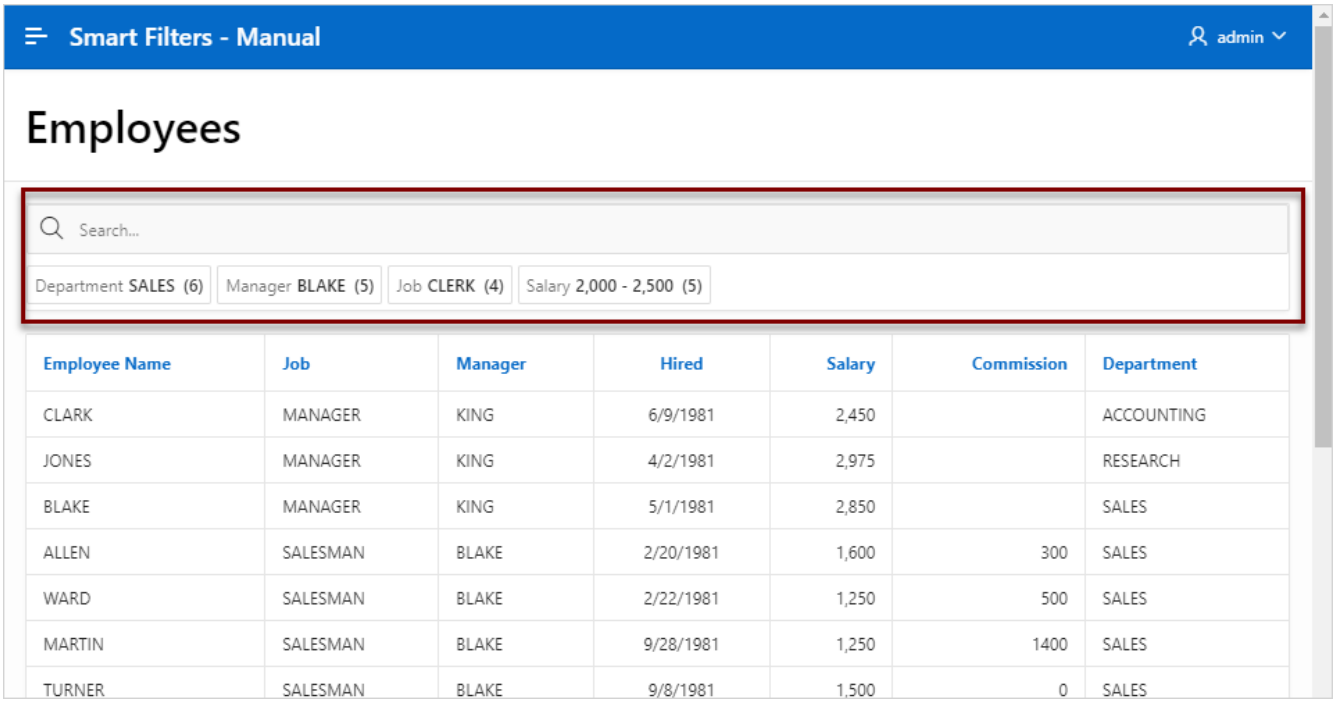
- Example Smart Filters
  View an example smart filters page.
- About Creating Smart Filters Pages
  Learn about creating smart filters pages.
- Creating Smart Filters Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains a smart filters page.
- Creating Smart Filters Using the Create Page Wizard
  Run the Create Page Wizard to add a new smart filters search page to an existing application.
- Creating a Smart Filters Page Manually
  Create a smart filters page by manually adding the page and smart filter components.
- Editing a Smart Filters Page
  Learn about how attributes can change the behavior of a smart filters page.

8.6.1 Example Smart Filters

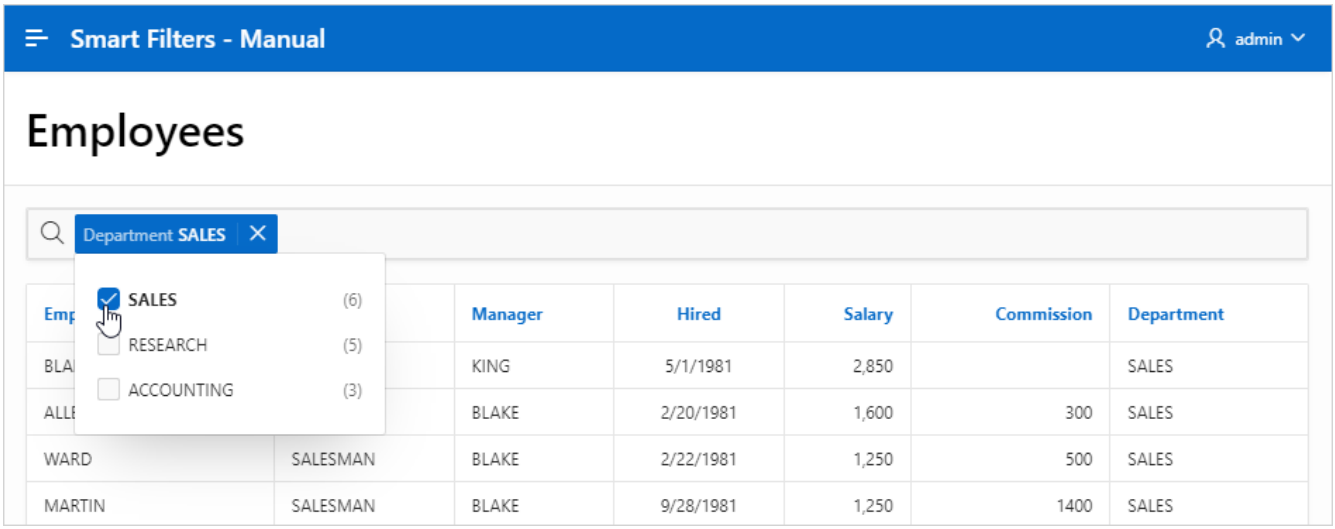
View an example smart filters page.

While a smart filters page behaves similarly to faceted search, it features a more space efficient layout. Smart filter pages include a single search field at the top of the page with filters that display as suggestion chips with counts that show how often a specific value occurs.

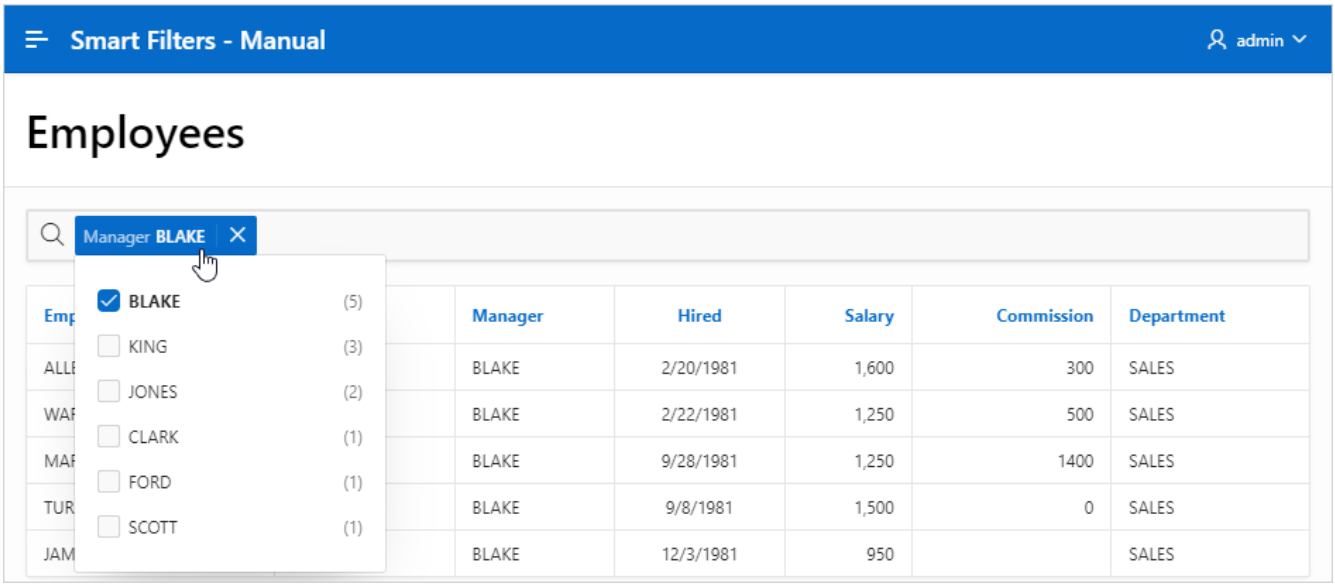
The following example shows a smart filters page built on the `EMP` table. Note the filter suggestion chips that display beneath the Search field.



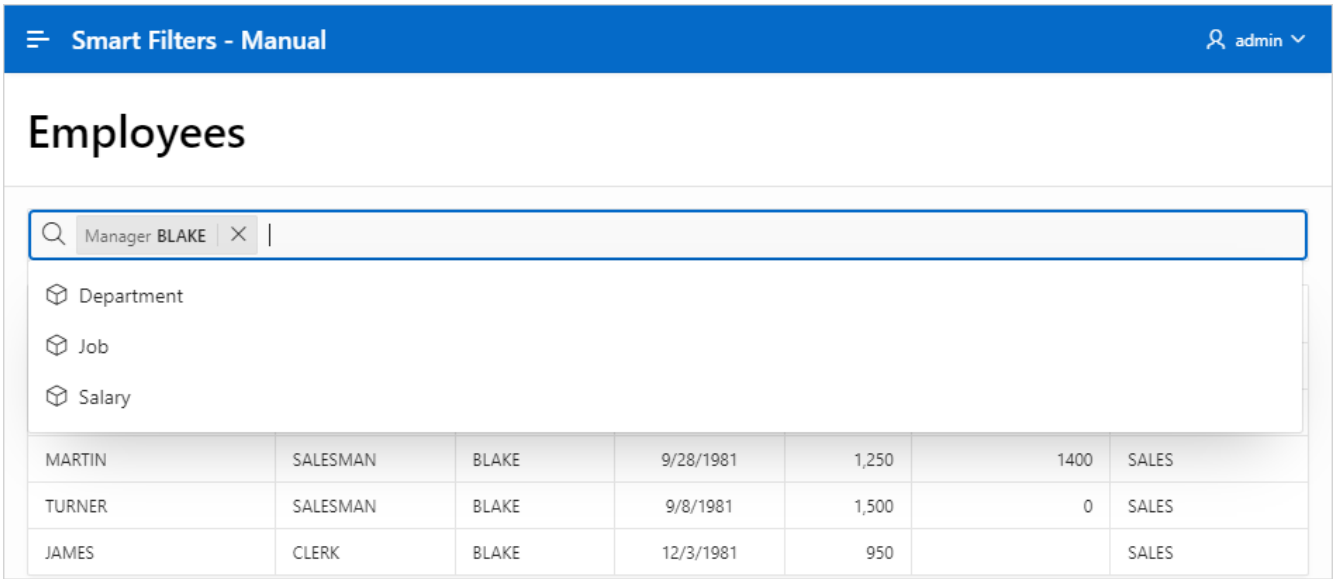
Clicking a the filter name displays a list of values. In this example the department, **Sales**, is selected and displays three values: SALES, RESEARCH, and ACCOUNTING.



If you click any place to right of the name, the suggestion chip becomes an applied filter chip and moves to the Applied Filters Area. Clicking the applied filter chip again displays the list. In this example, the Manager applied filter chip displays six managers.



To see other filters, click the Search field again. The other filters display.



To remove a filter, the click the **X** after the filter name.

> **Tip:**
>
> To view an example, install the sample app, *Sample Reporting*. To learn more, see Installing Apps from the Gallery.

8.6.2 About Creating Smart Filters Pages

Learn about creating smart filters pages.

Running the Create Application Wizard or Create Page Wizard is the easiest way to create a smart filters page. However, you can create smart filters page manually but the process involves multiple steps. To learn more, see Creating a Smart Filters Page Manually.

- About Creating Smart Filters Pages with a Wizard
  Create a smart filters page by running either the Create Application Wizard or Create Page Wizard.
- Search Filter Types
  Learn about supported smart filter types.

8.6.2.1 About Creating Smart Filters Pages with a Wizard

Create a smart filters page by running either the Create Application Wizard or Create Page Wizard.

Both the Create Application Wizard and Create Page Wizard support the creation of a smart filters page which contains a single search field at the top of the page with filters that display suggestion chips. Alternatively, you can add a smart filters region to an existing report in Page Designer. Each suggestion chip displays a single count for the specific suggestion values. The suggestion chip count only displays for filters that support counts, such as LOV-based filter types. Users can search, or click a suggestion chip to quickly filter the report. You choose if the report is a classic report or a cards report. If you create a smart filters page using the Create Application Wizard, you can also choose to include an update form.

Smart Filters, Breadcrumbs, and the Create Page Wizard

When adding a new smart filters page using the Create Page Wizard, breadcrumbs function a little differently. If breadcrumbs are disabled, the Smart Filters region is added to the Body container. However, if breadcrumbs are enabled, the Smart Filters region is added to the Breadcrumb container and a Breadcrumb displays at the top the page.

About Auto-Discovered Filters

When you build a smart filters page with either wizard, you can select a table or view on which to build the page. If you create the page based on a table, the filters are auto discovered using the Data Dictionary Cache. In contrast, if you build the page on a view or a SQL query, the facets (filters) are not auto-discovered. Instead, the wizard creates a filter region which only searches `VARCHAR2` columns.

About the Data Dictionary Cache

The Data Dictionary Cache report displays statistics and a data analysis of schema tables and caches in a workspace. Both the Create Application Wizard and Create Page Wizard use the Data Dictionary Cache when creating new applications and pages. You can preview how the wizards will render facets (filters) on the Data Dictionary Cache page by viewing the Table Column Cache report. The wizard uses the first top five facet (filter) search score columns with score greater than or equal to 20.

If a table changes due to DDL or DML modifications, the Oracle APEX nightly job `ORACLE_APEX_DICTIONARY_CACHE` refreshes the Data Dictionary Cache. However, you can refresh the cache manually by navigating to the Data Dictionary Cache and clicking either **Gather Statistics and Refresh Cache** or **Refresh Cache Only** at the workspace or table-level.

> **See Also:**
>
> Accessing the Data Dictionary Cache Report from Workspace Administration or Accessing the Data Dictionary Cache Report from Administration Services in *Oracle APEX Administration Guide*

8.6.2.2 Search Filter Types

Learn about supported smart filter types.

A smart filters page which contains a single search field at the top of the page with filters that display as suggestion chips.

You can specify a search filter type in Page Designer by selecting the filter in the Rendering tab and configuring the **Identification**, **Type** attribute in the Property Editor. Each filter has additional attributes that control behavior. To learn more about a specific attribute, select the attribute and click the **Help** tab in the central pane.

> **Tip:**
>
> To see an example of manually adding a Search filter field, Checkbox Group filters, and a Range filter, see Creating a Smart Filters Page Manually.

App Builder supports the following search filter Types:

- **Checkbox Group** - Displays multiple values as check boxes, enabling the end user to select multiple values. A list of values is required for items displayed as check boxes. The values corresponding to the checked boxes are stored in a single colon-delimited string.
- **Input Field** - Displays a text field that enables the user to filter by text input. The comparison operator is either set by the developer or can be chosen by the end user. Includes support for filtering by date, timestamp (with local time zone). For DATE or TIMESTAMP values, use the `YYYYMMDDHH24MISS` canonical format.
- **Radio Group** - Displays multiple values as radio group options, enabling the end user to select a single value.
- **Range** - Displays an item with a built-in list of values selector. Each option represents a range of values, from a lower to an upper boundary. The facet supports single or multiple selection, and manual entry.

  For filtering and count computation, the lower boundary of a specified range is always treated inclusive, and the upper boundary is exclusive. Example LOV Syntax:

  - Less than 100:

    ```
    |100
    ```

  - Between 100 and 500 (exclusive):

    ```
    100|500
    ```

  - 500 and more:

    ```
    500|
    ```

For a range with a Manual Input, filter by date, timestamp (with local time zone). For DATE or TIMESTAMP values, use the `YYYYMMDDHH24MISS` canonical format.

- **Search** - Displays a text field, enabling the end user to search the list entries of the filters in the Smart Filters region. A Smart Filters region supports one Search facet, which will always be at the top of the list of filters, regardless of its sequence.

8.6.3 Creating Smart Filters Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains a smart filters page.

A smart filters page can include a classic report or a cards report. If you select classic report, you can also choose to include an update form.

> **Tip:**
>
> If you are creating a smart filters page that displays as a report, clicking **Include Form** automatically creates a form page for creating or updating records. This option is not available in the Create Page Wizard.

To create an application that includes a smart filters search page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
7. Click **Add Page** and select **Smart Filters**.
8. In the Add Smart Filters Page dialog:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Select a report type: **Report** or **Cards**.
   d. Table - Select a table.
   e. What appears next depends upon whether you selected **Report** or **Cards**.
      - If **Report** - Select **Include Form** to include a form page for creating or updating records.
      - If **Cards** - Select the display: **Grid**, **Float**, and **Horizontal (Row)**.

        Select the information to display on each card: **Title Column**, **Body Column**, **Icon Initials Column**, and **Badge Column**.
   f. Advanced - Expand **Advanced** to configure these options:

- Set as Home Page - Enable this to make this page the home page for the application.
- Page Help - Enter text to be displayed when the user selects Page Help.

g. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

8.6.4 Creating Smart Filters Using the Create Page Wizard

Run the Create Page Wizard to add a new smart filters search page to an existing application.

If you are using the latest Universal Theme, breadcrumbs function a little differently when adding a new smart filters page using the Create Page Wizard. If breadcrumbs are disabled, the Smart Filters region is added to the Body container. However, if breadcrumbs are enabled, the Smart Filters region is added to the Breadcrumb container and a Breadcrumb displays at the top the page.

> **Tip:**
>
> To learn more about breadcrumbs, see Editing Breadcrumbs

To create a smart filters page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Smart Filters**.
5. Under Page Definition:

   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.
6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.

- **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table /View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.
8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Display As - Select how you want to display the filtered region. Options include: **Report** or **Cards**.
11. Select filters - Select the filters include in this report.

    Filter recommendations are based on statistics of data in your table. Click **Refresh** to refresh the column statistics to provide better recommendations.
12. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

8.6.5 Creating a Smart Filters Page Manually

Create a smart filters page by manually adding the page and smart filter components.

Using the Create Application Wizard or Create Page Wizard is the easiest way to create a smart filters page. However, you can create a smart filters page manually. At a minimum, creating a smart filters page manually involves the following steps:

- Create a report region that supports smart filters. Supported report regions include classic report, cards, map, or calendar.

  > **Tip:**
  >
  > You must create maps and calendars manually since the Create Application and Create Page Wizards only support creating smart filter pages with a classic report and cards report.

- Add a smart filters region.
- Add individual filters to the smart filters region.

The following example creates a smart filters page which contains a classic report built on the `EMP` table. The `EMP` table is available in the sample dataset, EMP / DEPT.

To create a filtered search manually.

1. Run the Create Application Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Click **Create**.

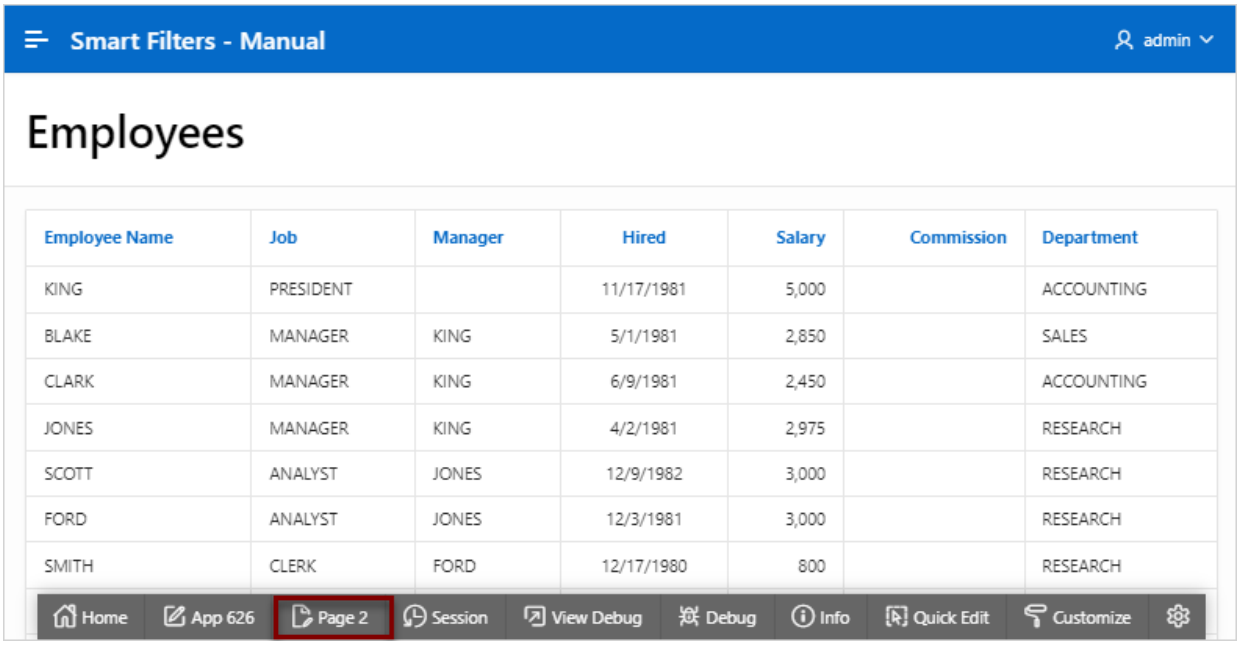
      The wizard prompts you to select a method of creating an application.
   c. Click **Use Create App Wizard**.
2. On Create an Application:
   a. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
   b. Name - Enter a name to identify the application. For example, enter `Smart Filters - Manual`
   c. Appearance - Accept the default Theme Style and menu layout (Vita, Side Menu). To change it, click the **Set Appearance** icon adjacent to the Appearance field.
   d. Under Pages, click **Add Page**. Expand **Additional Pages** at the bottom of the window and select **Classic Report**.
   e. On Add Report Page:
      - Page Name - Enter a name for this page. For example, enter `Employees`
      - Set Icon - Accept the default, or select an icon to display in the navigation menu for this page.
      - Table or View - Select a table or view on which to build the page. Select **EMP**.
      - Click **Add Page**.
   f. Click **Create Application**.

      The Application home page appears.

3. View the new page.
   a. On the Application home page, select the new page, **2 - Employees**.

      The page displays in Page Designer.
   b. Click **Save and Run Page** on the right side of the Page Designer toolbar.
   c. When prompted, enter your workspace username and password and click **Sign In**.

      A Employees report appears.



   d. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar where *X* is the page number.
4. Add a search region:

   When adding a search region, you can add it in the Body or Breadcrumb Bar. Adding it to the Breadcrumb Bar ensures it always displays at the top of the page. In this example, you add the search region to the Body.
   a. In the Rendering tab (or left pane), right-click **Body** and select **Create Region**.
   b. Select the new region. Edit the following Region attributes in the Property Editor (or right pane):
      - Identification, Title - Enter `Search`
      - Identification, Type - Select **Smart Filters**.
      - Source, Filtered Region - Select **Employees**.
   c. Update the Template Options:
      - Appearance, Template Options - Click **Use Template Defaults, Scroll - Default**.

        The Template Options dialog appears.
      - Common, General - Select **Remove Body Padding**.
      - Common, Header - Select **Hidden but accessible**.
      - Click **OK**.

5. Update the Search region appearance:
   a. Layout, Sequence - Enter `5`
   b. Layout, Column Span - Select **8**.
   c. Save your changes. Click **Save**.
6. Update the filter for searching multiple columns:
   a. Select `PX_SEARCH` where *X* represents the page number.
   b. Source, Database Column(s) - Enter `ENAME,JOB,MGR,DEPTNO`
   c. Save your changes. Click **Save**.
7. Add a filter for the `DEPTNO` column:
   a. In the Rendering tab, right-click the **Filters** and select **Create Filter**.
   b. Select the new filter and edit the following attributes in the Property Editor:
      - Identification, Name - Enter `PX_DEPTNO` where *X* represents the page number.
      - Identification, Type - Accept the default, **Checkbox Group**.
      - Source, Database Column - Verify that the column is **DEPTNO**.
      - Label, Label - Enter `Department`
      - List of Values, Type - Select **Shared Component**.

        The Create Application Wizard created a lookup list of values named `DEPT.DNAME` so that the `DEPT` column displays the department name instead of the department number.
      - List of Values, List of Values - Select `DEPT.DNAME`
   c. Save your changes. Click **Save**.
8. Add a filter for the `MGR` column:
   a. In the Rendering tab, right-click the **Filters** and select **Create Filter**.
   b. Select the new filter and edit the following attributes in the Property Editor:
      - Identification, Name - Enter `PX_MGR` where *X* represents the page number.
      - Identification, Type - Accept the default, **Checkbox Group**.
      - Source, Database Column - Verify that the column is **MGR**.
      - Label, Label - Enter `Manager`
      - List of Values, Type - Select **Shared Component**.

        The Create Application Wizard created a lookup list of values named `EMP.ENAME` so that the `MGR` column displays the manager name instead of the employee number.
      - List of Values, List of Values - Select **EMP.ENAME**.
   c. Save your changes. Click **Save**.
9. Add a filter for the `JOB` column:
   a. In the Rendering tab, right-click **Filters** and select **Create Filter**.
   b. Select the new filter and edit the following attributes in the Property Editor:
      - Identification, Name - Enter `PX_JOB` where *X* represents the page number.
      - Identification, Type - Accept the default **Checkbox Group**.

- Label, Label - Verify the label is `Job`.
- Source, Database Column - Verify that the column is **JOB**.
- List of Values, Type - Select **Distinct Values**.

c. Save your changes. Click **Save**.

10. Add a filter for the `SAL` column:

a. In the Rendering tab, right-click **Filters** and select **Create Filter**.

b. Select the new filter and edit the following filter attributes in the Property Editor:

- Identification, Name - Enter `PX_SAL` where *X* represents the page number.
- Identification, Type - Select **Range**.
- Label, Label - Enter `Salary`
- Source, Database Column - Verify that the column is **SAL**.
- Source, Data Type - Select **NUMBER**.
- Settings, Select Multiple - Enable **Select Multiple**.

c. Define the `PX_SAL` filter as a static list of values:

- List of Values, Type - Select **Static Values**.
- List of Values, Static Value - Click **Display1, Display2**.

  In the Static Values dialog, define the following:

  – Row 1:

    Display Value - Enter `<900`

    Return Value - Enter `|900`

  – Row 2:

    Display Value - Enter `900 - 1#G#300`

    Return Value - Enter `900|1300`

  – Row 3:

    Display Value - Enter `1#G#300 - 2#G#000`

    Return Value - Enter `1300|2000`

  – Row 4:

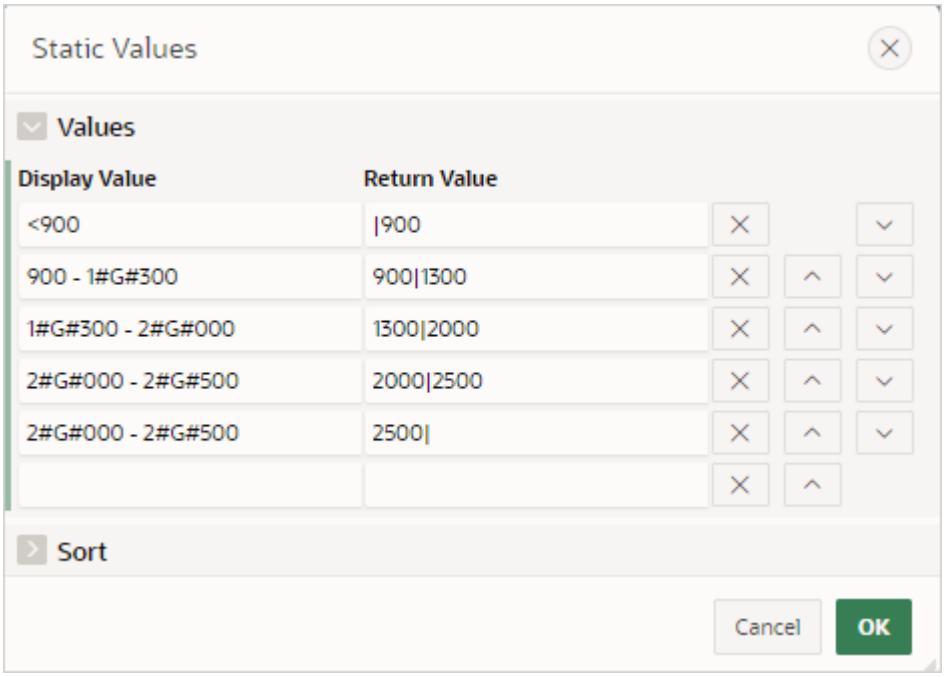
    Display Value - Enter `2#G#000 - 2#G#500`

    Return Value - Enter `2000|2500`

  – Row 5:

    Display Value - Enter `2#G#000 - 2#G#500`

    Return Value - Enter `2500|`



- Click **OK** to save your changes.

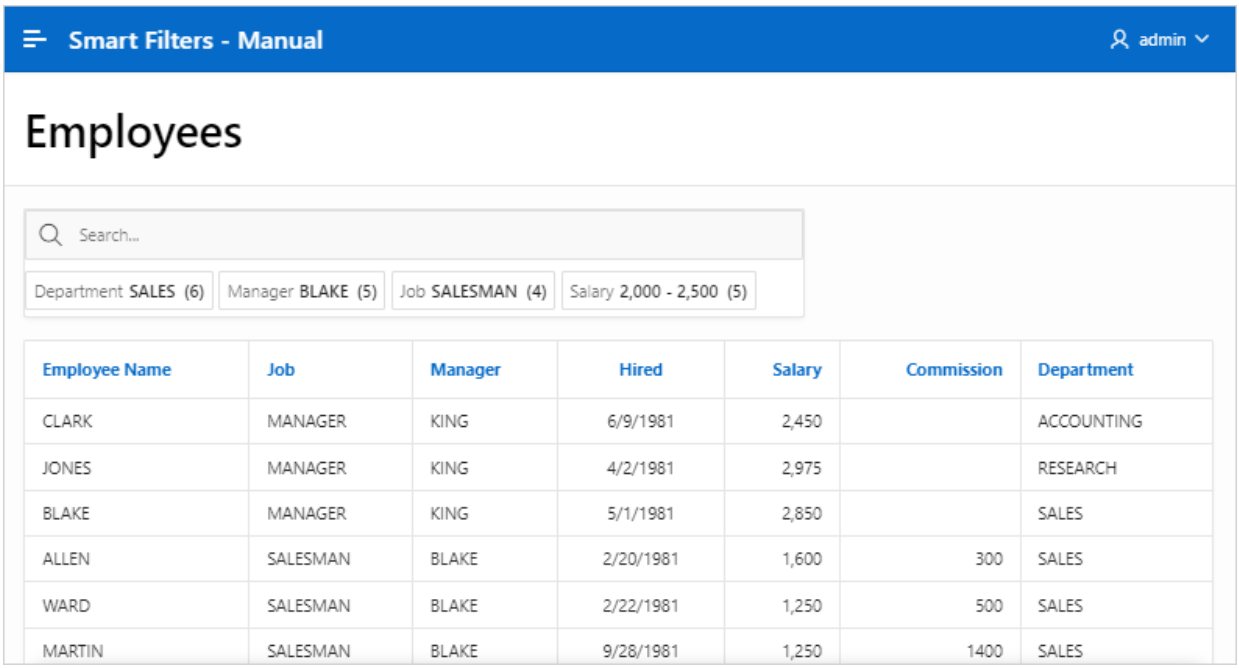
d. Save your changes. Click **Save**.

11. View the page.

a. Click **Save and Run Page** in the upper right corner.

b. If prompted, enter your workspace username and password and click **Sign In**.

The revised page appears. Notice four suggestion chips display under the search field.



c. Test the page. Select filters. To remove a filter, click the **X** after the filter name.

8.6.6 Editing a Smart Filters Page

Learn about how attributes can change the behavior of a smart filters page.

> **Tip:**
>
> Many examples in this section are built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data, or the `EMP` table available in the sample dataset, EMP / DEPT. To install a sample dataset, see Using Sample Datasets in *Oracle APEX SQL Workshop Guide*.

- Reviewing Smart Filters Page Components
  Review the components that comprise a smart filters page by viewing attributes in Page Designer.
- Understanding a Smart Filters Page
  Learn about the regions that comprise a smart filters page.
- Adding a New Filter
  Add new filters in Page Designer by right-clicking Filters and selecting Create Filter from the context menu.
- Editing Filter Attributes
  Edit attributes to change filter behavior.

8.6.6.1 Reviewing Smart Filters Page Components

Review the components that comprise a smart filters page by viewing attributes in Page Designer.

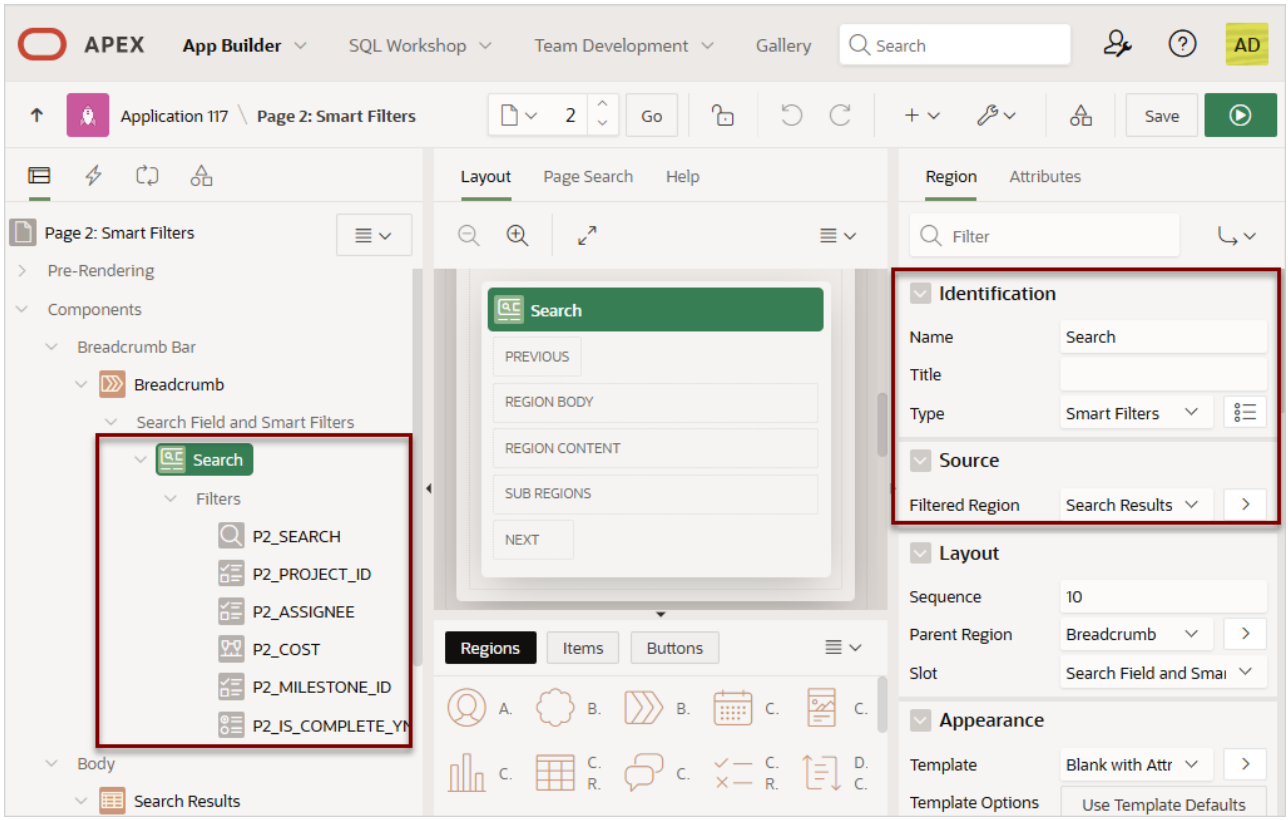
A smart filters page features a single search field at the top of the page with filters. Search Results can display as a classic report, cards, map, or calendar.

> **Tip:**
>
> You must create maps and calendars manually since the Create Application and Create Page Wizards only support creating smart filter pages with a classic report and cards report.

To review the components of a smart filters page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Review the Search region. In the Rendering tab, locate and select the **Search** region.



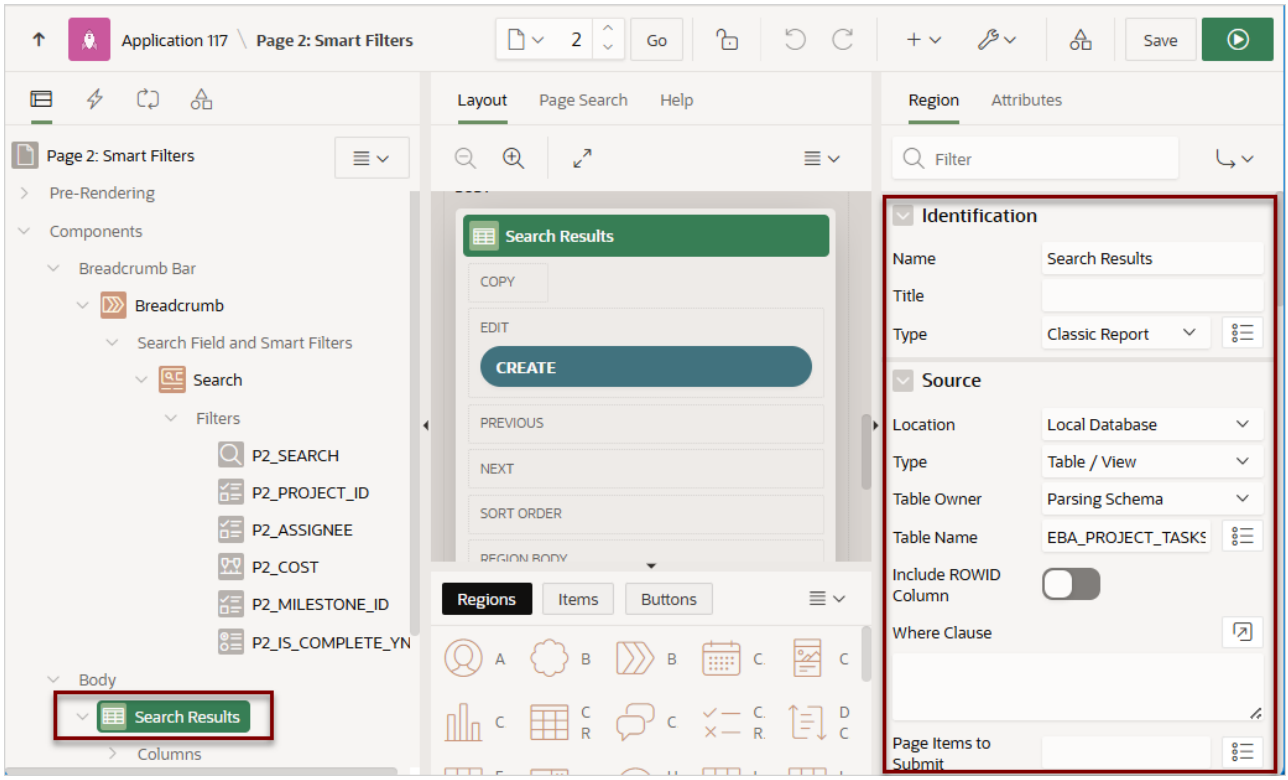
Note the Search region Type is **Smart Filter**. In this example, the Search region contains six filters. Filters display in runtime as suggestion chips with counts that show how often the specific value occurs. The value shown in the suggestion chip is configured with the attributes in the Suggestions group. Filters are similar to items but feature the following unique characteristics:

- **Naming Conventions** - Filters use the same naming convention as items. By default, wizards prefix filter names with `P<page no>_<item name>` (for example, `P2_SEARCH`).
- **Associated with a Column** - Filters are associated with a column from the Search Results (or report) region.
- **Checkbox Group Filters** - The filters with the Type **Checkbox Group** create a UI for selecting or entering the filter values. They are not implemented the same as page items even if they have the same type name. For this reason, most dynamic actions do not apply to filters.

3. Select a filter and explore the configurable options.

   To view Help, select the attribute and click the **Help** tab.

4. Review the report region. In the Rendering tab, locate and select the Search Results (or report).



In the previous example, note that the attributes indicate the report Type is a **Classic Report** built on the `EBA_PROJECT_TASKS` table in a local database:

> **See Also:**
>
> About Item Naming Conventions

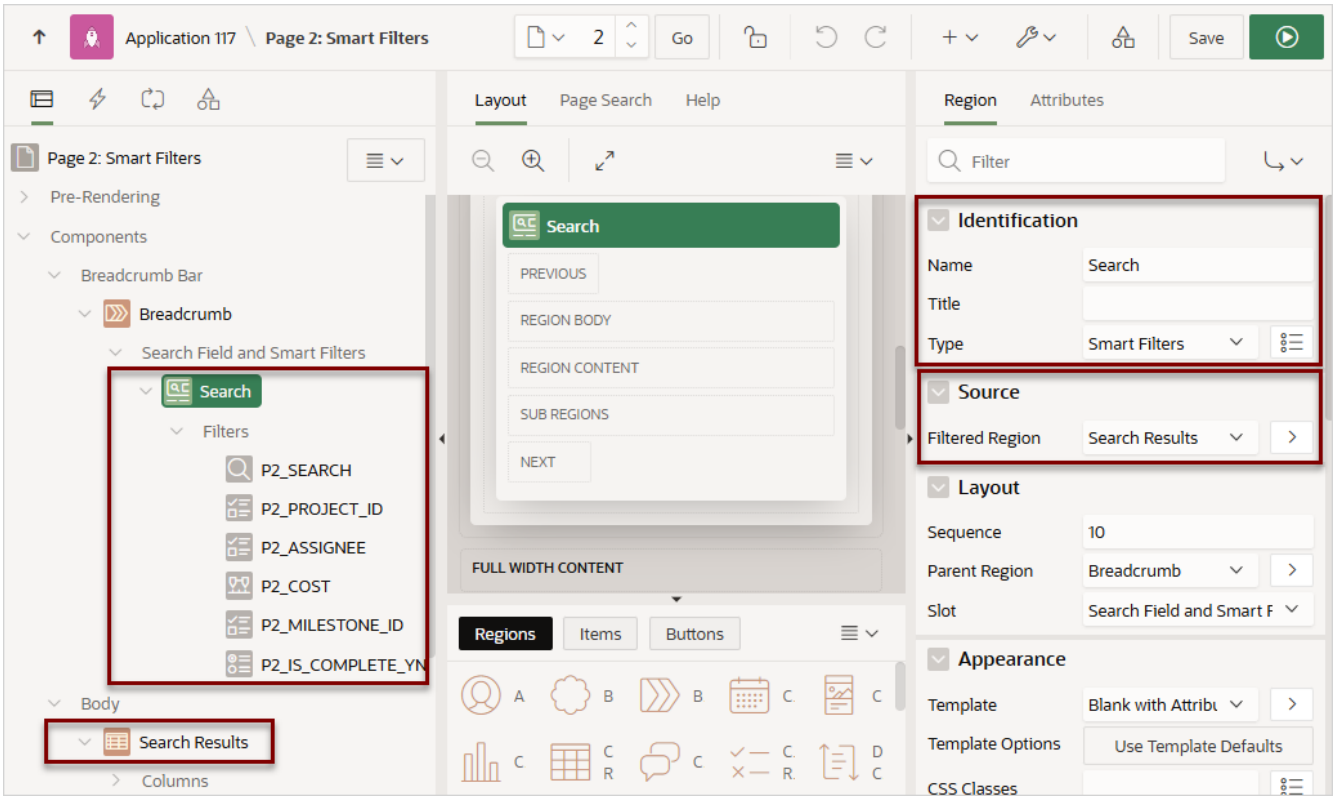
8.6.6.2 Understanding a Smart Filters Page

Learn about the regions that comprise a smart filters page.

About the Search Region

A smart filters page features a Search region which enables users to narrow down the search result and a report region such as classic report or cards report.

In Page Designer, the **Search** region displays in the left pane or Rendering tab. Selecting the Search region displays attributes in the Property Editor. In the following image, the Search region **Type** is defined as **Smart Filters**. The **Filtered Region** attribute identifies the Search Results region. In this example, **Filtered Region** is as a classic report named **Search Results**.



> **Tip:**
>
> Remember that each attribute in the Property Editor has associated Help. To view Help, select the attribute in the Property Editor and then click the **Help** tab in the central pane. Help describing the attribute appears in the central pane.

About Filters

Search filters display under **Filters** in the Rendering tab. When you select a filter, the associated attributes display in the Property Editor. Filters use the same naming convention as items. By default, wizards prefix filters names using the following convention: `P<page no>_<item name>`. For example, in the previous example, the first filter is named `P2_SEARCH)`.

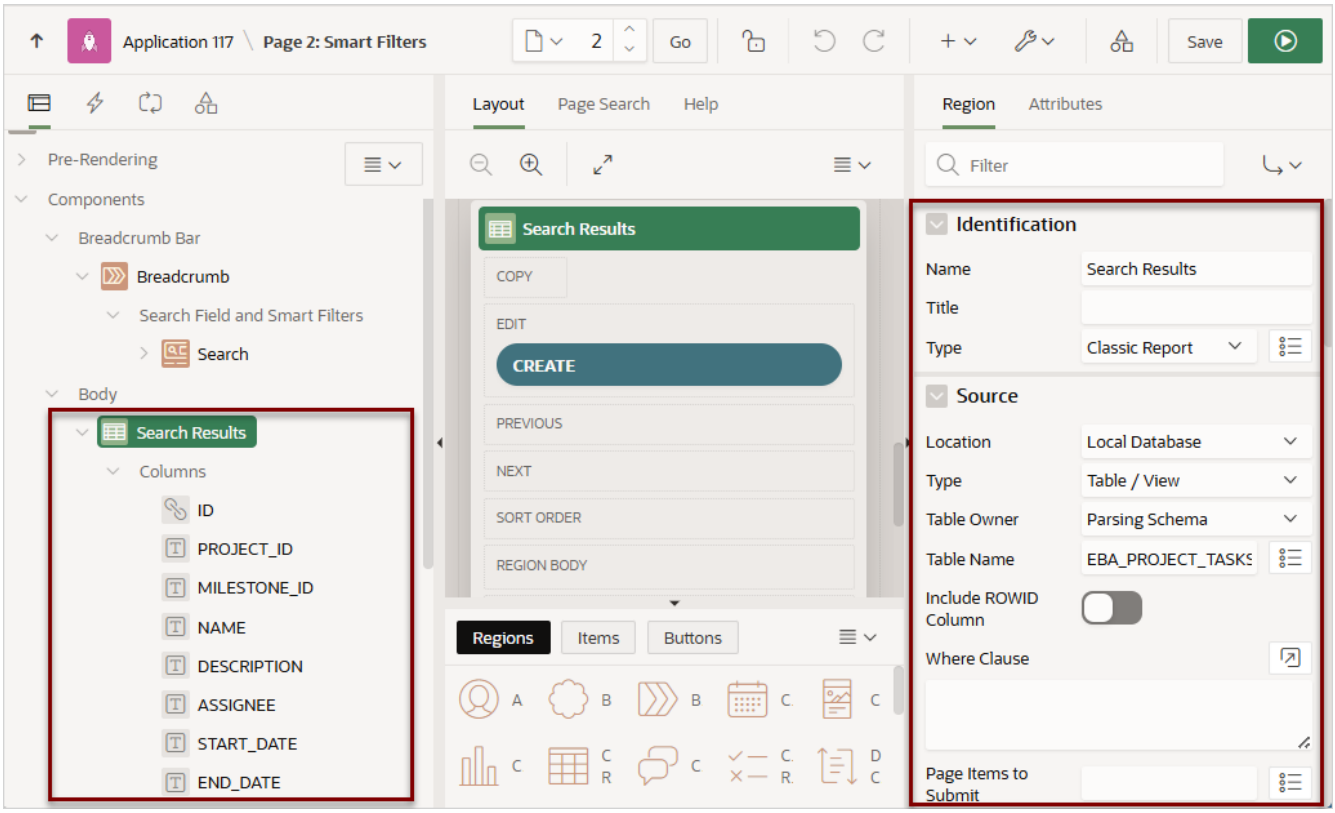
Filters are similar to items but feature the following unique characteristics:

- **Associated with a Column** - The filters are associated with a column from the report region.
- **Checkbox Group Filters** - Filters with the Type **Checkbox Group** create a UI for selecting or entering the filter values. They are not implemented the same as page items even if they have the same type name. For this reason, most dynamic actions do not apply to filters.

About the Search Results

A smart filters page includes a Search Results region. If you create a smart filters page using either the Create Application Wizard or Create Page Wizard, the Search Results region is either a classic report or cards report. If you create a smart filters page manually, the Search

Results region can also be a manually created map or calendar. In the following example, Search Results is a classic report built on the `EBA_PROJECT_TASKS` table.



About Cards Page Search Results

The Search Results on a smart filter search page can display as a cards report. A cards page functions as a colorful way to display a subset of information and then enable the user to link to more detail. You can create an action to link from a full card, a title, a subtitle, media, or a button. To see an example, see Using Actions to Link from a Cards Page.

> **See Also:**
>
> About Item Naming Conventions

8.6.6.3 Adding a New Filter

Add new filters in Page Designer by right-clicking Filters and selecting Create Filter from the context menu.

Filters map to specific database column. When creating a new filter, you can create the following types: Checkbox Group, Input Field, Radio Group, and Range.

You can control the filter order in the UI by reordering the filters in the Rendering tab. Note that there must always be one Search type filter.

The following example demonstrates how to create Range filter to filter employee salary. In this example, the filter maps to the `SAL` column in the `EMP` table available in the sample dataset, EMP / DEPT.

To create a Range filter:

1. View the smart filters page in Page Designer.
2. In the Rendering tab, right-click **Filters** and select **Create Filter**.
3. Select the new filter and edit the following filter attributes in the Property Editor:
   a. Identification, Name - Enter `PX_SAL` where *X* represents the page number.
   b. Identification, Type - Select **Range**.
   c. Label, Label - Enter `Salary`
   d. Source, Database Column - Verify that the column is **SAL**.
   e. Source, Data Type - Select **NUMBER**.
   f. Settings, Select Multiple - Enable **Select Multiple**.
4. Define the filter as a static list of values:
   a. List of Values, Type - Select **Static Values**.
   b. List of Values, Static Value - Click **Display1, Display2**.

      In the Static Values dialog, define the following:

      - Row 1:

        Display Value - Enter `<900`

        Return Value - Enter `|900`
      - Row 2:

        Display Value - Enter `900 - 1#G#300`

        Return Value - Enter `900|1300`
      - Row 3:

        Display Value - Enter `1#G#300 - 2#G#000`

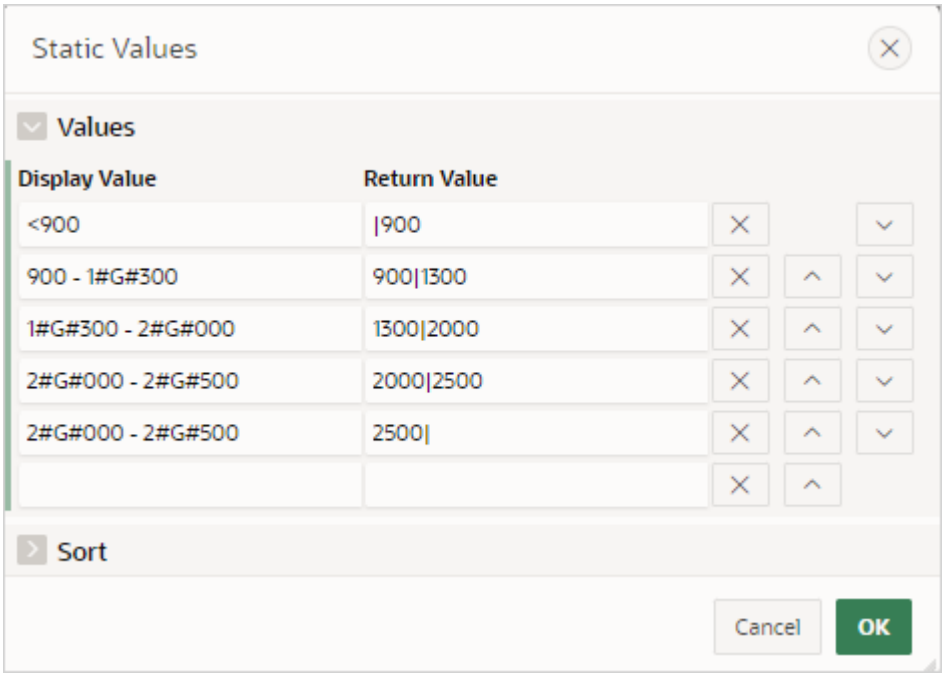
        Return Value - Enter `1300|2000`
      - Row 4:

        Display Value - Enter `2#G#000 - 2#G#500`

        Return Value - Enter `2000|2500`
      - Row 4:

        Display Value - Enter `2#G#000 - 2#G#500`

        Return Value - Enter `2500|`



- Click **OK** to save your changes.

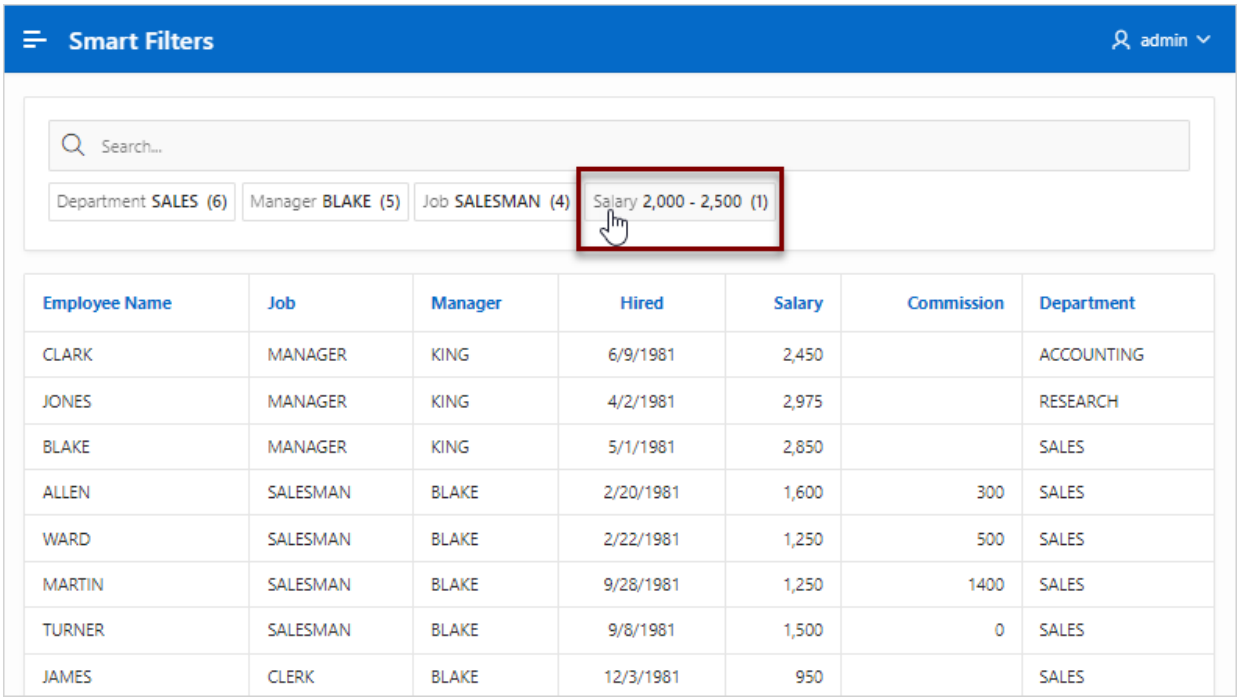
5. Configure the Suggestions attributes for the the `PX_SAL` filter:
   a. Select the the `PX_SAL` filter (if not already selected).
   b. Suggestions, Static Values - Select how suggestions are determined for this filter. Options include:
      - **Dynamic** - Picks the first visible entry from the List of Values.
      - **Static Values** - Enter a comma delimited list of values.
      - **SQL Query** - Values are returned from the SQL Query entered.

      For this example, select **Static Values**.
   c. Suggestions, Static Values - Enter `2000|2500`

      This will display **2,000 - 2,500** on the suggestion chip at runtime.
   d. Save your changes. Click **Save**.
6. View the page.
   a. Click **Save and Run Page** in the upper right corner.
   b. If prompted, enter your workspace username and password and click **Sign In**.

      The revised page appears. Notice that a Salary suggestion chip appears that reads:

      `Salary 2,000 - 2,500 (1)`



c. Select the **Salary** on the Salary suggestion chip.

   The revised page appears. Notice that the Salary suggestion chip has become an applied filter chip and moved to the Applied Filters Area.

d. Test the new filter. To remove a filter, click the **X** after the filter name.

8.6.6.4 Editing Filter Attributes

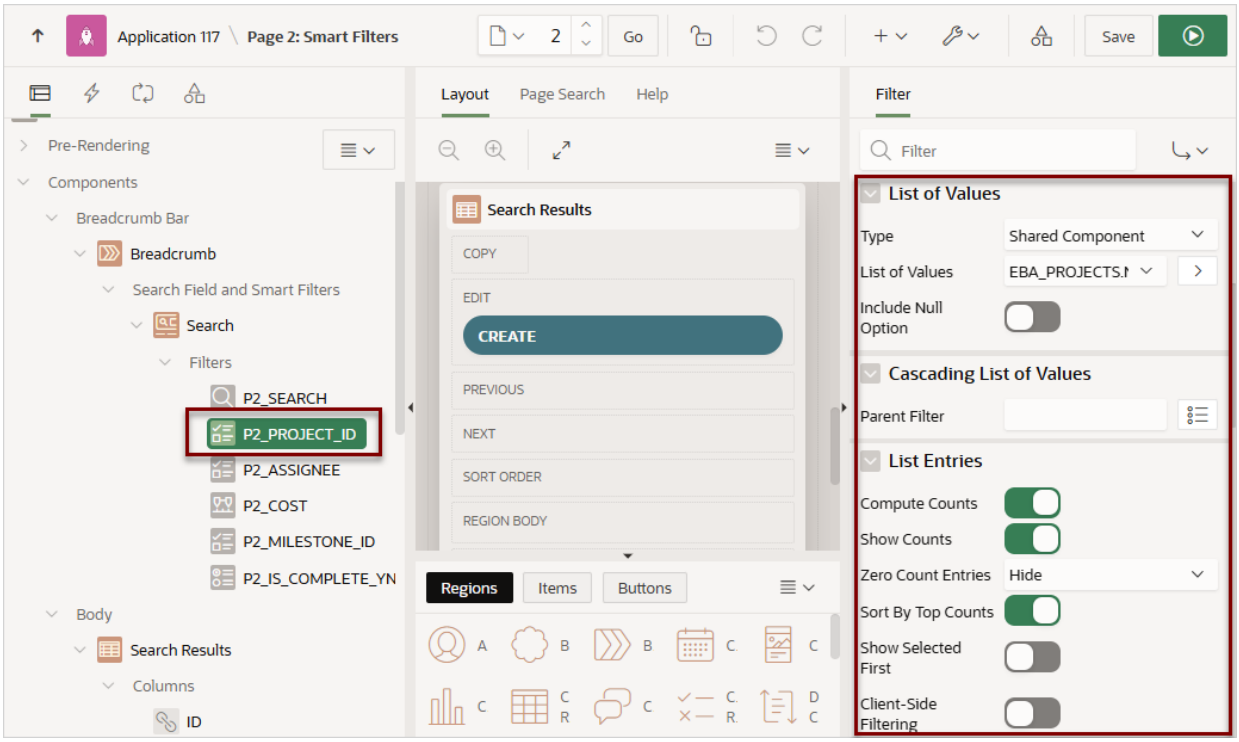
Edit attributes to change filter behavior.

The following example demonstrates how to review and edit the **Client-Side Filtering** and **Depending On** attributes and change filter behavior. Enabling **Client-Side Filtering** adds a search field which enables users to filter the list. **Depending On** makes the display of one filter dependent upon another. The following example is also built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.

To view and edit filter attributes:

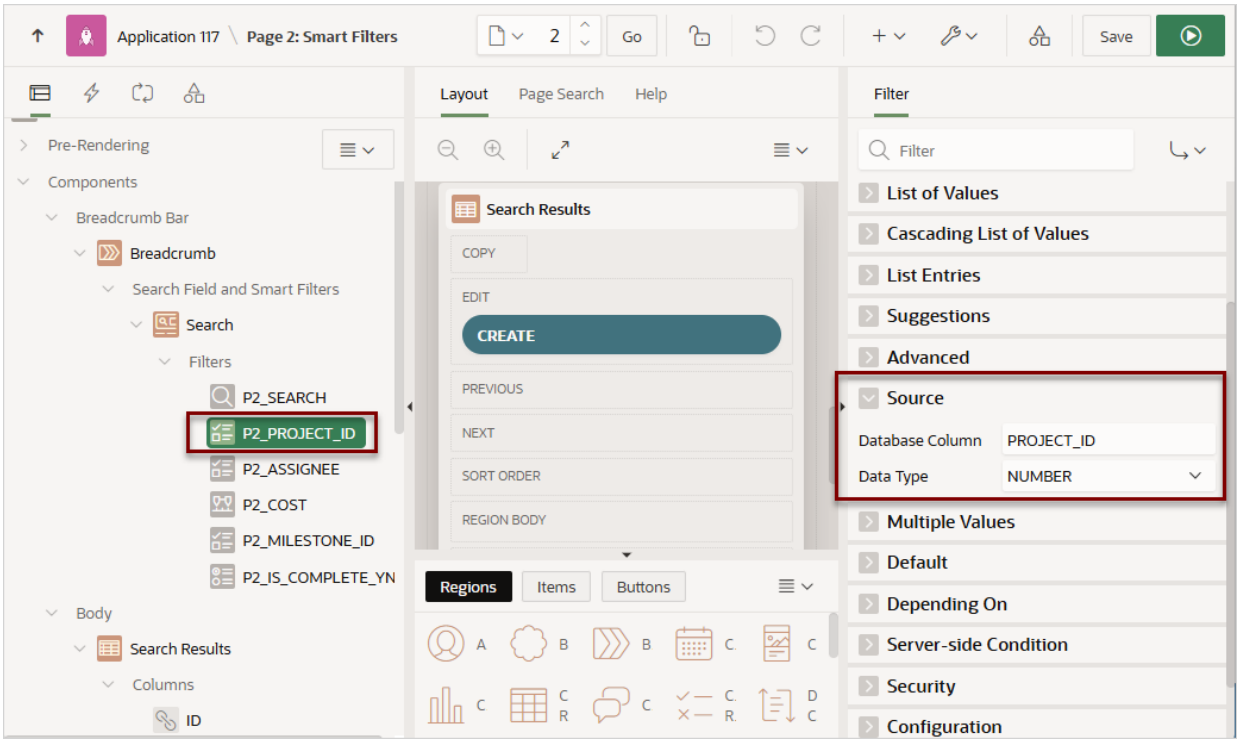
1. View the smart filters page in Page Designer.
2. View the filter attributes, **P2_PROJECT_ID**:
   a. In the Rendering tab (or left pane) locate the **Search** region and select the filter, **P2_PROJECT_ID**.
   b. In the Property Editor (or right pane) find the **List Entries** group.



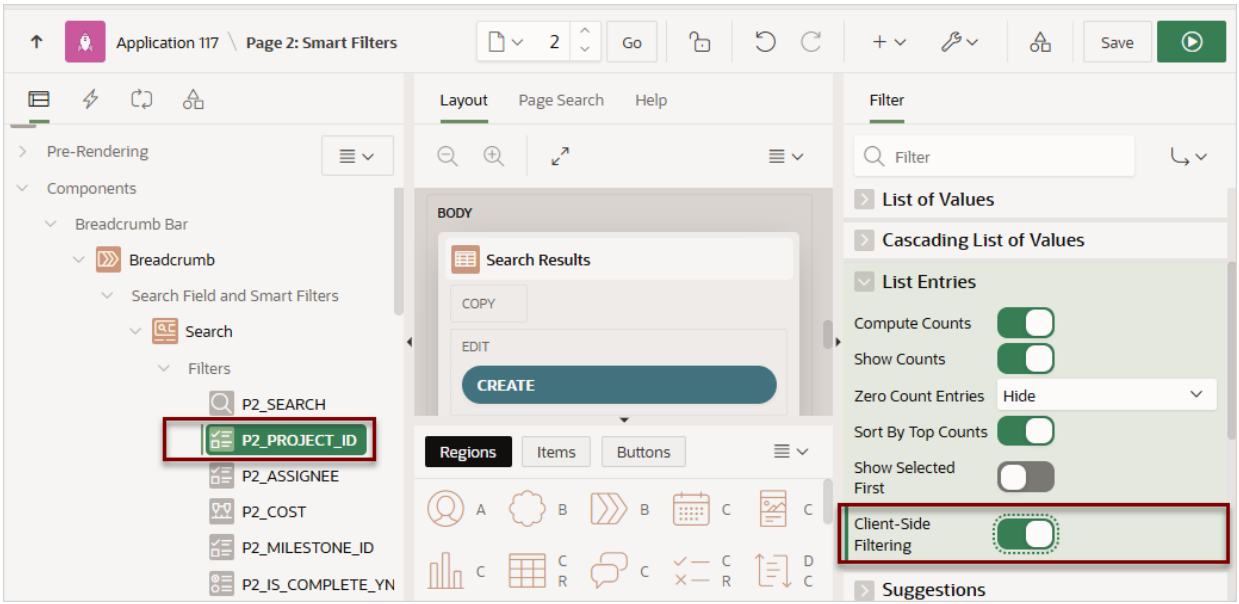
> **Tip:**
>
> The maximum amount of items which can be returned by an LOV, or by Distinct Values, is 5,000. If an LOV returns more than 5,000 items, search cuts the LOV silently at 5,000 values. When logged in as a developer, APEX raises a runtime error once more than 2,000 LOV items are returned. These errors indicate that the LOV returns too many values for a useful search UI.

c. In the Property Editor, find the **Source** group.

   The **Source** group indicates that the P5_PROJECT_ID filter is based on the `PROJECT_ID` column which has a data type of `Number`.

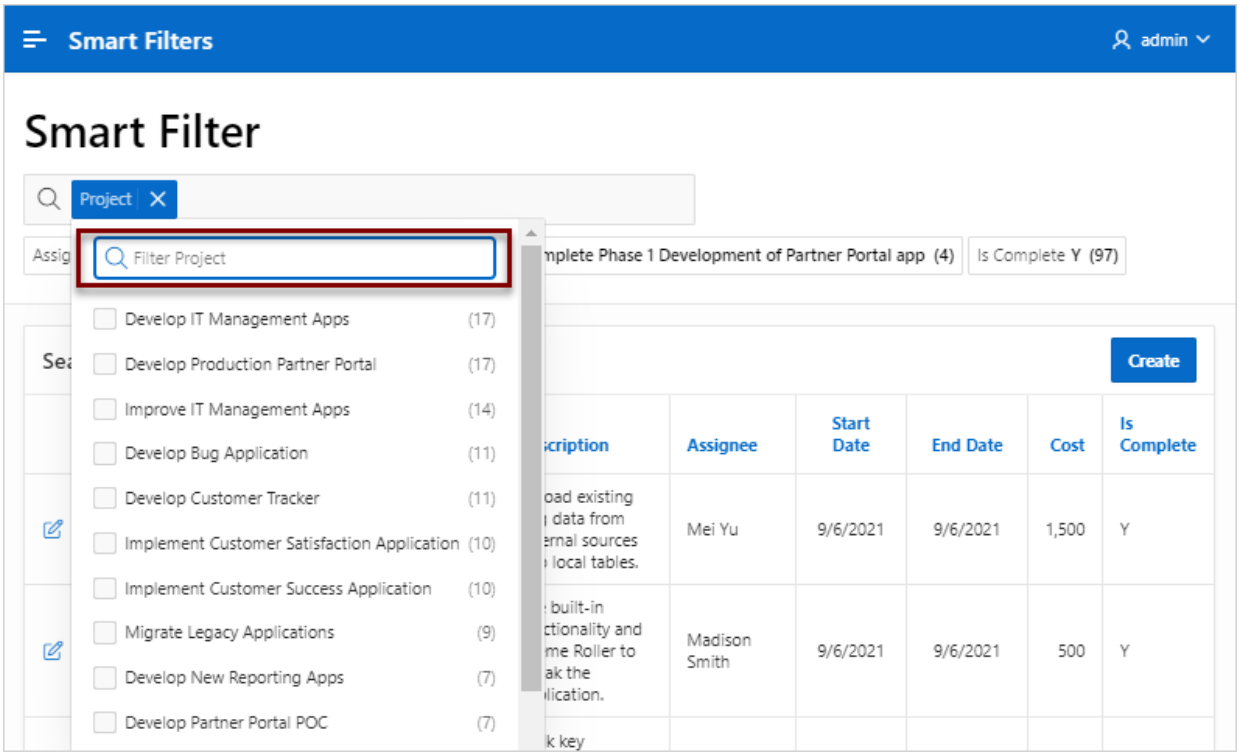


3. Update **P2_PROJECT_ID** and enable the Client-Side Filtering attribute:
   a. In the Rendering tab, select the **P2_PROJECT_ID** filter (if not already selected).
   b. In the Property Editor, find **List Entries** and enable **Client-Side Filtering**. Enabling this option renders a search field above the list entries.



   c. Review your changes. Click **Save and Run Page**.

      Click the name **Project** on the Project suggestion chip. Notice a **Filter Project** search field appears above the list.



d. Test the new Filter Project field. For example, entering the keyword "develop" will update the Project list to only include filters containing that phrase.

e. Click **Page XX** on the Runtime Developer Toolbar to return to Page Designer, where *XX* is the page number.

4. Only show Milestones for the selected Project by enabling the **Depending On** attribute.

   a. In the Rendering tab locate the **Search** region and select the **P5_MILESTONE_ID** filter.

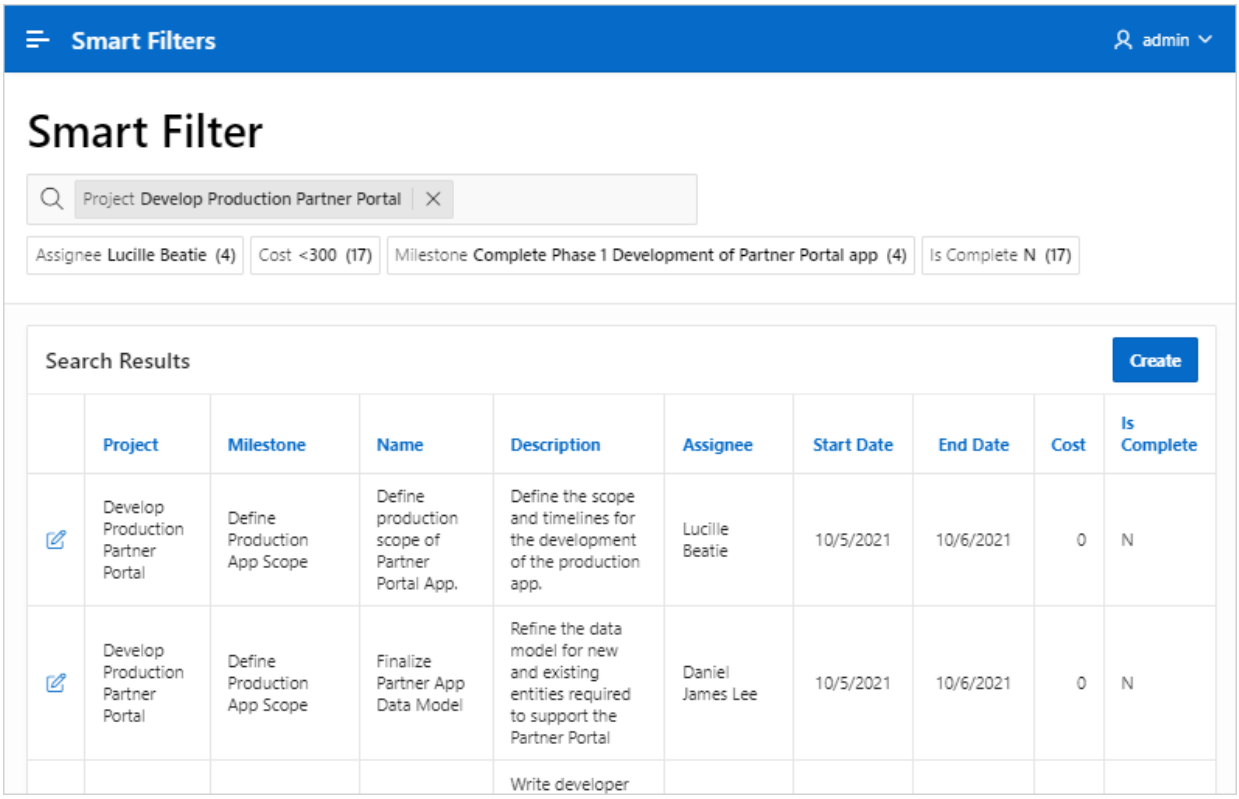
   b. In the Property Editor, find **Depending On**, **Filter** and select **P2_PROJECT_ID**.

   c. Review your changes. Click **Save and Run Page**.

      The Smart Filters page appears. Note that the Milestone suggestion chip no longer displays.

   d. Click **Project** on the Project suggestion chip and then select a project such as **Develop Production Partner Portal**

      Note the Milestone suggestion chip displays again.



8.7 Managing Faceted Search

A faceted search page displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). The report region displays search results as cards or a classic report.

- Example Faceted Search
  View an example faceted search page.
- About Creating Faceted Search Pages
  Learn about creating faceted search pages.
- Creating Faceted Search Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains a faceted search page.
- Creating Faceted Search Using the Create Page Wizard
  Run the Create Page Wizard to add a new faceted search page to an existing application.
- Creating a Faceted Search Manually
  Create a faceted search page by manually adding the page and facet components.
- Editing a Faceted Search Page
  Learn about how attributes can change faceted search page behavior.

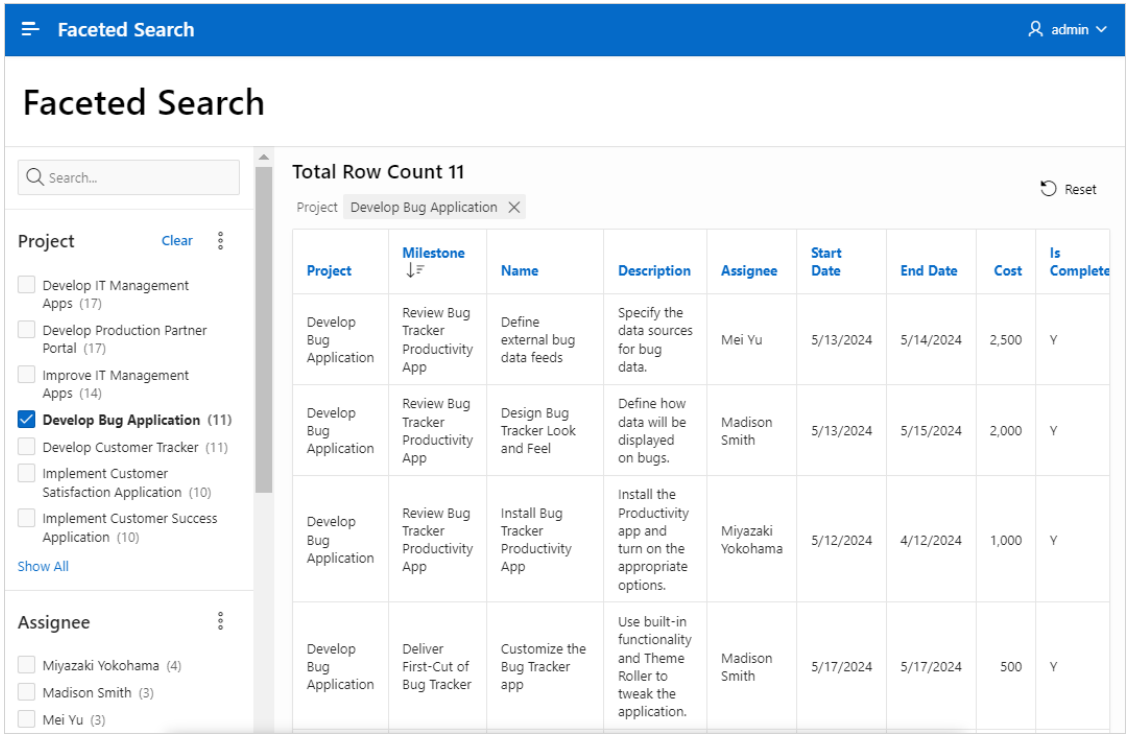
8.7.1 Example Faceted Search

View an example faceted search page.

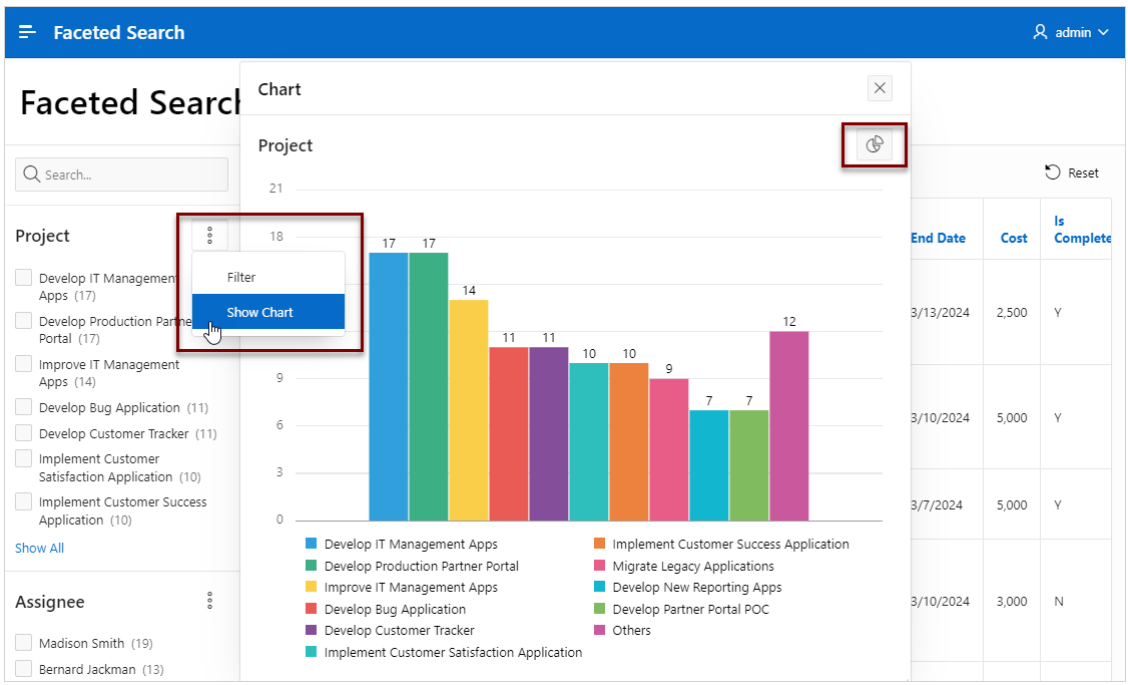
A faceted search page features a faceted Search region and report. The Search region displays on the left side of the page and enables users to narrow down the search result by selecting facet values. Narrowing the search result, makes it easier for users to find the data

they want. The right side of the page features a Search Results region which can display as a classic report or cards report.

The following shows a faceted search page built on the `EBA_PROJECT_TASKS` table. In this example the Project, **Develop Bug Application**, is selected and the report displays 11 rows.



By default, when you create a facet that computes counts with a wizard, each facet that computes counts includes a chart based on the facet counts. Clicking the **three dots** context menu on the facet and then **Show Chart** displays a chart based on the facet counts in a popup dialog. You can toggle between a bar chart or a pie chart by clicking the chart icon in the upper right of the popup window.



> **Tip:**
>
> To view an example, install the sample app, *Sample Reporting*. To learn more, see Installing Apps from the Gallery.

8.7.2 About Creating Faceted Search Pages

Learn about creating faceted search pages.

Running the Create Application Wizard or Create Page Wizard is the easiest way to create a faceted search page. However, you can create a faceted search page manually but the process involves multiple steps. To learn more, see Creating a Faceted Search Manually.

- About Creating Facet Search Pages with a Wizard
  Create a faceted search page by running either the Create Application Wizard or the Create Page Wizard.
- Search Facet Types
  Learn about supported facet types.

8.7.2.1 About Creating Facet Search Pages with a Wizard

Create a faceted search page by running either the Create Application Wizard or the Create Page Wizard.

Both the Create Application Wizard and Create Page Wizard support the creation of a faceted search page which contains a left Search region and a report. You choose if the report is a classic report or a cards report.

About Auto-Discovered Facets

When you build a faceted search page with either the Create Application Wizard and Create Page Wizard, you can select a table or view on which to build the page. If you create the page based on a table, the facets are auto-discovered using the Data Dictionary Cache. In contrast, if you build the page on a view or a SQL query, the facets are not auto-discovered. Instead, the wizard creates a facet region which only searches `VARCHAR2` columns.

About the Data Dictionary Cache

The Data Dictionary Cache report displays statistics and a data analysis of schema tables and caches in a workspace. Both the Create Application Wizard and Create Page Wizard use the Data Dictionary Cache when creating new applications and pages. You can preview how the wizards will render facets on the Data Dictionary Cache page by viewing the Table Column Cache report. The wizard uses the first top five facet search score columns with score greater than or equal to 20.

If a table changes due to DDL or DML modifications, the Oracle APEX nightly job `ORACLE_APEX_DICTIONARY_CACHE` refreshes the Data Dictionary Cache. However, you can refresh the cache manually by navigating to the Data Dictionary Cache and clicking either **Gather Statistics and Refresh Cache** or **Refresh Cache Only** at the workspace or table-level.

> **See Also:**
>
> Accessing the Data Dictionary Cache Report from Workspace Administration or Accessing the Data Dictionary Cache Report from Administration Services in *Oracle APEX Administration Guide*

8.7.2.2 Search Facet Types

Learn about supported facet types.

Search facets enable users to refine and narrow results by selecting multiple static filters simultaneously.

You specify a search facet type in Page Designer, by selecting the facet in the Rendering tab and configuring the **Identification**, **Type** attribute in the Property Editor. Each facet has additional attributes that control behavior. To learn more about a specific attribute, select the attribute and click the **Help** tab in the central pane.

> **Tip:**
>
> To see an example of manually adding a Search facet, Checkbox Group facets, and a Range facets, see Creating a Faceted Search Manually.

App Builder supports the following search facet Types:

- **Checkbox Group** - Displays multiple values as check boxes, enabling the end user to select multiple values. A list of values is required for items displayed as check boxes. The values corresponding to the checked boxes are stored in a single colon-delimited string.
- **Input Field** - Displays a text field that enables the user to filter by text input. The comparison operator is either set by the developer or can be chosen by the end user. Includes support for filtering by date, timestamp (with local time zone). For DATE or TIMESTAMP values, use the `YYYYMMDDHH24MISS` canonical format.
- **Radio Group** - Displays multiple values as radio group options, enabling the end user to select a single value.
- **Range** - Displays an item with a built-in list of values selector. Each option represents a range of values, from a lower to an upper boundary. The facet supports single or multiple selection, and manual entry.

  For filtering and count computation, the lower boundary of a specified range is always treated inclusive, and the upper boundary is exclusive. Example LOV Syntax:

  - Less than 100:

    ```
    |100
    ```

  - Between 100 and 500 (exclusive):

    ```
    100|500
    ```

  - 500 and more:

    ```
    500|
    ```

For a range with a Manual Input, filter by date, timestamp (with local time zone). For DATE or TIMESTAMP values, use the `YYYYMMDDHH24MISS` canonical format.

- **Search** - Displays a text field, enabling the end user to search the list entries of the facets in the Faceted Search region. A Faceted Search region supports one Search facet, which will always be at the top of the list of Facets, regardless of its sequence.
- **Select List** - Displays an item with a built-in list of values selector. When the end user clicks the item, the list of supported values displays directly inline with the current item.

  Select List is best suited for relatively small, discrete lists. End-Users can very quickly select a value from the list without needing to change focus to a popup dialog. For large lists *Popup LOV* is often better suited.

8.7.3 Creating Faceted Search Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains a faceted search page.

A faceted search page can include a classic report or a cards report. If you select classic report, you can also choose to include an update form.

> **Tip:**
>
> If you are creating a faceted search page that displays as a report, clicking **Include Form** automatically creates a form page for creating or updating records. This option is not available in the Create Page Wizard.

To create an application that includes a faceted search page:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Faceted Search**.
8. In the Add Faceted Search Page dialog:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Select a report type: **Report** or **Cards**.
   d. Table - Select a table.
   e. What appears next depends upon whether you selected **Report** or **Card**.
      - If **Report** - Select **Include Form** to include a form page for creating or updating records.

- If **Cards** - Select the display: **Grid**, **Float**, and **Horizontal (Row)**.

  Select the information to display on each card: **Title Column**, **Body Column**, **Icon Initials Column**, and **Badge Column**.

f. Advanced - Expand **Advanced** to configure these options:

- Set as Home Page - Enable this to make this page the home page for the application.
- Page Help - Enter text to be displayed when the user selects Page Help.

g. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

8.7.4 Creating Faceted Search Using the Create Page Wizard

Run the Create Page Wizard to add a new faceted search page to an existing application.

The Create Page Wizard offers a great deal of flexibility. For all Data Sources (Local Database, REST Enabled SQL Service, or REST Data Source) and Source Types (Table/View or SQL Query), you can select the facet columns you want to include. If you choose a local table, the facets are auto-discovered and the wizard pre-selects facets columns based on Data Dictionary Cache.

To create a faceted search page using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Faceted Search**.
5. Under Page Definition:

   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

6. Data Source - Select a Data Source for the page:

   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.

- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table /View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.
8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Display As - Select **Report** or **Cards**.
11. Select the facets to include in the report.

    Facet recommendations are based on statistics of data in your table. Click **Refresh** to refresh the column statistics to provide better recommendations.
12. Click **Create Page**.

> **See Also:**
>
> - Understanding Page Types in the Create Page Wizard
> - Managing Application Data

8.7.5 Creating a Faceted Search Manually

Create a faceted search page by manually adding the page and facet components.

Using the Create Application Wizard or Create Page Wizard is the easiest way to create a faceted search page. However, you can create a faceted search page manually. At a minimum, creating a faceted search page manually involves the following steps:

- Create a classic report or cards region.
- Add a faceted search region.
- Add individual facets or facet groups to the search region.
- Add a reset button.

The following example creates a faceted search page which contains a classic report built on the `EMP` table. The `EMP` table is available in the sample dataset, EMP / DEPT.

To create a faceted search manually.

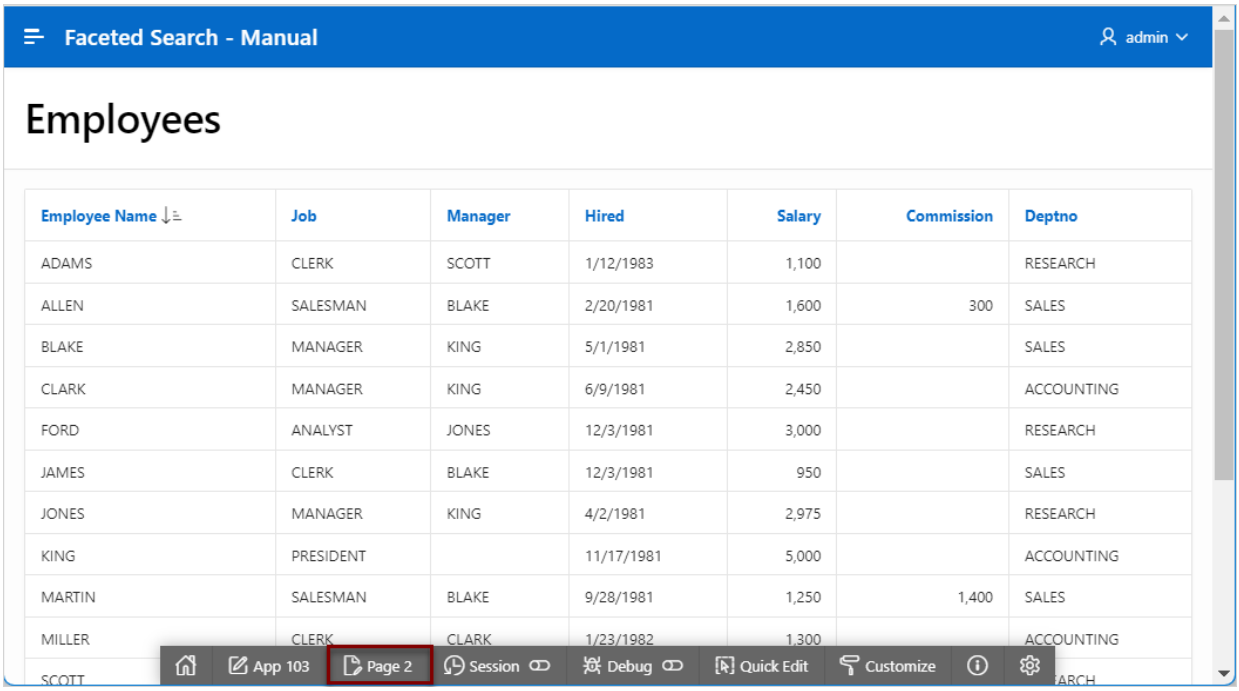
1. Run the Create Application Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Click **Create**.

      The wizard prompts you to select a method of creating an application.
   c. Click **Use Create App Wizard**.
2. On Create an Application:
   a. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
   b. Name - Enter a name to identify the application. For example, enter `Faceted Search - Manual`
   c. Appearance - Accept the default Theme Style and menu layout (Vita, Side Menu). To change it, click the **Set Appearance** icon adjacent to the Appearance field.
   d. Under Pages, click **Add Page**. Expand **Additional Pages** at the bottom of the window and select **Classic Report**.
   e. On Add Report Page:
      - Page Name - Enter a name for this page. For example, enter `Employees`
      - Set Icon - Accept the default, or select an icon to display in the navigation menu for this page.
      - Table or View - Select a table or view on which to build the page. Select **EMP**.
      - Click **Add Page**.
   f. Click **Create Application**.

      The Application home page appears.
3. View the new page.
   a. On the Application home page, select the new page, **2 - Employees**.

      The page displays in Page Designer.
   b. Click **Save and Run Page** on the right side of the Page Designer toolbar.
   c. When prompted, enter your workspace username and password and click **Sign In**.

      The Employees report appears.



d. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar where *X* is the page number.

4. Add a search region:

   a. In the Rendering tab (or left pane), right-click **Components** and select **Create Region**.

   b. Select the new region. Edit the following Region attributes in the Property Editor (or right pane):

      - Identification, Name - Enter `Search`
      - Identification, Type - Select **Faceted Search**.
      - Source, Filtered Region - Select **Employees**.

   c. Update the region appearance:

      - Appearance, Template Options - Click **Use Template Defaults, Scroll - Default**.

        The Template Options dialog appears.
      - Common, General - Select **Remove Body Padding**.
      - Common, Header - Select **Hidden but accessible**.
      - Click **OK**.

5. Update the facet for searching multiple columns:

   a. Select `PX_SEARCH` where *X* represents the page number.

   b. Source, Database Column(s) - Enter `ENAME,JOB,MGR,DEPTNO`

   c. Save your changes. Click **Save**.

6. Add a facet to filter the `DEPTNO` column:

   a. In the Rendering tab, right-click **Facets** and select **Create Facet**.

   b. Select the new facet and edit the following attributes in the Property Editor:

      - Identification, Name - Enter `PX_DEPTNO` where *X* represents the page number.

- Identification, Type - Accept the default, **Checkbox Group**.
- Label, Label - Enter `Department`
- List of Values, Type - Select **SQL Query**.
- List of Values, SQL Query - Enter `SELECT dname, deptno FROM dept`

c. Save your changes. Click **Save**.

7. Add a facet to filter the `MGR` column:

   a. In the Rendering tab, right-click **Facets** and select **Create Facet**.

   b. Select the new facet and edit the following attributes in the Property Editor:

   - Identification, Name - Enter `PX_MGR` where *X* represents the page number.
   - Identification, Type - Accept the default, **Checkbox Group**.
   - Source, Database Column - Verify that the column is **MGR**.
   - Label, Label - Enter `Manager`
   - List of Values, Type - Select **Shared Component**.

     The Create Application Wizard created a lookup list of values named `EMP.ENAME` so that the `MGR` column displays the manager name instead of the employee number.

   - List of Values, List of Values - Select **EMP.ENAME**.

   c. Save your changes. Click **Save**.

8. Add a facet to filter the `JOB` column:

   a. In the Rendering tab, right-click **Facets** and select **Create Facet**.

   b. Select the new facet and edit the following attributes in the Property Editor:

   - Identification, Name - Enter `PX_JOB` where *X* represents the page number.
   - Identification, Type - Accept the default **Checkbox Group**.
   - Source, Database Column - Verify that the column is **JOB**.
   - Label, Label - Enter `Job`
   - List of Values, Type - Select **Distinct Values**.

   c. Save your changes. Click **Save**.

9. Add a facet to filter the `SAL` column:

   a. In the Rendering tab, right-click **Facets** and select **Create Facet**.

   b. Select the new facet and edit the following attributes in the Property Editor:

   - Identification, Name - Enter `PX_SAL` where *X* represents the page number.
   - Identification, Type - Select **Range**.
   - Label, Label - Enter `Salary`
   - Source, Database Column - Verify that the column is **SAL**.
   - Source, Data Type - Select **NUMBER**.
   - Settings, Select Multiple - Enable **Select Multiple**.

   c. Define the facet as a static list of values:

   - List of Values, Type - Select **Static Values**.

- List of Values, Static Value - Click **Display1, Display2**.

  In the Static Values dialog, define the following:

  – Row 1:

    Display Value - Enter `< 1000`

    Return Value - Enter `|1000`

  – Row 2:

    Display Value - Enter `1000 to 3000`

    Return Value - Enter `1000|3000`

  – Row 3:

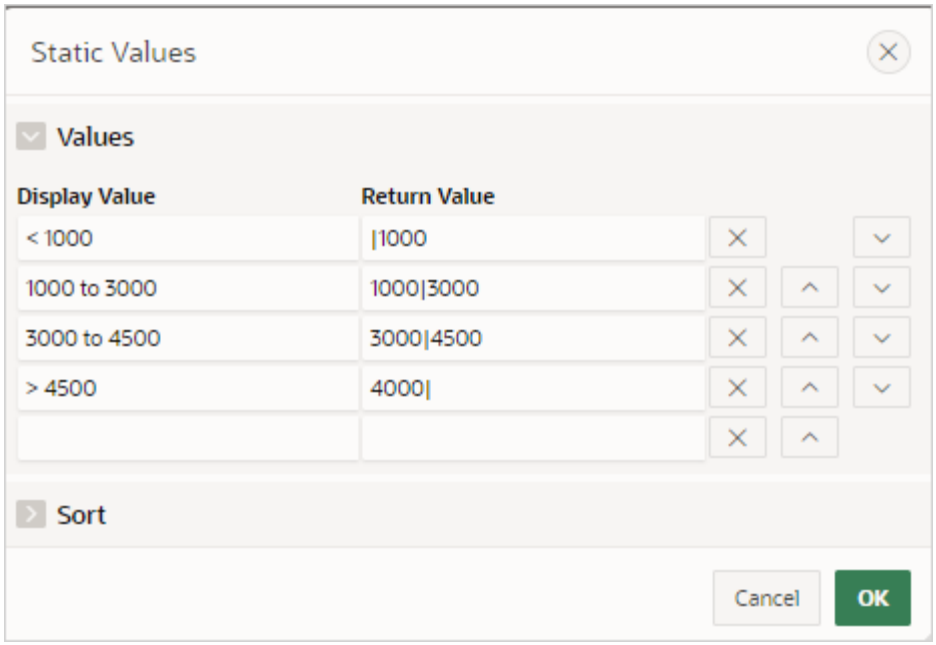
    Display Value - Enter `3000 to 4500`

    Return Value - Enter `3000|4500`

  – Row 4:

    Display Value - Enter `> 4500`

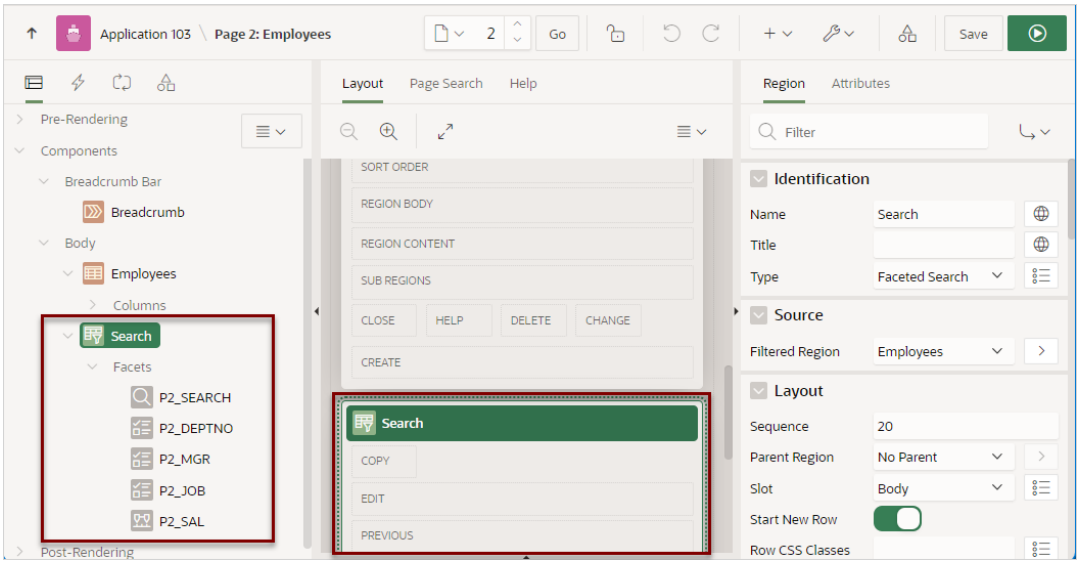
    Return Value - Enter `4000|`



- Click **OK** to save your changes.

d. Save your changes. Click **Save**.

Notice that the Search region displays below the Employees region.



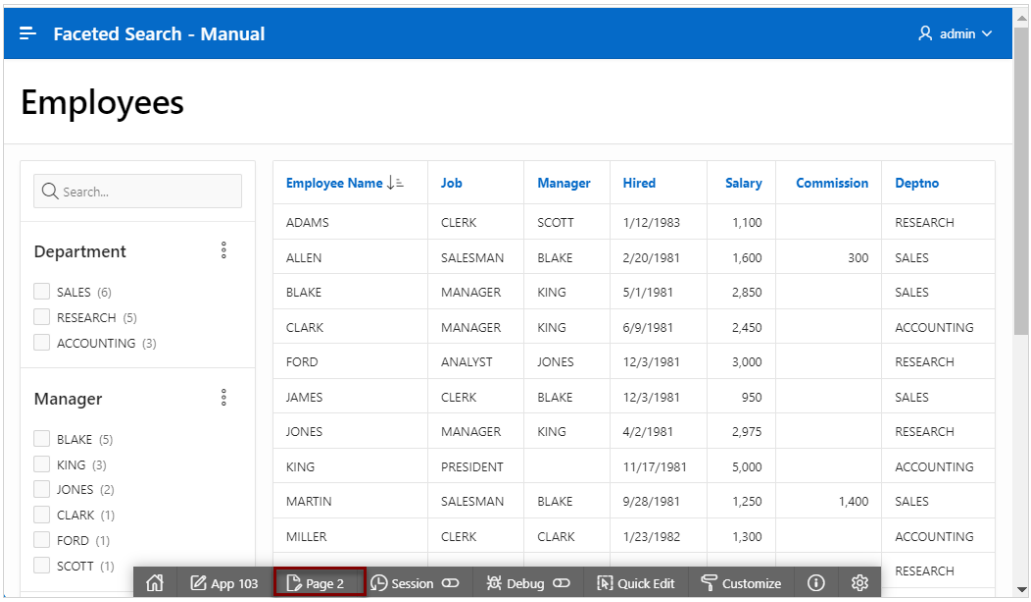
Next, move the Employees report so it displays adjacent to the Search region.

> **Tip:**
>
> The procedure that follows explains how to reposition a region by editing Region attributes. You can also move components by selecting them in the Layout tab (central pane) and dragging and dropping them to a new position.

10. Select the **Employees** region and update the following Region attributes:
    a. Layout, Sequence - Enter `20`
    b. Start New Row - Disable this option.

    In the Layout tab, notice the Search and Employees regions now display adjacent to one another.
11. Make the Search region narrower and then update region appearance. Select the **Search** region and update the following Region attributes:
    a. Layout, Column - Enter `1`
    b. Layout, Column Span - Select **3**.
    c. Save your changes. Click **Save**.
12. View the page.
    a. Click **Save and Run Page** in the upper right corner.
    b. If prompted, enter your workspace username and password and click **Sign In**.

    The revised page appears. Notice that the Search region contains the following facets: Search, Department, Manager, and Salary.



c. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar.

Next, add a button to refresh the page.

13. Add a region to contain a reset button:

    a. In the Rendering tab, right-click **Body** and select **Create Region**.

    b. Select the new region. In the Property Editor under **Region**, edit the following attributes:

    - Identification, Name - Enter `Button Bar`
    - Identification, Type - Accept the default **Static Content**.
    - Layout, Sequence - Enter `10`

    c. Update the Appearance attributes:

    - Appearance, Template - Select **Buttons Container**.
    - Appearance, Template Options - Click **Use Template Defaults**.

      The Template Options dialog appears.
    - Common, Body Padding - Select **No Padding**.
    - Common, Style - Select **Remove UI Decoration**.
    - Click **OK**.

14. Add a button:

    a. In the Rendering tab, right-click the Button Bar region and select **Create Button**.

    b. Edit the button. Select the button and edit the following attributes in the Property Editor:

    - Identification, Button Name - Enter `RESET`
    - Layout, Slot - Select **Next**.

    c. Update the Appearance attributes:

    - Appearance, Button Template - Select **Text with Icon**.
    - Appearance, Template Options - Click **Use Template Defaults, Left**.

In the Template Options dialog:

– Common, Style - Select **Remove UI Decoration**.

– Click **OK**.

- Appearance, Icon - Click **List of Values - Icon** and find and select **fa-undo**.

d. Update the Behavior attributes:

- Behavior, Action - Select **Redirect to Page in this Application**.
- Behavior, Target - Click **No Link Defined**.

  In the Link Builder - Target dialog:

  – Target, Page - Select the current page. For example, select **2**.

  – Clear/Reset, Clear Cache - Select the current page. For example, select **2**.

  – Clear/Reset, Action - Select **Reset Regions**.

  – Click **OK**.

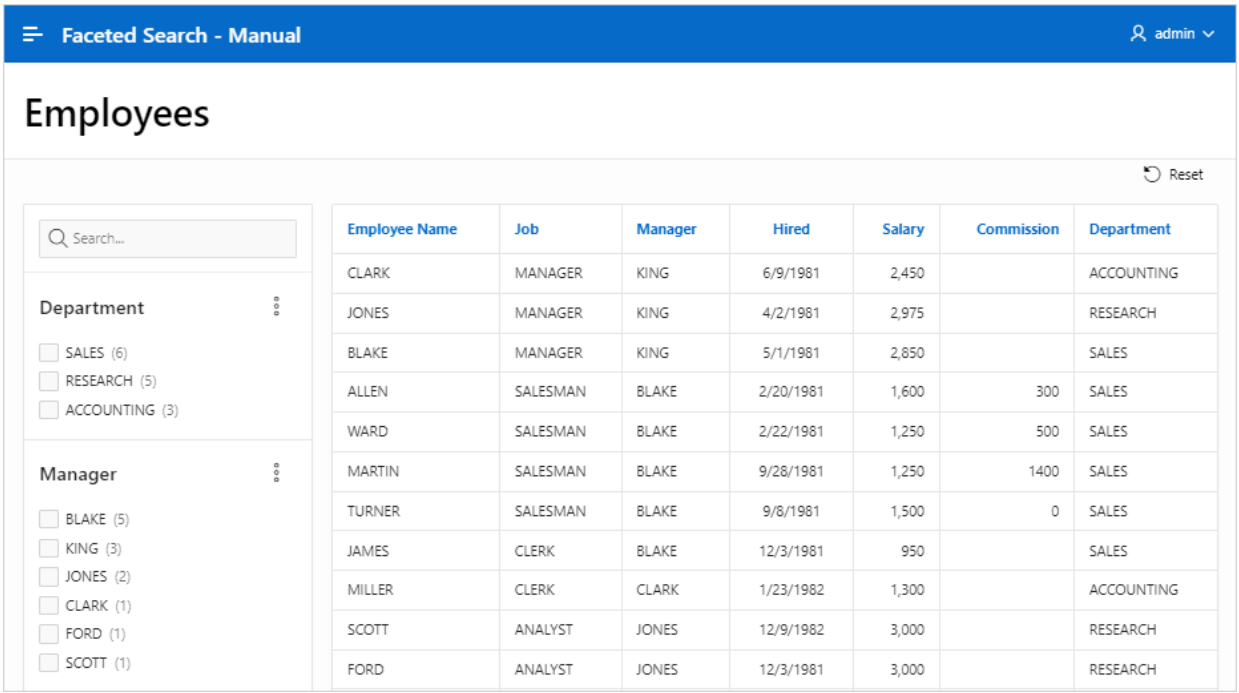
e. Save your changes. Click **Save**.

15. View the revised page.

a. Click **Save and Run Page**.

b. If prompted, enter your workspace username and password and click **Sign In**.

The faceted search page appears. Notice the new Reset button displays in the upper right above the report.



c. Test the page. Select facets and then click **Reset**.

8.7.6 Editing a Faceted Search Page

Learn about how attributes can change faceted search page behavior.

> **Tip:**
>
> Many examples in this section are built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data, or the `EMP` table available in the sample dataset, EMP / DEPT. To install a sample dataset, see Using Sample Datasets in *Oracle APEX SQL Workshop Guide*.

- Reviewing Faceted Search Page Components
  Review the components that comprise a faceted search page by viewing attributes in Page Designer.
- Understanding a Faceted Search Page
  Learn about the regions that comprise a faceted search page.
- Editing Faceted Search Region Attributes
  Edit region Attributes to change general search facet region behavior.
- Editing Facet Attributes
  Edit facet attributes to change facet behavior.
- Adding a New Facet
  Add new facets in Page Designer by right-clicking Facet and selecting Create Facet from the context menu.
- Adding a Facet on a Column Containing Multiple Values
  Add a facet to filter a column storing multiple values.
- Creating Facet Groups
  Create facet groups for data models having multiple flag columns.
- Configuring Single Facet Charts
  Configure region and facet attributes to declaratively control how and if single facet charts display in the facet search region.
- Editing the Static List of Values for a Range Facet
  Edit a range facet that displays as a static list of values.

8.7.6.1 Reviewing Faceted Search Page Components

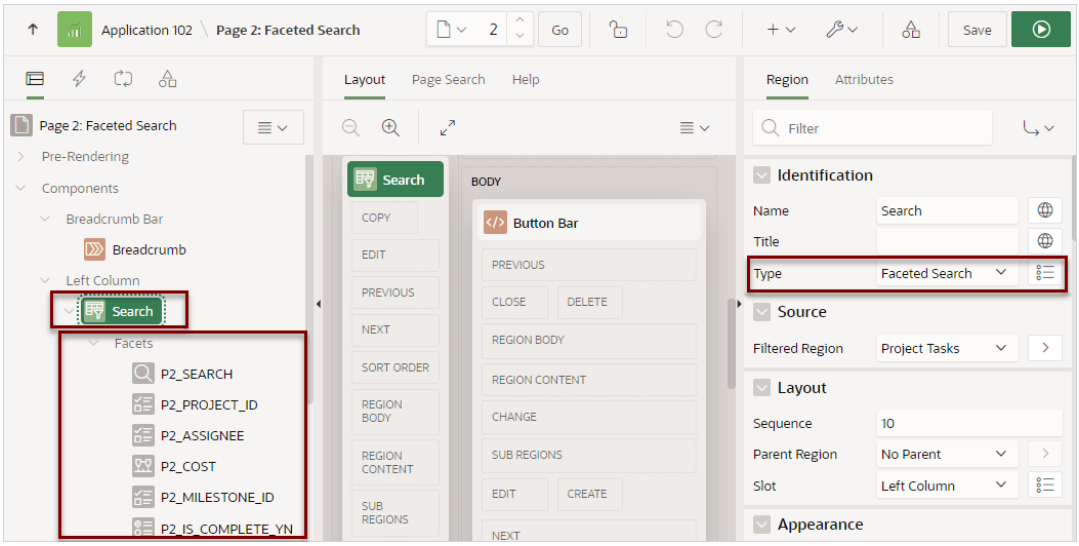
Review the components that comprise a faceted search page by viewing attributes in Page Designer.

A faceted search page features a faceted region that enables users to narrow down the search results and displays the results in either a report or cards view.

To review the components of a faceted search page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Review the Search region. In the Rendering tab under Left Column, locate and select the **Search** region.



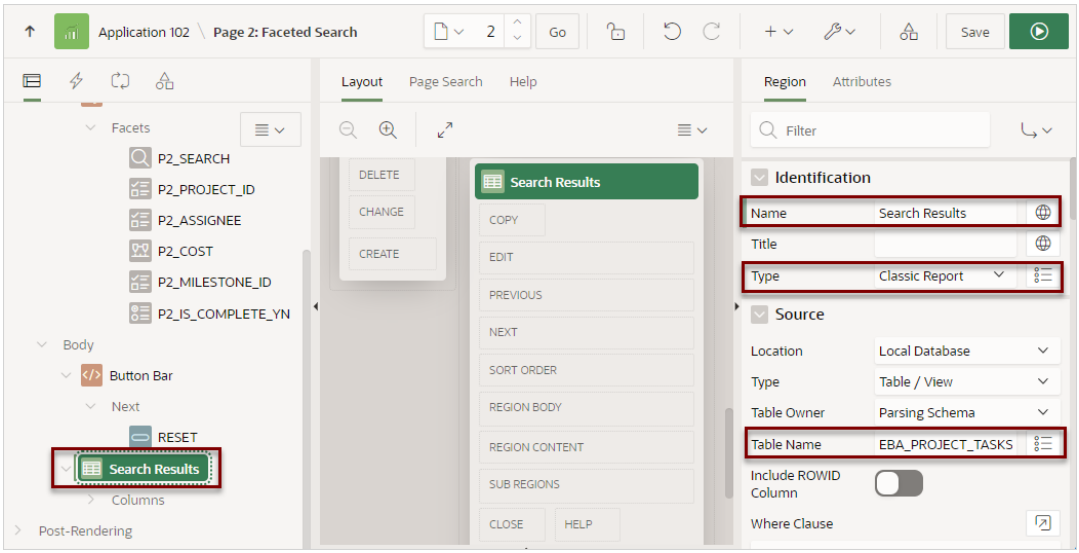
Note the Search region Type is **Faceted Search**. The search fields display under **Facets**. Facets are similar to items but feature the following unique characteristics:

- **Naming Conventions** - Facets use the same naming convention as items. By default, wizards prefix facet names with `P<page no>_<item name>` (for example, `P2_SEARCH`).
- **Filtering on More Than One Column** - Search facet type can filter based on more than one column and, when not external, is always at the top of the facets region.
- **Associated with a Column** - Facets are associated with a column from the report region.
- **Differences with Checkbox or Select List Facets** - The facet types such as **Checkbox** or **Select List** specify the UI for selecting or entering the facet values. They are not implemented the same as page items even if they have the same type name. For this reason, most dynamic actions do not apply to facets.

3. Select a facet and explore the configurable options.

   To view Help, select the attribute and click the **Help** tab.

4. Review the report region. In the Rendering tab, locate and select the report region (for example, **Search Results**).



In the previous example, note that the attributes indicate the report Type is a **Classic Report** built on the `EBA_PROJECT_TASKS` table in a local database.

> **See Also:**
>
> About Item Naming Conventions

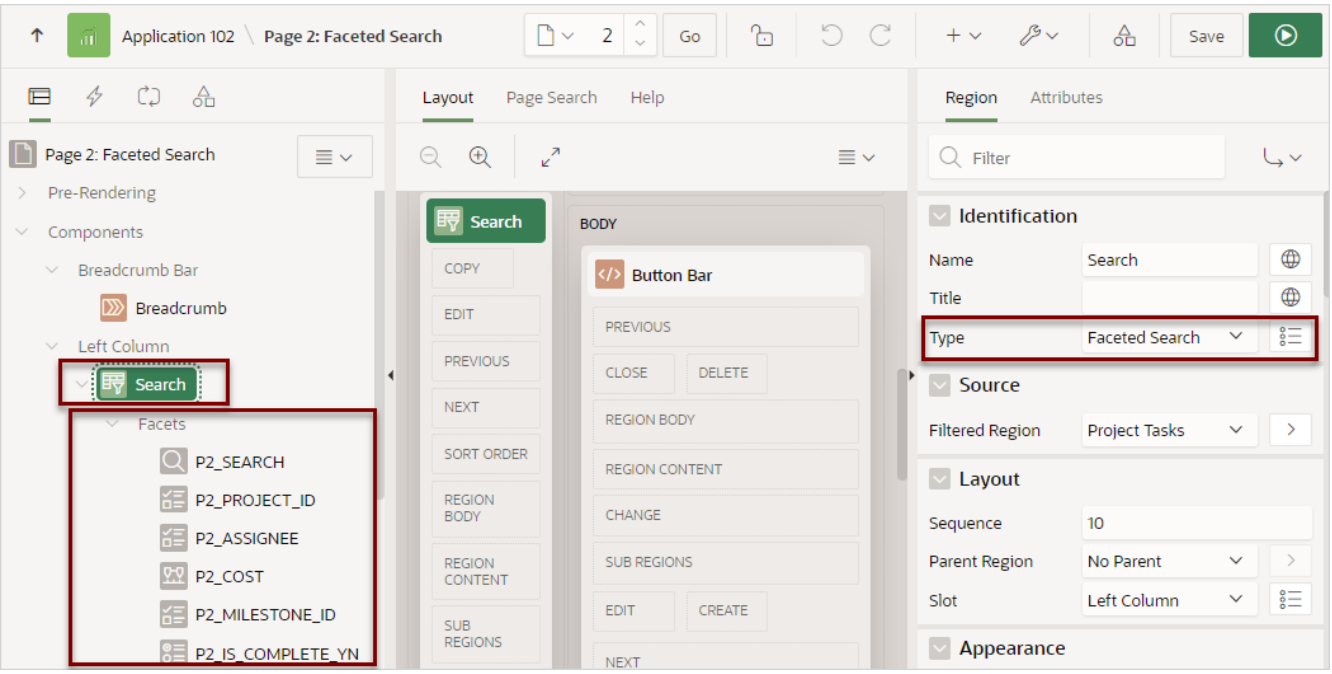
8.7.6.2 Understanding a Faceted Search Page

Learn about the regions that comprise a faceted search page.

About the Search Region

A faceted search page features a Search region which enables users to narrow down the search result and a Search Results region which displays as either a classic report or cards report.

The **Search** region displays in the left pane or Rendering tab. Selecting the Search region displays attributes in the Property Editor. In the following image, the Search region **Type** is defined as **Faceted Search**. The **Filtered Region** attribute identifies the Search Results region. In this example, **Filtered Region** is as a classic report named **Search Results**.



> **Tip:**
>
> Remember that each attribute in the Property Editor has associated Help. To view Help, select the attribute in the Property Editor and then click the **Help** tab in the central pane. Help describing the attribute appears in the central pane.

About Facets

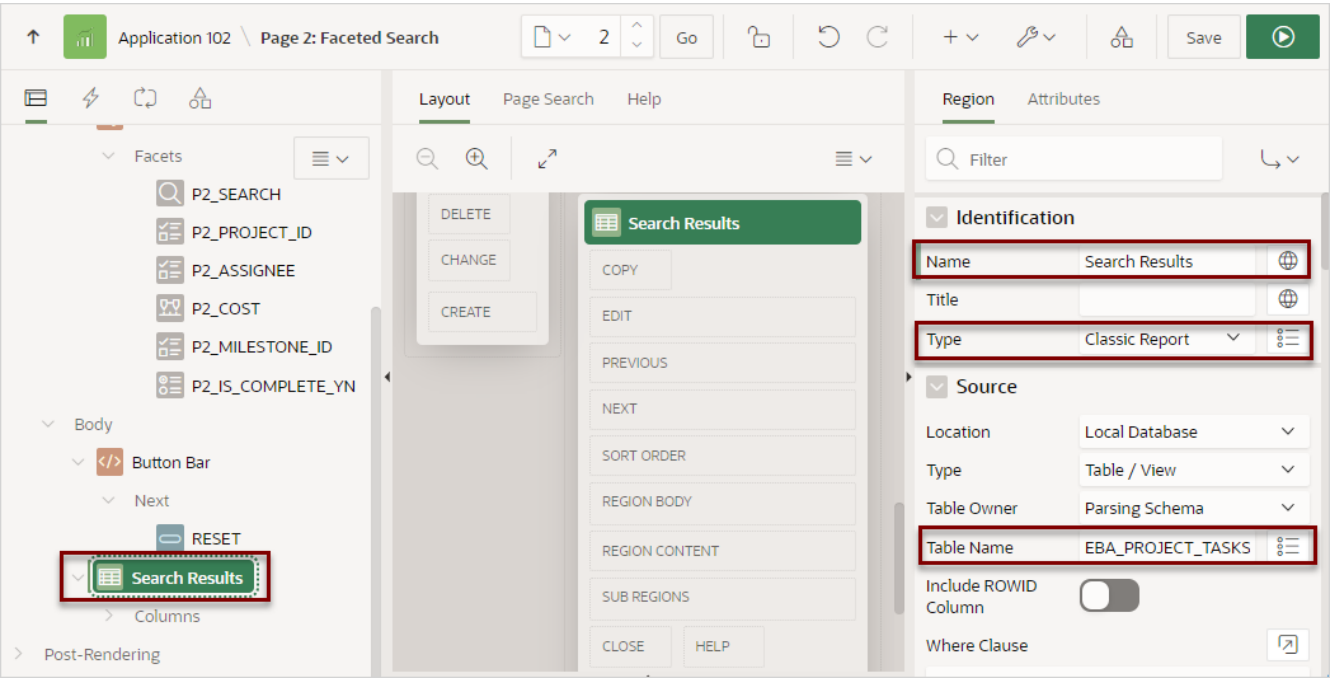
Search facets display under **Facets** in the Rendering tab. When you select a facet, the associated attributes display in the Property Editor. Facets use the same naming convention as items. By default, wizards prefix facet names using the following convention: `P<page no>_<item name>`. For example, in the previous example, the first facet is named `P2_SEARCH)`.

Facets are similar to items but feature the following unique characteristics:

- **Filtering on More Than One Column** - The facet of type Search, generally called **Search**, can filter based on more than one column and, when not external always displays at the top of the facets region.
- **Associated with a Column** - The facets are associated with a column from the report region.
- **Differences with Checkbox or Select List Facets** - The facet types, such as Checkbox or Select List, indicate how UI enables the user to select or filter data. Facets are not implemented the same as page items even if they have a similar naming convention. For this reason, most dynamic actions do not apply to facets.

About the Search Results

A faceted search page includes a Search Results region which displays as either a classic report or cards report. In the following example, Search Results is a **Classic Report** built on the `EBA_PROJECT_TASKS` table.



About Cards Page Search Results

The Search Results in a faceted search page can display as a cards report. A cards page functions as a colorful way to display a subset of information and then enable the user to link to more detail. To create a link from a cards page, you create an action and then select an action type. You can create an action to link from a full card, a title, a subtitle, media, or a button. To see an example, see Using Actions to Link from a Cards Page.

> **See Also:**
>
> About Item Naming Conventions

8.7.6.3 Editing Faceted Search Region Attributes

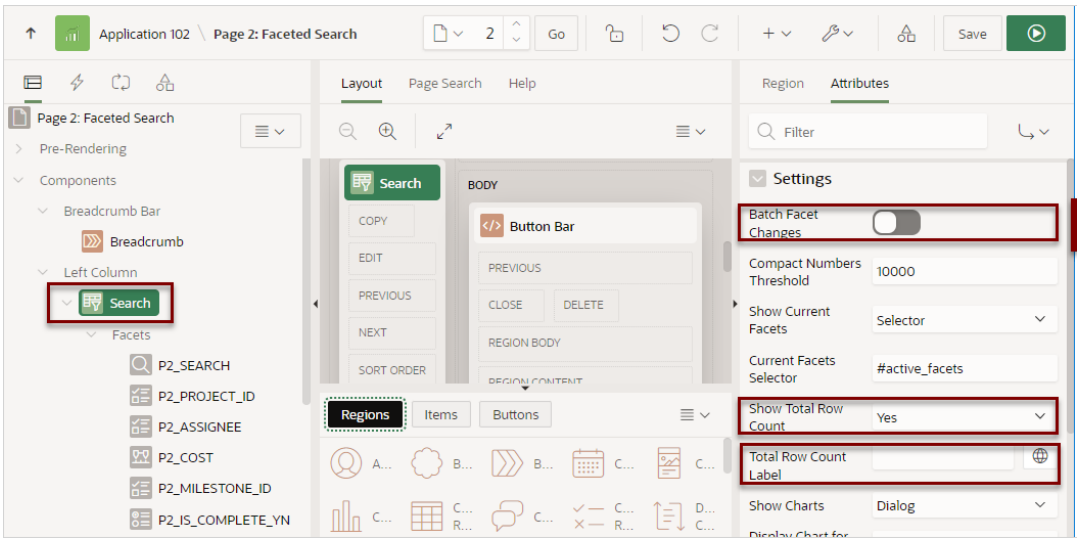
Edit region Attributes to change general search facet region behavior.

The following example demonstrates how to review and edit the region Attributes **Batch Facet Changes** and **Show Total Row Count**. **Batch Facet Changes** controls whether the report filtering executes as soon as a facet value changes, or individually by clicking an Apply button. **Show Total Row Count** determines if a row count value displays on the page. The following example built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.

To edit Search facet region Attributes:

1. View the faceted search page in Page Designer.
2. View the Search region Attributes:
   a. In the Rendering tab (or left pane), select the **Search** region.
   b. In the Property Editor, select the **Attributes** tab.



3. Enable **Batch Facet Changes**:

   Batch Facet Changes controls how report filtering executes. When enabled and a user selects a facet, an Apply button appears under the facet. The user must click the Apply button to apply the facet and filter the report. When disabled, the report filtering executes as soon as the facet value changes.

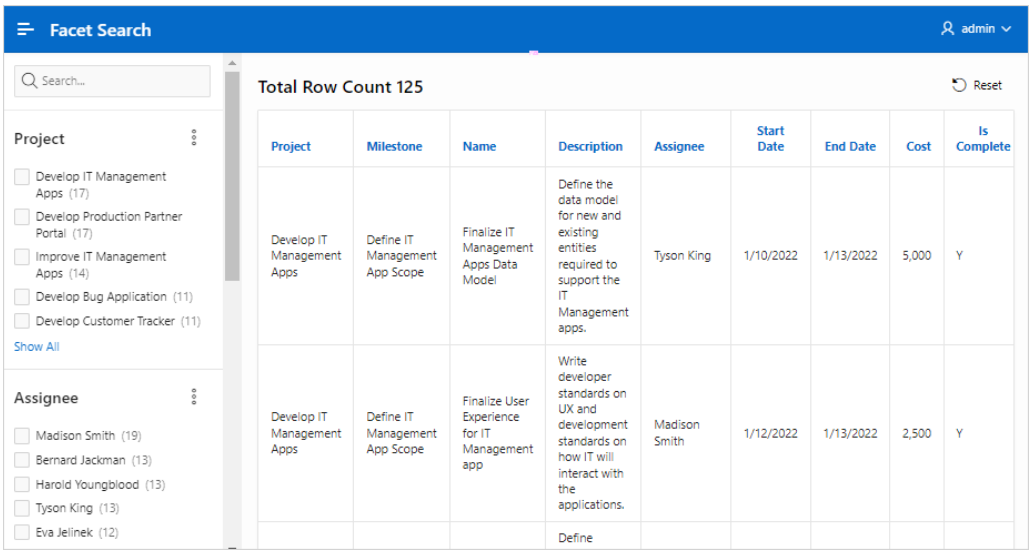
   a. In the Rendering tab (or left pane), select the **Search** region.

b. In the Property Editor, select the **Attributes** tab.

c. Under **Settings**, enable **Batch Facet Changes**.

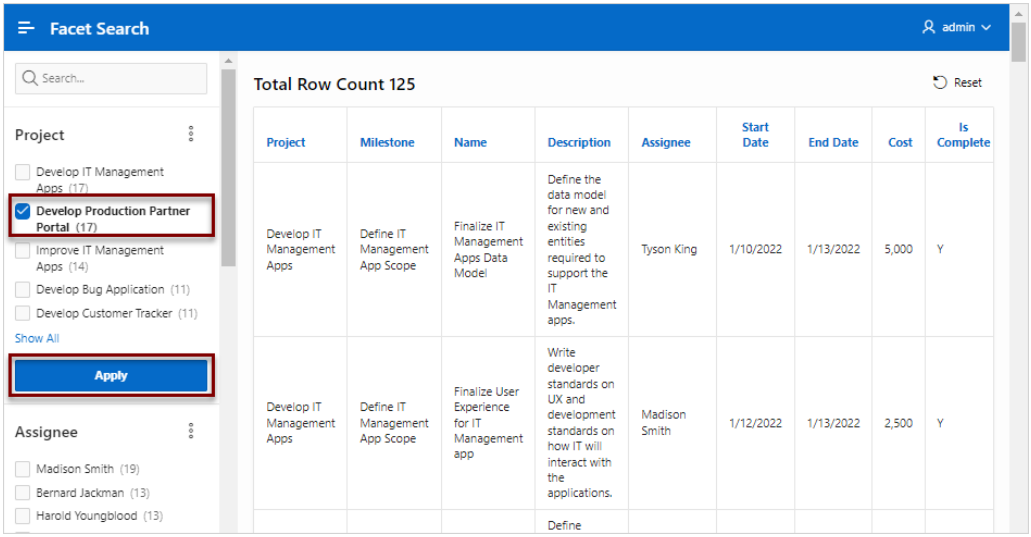
d. Click **Save and Run Page**.

A rendered version of the application appears.



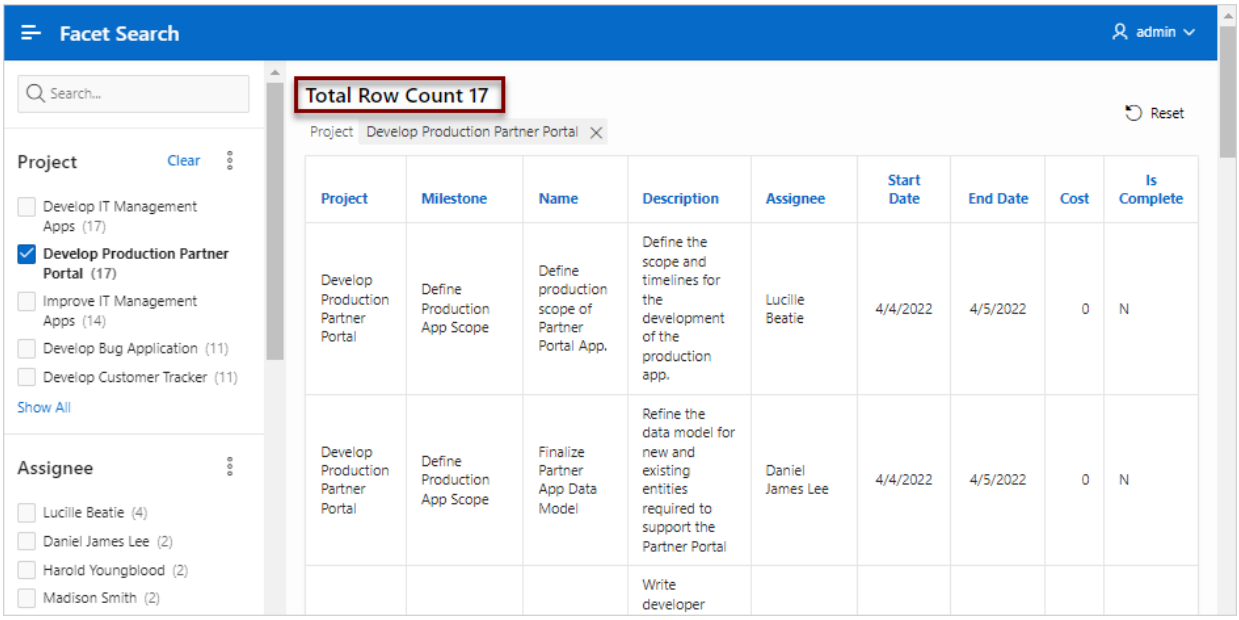
e. In the Search region, select a project, such as **Develop Production Partner Portal**.

An Apply button appears.



f. Click **Apply**.

A revised report appears and a filter displays above report.



g. Return to Page Designer by clicking **Page XX** on the Runtime Developer Toolbar.

4. Review the **Show Total Row Count** attribute.

   When set to Yes, the total row count displays above the report. In the prevous illustration, this displays as `Total Row Count 17`.

   a. In the Rendering tab (or left pane), select the **Search** region.

   b. In the Property Editor, select the **Attributes** tab.

   c. Find **Settings**, **Show Total Row Count**. Options include:

      - **No** - If **No**, the total row count does not display.
      - **Yes** - If **Yes**, the total row count displays at the top of the current facets as shown in the previous illustration.

        If Show Current Facets is a selector, the total row count displays whereever that selector is defined. The wizards by default create an element on the page called `active_facets` and that is above the report. If Show Current Facets is **No** and Show Total Row Count is **Yes**, the total row count displays at the top of the facets region.
      - **Selector** - If **Selector**, the total row count displays wherever the selector is defined. This option enables you to place the total row count anywhere on the page.

5. Click **Save** or **Save and Run Page**.

8.7.6.4 Editing Facet Attributes

Edit facet attributes to change facet behavior.

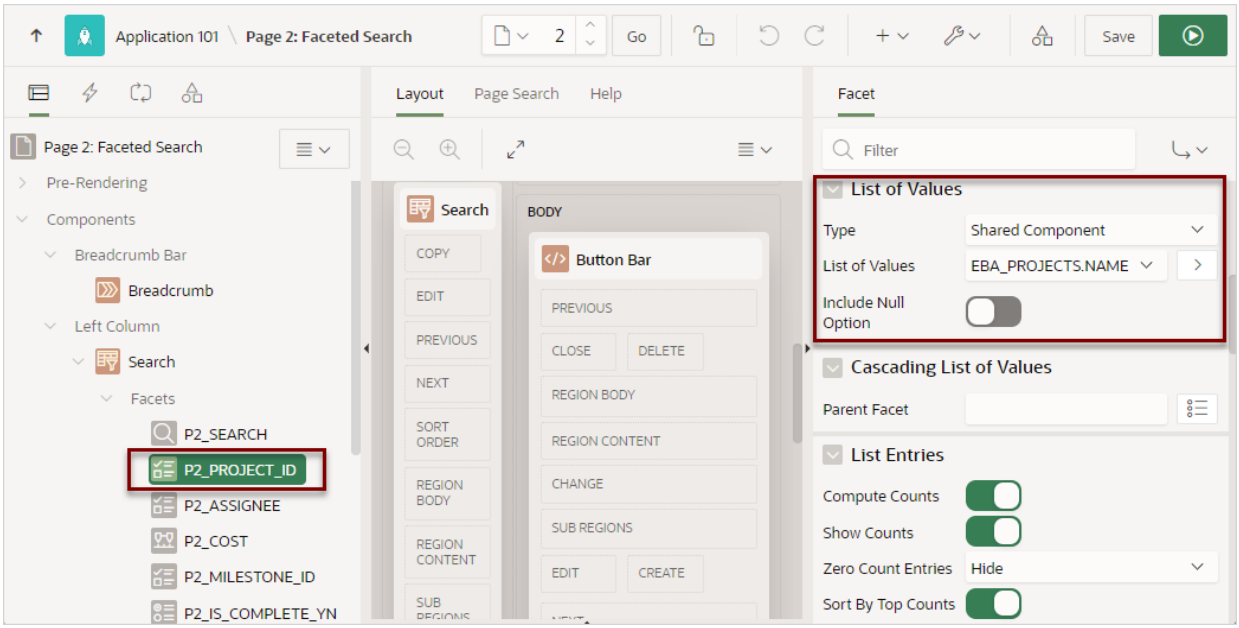
The following example demonstrates how to edit facet attributes to change facet behavior. Enabling **Display Filter Initially** adds a filter field which displays above the list entries. **Depending On** makes the display of one facet dependent upon another. Configuring **Appearance**, **Display** enables you to specify whether the facet displays inline or in a filter dialog. The following example is also built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.

To view and edit facet attributes:

1. View the faceted search page in Page Designer.
2. View the facet attributes for the facet, **P2_PROJECT_ID**:
   a. In the Rendering tab (or left pane) locate the **Search** region and select the facet, **P2_PROJECT_ID**.
   b. In the Property Editor (or right pane) find the **List of Values** group.

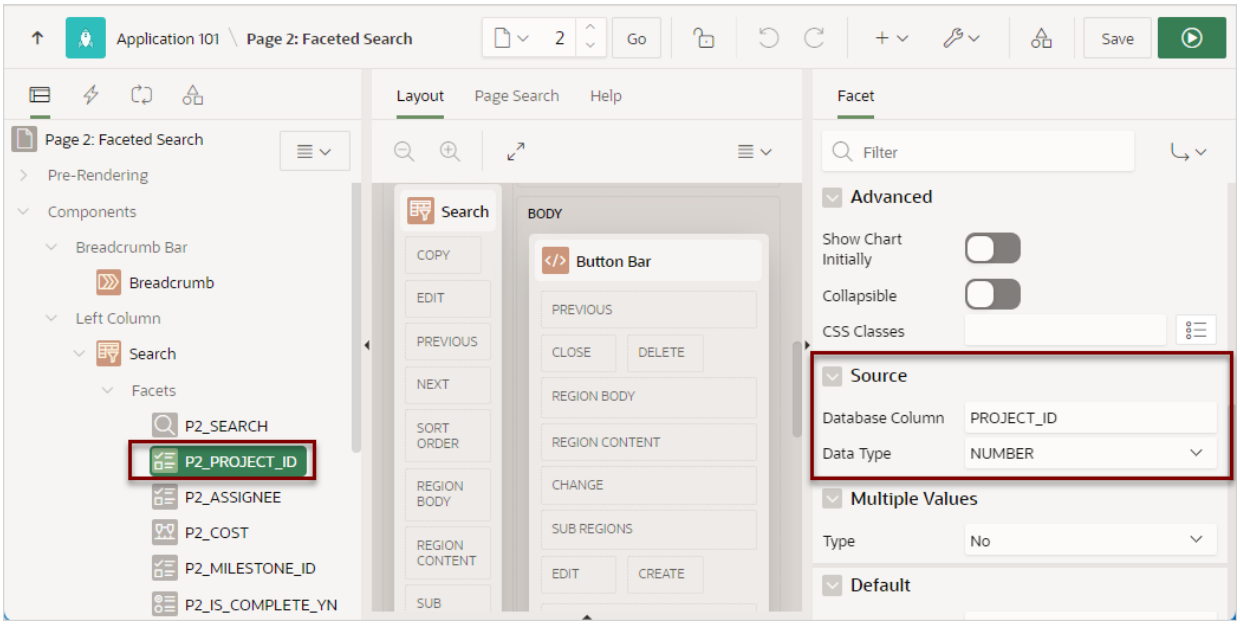
      The **List of Values** group indicates that the P2_PROJECT_ID facet is a List of Values created as a Shared Component.



> **Tip:**
>
> The maximum amount of items which can be returned by an LOV, or by Distinct Values, is 5,000. If an LOV returns more than 5,000 items, faceted search cuts the LOV silently at 5,000 values. When logged in as a developer, Oracle APEX raises a runtime error once more than 4,000 LOV items are returned. These errors indicate that the LOV returns too many values for a useful faceted search UI.

   c. In the Property Editor, find the **Source** group.

      The **Source** group indicates that the P2_PROJECT_ID facet is based on the `PROJECT_ID` column which has a data type of `Number`.

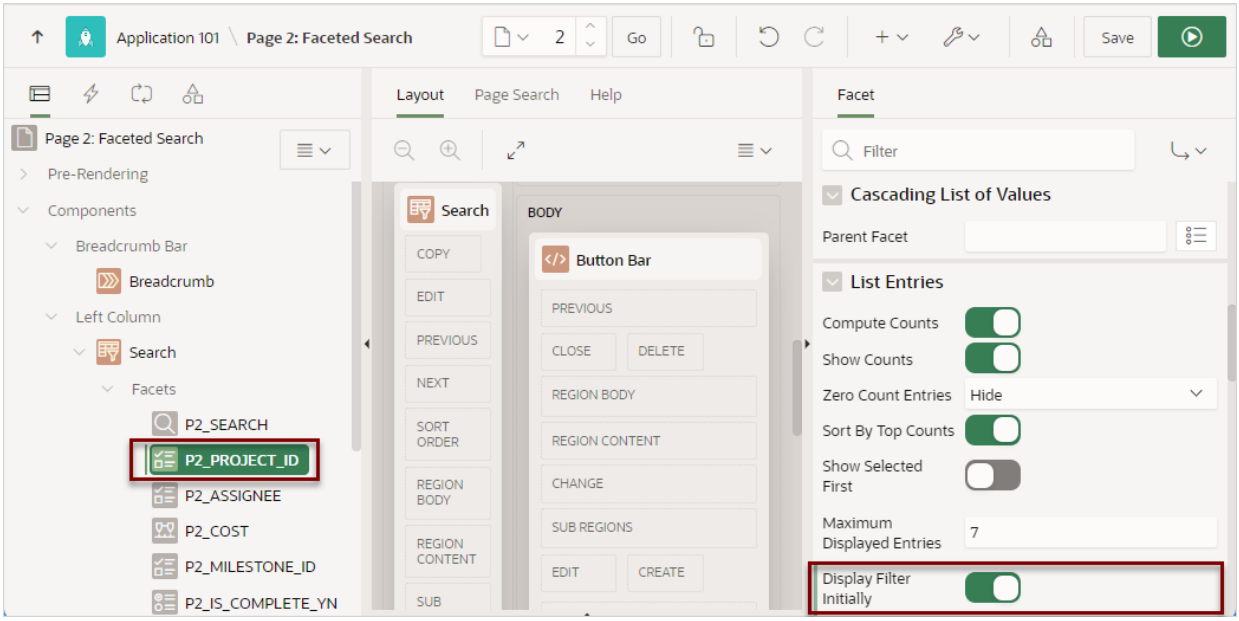


3. Update **P2_PROJECT_ID** and enable the **Display Filter Initially** attribute:

   a. In the Rendering tab, select the **P2_PROJECT_ID** facet (if not already selected).

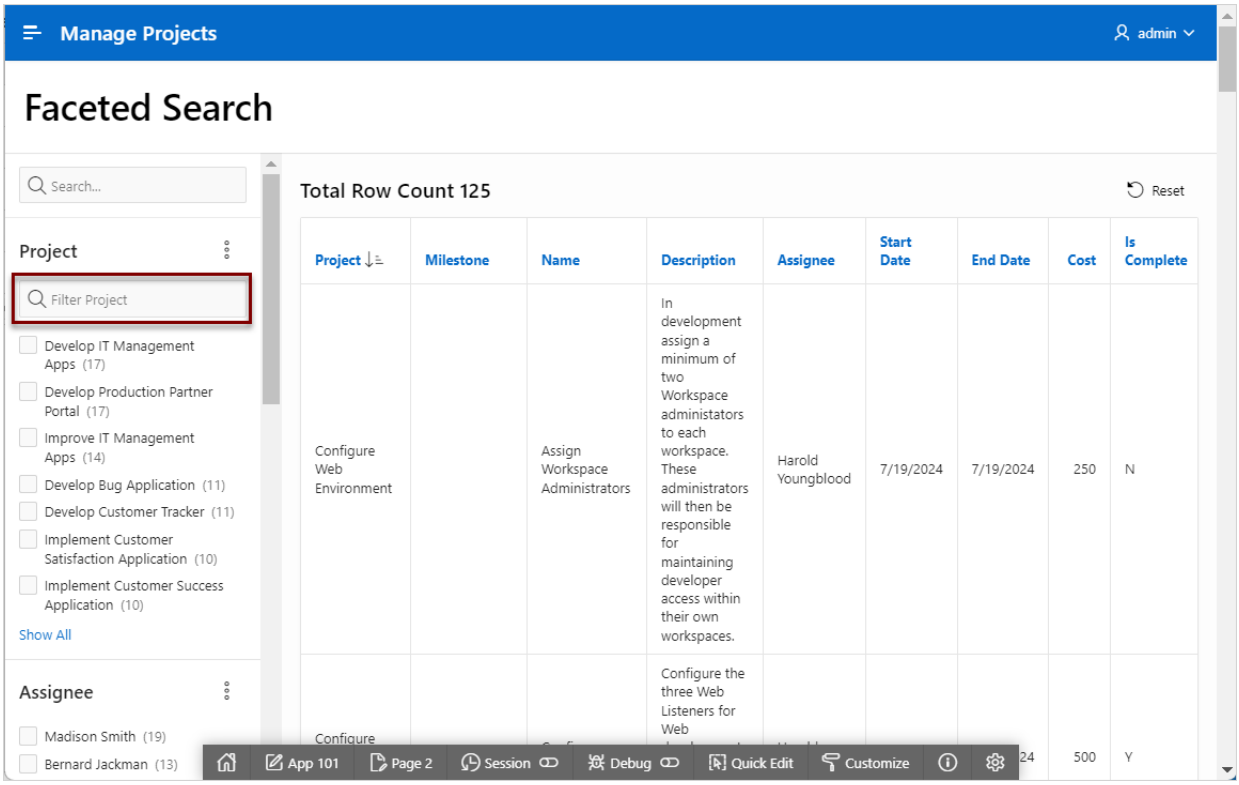
   b. In the Property Editor, find **List Entries** and enable **Display Filter Initially**.

   When **Display Filter Initially** is enabled, a filter field displays above the list entries. If the number of entries in the facet is less than the value on Maximum Displayed Entries, the filter field will not display.



   c. Review your changes. Click **Save and Run Page**.

   The faceted search page appears and a new **Filter Project** field appears.



d. Test the new Filter Project field. For example, entering the keyword "develop" will update the Project list to only include search facets containing that phrase.

e. Click **Page X** on the Runtime Developer Toolbar to return to Page Designer, where *X* is the page number.

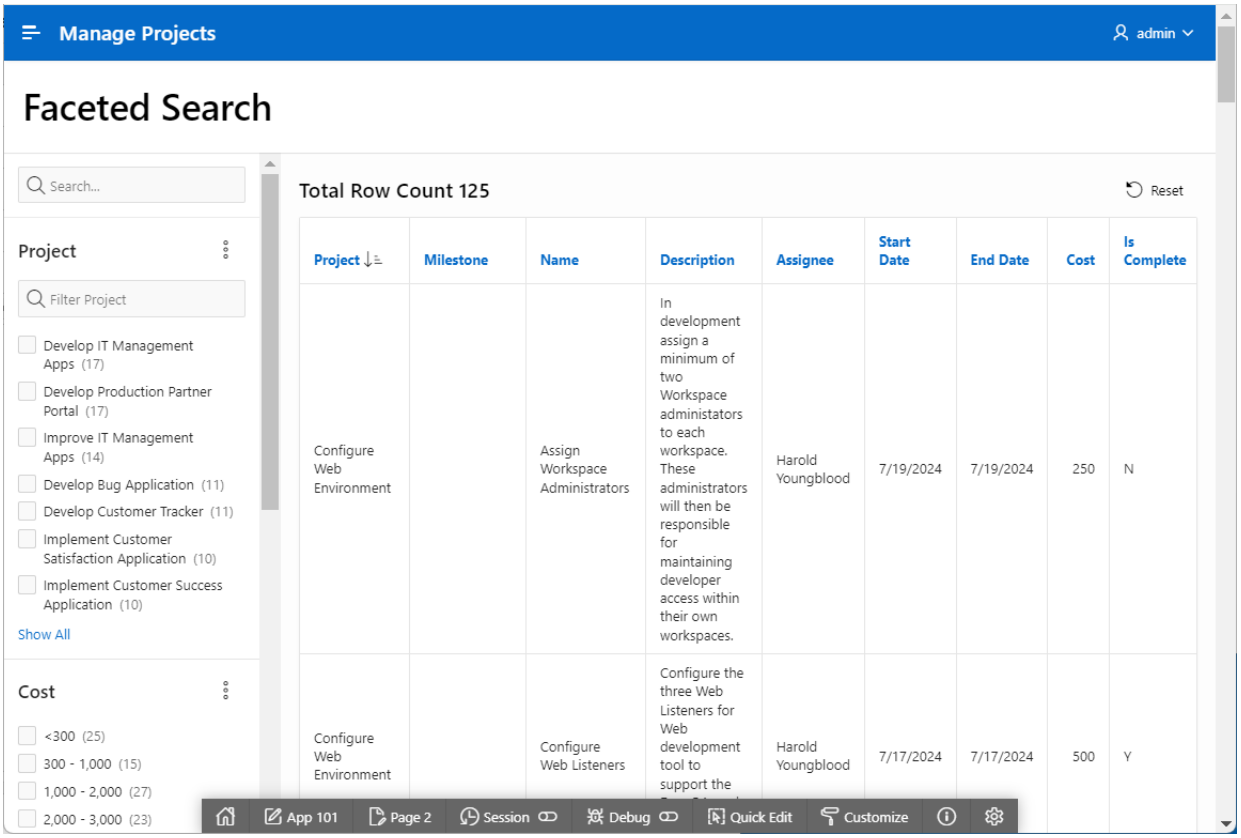
4. Only show Assignees for the selected Project by enabling the **Depending On** attribute.

   a. In the Rendering tab locate the **Search** region and select the **P2_ASSIGNEE** facet.

   b. In the Property Editor, find **Depending On**, **Facet** and select **P2_PROJECT_ID**.

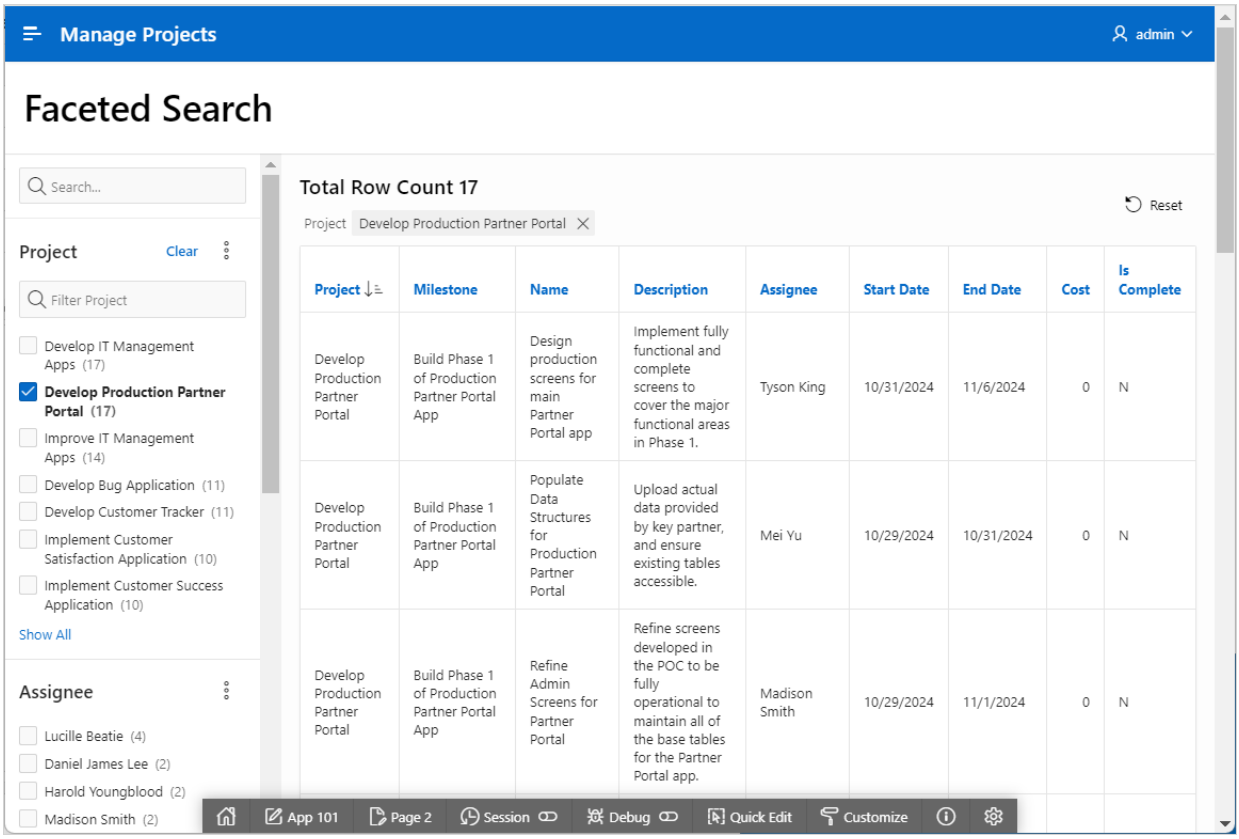
   c. Review your changes. Click **Save and Run Page**.

   The faceted search page appears. Note that the Assignee list no longer displays.



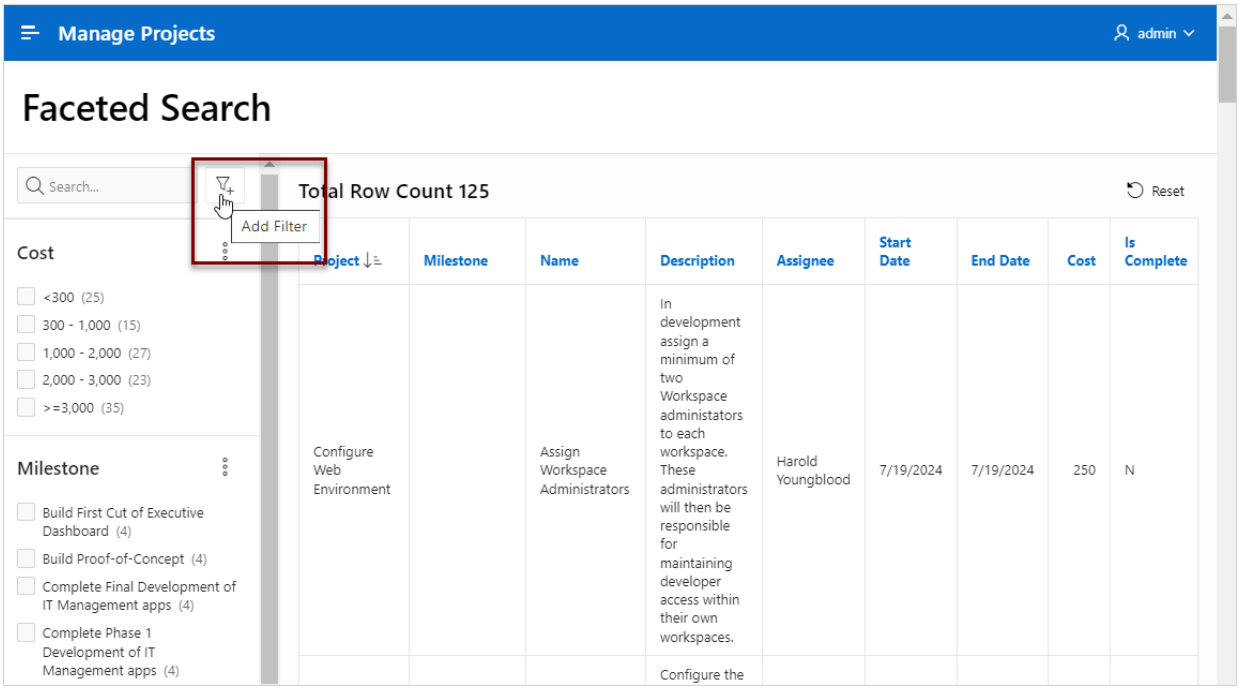
d. Select a project such as **Develop Production Partner Portal**.

Note that the Assignee list now appears but only displays assignees associated with the **Develop Production Partner Portal** project.



5. Update **P2_PROJECT_ID** so that Projects list displays in a dialog:
   a. In the Rendering tab, select the **P2_PROJECT_ID**.
   b. In the Property Editor, find **Appearance**, **Display** and select **Add Filter Dialog**.
   c. Review your changes. Click **Save and Run Page**.

      The faceted search page appears. Note that the Projects list no longer displays and a new **Add Filter** icon displays next to the Search field.



d. Click **Add Filter** icon to view the Projects list.

8.7.6.5 Adding a New Facet

Add new facets in Page Designer by right-clicking Facet and selecting Create Facet from the context menu.

Facets map to specific database columns. When creating a new facet, you can create the following types: Checkbox Group, Input Field, Radio Group, Range, Search field, or Select List.

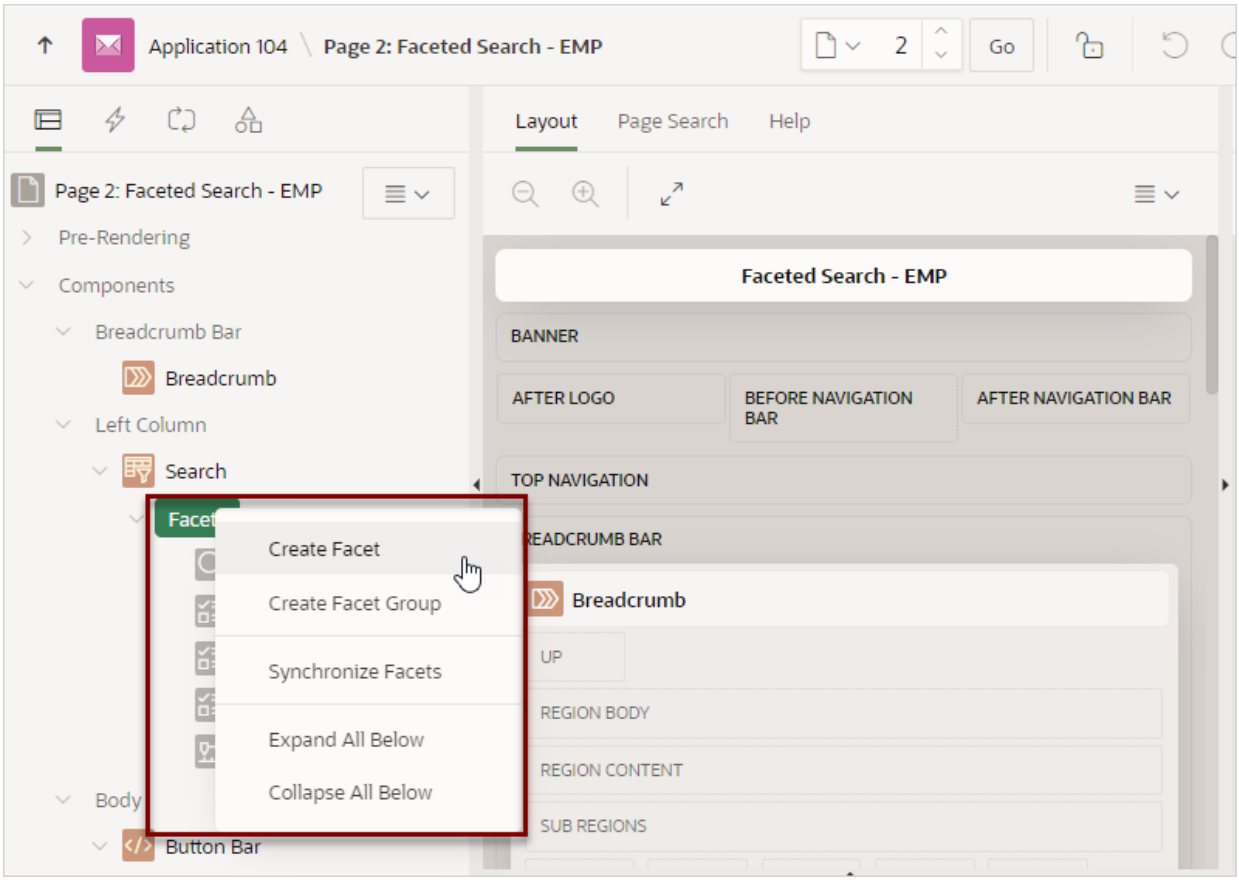
You can control the facet order in the UI by reordering the facets in the Rendering tab. Note there can be at most one Search type facet and if present it displays at the top of the faceted search region no matter where it resides in the Rendering tab.

> **Tip:**
>
> To quickly synchronize facets for existing regions, right-click **Facets** in the Rendering tab, and select **Synchronize Facets**.

The following example demonstrates how to create Input Field facet to filter employee salary that is less than or equal to the provided dollar amount. In this example, the Input Field facet maps to the `SAL` column in the `EMP` table available in the sample dataset, EMP / DEPT.

To create an Input Field facet:

1. View the faceted search page in Page Designer.
2. Create the new facet:
   a. In the Rendering tab (or left pane), find the **Search** region.
   b. Right-click **Facets** and select **Create Facet** from the context menu.



A new facet appears in the Rendering tab. In this example, the facet is named **P2_NEW**.

3. In the Rendering tab, select the new facet and then update the following attributes in the Property Editor (right pane):
   a. Identification, Type - Specify how this item is rendered. Select **Input Field**.
   b. Label, Label - Enter the label for the facet. The label displays on the page only if the facet displays. Enter:

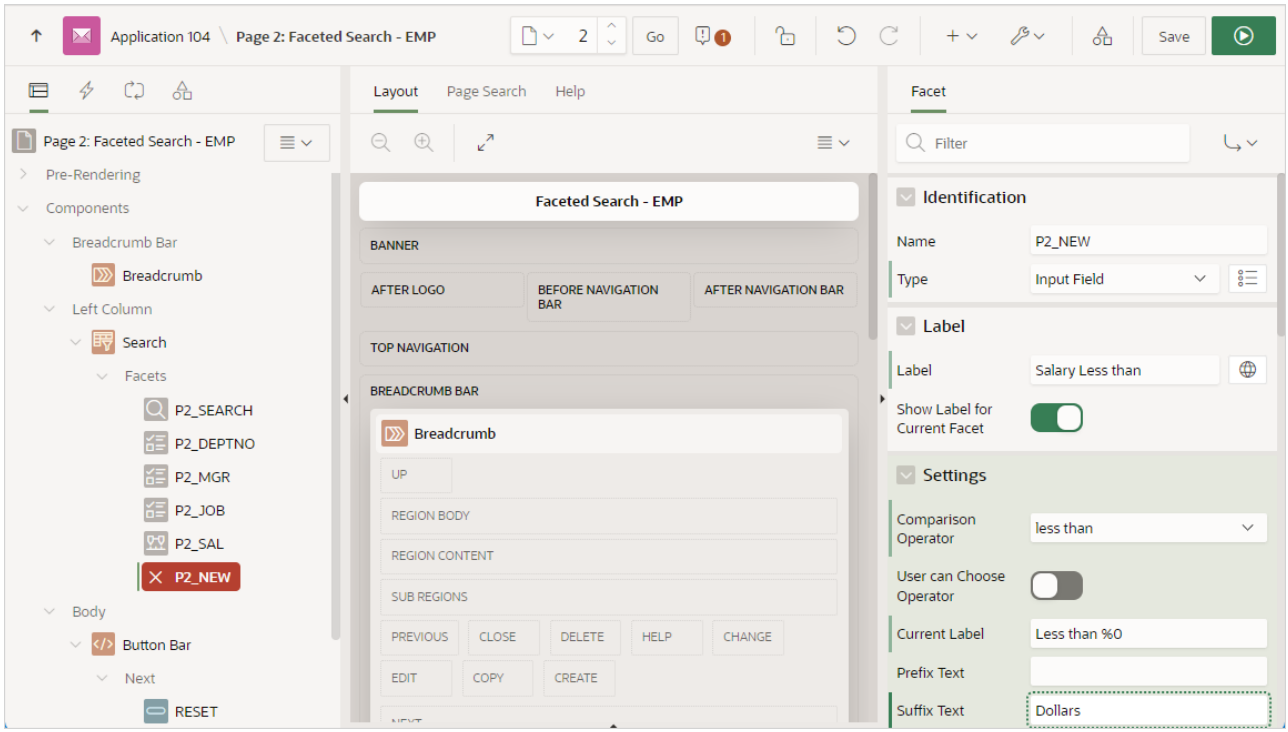
      ```
      Salary Less than
      ```

   c. Settings, Comparison Operator - Select a comparison operation. Select **Less than**.
   d. Settings, Current Label - Provide the text to be displayed for this facet in the current facets area. The current value must be referenced with the `%0` placeholder. Enter:

      ```
      Less than %0
      ```

   e. Settings, Suffix Text - Provide text to be displayed after the input field. Enter:

      ```
      Dollars
      ```

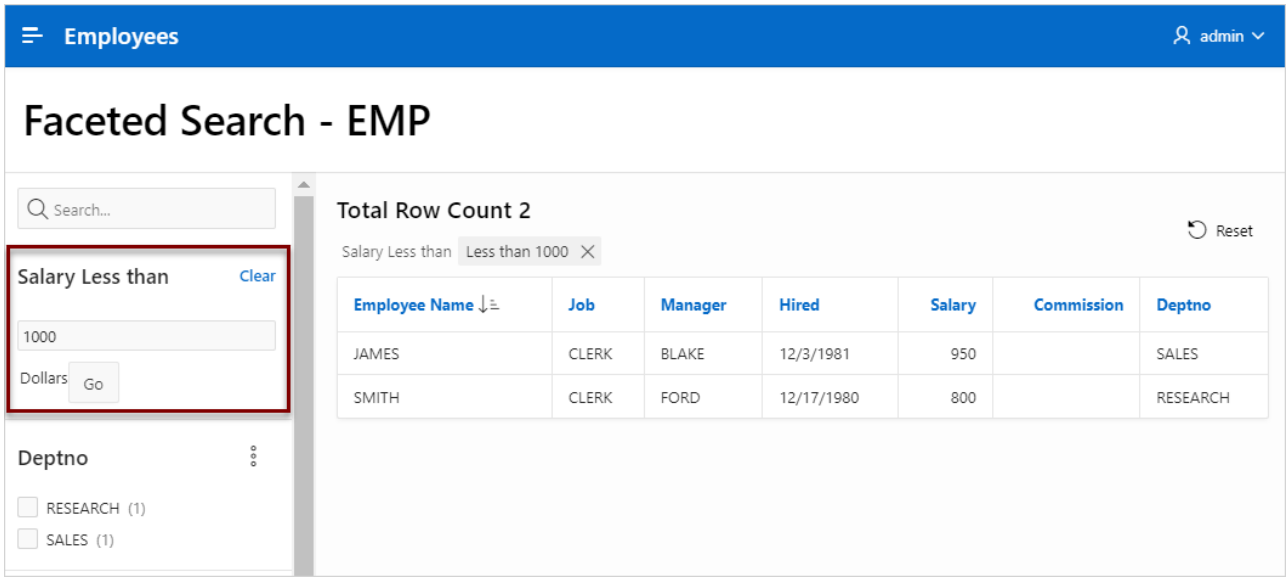


4. Map the facet to the appropriate database column.
   - Source, Database Column - Enter the case sensitive database column name used as the source for this facet. Since this facet maps to the `SAL` column, enter: `SAL`
   - Source, Data Type - Select `NUMBER`.
5. Change the facet position so it displays after the **Search** facet.
   - Layout, Sequence - Enter: `10`

> **Tip:**
>
> You can also drag and drop the **P3_NEW** facet in the Rendering tab to the appropriate location.

6. Review your changes. Click **Save and Run Page**.

   The following example shows the new Input Facet in the Search region. In this example, the value 1000 is entered in the **Salary Less than** facet. The report in the Search Results region displays two employees that have a salary of less than $1000.



8.7.6.6 Adding a Facet on a Column Containing Multiple Values

Add a facet to filter a column storing multiple values.

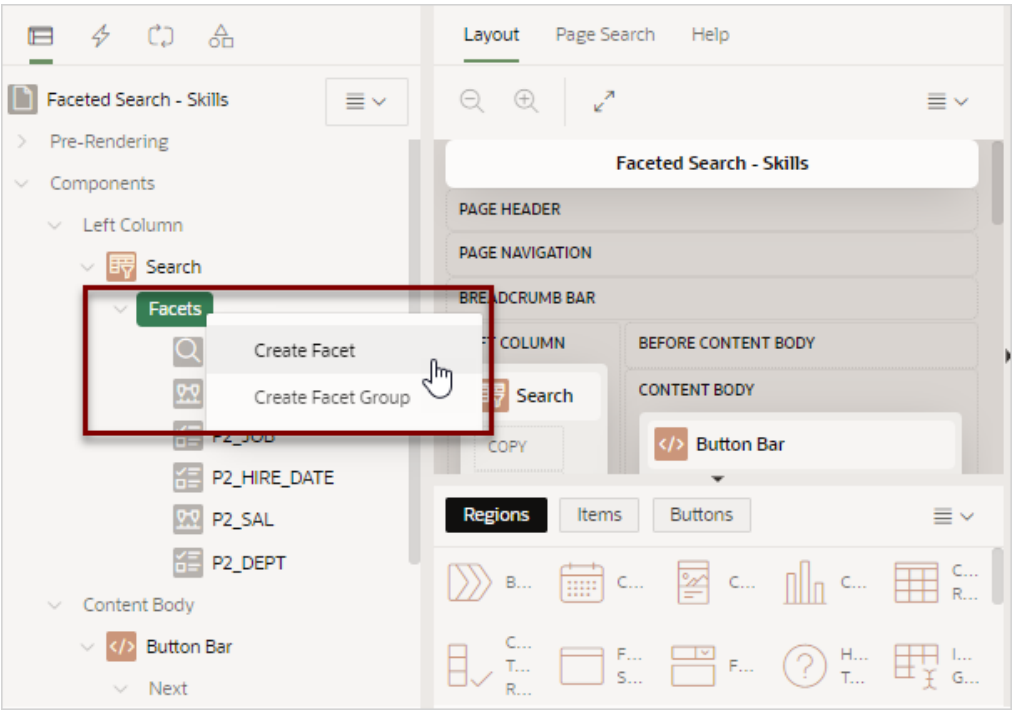
> **Note:**
>
> Creating a facet on a column containing multiple values requires Oracle Database release 12.2 or later. Using an earlier release raises an error.

Faceted Search supports the ability to filter columns storing multiple values as one string.

The following example demonstrates how to create a Checkbox Group facet which filters a column containing employee skills. The following example includes a Faceted Search page built on a table named `EMP_SKILLS`. This example demonstrates how to create a Checkbox Group facet that maps to the `SKILLS` column which contains multiple values using a colon separator (for example, `PLSQL:Spatial`).

To add facet to filter a column storing multiple values:

1. View the faceted search page in Page Designer.
2. Create the new facet:
   a. In the Rendering tab (or left pane), find the **Search** region.
   b. Right-click **Facets** and select **Create Facet** from the context menu.



A new facet appears in the Rendering tab. In this example, the facet is named **P2_NEW**.

3. In the Rendering tab, select the new facet and then update the Identification and Label attributes in the Property Editor (right pane):

   a. Identification, Name - Enter the name of this item (for example, `P2_SKILLS`).

      Item names must be valid Oracle identifiers. Oracle recommends that item names be no longer then 30 bytes so they can be used as bind variables in SQL Queries.

   b. Identification, Type - Specify how this item is rendered. Select **Checkbox Group**.

   c. Label, Label - Enter the label for the facet. The label displays on the page only if the facet displays. Enter:

      ```
      Skills
      ```

4. Update List of Values attributes:

   a. List of Values, Type - Select **Distinct Values**.

      Distinct Values creates a list of values is based on an automatically generated query that selects the distinct column values.

   b. List of Values, Include Null Option - Enable this option to include a NULL option at the top of the list.

   c. List of Values, Null Display Value - Enter:

      ```
      No Skills
      ```

5. Update the Multiple Values attributes:

   a. Multiple Values, Type - Determines whether the Source Column contains multiple values, and how these are encoded. For this example, select **Delimited List**.

      Options include:

      - **No** - Column does not contain multiple values.

- **Delimited List** - Column contains values as a delimited list. Example:

  `APEX,SQL,PL/SQL`

- **JSON Array** - The Source column contains a JSON array of scalar values. APEX will raise error messages if the column contains invalid JSON syntax. Example:

  `APEX,SQL,PL/SQL`

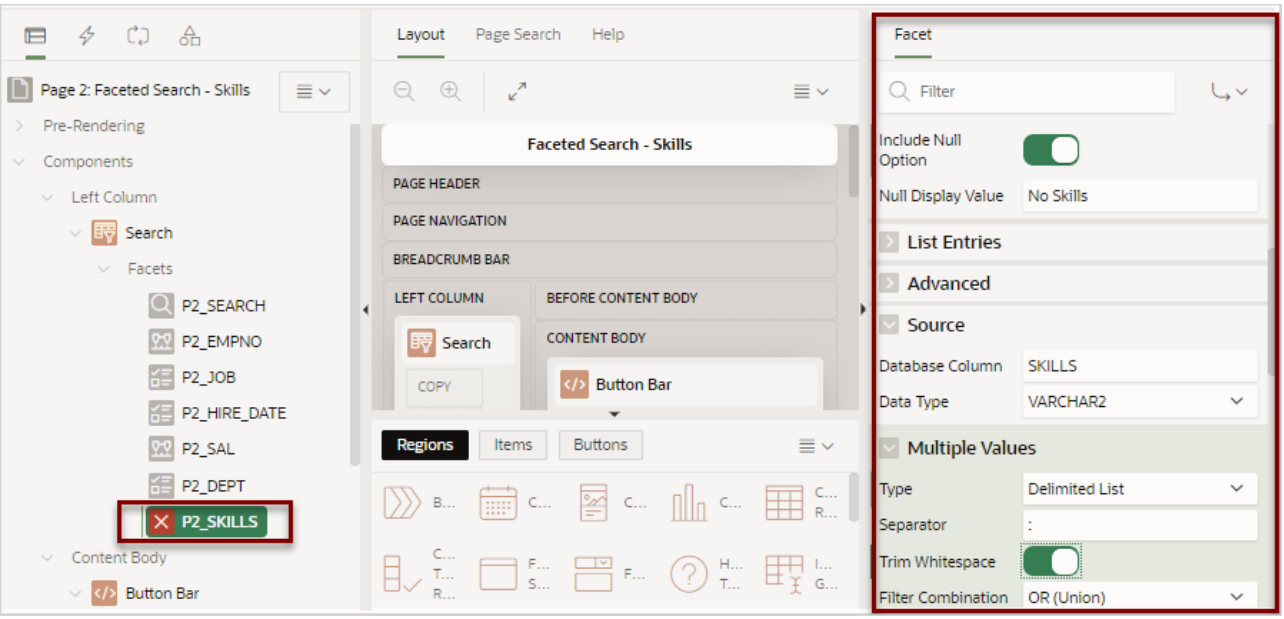
b. Multiple Values, Separator - Identifies the character which separates values from each other, such as a colon (`:`).

   Double-quote (`"`) and backslash (`\`) characters are not supported as a Value Separator.

c. Multiple Values, Trim Whitespace - Specify whether to trim leading and trailing whitespace from individual values after splitting.

   Enabling **Trim Whitespace** increases the parsing effort and results in slower performance. Oracle recommends disabling this option. For larger data sets, consider data cleansing instead.

d. Multiple Values, Filter Combination - Specify how to combine individual values when filtering. For this example, select **OR (Union)**.



6. Source, Database Column - Verify the case sensitive database column name used as the source for this facet. Since this facet maps to the `SKILLS` column, this attribute should display:

   `SKILLS`

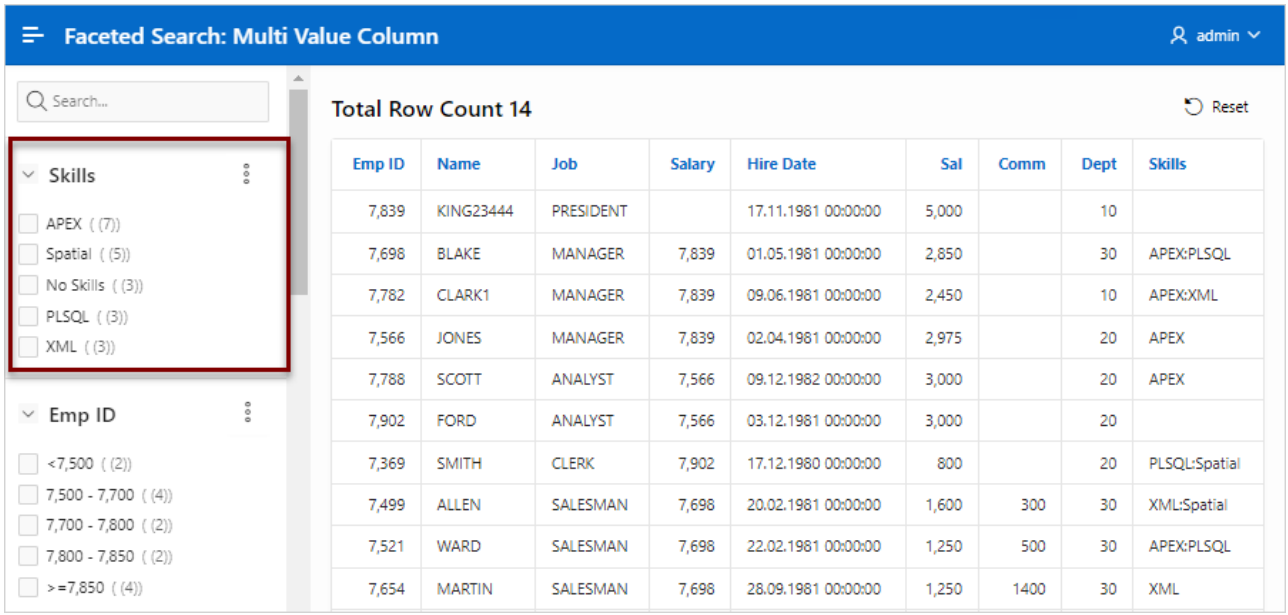
7. Change the facet position so it displays after the **P2_SEARCH** facet.

   - Layout, Sequence - Enter: `20`

> **Tip:**
>
> You can also drag and drop the **P2_SKILLS** facet in the Rendering tab to the appropriate location.

8. Review your changes. Click **Save and Run Page**.

   The following example shows a new facet labeled **Skills** which contains five checkboxes. Clicking a checkbox filters the report to only display only employees having the selected skill. In this example, no skills are selected.



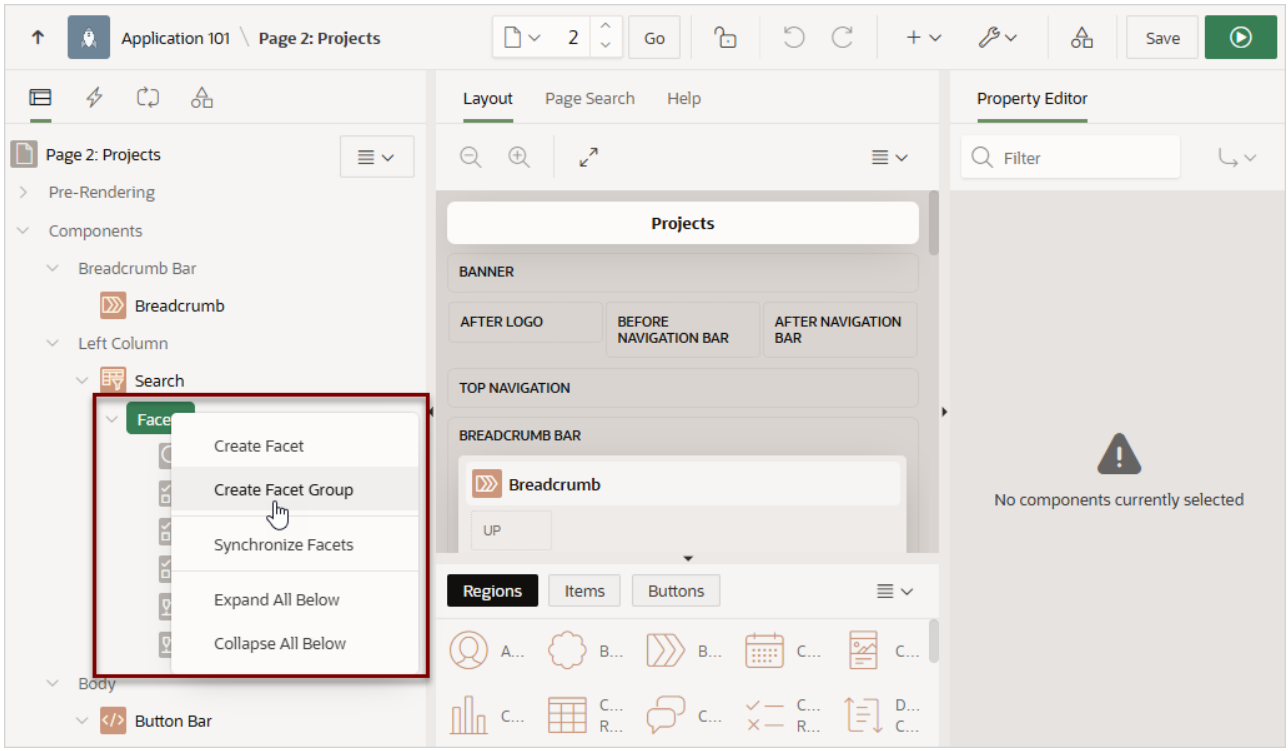
8.7.6.7 Creating Facet Groups

Create facet groups for data models having multiple flag columns.

Data models having multiple flag columns can result in a busy faceted search UI. Flag columns typically display two values such as `Yes` or `No` (for example, `In Stock`, `Orderable`, and so on). To streamline your faceted search UI, you can create a facet group which displays one facet that contains multiple checkboxes.

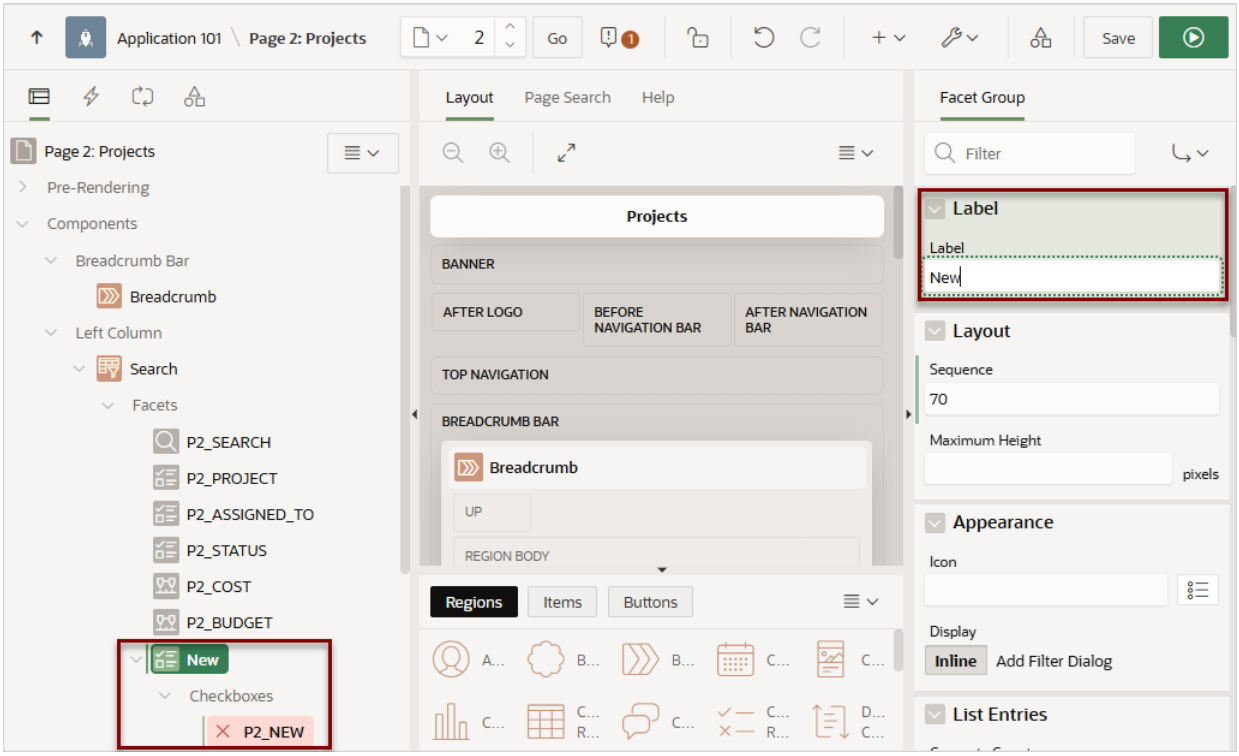
To create a facet group:

1. View the faceted search page in Page Designer.
2. In the Rendering tab (or left pane), find the **Search** region.
3. In the Rendering tab (or left pane), right-click **Facets** and select **Create Facet Group** from the context menu.

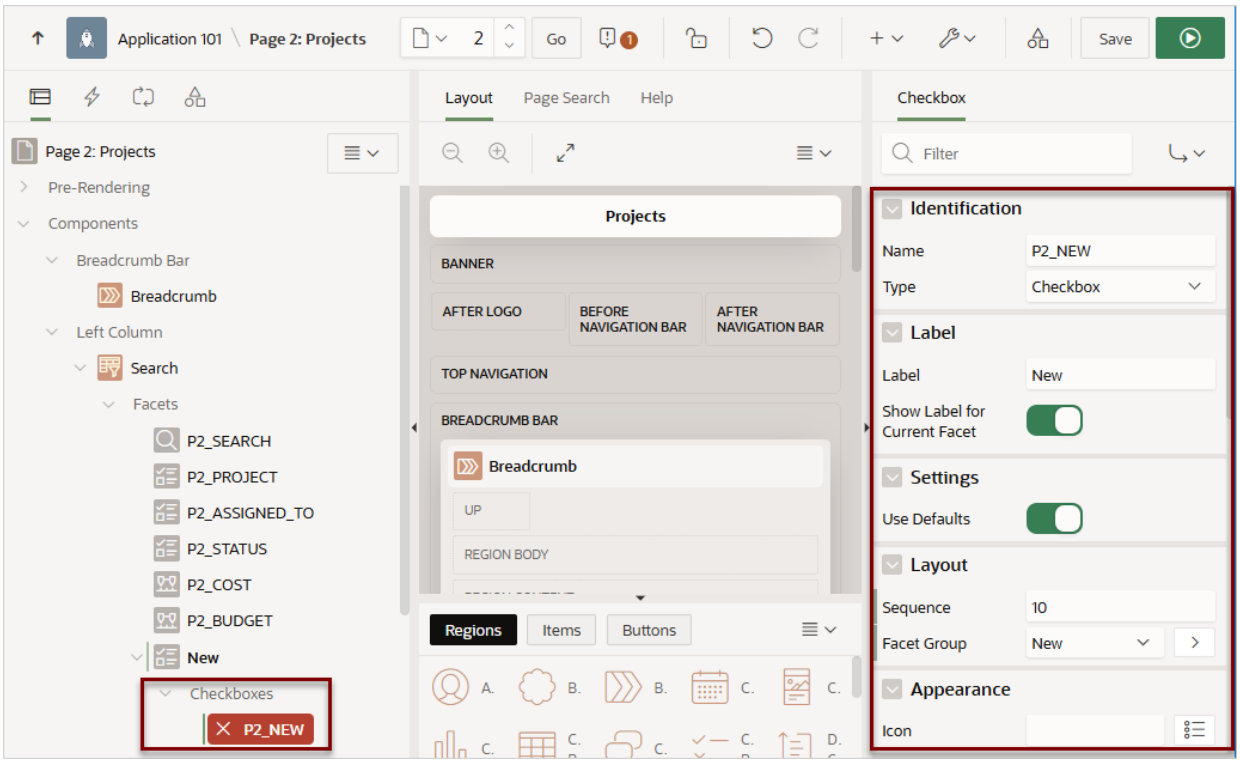


A Facet Group appears in the Rendering tab and a new checkbox named PX_NEW.

4. Update the label for the new facet group:
   a. In the Rendering tab, select the new facet group, **New** (if not already selected).
   b. In the Property Editor, edit the **Label** attribute.



5. Edit the checkbox, **PX_NEW**
   a. In the Rendering tab, select the checkbox facet, **PX_NEW**.



   b. In the Property Editor, edit the following Checkbox attributes:
      - Identification, Name - Update the name of this item.
      - Settings, Use Defaults - Disable **Use Defaults**.

        The Checked Value attribute appears.
      - Settings, Checked Value - Enter the value stored in the page item if the user checks the checkbox. The value can be up to 255 characters long.
      - Source, Database Column - Enter the case sensitive database column name used as the source for this facet.
      - Source, Data Type - Select the data type of the underlying database column.

6. Add more facet checkboxes:
   a. In the Rendering tab (or left pane), right-click **Checkboxes** and select **Create Checkbox**.

      A new checkbox named **PX_NEW** appears.
   b. In the Property Editor (or right pane), edit the following Checkbox attributes:
      - Identification, Name - Update the name of this item.
      - Settings, Use Defaults - Disable **Use Defaults**.

        The Checked Value attribute appears.
      - Settings, Checked Value - Enter the value stored in the page item if the user checks the checkbox. The value can be up to 255 characters long.
      - Database Column - Enter the case sensitive database column name used as the source for this facet.

- Data Type - Select the data type of the underlying database column.

7. Repeat the previous step until all facet checkboxes have been added.
8. Review your changes. Click **Save and Run Page**.

8.7.6.8 Configuring Single Facet Charts

Configure region and facet attributes to declaratively control how and if single facet charts display in the facet search region.

By default, when you create a facet that computes counts with a wizard, each facet that computes counts includes a chart based on the facet counts. When users click the three dots context menu on the facet and then **Show Chart**, a chart displays in a popup dialog. Users can toggle between a bar chart and pie chart by clicking the chart icon on the right side of the dialog. You can configure region and facet attributes to control if these charts appears.

> **Note:**
>
> To view an example of a single chart based on the facet counts, see Example Faceted Search.

To configure region and facet attributes to control facet charts:

1. View the faceted search page in Page Designer.
2. Edit faceted search region attributes:
   a. In the Rendering tab (or left pane), select the **Search** region.
   b. In the Property Editor (right pane), select the **Attributes** tab.
   c. Settings, Show Charts - Specifies where facet value charts are shown. Options include:
      - **Dialog** - Facet value charts are shown in a non-modal dialog. Adds the Show Chart action on the Facet Context Menu in the facet header. When the user clicks the Show Chart action, a popup dialog displays the chart.
      - **No** - The Show Chart action is not displayed on Facet Context Menu in the facet header.

        Also, when this attribute is set to **No**, the facet attributes **Chart** and **Show Chart Initially** are not applicable.
      - **Selector** - Specify the selector for an element you place somewhere on the page. When the user clicks the **Show Chart** action, the chart is added to this element. This option enables you to place the charts in a dashboard like area just about anywhere on the page.

        For example, suppose you added a static content region somewhere on the page and added a `<div>` with an `id` like this:

        ```
        <div id='facetDashboardArea'></div>
        ```

        Then in the Charts Selector attribute you would enter:

        ```
        #facetDashboardArea
        ```
   d. Settings, Display Chart for Top N Values - Use this attribute to control the maximum number of values to display in the chart for each facet. A value of 10 means that the

facet charts will show at most the first 10 items. If the facet has the Sort By Top Counts attribute **On** then this will be the 10 largest count values.

> **Tip:**
>
> You may need to reset the region before changes to **Display Chart for Top N Values** appear because the current set of visible charts is stored in browser session storage.

3. Configure facet attributes.
   a. In the Rendering tab locate the **Search** region and select a facet.
   b. In the Property Editor, review the facet attributes. Some available attributes include:
      - List Entries, Compute Counts - Single facet charts are based on facet counts. Note that **Compute Counts** must be enabled for the **Show Chart** action to display on the facet context menu in the facet header.
      - List Entries, Show Counts - Enable or disable this option to show or hide the count of occurrences of the list entries for the facet or filter.
      - List Entries, Display Filter Initially - Enable this option to have a filter field is display above the list entries. If the number of entries in the facet is less than the value on Maximum Displayed Entries, the filter field will not display.

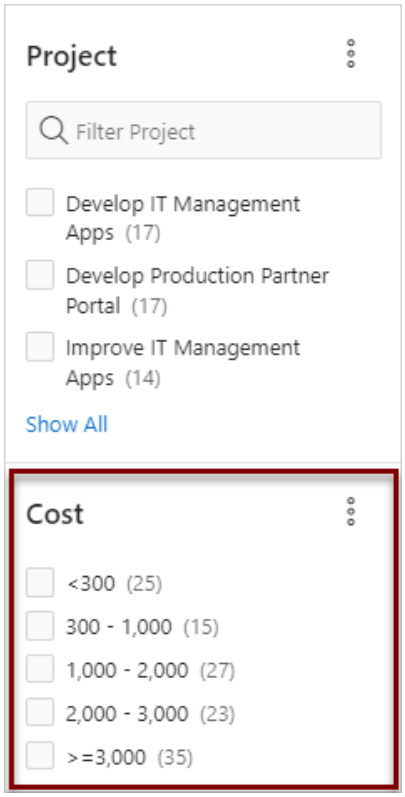
> **Tip:**
>
> Remember that each attribute in the Property Editor has associated Help. To view Help, select the attribute in the Property Editor and then click the Help tab in the central pane. Help describing the attribute appears in the central pane.

4. Click **Save** or **Save and Run Page**.

8.7.6.9 Editing the Static List of Values for a Range Facet

Edit a range facet that displays as a static list of values.

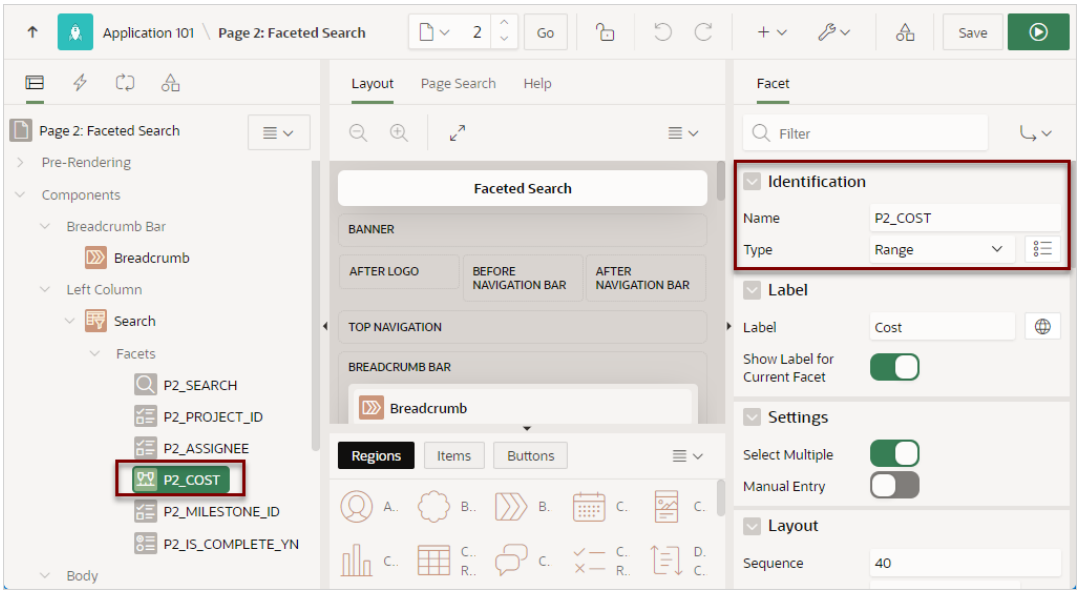
The following example demonstrates how to edit a facet that renders as a static list of values. The following example is built on the `COST` column of the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data. This example demonstrates how to update the Cost list shown in the following illustration so that the first two rows display in increments of 500 instead of 300.



To edit a facet that displays as a static list of values:

1. View the faceted search page in Page Designer.
2. In the Rendering tab (or left pane) locate the **Search** region and select the facet, for example **P2_COST**.

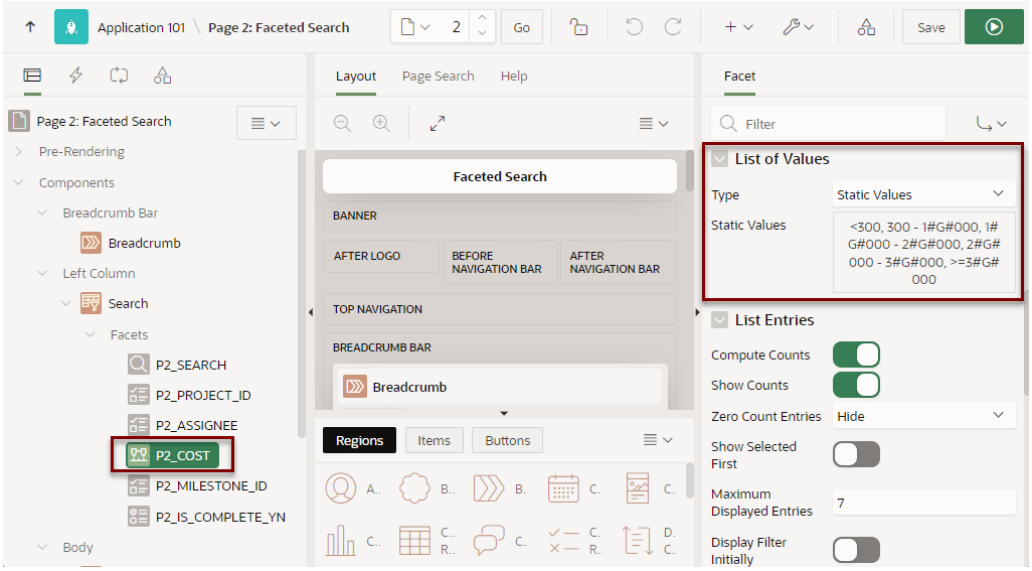
   In the Property Editor (or right pane) note that the **Type** attribute indicates that P2_COST is defined as a **Range**.



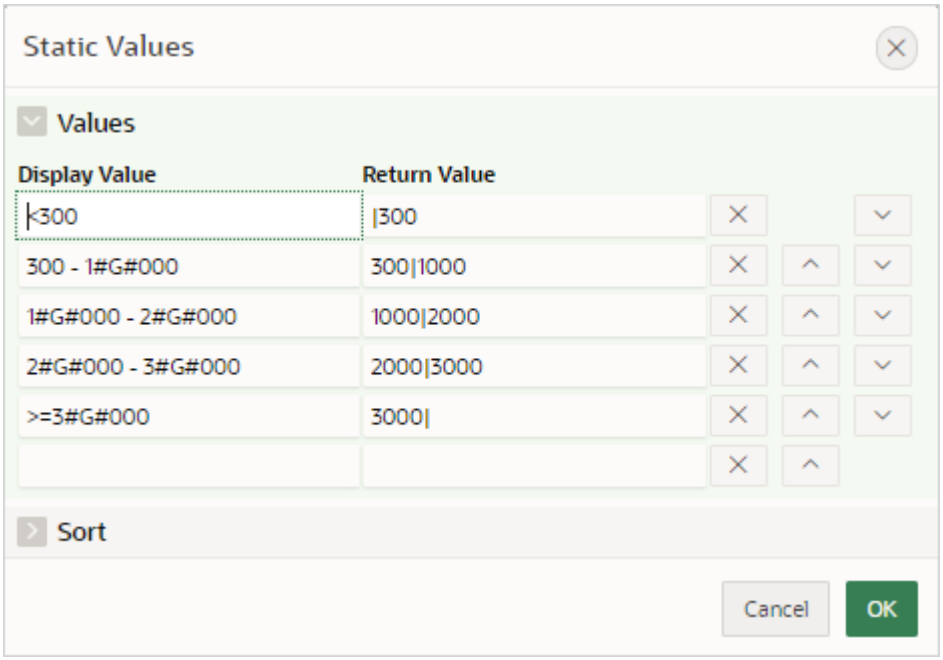
3. Edit the List of Values:

   a. Find the **List of Values** group.

b. Click **Static Values**.



The Static Values dialog appear.



To avoid hard-coding a specific separator in the Display Value, the Static Values dialog uses substitution strings which are replaced with the appropriate NLS separator at runtime.

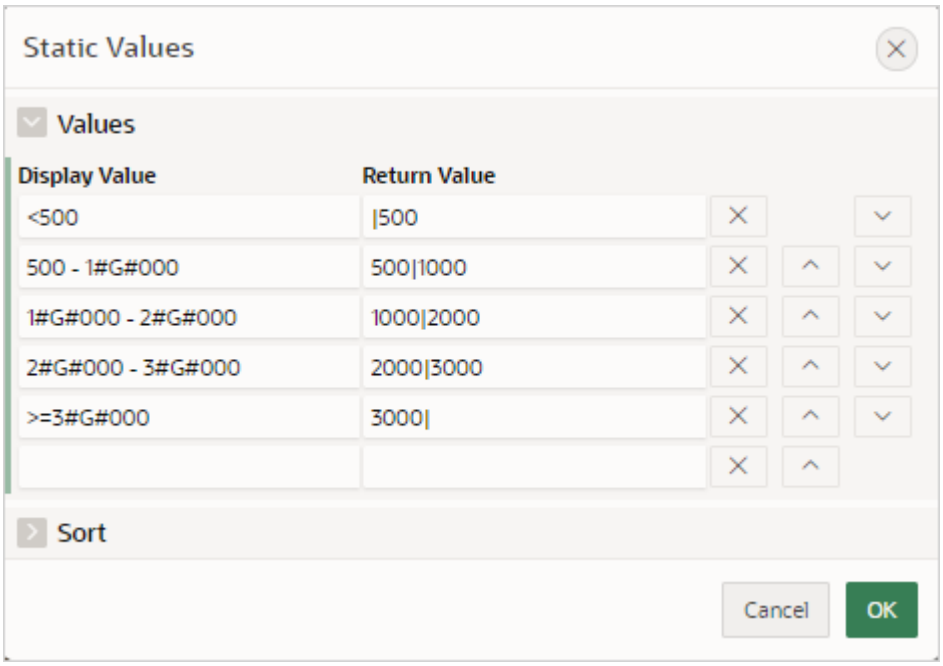
In this example, the substitution string `#G#` is replaced with the NLS Group Separator, comma (`,`). Other lists can include a `#D#` substitution string which is replaced at runtime with the NLS Decimal Separator.

Therefore, in this example `1#G#000 - 2#G#000` displays at runtime as `1,000 - 2,000`.

c. Update the Display Values and Return Values of the first two facets to use increments of 500:

- Row 1:
  - **Display Value** - Update `<300` to `<500`

  - **Return Value** - Update `|300` to `|500`
- Row 2:
  - **Display Value** - Update `300 - 1#G#000` to `500 - 1#G#000`
  - **Return Value**: Update `300|1000` to `500|1000`



d. Click **Ok** to accept the changes.

4. Review your changes. Click **Save and Run Page**.

   The revised facets search page displays. Note the first two facets in the Cost list display as **<500** and **500 - 1,000**.

8.8 Managing Interactive Grids

An interactive grid presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard.

- Example Interactive Grid
  View interactive grid examples.
- Supported Columns in Interactive Grids
  Learn about column types supported by interactive grids.

- Creating an Interactive Grid Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains one or multiple interactive grid report pages.
- Creating an Interactive Grid Using the Create Page Wizard
  Run the Create Page Wizard to add a new interactive grid page to an existing application.
- Creating Master Detail from an Existing Interactive Grid
  Create a single page master detail which includes two editable interactive grids based on two related tables or views.
- Managing Interactive Grid Region Attributes
  Edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- Managing Interactive Grid Attributes
  Edit interactive grid Attributes to control how an interactive grid works.
- Managing Interactive Grid Column Attributes
  Edit column attributes to control the display, features, and behavior of interactive grid columns.
- About Customizing Interactive Grids in a Running Application
  When running an application, users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu.
- Saving Interactive Grid Reports
  After users customize an interactive grid they can save it as either a Private or Public report.
- Linking to Interactive Grid Reports
  Developers can use the syntax in the `Request` value and `ItemNames` section of the URL to link to interactive grid report.

See Also:

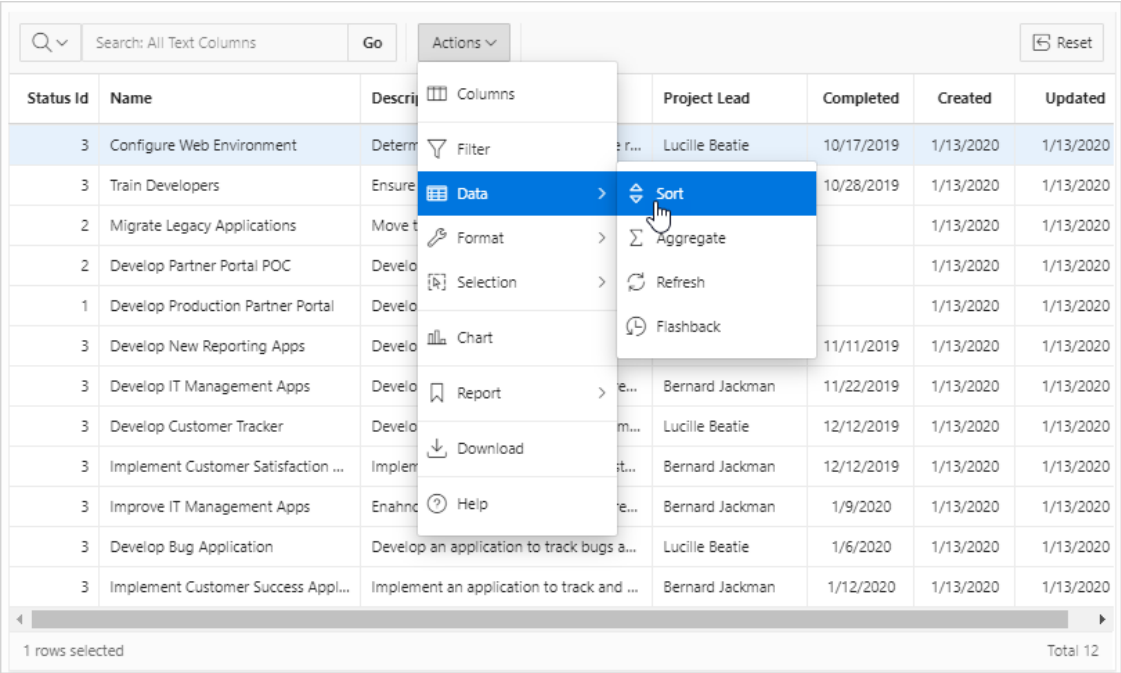
- Creating a Report Using the Create Page Wizard
- Using Interactive Grids in *Oracle APEX End User's Guide*
- Making an Existing Interactive Grid Editable

8.8.1 Example Interactive Grid

View interactive grid examples.

An interactive grid presents users with a set of data in a searchable, customizable report. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse and keyboard. Users can lock, hide, filter, freeze, highlight, sort individual columns, and create control breaks on specific columns using the **Actions** and **Column Heading** menus. Advanced users can define breaks and aggregations against columns. Users can also directly customize the appearance of an interactive grid. Users can use the mouse and keyboard to resize the width of a column and drag and drop columns into different places in the grid. Once customized, the report can be saved as either a private or public report. Both the Create Application Wizard and Create Page Wizard support the creation of interactive grids.

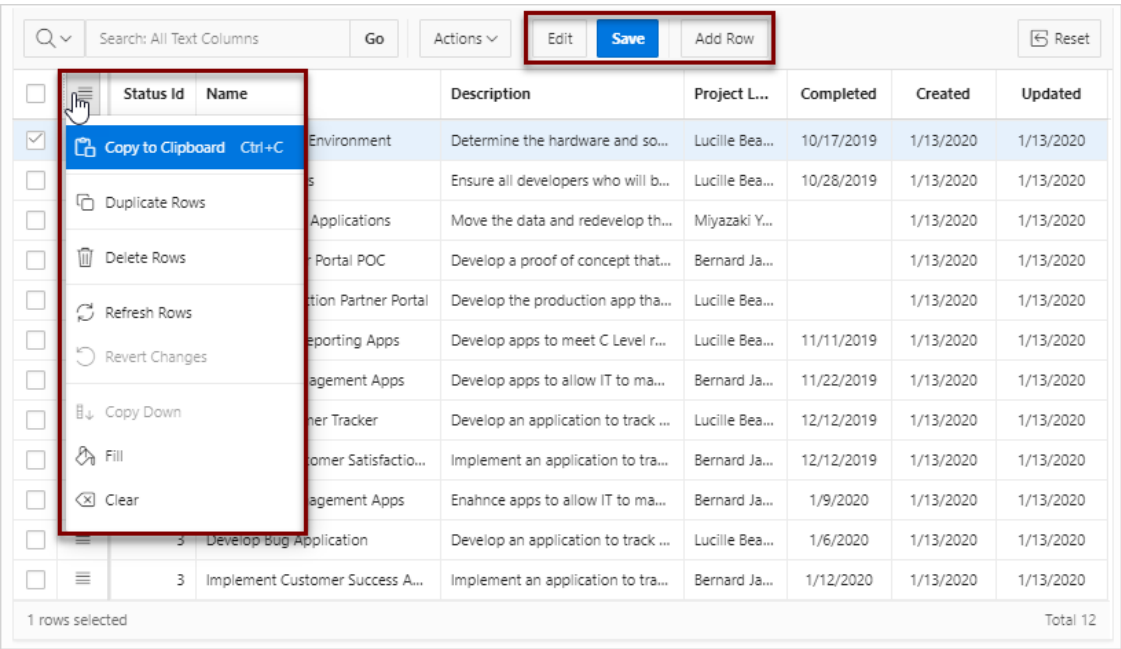
The following is an example of an interactive grid.



Editable Interactive Grid

Developers have the option of making an interactive grid editable. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Editable interactive grids include additional controls. A Row Actions menu displays at the start of each row and enables users to add, edit, and refresh rows. Edit, Save, and Add Row buttons also display to the right of the Actions menu.

The following is an example of an editable interactive grid.



Tip:

To view an example, install the sample app, *Sample Interactive Grids*. To learn more, see Installing Apps from the Gallery.

See Also:

- Using Interactive Grids in *Oracle APEX End User's Guide*
- Making an Existing Interactive Grid Editable

8.8.2 Supported Columns in Interactive Grids

Learn about column types supported by interactive grids.

Interactive grids support the following built-in column types. Depending on your application configuration, you may also have additional column type plug-ins available here. These display as *<My Plug-in> [Plug-in]*. To view the column type in Page Designer, select the column and find the **Identification**, **Type** attribute in the Property Editor.

Note:

You can also view this information in Page Designer Help. View the page containing the report in Page Designer. Select a column and then the **Identification**, **Type** attribute. Click the **Help** tab in the central pane.

Column Type	Description
Actions Menu	Displays the Row Actions menu on each row of the interactive grid. Depending on the grid attributes and end user authorizations, the Row Actions menu includes options such as Single Row View, Add Row, Delete Row, and Refresh Row, and so on. These menu options enable end users to perform specific actions on the current row in the grid. The Row Actions menu can be used in both editable and non-editable interactive grids, but the menu options differ slightly. For non-editable interactive grids, the Row Actions menu has a subset of options, for example Single Row View and Refresh Row, and does not include editable actions like Add Row, Delete Row and so on. When an interactive grid is editable, the `APEX$ROW_ACTION` column is automatically added and set to a type of Actions Menu. For non-editable interactive grids, this column must be added manually.

Column Type	Description
Checkbox	Displays a single checkbox that the user can check or uncheck. The defined checked or unchecked value is stored in session state when submitted.
Checkbox Group	Displays multiple values as check boxes, enabling the end user to select multiple values. A list of values is required for items displayed as check boxes. The values corresponding to the checked boxes are stored in a single colon-delimited string.
Color Picker	Enables users to choose or enter a color. Colors can be selected from a color spectrum.
Date Picker	Enables users to enter a date. You can either use a text field that opens a date picker as a popup, or you can use an inline or native date picker.
Display Only	Displays a non-enterable text item.
Hidden	Items that are included within the page source but are not rendered. Hidden item values are saved in session state. They are generally used to store values required by page processing or other page items, but are not displayed to the end user.
HTML Expression	Displays a non-enterable item based on a HTML expression.
Link	Displays a non-enterable item as a link.
Markdown Editor	Displays a text area supporting markdown input, with text formatting options, support for including images, and a preview option to view the formatted text. The Markdown Editor enables you to write using easy-to-read, easy-to-write plain text, then convert it to structurally valid HTML using the various options provided and supported syntax.
Number Field	Displays a number field. This item type automatically validates that the value is a number.
Password	Displays an HTML password form element. As the end user enters text a black dot is displayed for that character instead of the actual character entered.
Percent Graph	Displays the value as a percentage graph. The value retrieved must be between 0 and 100.
Popup LOV	Displays a text item with a popup list of values icon. When the end user clicks the icon, a popup window appears with a search field and a list of supported values. Popup LOV is best suited for large lists since end users can enter search criteria to reduce the available values displayed. For relatively small lists, a Select List is often better choice.
Radio Group	Displays multiple values as radio group options, enabling the end user to select a single value.
Rich Text Editor	Displays a rich text editor with comprehensive text formatting options. End users can enhance the content displayed in a similar fashion to using a word processor, such as Microsoft Word.

Column Type	Description
Row Selector	Displays a check box on each row of an interactive grid, enabling the selection of specific row(s) for row manipulation. By default the `APEX$ROW_SELECTOR` column is set to a type of Row Selector. This column is automatically added when the interactive grid is made editable.
Select List	Displays an item with a built-in list of values selector. When the end user clicks the item, the list of supported values displays directly inline with the current item. Select List is best suited for relatively small, discrete lists. End users can very quickly select a value from the list without needing to change focus to a popup dialog. For large lists, a Popup LOV is often a better choice.
Shuttle	Displays as a multiple select list that includes two boxes containing lists. The left list displays the source list of values that have not been selected and the right list shows the currently selected values. End users can select one or more values in a list then use the shuttle controls to move the selected values or all values. The current values are stored in a single colon-delimited string.
Star Rating	Displays a numeric value as a number of stars or other icons. The end user can change the value by selecting the desired number of stars. The value can be adjusted by clicking the number of stars with the mouse or by using the keyboard. The icon to be displayed, the background and foreground colors can be adjusted either in the component settings or in the individual item settings. To use individual items and colors, set the attribute **Use Component Defaults** to **No**. As icons, you may use any icon from the Icon Font available in your application. For Universal Theme applications, use an icon from the Font APEX library. See Accessing Font APEX.
Switch	Displays as a flip toggle switch.
Text Field	Displays a text field.
Text Field with autocomplete	Displays a text field that displays a list of possible values, based on the text already entered by the end user, inline with the text item. The list is further refined as the end user types in more text.
Textarea	Displays a multiple-row text area.

8.8.3 Creating an Interactive Grid Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains one or multiple interactive grid report pages.

To create an interactive grid using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. To add a report, click **Add Page** and select **Interactive Grid**.
8. On Add Interactive Grid Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Page Source - Select the source for the report:
      - **Table or View**
      - **SQL Query**

      What you select determines what displays next. Follow the on-screen instructions. To learn more about an attribute, click the Help icon in the lower left corner.
   d. Determine if the report is editable. Select either:
      - **Allow Editing**
      - **Read Only**
   e. Advanced - Expand **Advanced** to configure these options:
      - Set as Home Page - Enable this to make this page the home page for the application.
      - Set as Administration Page - Enable this to have this page display under Administration.
      - Page Help - Enter text to be displayed when the user selects Page Help.
   f. Click **Add Page**.
9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

> **See Also:**
>
> Creating a Report Using the Create Page Wizard

8.8.4 Creating an Interactive Grid Using the Create Page Wizard

Run the Create Page Wizard to add a new interactive grid page to an existing application.

To create an interactive grid page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Interactive Grid**.
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
   d. Include Form Page - Enable this flag to create a form page along with the report page.

      Form pages are always created as Modal Dialog pages. Enabling this flag displays additional attributes. To learn more about an attribute, see field-level Help.
6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
7. Source Type:
   a. Specify the source of the new page. Options include:
      - **Table**:

  - Table /View Owner - Select the owner of the table on which you are building the page.
  - Table/View Name - Select the table or view on which the page will be based.
- **SQL Query**:
  - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

b. Editing Enabled - Select whether to enable editing for this interactive grid.

8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

9. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

8.8.5 Creating Master Detail from an Existing Interactive Grid

Create a single page master detail which includes two editable interactive grids based on two related tables or views.

You can define any number of master detail relationships on a single page by simply adding a related interactive grid to the page. If you have an existing interactive grid, you can add another related interactive grid by doing the following:

- Create the parent interactive grid.
- Define the primary key (if needed).
- Create the child interactive grid.
- For the child interactive grid, specify the Master Region and Master Column.

The following example demonstrates how to create an interactive grid on the departments table, `OEHR_DEPARTMENTS` by running the Create Page Wizard. This wizard creates the parent interactive grid and enables you to define the primary key. Then, you create a child interactive grid on the employees table, `OEHR_EMPLOYEES`. The `OEHR_DEPARTMENTS` and `OEHR_EMPLOYEES` tables are available in the sample data set, HR Data.

To add a related interactive grid below an existing grid:

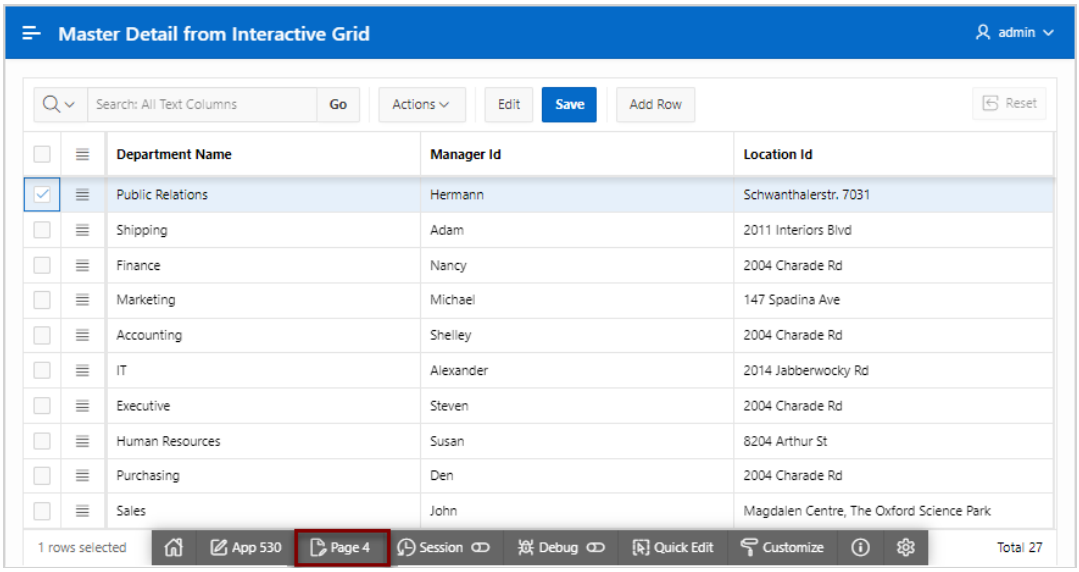
1. Create an application.

2. Create the first interactive grid on the `OEHR_DEPARTMENTS` table:
   a. On the Application home page, click **Create Page**
   b. Under Component, select **Interactive Grid**.
   c. Name - Specify a text name for this page (for example, `Departments`).

   > **Tip:**
   > This text is also used for report region Title. After page creation, you can modify the Title in Page Designer.

   d. Page Mode - Select **Normal**.
   e. Include Form Page - Disable this flag.
   f. Data Source - Select **Local Database**.
   g. Source Type - Select **Table**.
   h. Table /View Name - Select **OEHR_DEPARTMENTS (table)**.
   i. Editing Enabled - Enable this flag to make this interactive grid editable.
   j. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   k. Click **Next**.
3. Primary Key Column 1 - Accept the default.
4. Click **Create Page**.
5. Click **Save and Run Page** to view a rendered version of the parent interactive grid.

   This following image shows an editable interactive grid created on the `OEHR_DEPARTMENTS` table.



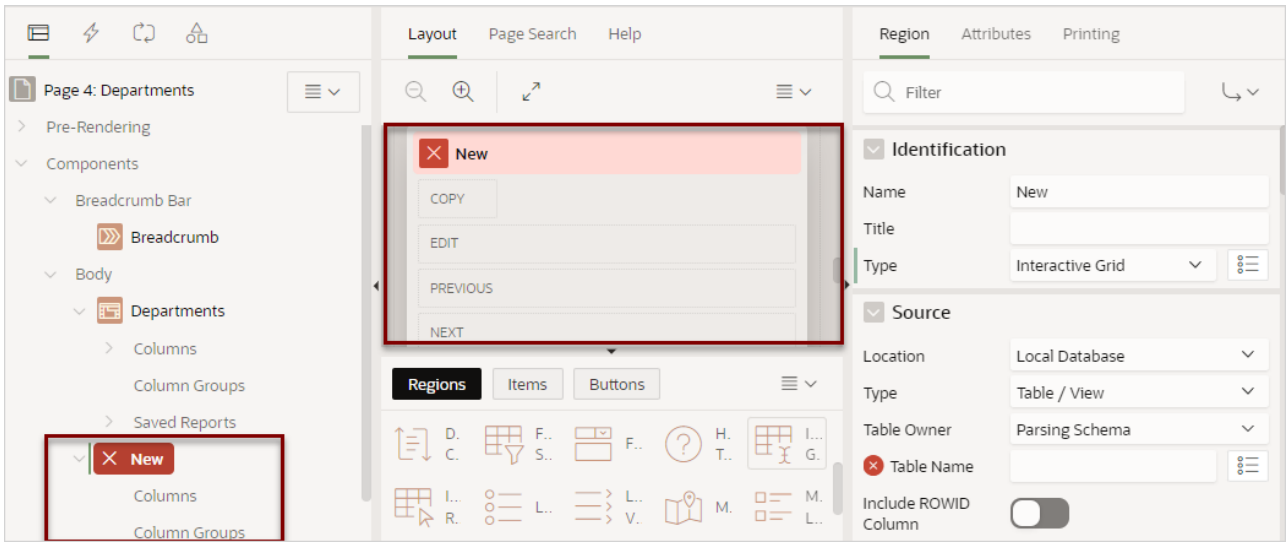
6. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar, where *X* is the page number.
7. Create the child interactive grid on the `OEHR_EMPLOYEES` table:

a. In the Gallery, locate the Regions tab.

b. Under Regions, find and then right-click **Interactive Grid**.

c. Select **Add To**, **Body**, the region, **Departments**, and then **After**.

The Rendering tab and Layout tab both display the new region.



8. Edit the New child region:

a. In the Rendering tab, select the **New** region (if not already selected).

b. In the Property Editor, edit the region attributes:

> **Note:**
>
> To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.

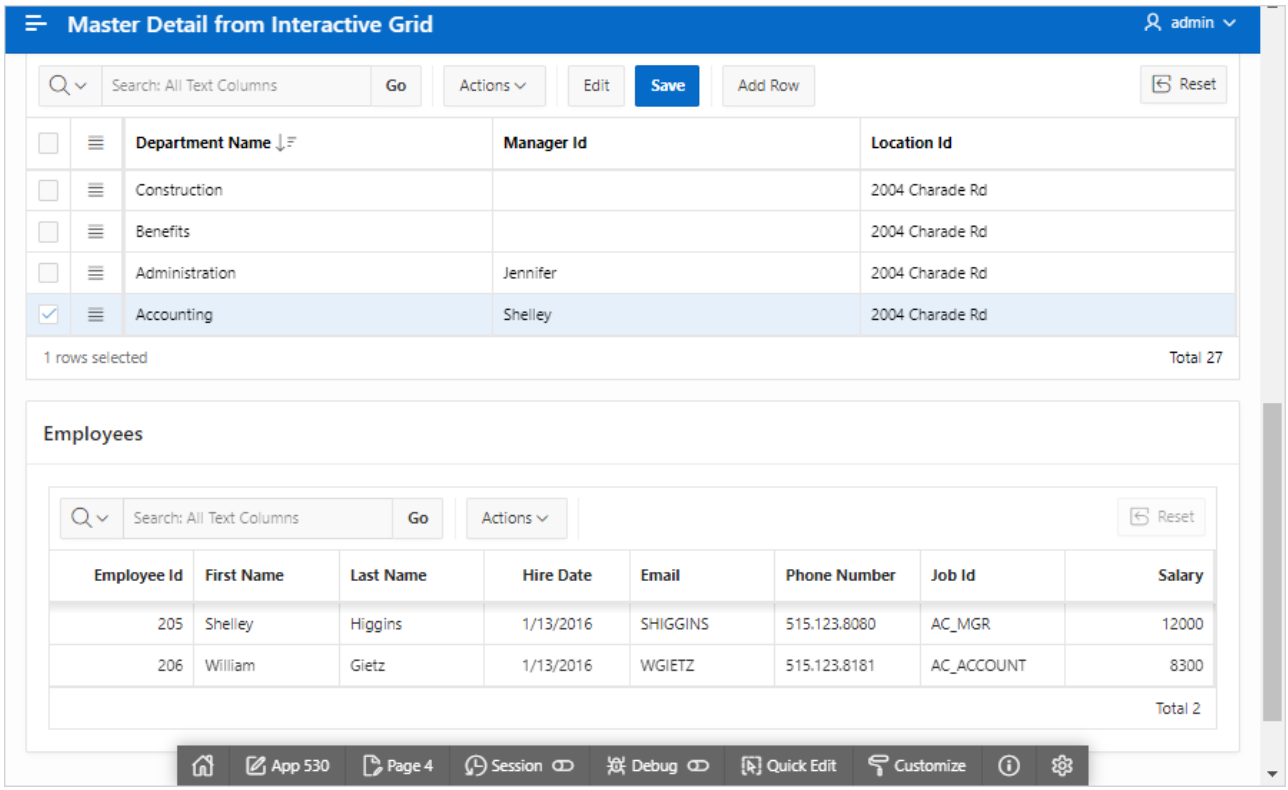
- Identification, Name - Enter `Employees`.
- Source, Type - Select **SQL Query**.
- Source, SQL Query - Enter the query:

```
select OEHR_EMPLOYEES.EMPLOYEE_ID as EMPLOYEE_ID,
    OEHR_EMPLOYEES.FIRST_NAME as FIRST_NAME,
    OEHR_EMPLOYEES.LAST_NAME as LAST_NAME,
    OEHR_EMPLOYEES.HIRE_DATE as HIRE_DATE,
    OEHR_EMPLOYEES.EMAIL as EMAIL,
    OEHR_EMPLOYEES.PHONE_NUMBER as PHONE_NUMBER,
    OEHR_EMPLOYEES.JOB_ID as JOB_ID,
    OEHR_EMPLOYEES.SALARY as SALARY,
    OEHR_EMPLOYEES.DEPARTMENT_ID as DEPARTMENT_ID
 from OEHR_EMPLOYEES
```

- Layout, Start New Row - Select **On**.

- Appearance, Template - Select **Standard**
- Master Detail, Master Region - Select the region, **Departments**.

9. For the child interactive grid, Employees, define `DEPARTMENTS` as the Master Column:
   a. In the Rendering tab, under the child interactive grid, **Employees**, expand the Columns node and select **DEPARTMENT_ID**.
   b. In Property Editor, edit the following:
      - Find **Master Detail**.
      - For Master Column, select **DEPARTMENT_ID**.
10. Click **Save**. To run the page, click **Save and Run Page**.



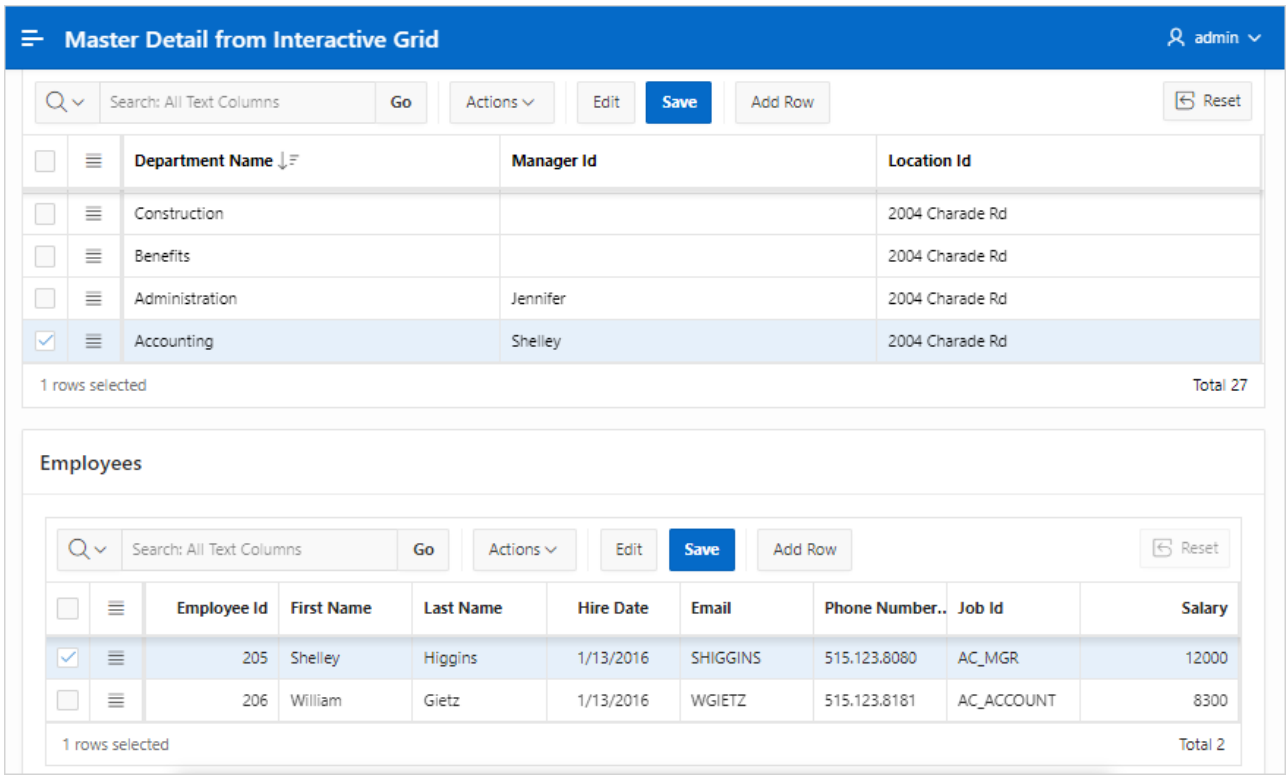
Note that parent the parent report, **Department**, filters the results in **Employees**, the child report.

Next, make the child report, Employees, editable.

11. Make Employees report editable:
    a. Access Page Designer. Click **Page X** on the Runtime Developer toolbar, where *X* is the page number.
    b. In the Rendering tab, select the **Employees** region.
    c. In the Property Editor, click the **Attributes** tab.
    d. In the Property Editor, edit the following:
       - Edit, Enabled - Select **On**.
       - Allowed Operations - Verify the following operations are enabled: **Add Row**, **Update Row**, and **Delete Row**.

12. Define the `EMPLOYEE_ID` column as the Primary Key Column:
    a. In the Rendering tab, under **Employees**, expand **Columns** and select **EMPLOYEE_ID**.
    b. In the Property Editor, find **Source**, and turn **Primary Key** to **On**.
13. Click **Save**. To run the page, click **Save and Run Page**.

    The child interactive grid, Employees, is now editable



8.8.6 Managing Interactive Grid Region Attributes

Edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

- Editing Interactive Grid Region Attributes
  Edit interactive grid region attributes in Page Designer.
- Displaying Interactive Grids Conditionally
  Dispay interactive grids conditionally by editing region attributes.

> **See Also:**
>
> Editing Pages in Page Designer

8.8.6.1 Editing Interactive Grid Region Attributes

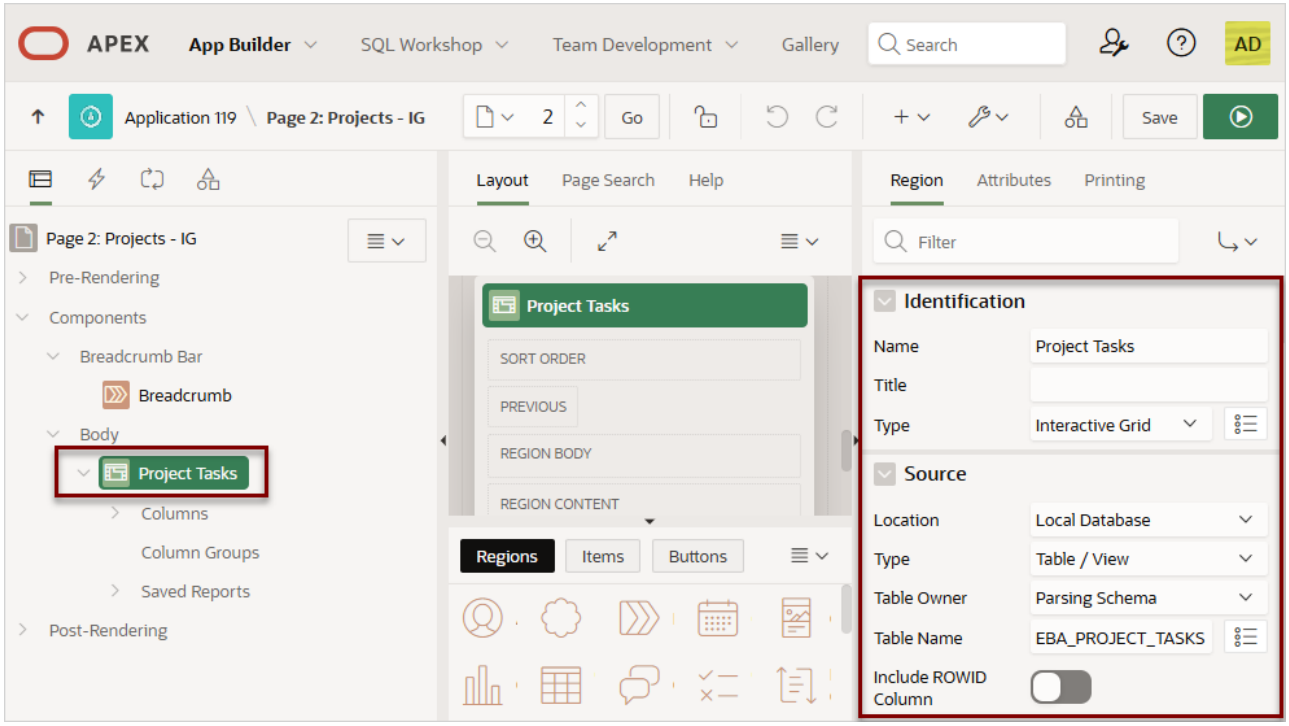
Edit interactive grid region attributes in Page Designer.

To edit interactive grid region attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

   In the Property Editor, the **Region** tab displays the attributes. Attributes are organized in groups.



   Use region attributes to alter the SQL source, change the region layout and appearance, create a master detail, define a region displays selector, or create conditions.
3. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

8.8.6.2 Displaying Interactive Grids Conditionally

Dispay interactive grids conditionally by editing region attributes.

To display an interactive grid conditionally:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

   The Property Editor displays the **Region** tab. Attributes are organized in groups.
3. Edit Server-side Condition:
   a. Find the **Server-side Condition** group.
   b. Type - Select a condition type and follow the on-screen instructions.

   The UI changes based on the Type you select.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7 Managing Interactive Grid Attributes

Edit interactive grid Attributes to control how an interactive grid works.

- Editing Interactive Grid Attributes
  Edit interactive grid Attributes in Page Designer.
- Making an Existing Interactive Grid Editable
  Make an existing interactive grid editable by configuring attributes in the Edit group.
- Controlling Interactive Grid Pagination
  Pagination provides the end user with information about the number of rows displayed and the current position within the result set.
- Customizing the Interactive Grid Toolbar
  Configure what controls display on the interactive grid toolbar.

- Enabling Users to Save Interactive Grid Public Reports
  Enable users to save an interactive grid as a Public report.
- Enabling Flashback and Charts
  Enable users to utilize flashback queries and define a chart in an interactive grid.
- Configuring Interactive Grid Download Formats
  Enable users to download an interactive grid to specified download formats.
- Enabling Icon View in an Interactive Grid
  Learn how to enable Icon View in an interactive grid.
- Enabling Detail View in an Interactive Grid
  Learn how to enable Detail View in an interactive grid.

8.8.7.1 Editing Interactive Grid Attributes

Edit interactive grid Attributes in Page Designer.

To edit interactive grid Attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.

   > **Tip:**
   >
   > To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

8.8.7.2 Making an Existing Interactive Grid Editable

Make an existing interactive grid editable by configuring attributes in the Edit group.

End users can alter the report layout of an interactive grid using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu. However, developers determine whether the underlying data is read-only or editable by users. When an interactive grid is editable, end users can edit the underlying data, add rows, delete rows, and refresh rows. By default, editable interactive grids include a Save button on the toolbar.

To make an interactive grid editable:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the interactive grid.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups

4. In the Property Editor **Attributes** tab, find the **Edit** group and update the following attributes:

   a. Enabled - Select **On**.

      Selecting **On** automatically creates a process to process the records.

   b. Allowed Operations - Select supported data manipulation operations. Options include: **Add Row**, **Update Row**, **Delete Row**

   c. Allowed Row Operations Column - Select a column where the column value indicates if the row can be updated and/or deleted.

   d. Lost Update Type - Select how to protect data against lost updates.

      Lost updates occur when two or more users try to update the same data, and the last update overwrites updates made by the previous user. To prevent this, updates can check the existing column value(s) in the database with the value(s) retrieved when the record was originally queried. For an example, see Page Designer Help.

      Lost Update Type options include:

      - **Row Values** - A checksum value is calculated for each row, when initially querying the data. The checksum is calculated by concatenating all of the updateable columns into a string and then generating a unique value. When committing the updated record, this checksum is compared to the checksum value.
      - **Row Version Column** - If your database table includes a column which is incremented every time a record is updated, preferably by a database trigger, then this column can be used instead of calculating checksums. Oracle does not recommend this option if your interactive grid updates data in multiple tables.

   e. Add Row If Empty - Specify whether to add a new row if the interactive grid currently has no records returned. When enabled, the `No Data Found` message never displays since there will always be at least one record. If no updates are made to the record that was added it will not be inserted when the region is saved.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.3 Controlling Interactive Grid Pagination

Pagination provides the end user with information about the number of rows displayed and the current position within the result set.

To change interactive grid pagination:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

   The Property Editor displays the attributes. Attributes are organized in groups.
4. Find **Pagination**.

   Pagination attributes provide the end user with information about the number of rows displayed and the current position within the result set.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Under Pagination, edit the following attributes:
   a. Type - Select an option:
      - **Page** - Rows display based on what the end users select in the Actions menu, Format, Rows per Page option. If there are additional rows, controls are added to the report footer which enables users to navigate forwards and backwards between row sets.
      - **Scroll** - Rows display to fill the height of the interactive grid. As the end user scrolls, additional rows display as needed.
   b. Show Total Row Count - Select **Yes** or **No**.

      Selecting **Yes** displays the total row count in the report footer. Enabling this option requires an additional query which may hinder performance on very large data sets
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.4 Customizing the Interactive Grid Toolbar

Configure what controls display on the interactive grid toolbar.

All interactive grids include a toolbar at the top of the page. Developers can customize what controls display on the toolbar or remove it.

> **Note:**
>
> Some of the options described in this procedure only apply if the interactive grid is editable. See Making an Existing Interactive Grid Editable.

To customize the interactive grid toolbar:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region containing the interactive grid.
3. Under the region, select the **Attributes** node.

   The Property Editor displays the attributes. Attributes are organized in groups.

   > **Tip:**
   >
   > To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.
4. In the Property Editor, find the **Toolbar** group and edit the following:
   a. Show - Select **Yes** or **No** to determine if a toolbar displays above the report.
   b. Controls - Configure the following:
      - Search Column Selection - Displays a drop down list of displayed non-numeric columns. If a column is selected, then the search is only performed against the values in the selected column.
      - Search Field - Displays a text field in the search bar used to enter search criteria.
      - Actions Menu - Displays the Actions menu button, which generally includes a comprehensive set of options. These options enables the end user to customize the report based upon the functions you make available.
      - Reset Button - Displays the Reset button. When clicked, any customizations, such as filters, column width, and ordering, are removed, and the report definition is reloaded from the server.
      - Save Button - Saves changes made to the interactive grid, without needing to save the whole page. The **Save Button** only displays when an interactive grid is editable and the end user has authorization to add, update or delete.
   c. Add Button Label - Only displays when the interactive grid is editable. For **Add Button Label**, enter an alternative label for the Add button. The default label is **Add Row**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.5 Enabling Users to Save Interactive Grid Public Reports

Enable users to save an interactive grid as a Public report.

Users can save an interactive grid as a Public report if the report developer enables that capability. Public reports display in a select list of available reports on the toolbar and are available to all users. However, only the user who creates a Public report can save, rename, or delete it. You can further restrict who can save Public reports by selecting an authorization scheme.

> **Tip:**
>
> The Save Public Report attribute is only available to authenticated users and can be further restricted by the defined Authorization scheme. See Establishing User Identity Through Authentication.

To enable end users to save Public reports:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor, find the **Enable Users To** group and edit the following:
   a. Save Public Report - Select **On**. Selecting **Off** enables users to save the current report as a Public report.
   b. Save Public Report Authorization - Only displays if **Save Public Report** is set to **On**. Select an authorization scheme to restrict who can save public report definitions. To enable an end user to save a public report, this authorization scheme must evaluate to TRUE for that user. If you do not select an authorization scheme, then any user may save public reports.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.6 Enabling Flashback and Charts

Enable users to utilize flashback queries and define a chart in an interactive grid.

Developers can enable flashback queries and support for Chart view by editing the interactive grid Attributes, Flashback and Define Chart View

To enable Flashback or Define Chart View:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor, find the **Enable Users To** group and edit the following:

a. Flashback - Specify whether end users can utilize flashback queries for this interactive grid.

   Flashback queries utilize the Oracle Flashback Technology and Database Point-in-Time Recovery to view data as it existed at a previous point in time. Users can only query this data, but can not rewind any data changes. How far back in time they can successfully perform a flashback query is determined by when the redo log files were last switched within the Oracle database, as this technology utilizes the RMAN backup.

b. Define Chart View - Specify whether end users can define a chart for this interactive grid.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.7 Configuring Interactive Grid Download Formats

Enable users to download an interactive grid to specified download formats.

To configure interactive grid download formats:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups

4. In the Property Editor Attributes tab, find **Enable User To** and edit the following:
   - Download - Select **On**.

   > **Tip:**
   >
   > Setting **Download** to **Off** prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.

5. Find **Download** and edit the following:

   a. Authorization - Select an authorization scheme to restrict who can download. To enable an end user to download, this authorization scheme must evaluate to TRUE for that user.

   b. Formats - Select download formats:
      - **CSV**
      - **HTML**
      - **PDF**
      - **Excel**

   c. Send as Email - Specify whether end users can send an email, with the report download included as a HTML attachment.

      Email is only available to authenticated users, and requires the SMTP to be configured within APEX Administration Services application (Administration Services), by an Instance Administrator.

d. CSV Separator - (CSV only) Define the column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.

e. CSV Enclosed By - (CSV only) Enter a delimiter character. This character delineates the starting and ending boundary of a data value. Default delimiter is double quotation marks (" ").

f. Filename - Enter a name for the downloaded document. Do not specify a file extension, the extension is added automatically. If you leave this blank, the region name is used as the file name.

g. Supplemental Text - Enter text that displays at the top of the download file. Use this attribute to for data classification, legal, or warning messages about the data in the file.

> **WARNING:**
>
> Do not use the Supplemental Text attribute with CSV download. If you enable **CSV** download, the defined text will invalidate the CSV file format.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.8 Enabling Icon View in an Interactive Grid

Learn how to enable Icon View in an interactive grid.

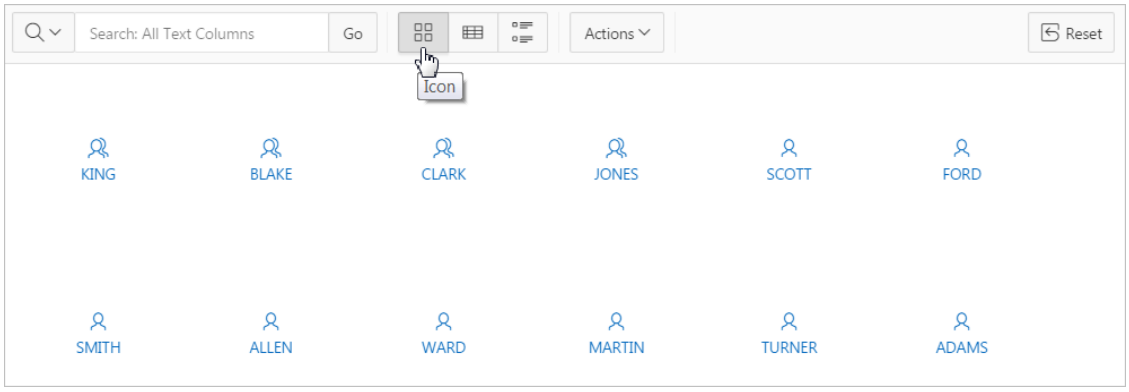
Interactive grids support different views of data. By default, interactive grids display data in a Grid view. Developers can optionally display data as icons. Enabling Icon view adds an Icon button to the select view button group.

- About Icon View in an Interactive Grid
  Learn about interactive grid Icon View.
- Enabling Icon View in an Interactive Grid
  Learn how to enable Icon View in an interactive grid.
- Adding a Custom Icon View Link to an Interactive Grid
  Learn how to add custom Icon View link in an interactive grid.

8.8.7.8.1 About Icon View in an Interactive Grid

Learn about interactive grid Icon View.

Interactive grids support different views of data. Users can switch between these views by clicking the buttons in the select view button group. The following illustration shows an interactive grid with Icon view enabled and the Icon button selected. To enable Icon view, you must identify the columns used to identify the icon, the label, and the target (that is, the link).



Viewing a Working Example of Icon View

To see a working example of this Icon view, install the sample application, *Sample Interactive Grids* and review **Reporting**, **Icon and Detail Views** .

8.8.7.8.2 Enabling Icon View in an Interactive Grid

Learn how to enable Icon View in an interactive grid.

An important requirement of icon view is that each item in the view must have the same width and height and the item content must not overflow. The width and height are set with CSS rules on elements with a class of `a-IconList-item`.

To add an Icon view to an interactive grid:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups

4. In the Property Editor Attributes tab, find the **Icon View** group and edit the following:
   a. Show - Specify whether the Icon View is enabled. Select **On**.
   b. Custom - Select **On**. Selecting **On** provides full control over the icon view item markup but also requires a full understanding of HTML markup.
   c. Icon Type - Select the column type that holds the icon image source. Select **Icon CSS Classes**. You must also add a column to your SQL statement that has a value that is a CSS class for an icon (for example, a Font APEX icon class name). As an alternative, you can also select **Image in URL** and then specify the column that contains the URL and Icon Attributes.
   d. Icon - Select the interactive grid column that returns the CSS classes for the icon image source.
   e. Link Target - Click **No Link Defined** to display the Link Builder - Link Target dialog. The link target defined is followed when the user activates the icon link with mouse or keyboard.
   f. Link Attributes - Enter HTML attributes that are associated with the display of each Link Target.
   g. Label Column - Select the report column that returns the image label.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.8.3 Adding a Custom Icon View Link to an Interactive Grid

Learn how to add custom Icon View link in an interactive grid.

A custom Icon View link provides greater flexibility in defining the content of the Icon View, including the ability to utilize more than one column.

To add custom Icon View link to an interactive grid:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor Attributes tab, find the **Icon View** group and edit the following:
   a. Show - Select **On**.
   b. Custom - Select **Off**. Selecting **On** provides full control over the icon view item markup but also requires a full understanding of HTML markup..
   c. Icon Type - Select the column type that holds the icon image source. Select **Icon CSS Classes**. You must also add a column to your SQL statement that has a value that is a CSS class for an icon (for example, a Font APEX icon class name). As an alternative, you can also select **Image in URL** and then specify the column that contains the URL and Icon Attributes.
   d. Icon - Select the interactive grid column that returns the CSS classes for the icon image source.
   e. Link Target - Click **No Link Defined** to display the Link Builder - Link Target dialog. The link target defined is followed when the user activates the icon link with mouse or keyboard.
   f. Link Attributes - Enter HTML attributes that are associated with the display of each Link Target.
   g. Label Column - Select the report column that returns the image label.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.7.9 Enabling Detail View in an Interactive Grid

Learn how to enable Detail View in an interactive grid.

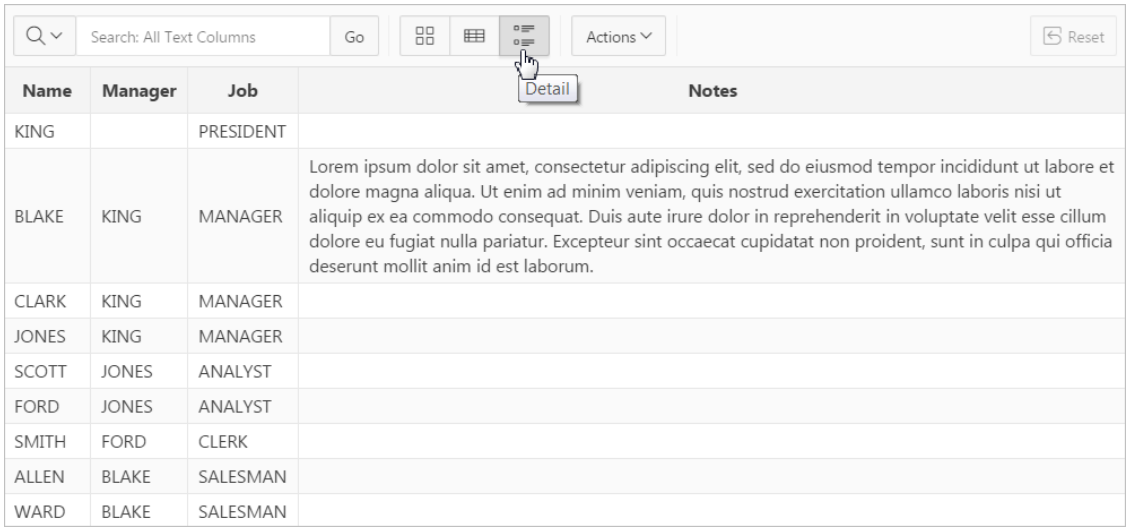
Interactive grids support different views of data. By default, interactive grids display data in a Grid view. Developers can optionally display data as in a Detail view. Enabling Detail view adds an Detail button to the select view button group.

- About Detail View in an Interactive Grid
  Learn about interactive grid Detail View.
- Adding a Detail View to an Interactive Grid
  Learn how to enable Detail View in an interactive grid.

8.8.7.9.1 About Detail View in an Interactive Grid

Learn about interactive grid Detail View.

Interactive grids support different views of data. Users can switch between these views by clicking the buttons in the select view button group. The following illustration shows an interactive grid with Detail view enabled and the Detail button selected. Detail view displays column values using developer defined HTML markup as shown in the following illustration.



8.8.7.9.2 Adding a Detail View to an Interactive Grid

Learn how to enable Detail View in an interactive grid.

To add a Detail view to an interactive grid:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive grid.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor Attributes tab, find the **Detail View** group and edit the following:
   a. Show - Select **On**.
   b. Before Rows - Enter the HTML to be displayed before report rows.
   c. For Each Row - Enter the HTML template to use for each report row in the Detail View. Use `#COLUMN_NAME#` substitution strings for column names and column labels.

      The following substitution strings are available:

      - `&COLUMN_NAME.` - substitution strings for column labels and names.
      - `&APEX$ROW_ID.` - the record identifier for the row.
      - `&APEX$ROW_INDEX.` - the record display sequence number.

      Example:

      ```
      <tr><td align="right">&ENAME_LABEL.:</td><td>&ENAME.</td></tr> <tr><td
      align="right">&JOB_LABEL.:</td><td>&JOB.</td></tr>
      ```

   d. After Rows - Enter the HTML to be displayed after report rows.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.8 Managing Interactive Grid Column Attributes

Edit column attributes to control the display, features, and behavior of interactive grid columns.

> **Note:**
>
> Some settings are configured at runtime by the report developer and saved as part of the primary or alternate report (for example, column display order, the columns the report is sorted on, and column widths. To learn more about save options, see Saving Interactive Grid Reports.

- Editing Common Grid Column Attributes
  Edit columns attributes to change the column heading, determine how a column renders, specify if users can edit a column value, and control how users can manipulate a column.
- Enabling Hide, Sort, Control Breaks, and Aggregates
  Configure the attributes in the Enable Users To group to control whether end users can hide, sort, or use a column in control breaks and aggregates.
- Defining a Column as a List of Values for Filtering
  Define an interactive grid column as a list of values to improve the speed of built-in filter tools.
- Creating a Column Link in an Interactive Grid
  Use Column Attributes to create a link from a report to another page in your application or to a URL.
- About Configuring Interactive Grids to Support Multi Select
  Learn about configuring interactive grids must to support multi select.

> **See Also:**
>
> Editing Pages in Page Designer

8.8.8.1 Editing Common Grid Column Attributes

Edit columns attributes to change the column heading, determine how a column renders, specify if users can edit a column value, and control how users can manipulate a column.

To edit interactive grid Column attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.

3. Select a column to edit.

   The Property Editor displays the attributes.

   > **Note:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Edit the appropriate attributes.
5. To edit the column heading:
   a. Find **Heading**.
   b. Heading - Enter the report column heading text.
   c. Alignment - Select the heading alignment. Note that your alignment selection must be supported by the selected report template..
   d. Alternative Label - Enter the alternative label to use in dialogs and in the Single Row View. Use an alternative label when the heading contains extra formatting, such as HTML tags, which do not display properly.
6. To control how column values display and for editable grids how the column value is edited:
   a. Find **Identification**.
   b. Type - Select how the column is rendered.

      Different settings are available depending on the selected Type. For example, for an editable grid, select the Type **Display Only** to make a column uneditable. Choose the Type **Hidden** for primary key columns that should not be seen or other columns that are only used in templates such as icon or detail view, link targets, or columns of type HTML Expression.
7. To change how users interact with column widths:
   a. Find **Layout**.
   b. Stretch - Select from the available options.

      The default option is **Use Report Setting**, which uses the **Stretch Column Width** option to determine whether or not columns automatically stretch. You can also select **Never**, which does not stretch the column even if the report setting says to, or **Always**, which stretches the column even if the report says not to.
8. To control how users can manipulate a column:
   a. Find **Enable Users To**.
   b. Sort/Control Break/Aggregate - Select whether end users can use this column in sorts, control breaks, and aggregates. Select **On** or **Off**.

   > **Tip:**
   >
   > Enabling these operations can cause performance issues, especially on large data sets or where appropriate column indexes are not available.

9. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.8.2 Enabling Hide, Sort, Control Breaks, and Aggregates

Configure the attributes in the Enable Users To group to control whether end users can hide, sort, or use a column in control breaks and aggregates.

To enable hide, sort, control breaks, and aggregates:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. In the Property Editor, find **Enable Users To**. Configure the following:
   a. Sort - Select **On** to enable end users can use this column in sorts.
   b. Control Break/Aggregate - Select **On** to enable end users can use the column in control breaks and aggregates. This options only displays if **Sort** is enabled.
   c. Hide - Select **On** to enable end users yo hide this column.

   Be aware that enabling **Sort** or **Control Break/Aggregate** operations can lead to performance issues, especially on large data sets or where appropriate column indexes are not available.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.8.3 Defining a Column as a List of Values for Filtering

Define an interactive grid column as a list of values to improve the speed of built-in filter tools.

To define a report column as a list of values:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. In the Property Editor, find the **Column Filter** group. Specify the following:

a. Enabled - Specify whether end users can filter this column. Select **On** or **Off**.

b. LOV Type - Select how to derive the list of values for the filter when a user clicks on the column header or in filter dialogs. Options include:

- **None** - Disable any values being displayed in the filter. This option is appropriate on large columns, generally with most values being over 50 characters, or when a list would not be very usable for selecting a specific record. Instead the end user can just type in a partial value and then view the results.
- **Distinct Column** - Distinct column values are displayed for VARCHAR2 and NUMBER columns. DATE and TIMESTAMP columns utilize the built-in predefined date ranges.
- **SQL Query** - Values are derived from the SQL query definition.
- Shared Component - Values are derived from an existing list of values.
- **Static Values** - Values are derived from a static list of values.
- **PL/SQL Function Body returning SQL Query**- Values are derived from the PL/SQL function body that returns a SQL Query.
- **Use List of Values** - Values are derived from list of values.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.8.4 Creating a Column Link in an Interactive Grid

Use Column Attributes to create a link from a report to another page in your application or to a URL.

To create a column link to another page:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.

3. Select the desired column.

   The Column attributes appear in the Property Editor. Attributes are organized in groups.

   > **Tip:**
   >
   > To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, click **Go to Group** and select the group.

4. Find **Identification**. From Type, select **Link**.

5. Find **Link** and click **No Link Defined**.

   The Link Builder - Target dialog appears. The Target Type you select determines the steps that follow.

6. To create a link to another page, in the Link Builder - Target dialog:

a. Type - Select **Page in this application**.
b. Page - Specify the target page number.
c. Set Items - Select a Name and Value to specify session state for an item.
d. Clear/Reset, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
e. Clear/Reset, Action - Select one of the following:
   - **None** - No clear or reset action when page loads.
   - **Clear Regions** - Clears all region settings on the page.
   - **Reset Regions** - Resets all regions on the page back to the default state, including sort orders, hidden columns and so on.
   - **Reset Pagination** - This option enables backwards compatibility.
f. Advanced, Request - Specify the request to be used.
g. Click **OK**.

7. To create a link to another page in another application, in the Link Builder - Target dialog:
   a. Type - Select **Page in a different application**.
   b. Application - Select the application.
   c. Page - Specify the target page number.
   d. Set Items - Select a Name and Value to specify session state for an item.
   e. Clear/Reset, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
   f. Clear/Reset, Action - Select one of the following:
      - **None** - No clear or reset action when page loads.
      - **Clear Regions** - Clears all region settings on the page.
      - **Reset Regions** - Resets all regions on the page, including sort orders, hidden columns and so on, back to default state.
      - **Reset Pagination** - This option is for backwards compatibility.
   g. Advanced, Request - Specify the request to be used.
   h. Click **OK**.
8. To create a link to a URL, in the Link Builder - Target dialog:
   a. Type - Select **URL**.
   b. URL - Enter the URL address.
   c. Click **OK**.
9. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.8.5 About Configuring Interactive Grids to Support Multi Select

Learn about configuring interactive grids must to support multi select.

Interactive grids must be configured to support multi select. When multi select is enabled, users can use arrow keys to move around an entire grid, use the Space key to select, or use Ctrl+Space to toggle selections.

To configure an interactive grid to support multi-select:

- Editable interactive grid - Configure the `APEX$ROW_SELECTOR` column which is automatically created for you when you make an interactive grid editable. Under **Settings**, set the **Enable Multi Select** attribute to **Yes** (the default).
- Non-editable interactive grid - Use an advanced JavaScript configuration. To see an example, install the *Sample Interactive Grids* sample application, and see the **Multiple Selection** example under **Reporting**.

> **See Also:**
>
> - Making an Existing Interactive Grid Editable
> - Installing Apps from the Gallery

8.8.9 About Customizing Interactive Grids in a Running Application

When running an application, users can alter the report layout using Column menus and sort options, rearrange columns by dragging and dropping, and change how data displays using options on the Actions menu.

When an interactive grid is editable, users can also edit the underlying data, add rows, delete rows, and refresh rows. After users customize an interactive grid they can save it. However, the available save capabilities differ depending upon the user type.

> **See Also:**
>
> - Making an Existing Interactive Grid Editable
> - Saving Interactive Grid Reports
> - *Oracle APEX End User's Guide*

8.8.10 Saving Interactive Grid Reports

After users customize an interactive grid they can save it as either a Private or Public report.

Only the user who creates a **Private** report can view, save, rename, or delete it. Report developers have additional save capabilities in that they can save the **Primary** report (that is, the report that displays to all users), or create an **Alternative** report to test multiple report layouts.

- How the User Type Effects Available Save Options
  Learn how user type determines the save capabilities available in an interactive grid.
- About Configuration Dependencies When Saving Interactive Grid Reports
  Learn about configuration dependencies when saving interactive grid reports.
- Saving a Private Interactive Grid Report
  Save a private interactive grid report.
- Saving a Public Interactive Grid Report
  Save a public interactive grid report.

- Saving or Renaming a Primary Interactive Grid Report
  Developers can save or rename a primary interactive grid report.
- Saving an Alternative Interactive Grid Report
  Alternative reports enable a developers to optionally define multiple report layouts, which are available to all users (subject to any authorization scheme, if defined).
- Renaming a Saved Interactive Grid Report
  All users can rename the interactive grid reports they create.
- Deleting a Saved Interactive Grid Report
  All users can delete the interactive grid reports they create.
- Resetting a Saved Interactive Grid Report
  All users can reset an interactive grid report to the last saved version.

8.8.10.1 How the User Type Effects Available Save Options

Learn how user type determines the save capabilities available in an interactive grid.

After users customize an interactive grid they can save it. However, the available save capabilities differ depending upon the user type.

End Users

End users can save an interactive grid as follows:

- **Private** report. Only the user who creates a Private report can view, save, rename, or delete it. All users (that is, end users and developers) can create interactive grid Private reports.
- **Public** report. All users can view a Public report. However, only the user who creates a Public report can save, rename, or delete it. All other users can view a Public report and save it under a new report name.

> **Tip:**
>
> End users can only save Public reports if the report developer has enabled that capability. See Enabling Users to Save Interactive Grid Public Reports.

Developers

In addition to Private and Public reports, a developer can save two other report formats:

- **Primary** report. The Primary report displays to all users. Only a developer can save a new Primary report, rename it, or delete it.
- **Alternative** report. An Alternative report enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative report.

8.8.10.2 About Configuration Dependencies When Saving Interactive Grid Reports

Learn about configuration dependencies when saving interactive grid reports.

To save a customized version of an interactive grid, users select the Actions menu and then click **Report** and then **Save** or **Save As**. End users can only access Save options if the application developer defines an authentication scheme at the application-level.

Other configuration issues that impact a user's ability to save an interactive grid include:

- **Interactive Grid Toolbar Availability**. Developers can edit the interactive grid Attribute, **Toolbar** to determine if an interactive grid includes a toolbar. If the toolbar is disabled, all the options on the Actions menu, including Save and Save As, are unavailable.
- **Support for Saving Public Reports**. By default, end users cannot save Public reports. To enable support for Public reports, developers edit the report Attribute, **Enables Users To**, **Save Public Report**.

When creating a public report, Oracle suggests you:

1. Configure the **Stretch** settings for each column. See Editing Common Grid Column Attributes.
2. Run the page.
3. Adjust all columns to their smallest reasonable maximum width.
4. Save the report settings.

See Also:

- Establishing User Identity Through Authentication
- Customizing the Interactive Grid Toolbar
- Enabling Users to Save Interactive Grid Public Reports

8.8.10.3 Saving a Private Interactive Grid Report

Save a private interactive grid report.

All users (that is, both end users and developers) can save a Private interactive grid. Only the user who creates a Private interactive grid can view, save, rename, or delete it.

To save a Private interactive grid:

1. Run the interactive grid.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).
3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
5. In the Report - Save As dialog:
   a. Type - Select **Private**.
   b. Name - Enter a name for the report.
   c. Click **Save**.

      Private reports display on the Saved Reports list on the toolbar under the heading, Private.

8.8.10.4 Saving a Public Interactive Grid Report

Save a public interactive grid report.

All users (that is, both end users and developers) can save a Public interactive grid. Every user who has privileges to save public reports can modify any public report or make their own copy of it using the Save As option.

> **Note:**
>
> End users can only save Public interactive grids if the report developer has enabled that capability. See Enabling Users to Save Interactive Grid Public Reports.

To save a Public interactive grid:

1. Run the interactive grid.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).
3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
5. In the Report - Save As dialog:
   a. Type - Select **Public**.
   b. Name - Enter a name for the report.
   c. Click **Save**.

      Public reports display on the Saved Reports list on the toolbar under the heading, Public.

8.8.10.5 Saving or Renaming a Primary Interactive Grid Report

Developers can save or rename a primary interactive grid report.

A Primary interactive grid displays to all users. Only developers can save or rename Primary reports. It is impossible to delete a primary report.

To save or rename a Primary interactive grid:

1. Run the interactive grid as a developer.
2. To customize and then save a primary interactive grid:
   a. Customize the report (for example, show, hide, or move columns, create filters, and so on).
   b. Click the **Actions** menu.
   c. Select **Report** and then **Save**.
3. To rename a primary report:
   a. Click the **Actions** menu.
   b. Select **Report** and then **Edit**.

      The Report - Edit dialog appears.
   c. In Name, enter a new name and click **Save**.
4. Select **Report** and then **Save**.

   If multiple reports have been redefined, the Primary report displays on the Saved Reports list on the toolbar under the heading, Default.

8.8.10.6 Saving an Alternative Interactive Grid Report

Alternative reports enable a developers to optionally define multiple report layouts, which are available to all users (subject to any authorization scheme, if defined).

To save an Alternative interactive grid:

1. Run the interactive grid as a developer.
2. Customize the report (for example, show, hide, or move columns, create filters, and so on).
3. Click the **Actions** menu.
4. Select **Report** and then **Save As**.
   - Type - Select **Alternative**.
   - Name - Enter a name for the report.
   - Authorization - Select an alternative authorization scheme to restrict who can view this report.
   - Click **Save**.

   An Alternative report displays on the Saved Reports list on the toolbar under the heading, Default.

8.8.10.7 Renaming a Saved Interactive Grid Report

All users can rename the interactive grid reports they create.

To rename an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.
3. Click the **Actions** menu and then select **Report** and then **Edit**.
4. In the Report - Edit dialog:
   a. Type - Select the type of report to be edited.
   b. Name - Enter a new name.
   c. Click **Save**.

   The new report name displays in the Saved Reports list on the toolbar.

8.8.10.8 Deleting a Saved Interactive Grid Report

All users can delete the interactive grid reports they create.

To delete an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.
3. Click the **Actions** menu and then select **Report** and then **Delete**.

   A confirmation dialog appears.
4. Confirm your request and click **OK**.

   The report disappears from the Saved Reports list on the toolbar.

8.8.10.9 Resetting a Saved Interactive Grid Report

All users can reset an interactive grid report to the last saved version.

To reset an interactive grid report:

1. Run the interactive grid.
2. From the Saved Report list on the toolbar, select the report.
3. Customize the report (for example, show, hide, or move columns, create filters, and so on).
4. Reset the report to the last saved version. Click the **Actions** menu and then select **Report** and then **Reset**.

> **Tip:**
>
> Alternatively, you can also click the **Reset** button on the toolbar.
>
> The revised report appears.

8.8.11 Linking to Interactive Grid Reports

Developers can use the syntax in the `Request` value and `ItemNames` section of the URL to link to interactive grid report.

Data in an interactive grid is driven by declarative filters. Developer can set report parameters by modifying interactive grid report settings for an end user. However, these settings only affect the end user's working report and do not affect any saved report settings. Developers can use the syntax described in this section to link to interactive grid reports.

> **Note:**
>
> When you add additional interactive grid report regions, be sure to change the old single interactive grid report link syntax. Otherwise, the link generates an error. A link to a page with more than one interactive grid report region, the request or name syntax needs to fully qualify with region static ID. To open a link to an interactive grid report from outside of an application (for example, from an email), you may need to enable the Session Management, Deep Linking attribute in your application. See Session Management.

- Interactive Grid Request Syntax
  Learn about interactive grid request syntax.
- Interactive Grid Name and Value Syntax
  Learn about interactive grid name and value syntax.
- Interactive Grid ClearCache Syntax
  Learn about interactive grid clearcache syntax.
- Restrictions When Linking to Interactive Grid Reports
  Learn about restrictions when linking to interactive grids.
- Valid Interactive Grid Linking Examples
  View valid linking examples.
- Invalid Interactive Grid Linking Examples
  View invalid linking examples.
- Linking to Shared Interactive Grid Reports
  learn about linking to shared interactive grid reports.

8.8.11.1 Interactive Grid Request Syntax

Learn about interactive grid request syntax.

To link to a specific saved interactive grid report, use the following `Request` syntax:

```
IG[region static ID]_<report_alias>
```

Consider the following example:

```
IG[EMP]_dept20
```

> **Tip:**
>
> The use of `Request` in the URL is optional. To link to multiple saved reports, separate each `Request` with comma.

Passing Values for DATE and TIMESTAMP Columns

For interactive grids, the value for `DATE` or `TIMESTAMP` columns must be passed in canonical format without blanks or separators. Use the following syntax:

```
YYYYMMDDHH24MISS
```

Consider the following example:

```
IG[EMP]EQ_HIREDATE = 19811117000000
```

8.8.11.2 Interactive Grid Name and Value Syntax

Learn about interactive grid name and value syntax.

To create a filter, use the following `itemNames` and `itemValues` syntax:

```
IG[region static ID]<operator>_<target column alias>
```

Consider the following example:

```
IG[EMP]C_ENAME:KING
```

> **Note:**
>
> If there is only one interactive grid report region on the page, the `itemName` does not need to be fully qualified with the region static ID. If an operator is not defined, a filter with equal operator is added.

To create a filter on an interactive grid report in a link, use the `itemNames` syntax described previously and add the filter value in the corresponding location in the item names section of the URL.

To create row text which contains a filter on an interactive grid, use `ROWFILTER` as the target column alias in the item name syntax and add a filter value in the item `itemValues` of the URL. The link will add a Row Text Contains filter which searches all columns displayed in the report with type `STRING` or `NUMBER`.

Consider the following example:

```
IG[EMP]_ROWFILTER:SALES
```

Valid operators include:

- `C` = Contains
- `EQ` = Equals (this is the default)
- `GTE` = Greater than or equal to
- `GT` = Greater Than
- `LT` = Less than
- `LTE` = Less than or equal to
- `N` = Null
- `NC` = Not Contains
- `NEQ` = Not Equals
- `NN` = Not Null
- `NIN` = Not In (escape the comma separated values with a leading and trailing backslash, \)
- `IN` = In (escape the comma separated values with a leading and trailing backslash, \)
- `ROWFILTER` = Row Text Contains (this searches all columns displayed in the report with type STRING or NUMBER)

8.8.11.3 Interactive Grid ClearCache Syntax

Learn about interactive grid clearcache syntax.

> **Tip:**
>
> Using `ClearCache` without specifying a saved report alias in the `Request`, clears the primary default report. To clear a specific interactive grid report region on a page with more than one interactive grid reports, specify the region static ID along with saved report alias in the `Request`.

Use the following syntax in the `ClearCache` section of the URL to reset or clear pagination in an interactive grid report:

- `RR` - Resets an interactive grid report. This is equivalent to the end user choosing the Reset option from the interactive grid report Actions menu on the target page. The report is returned to the default report settings specified by the developer or saved by the user.

- `CR` - Clears interactive grid report. This clears all of the report filters.

8.8.11.4 Restrictions When Linking to Interactive Grid Reports

Learn about restrictions when linking to interactive grids.

When linking to interactive grid reports, remember the following restrictions:

1. If a `Request` value is not defined in the URL, all of the primary default report settings will change.
2. If a `Request` value is not defined and `itemNames` section does not include a fully qualified filter column name with region static ID, an error occurs if there are more than one interactive grid report in the target page.
3. If a `Request` value is not defined and `itemNames` section do not include a fully qualified filter column name with region static ID, it will create a filter on the primary default report if there is only one interactive grid report region in the target page.

8.8.11.5 Valid Interactive Grid Linking Examples

View valid linking examples.

Valid Linking Example 1

This example includes a link with no `Request` value, but includes a qualified filter column with region static ID in the `itemNames` section of the URL. This example creates a filter on the primary default report in the referenced region item name.

```
f?p=&APP_ID.:2:&SESSION.::::IG[EMP]_ENAME:KING
```

Valid Linking Example 2

In this example the link has no `Request` value. The clear cache section has `RR` or `CR` and will reset, or clear the primary default reports of all interactive grid report regions.

```
f?p=&APP_ID.:1:&SESSION.:::RR:
```

8.8.11.6 Invalid Interactive Grid Linking Examples

View invalid linking examples.

Invalid Linking Example 1

This link has no `Request` value and does not include a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error if there is more than one interactive grid report region in the target page.

```
f?p=&APP_ID.:2:&SESSION.::::ENAME:KING
```

Invalid Linking Example 2

This example has a link with two request values separated by commas and does not contain a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error since it is not clear where the filter needs to be created.

```
f?p=&APP_ID.:2:&SESSION.:IG[test]_employee,IG[test2]_dept10:::ENAME:KING
```

8.8.11.7 Linking to Shared Interactive Grid Reports

learn about linking to shared interactive grid reports.

You can link to saved primary default, alternative default, and public reports using `IG[region static ID]_<report_alias>` in the `Request` value of the URL.

- Editing the Interactive Grid Saved Report Alias
  Edit the Alias for a saved reports to create a more meaningful URL link to the saved report.
- Link Examples to a Primary Report
  View examples that demonstrate how to link to a primary report.
- Link Examples to Saved Reports
  View examples that demonstrate how to link to a saved report.

> **See Also:**
>
> - About Using f?p Syntax to Link Pages
> - Saving Interactive Grid Reports
> - Deleting a Saved Interactive Grid Report

8.8.11.7.1 Editing the Interactive Grid Saved Report Alias

Edit the Alias for a saved reports to create a more meaningful URL link to the saved report.

To edit the Alias for saved reports:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, find the region containing the report.
3. Under the region, expand **Saved Reports** and select the saved report.

   The Property Editor displays the attributes for the Saved Report.
4. **Identification**, **Alias** - Enter a new alias for the report. Use this alias to define a more meaningful URL link to this saved report. This link can be used to link directly from elsewhere in the application.

5. **Link**, **Link Example** - View the URL link for this saved report. This attribute is view only and cannot be updated. Use this attribute to quickly copy the displayed URL. This URL can be used to link directly to this particular saved report from anywhere else in the application.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.8.11.7.2 Link Examples to a Primary Report

View examples that demonstrate how to link to a primary report.

The following examples demonstrate how to link to a primary report (`report_alias=PRIMARY`):

This example links to a primary report:

```
f?p=100:1:&APP_SESSION.:IG_PRIMARY
```

This example links, resets, and clears primary report settings:

```
f?p=100:1:&APP_SESSION.:IG_PRIMARY::RR,CR:
```

This example links, resets, and clears primary report settings. It additionally creates a `ENAME = 'KING'` filter on the primary report:

```
f?p=100:1:&APP_SESSION.:IG_PRIMARY::RR,CR:IG_ENAME:KING
```

8.8.11.7.3 Link Examples to Saved Reports

View examples that demonstrate how to link to a saved report.

The following examples demonstrate how to link to a saved report (`report_alias=12345`):

This example links to a saved report:

```
f?p=100:1:&SESSION.:IG_12345
```

This example links, resets, and clears settings for saved report `12345`:

```
f?p=100:1:&SESSION.:IG_12345::RR,CR:
```

This example links, resets, and clears saved report `12345` settings. It additionally creates a `ENAME = 'KING'` filter on saved report `12345`.

```
f?p=100:1:&SESSION.:IG_12345::RR,CR:IG_ENAME:KING
```

8.9 Managing Interactive Reports

An interactive report page features a searchable, easily customizable report.

- Example Interactive Report
  View an interactive report example.
- Supported Columns in Interactive Reports
  Learn about column types supported by interactive reports.
- Creating an Interactive Report Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains one or multiple interactive report pages.
- Creating an Interactive Report Using the Create Page Wizard
  Run the Create Page Wizard to add a new interactive report page to an existing application.

- Managing Interactive Report Region Attributes
  Edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- Managing Interactive Report Attributes
  Edit report Attributes to control how an interactive report works.
- Managing Interactive Report Column Attributes
  Edit column attributes to control the display, features, and behavior of interactive report columns.
- Customizing Interactive Reports in a Running Application
  When running an application, end users and developers can alter the layout of interactive reports data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations.
- Linking to Interactive Reports
  Link to interactive reports using the syntax in the `Request` value and `ItemNames` section of the URL.

> **See Also:**
>
> Using Interactive Reports in *Oracle APEX End User's Guide*

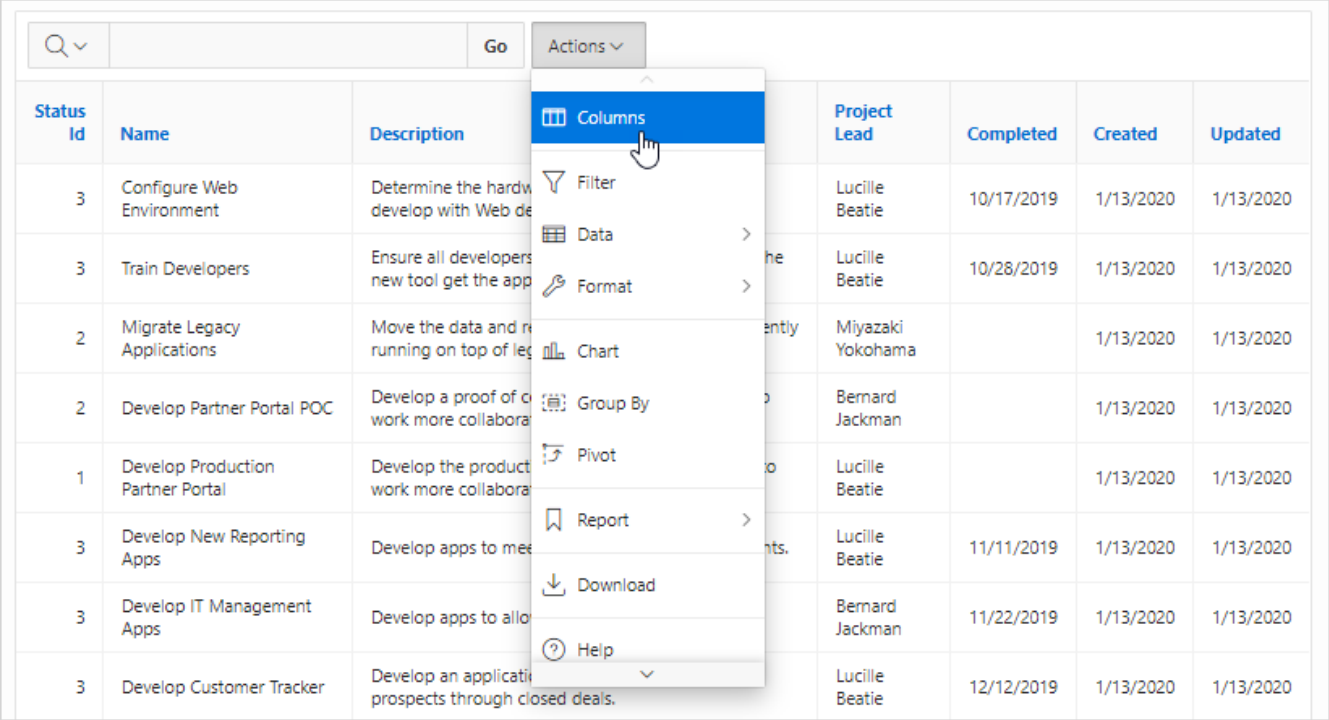
8.9.1 Example Interactive Report

View an interactive report example.

An interactive report is a formatted result of a SQL query. End users can customize the both report layout and control how the data that displays. Both the Create Application Wizard and Create Page Wizard support the creation of interactive reports. You choose a table on which to build a report, or provide a custom SQL SELECT statement. Interactive reports are only supported for Desktop applications.

End users can customize the report layout and data displayed by selecting options on the Actions menu. For example, end users can alter the report layout by hiding or exposing specific columns and applying filters, highlighting, and sorting. Advanced end users can also define breaks, aggregations, charts, group data, and add computations. Once customized, the report can be saved as either a private or public report.

The following is an example of an interactive report.



Developers can include multiple interactive reports per page and can restrict the capabilities available to end users (such as disabling download or support for hiding column). When the end user views the report, report functionality is the same across all reports in the application.

When viewing an interactive report, end users can customize how and what data displays. By default, interactive reports include a search bar, an Actions menu, column heading menus, and Edit icons in the first column of each row. Using options on the Actions menu, users can alter the report layout by hiding or exposing specific columns and applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations. Once customized, the report can be saved as either a private or public report.

Tip:

To view an example, install the sample app, *Sample Reporting*. To learn more, see Installing Apps from the Gallery.

See Also:

Using Interactive Reports in *Oracle APEX End User's Guide*

8.9.2 Supported Columns in Interactive Reports

Learn about column types supported by interactive reports.

Interactive reports support the following column types. Column types can be categorized as **Built-in Components** and **Theme Components**. **Theme Components** are Universal Theme Template Component plug-ins installed in the application's Shared Components. To view the column type in Page Designer, select the column and find the **Identification**, **Type** attribute in the Property Editor.

> **Note:**
>
> You can also view this information in Page Designer Help. View the page containing the report in Page Designer. Select a column and then **Identification**, **Type** attribute. Click the **Help** tab in the central pane.

Column Type	Description	Category
Display Image	Displays the column as an image.	Built-in Components
Download BLOB	Displays the column as a link to download the file.	Built-in Components
Hidden	Does not display the column, but it is referenceable.	Built-in Components
Link	Displays the column as a link.	Built-in Components
Percent Graph	Displays the column as a percentage graph. The value retrieved must be between 0 and 100.	Built-in Components
Plain Text	Displays the column value.	Built-in Components
Plain Text (based on List of Values)	Displays the value returned from the list of values specified.	Built-in Components
Remove HTML	Displays the column value, after removing HTML tags and escaping special characters.	Built-in Components
Rich Text	Renders the column using HTML or Markdown.	Built-in Components
Avatar	Display an image, icon, or initials. Available as a partial template to display a single row.	Theme Components
Badge	Display content within a badge. Available for multiple report rows or a single row.	Theme Components
Comments	Display user comments and status updates. Supports avatars. Available for a single row or as a report with multiple rows.	Theme Components
Content Row	Display content in a formatted row with a title, description, and more. Supports avatars and badges. Available for a single row or as a report with multiple rows.	Theme Components

Column Type	Description	Category
Media List	Display report content in a formatted media list. Supports avatars and badges. Available for a single row or as a report with multiple rows.	Theme Components
Timeline	Display a series of events. Supports avatars and badges. Available for a single row or as a report with multiple rows.	Theme Components

8.9.3 Creating an Interactive Report Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains one or multiple interactive report pages.

To create an application that contains one or multiple interactive report pages:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Interactive Report**.
8. On Add Report Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Select the source for the report:
      - **Table or View**
      - **SQL Query**

      What you select determines what displays next. Follow the on-screen instructions. To learn more about an attribute, click the Help icon in the lower left corner.
   d. Select the report type **Interactive Report**.
   e. Include Form - Select **Include Form** to include a form page for creating or updating records.
   f. Lookup Columns - Expand to see attributes for defining a lookup to another table. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number. Specify the following:
      - Lookup Key - Select the column to change to output the display column.
      - Display Column - Select the column (from the lookup table) to be displayed.

> **Tip:**
>
> **Lookup Columns** only appear if the current page is based on a table and has foreign key constraints to another table.

g. Advanced - Expand **Advanced** to configure these options:
   - Set as Home Page - Enable this to make this page the home page for the application.
   - Set as Administration Page - Enable this to have this page display under Administration.
   - Page Help - Enter text to be displayed when the user selects Page Help.

h. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

> **See Also:**
>
> Creating a Report Using the Create Page Wizard

8.9.4 Creating an Interactive Report Using the Create Page Wizard

Run the Create Page Wizard to add a new interactive report page to an existing application.

To create an interactive report page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Interactive Report**.
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

> **Tip:**
>
> This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

c. Page Mode - Identify the page mode. To learn more, see field-level Help.

d. Include Form Page - Enable this flag to create a form page along with the report page.

   Form pages are always created as Modal Dialog pages. Enabling this flag displays additional attributes. To learn more about an attribute, see field-level Help.

6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table /View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

9. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

8.9.5 Managing Interactive Report Region Attributes

Edit region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

- Editing Interactive Report Region Attributes
  Edit interactive report region attributes in Page Designer.
- Displaying Interactive Reports Conditionally
  Display interactive reports conditionally by editing region attributes.

> **See Also:**
>
> Editing Pages in Page Designer

8.9.5.1 Editing Interactive Report Region Attributes

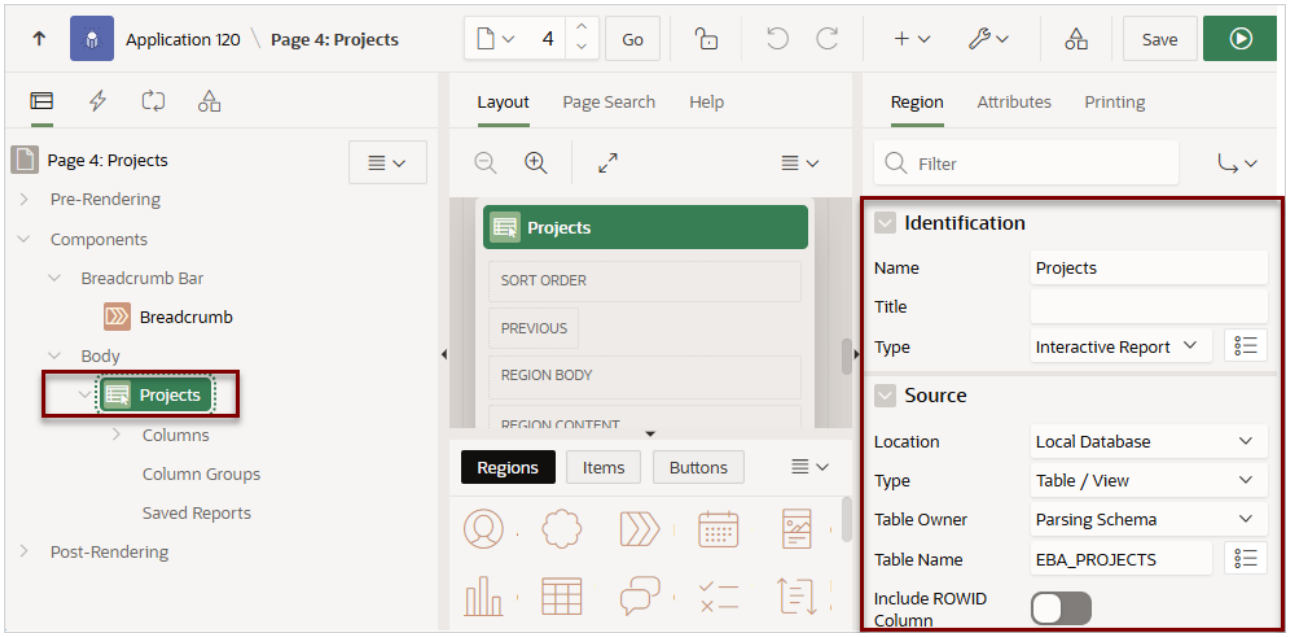
Edit interactive report region attributes in Page Designer.

To edit interactive report region attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

   In the Property Editor, the **Region** tab displays the attributes which are organized into groups.



3. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. Edit the appropriate attributes.

> **Tip:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

8.9.5.2 Displaying Interactive Reports Conditionally

Display interactive reports conditionally by editing region attributes.

To display an interactive report conditionally:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region.

   The Property Editor displays the **Region** tab. Attributes are organized in groups.

3. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
4. Configure **Server-side Condition**:
   a. Find the **Server-side Condition** group.
   b. **Type** - Select a condition type and follow the on-screen instructions
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Tip:**
>
> Asynchronous JavaScript and XML (Ajax) is used throughout interactive reports. Because Ajax is asynchronous, the value evaluated for conditional display must be in the session. To display an interactive report conditionally based on specified value, use a page item instead of using a `:REQUEST` value. Using `:REQUEST` in a conditional display causes the interactive report to function incorrectly.

8.9.6 Managing Interactive Report Attributes

Edit report Attributes to control how an interactive report works.

- Editing Interactive Report Attributes
  Edit interactive report Attributes in Page Designer.
- Managing Link Columns
  A Link Column displays on the left side of an interactive report. Configure a Link Column to link to a single row view, a custom target, or be excluded from the report.
- Customizing an Interactive Report Search Bar
  Customize or remove an interactive report search bar.
- Customizing the Interactive Report Actions Menu
  Customize or disable an interactive report Actions menu.
- Configuring Interactive Report Download Options
  Configure interactive report download options available on the Actions menu..
- Controlling Interactive Report Pagination
  Control interactive report pagination.
- Enabling Icon View
  Learn about interactive report Icon View.
- Enabling Detail View
  Add Details view to an interactive report.
- Configuring Advanced Attributes for Interactive Reports
  Configure advanced interactive report attributes to define the region alias, specify an item whose value stores the saved report ID, and define a from address in an interactive report subscription or email download

> **See Also:**
>
> Editing Pages in Page Designer

8.9.6.1 Editing Interactive Report Attributes

Edit interactive report Attributes in Page Designer.

To edit interactive report Attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region containing the interactive report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. Edit the appropriate attributes.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

8.9.6.2 Managing Link Columns

A Link Column displays on the left side of an interactive report. Configure a Link Column to link to a single row view, a custom target, or be excluded from the report.

> **Tip:**
>
> A Link Column cannot be sorted, hidden, or moved by an end user and does not interact with the standard column link defined on the Column Attributes page.

- Linking to a Single Row View
  Configure a link column to link to a single row view.
- Linking to a Specific Page
  Configure a link column to link to a specific page.
- Linking to a URL
  Configure a link column to link to a URL.
- Excluding a Link Column
  Exclude a Link Column from a report.

8.9.6.2.1 Linking to a Single Row View

Configure a link column to link to a single row view.

To link to a single row view in an interactive report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find the **Link** group and edit the following:
   a. **Link Column** - Select **Link to Single Row View**.
   b. **Uniquely Identify Rows by** - Select **ROWID** or **Unique Column**.

      If you select, **Unique Column**, specify the column in the Unique Column field.
   c. **Link Icon** - Accept the default or specify the path to another icon.
   d. **Link Attributes** - Optionally specify additional column link attributes to be included in the `<a href= >` tag (for example, a link target, classes, or styles).
   e. **Authorization Scheme** - Select an authorization scheme to associate with the Link Column. This authorization scheme must evaluate to TRUE in order for the Link Column to be rendered in the report.
   f. **Condition Type** - Select a condition type from the list that must be met in order for this link to be rendered.
5. Find the **Single Row View** group and edit the following:
   a. **Exclude Null Values** - Specify whether null columns are hidden on the Single Record View
   b. **Only Displayed Columns** - Specify whether only those columns currently displayed are displayed in the Single Record View. If you select **Off**, then hidden report columns may also be shown on the Single Record View.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Tip:**
>
> If an interactive report links to single row view and the link icon is missing `ALT` text or has empty `ALT` text, the `alt="Single Row View"` gets included automatically to improve accessibility. This can be further improved by making the `ALT` text something specific to the row, for example `alt="View Row: #COLUMN_NAME#"` (where `COLUMN_NAME` should be a column that helps uniquely identify the current row).

8.9.6.2.2 Linking to a Specific Page

Configure a link column to link to a specific page.

To link to a specific page:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, locate the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find the **Link** group and edit the following:
   a. **Link Column** - Select **Link to Custom Target**.
   b. **Target** - Click **No Link Defined**.

      The Link Builder - Target dialog appears.
      - Target Type - Select **Page in this Application**.
      - Page - Select the target page number.
      - Set Items - Select a **Name** and **Value** to specify session state for an item.
      - Clear/Reset, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
      - Clear/Reset, Action - Select one of the following:
        – **None** - No clear or reset action when page loads.
        – **Clear Regions** - Clears all region settings on the page.
        – **Reset Regions** - Resets all regions on the page back to the default state, including sort orders, hidden columns and so on.
        – **Reset Pagination** - This option enables backwards compatibility.
      - Advanced, Request - Specify the request to be used.
      - Click **OK**.
   c. **Link Icon** - Enter the HTML to be used for the link. Use an image tag to display images, or pick from the list of default images. If not defined,`#APEX_FILES#ws/small_page.gif` is used.
   d. **Link Attributes** (Optional) - Enter additional column link attributes to be included in the `<a href= >` tag (for example, a link target, classes, or styles).

      To view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.2.3 Linking to a URL

Configure a link column to link to a URL.

To link to a URL:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find the **Link** group and edit the following:
   a. **Link Column** - Select **Link to Custom Target**.
   b. **Target** - Click the **No Link Defined**.

      The Link Builder - Target dialog appears.

      - Target Type - Select **URL**.
      - Click **OK**.
   c. **Link Icon** - Enter the HTML to be used for the link. Use an image tag to display images, or pick from the list of default images. If not defined, `#APEX_FILES#ws/small_page.gif` is used.
   d. **Link Attributes** (Optional) - Enter additional column link attributes to be included in the `<a href= >` tag (for example, a link target, classes, or styles).

      To view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.2.4 Excluding a Link Column

Exclude a Link Column from a report.

To exclude a column link:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Configure Link attributes:
   a. Find the **Link** group.
   b. **Link Column** - Select **Exclude Link Column**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.3 Customizing an Interactive Report Search Bar

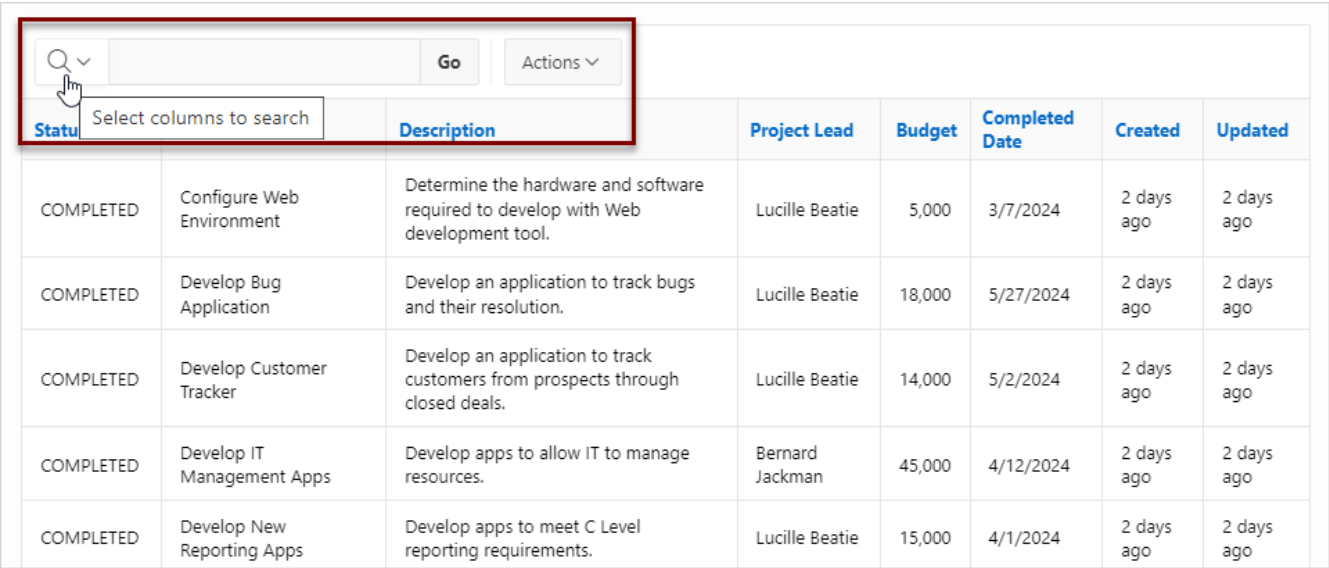
Customize or remove an interactive report search bar.

- Interactive Report Search Bar
  Learn about the interactive report search bar.
- Customizing the Search Bar
  Customize an interactive report search bar.
- Removing the Interactive Report Search Bar
  Disable an interactive report search bar.

8.9.6.3.1 Interactive Report Search Bar

Learn about the interactive report search bar.

All interactive reports include a search bar at the top of the page which includes the following controls: a Select columns to search button, Text area, Go button, and Actions menu.



If users have saved multiple reports, a saved report select list displays to the right of the **Go** button. Users can use this list to select a specific saved report. If multiple views are configured, a view selection button group also displays which enables the end user to switch views. You can remove or customize the search bar in Page Designer by configuring report attributes in the Property Editor.

> **See Also:**
>
> Customizing the Search Bar and Removing the Interactive Report Search Bar

8.9.6.3.2 Customizing the Search Bar

Customize an interactive report search bar.

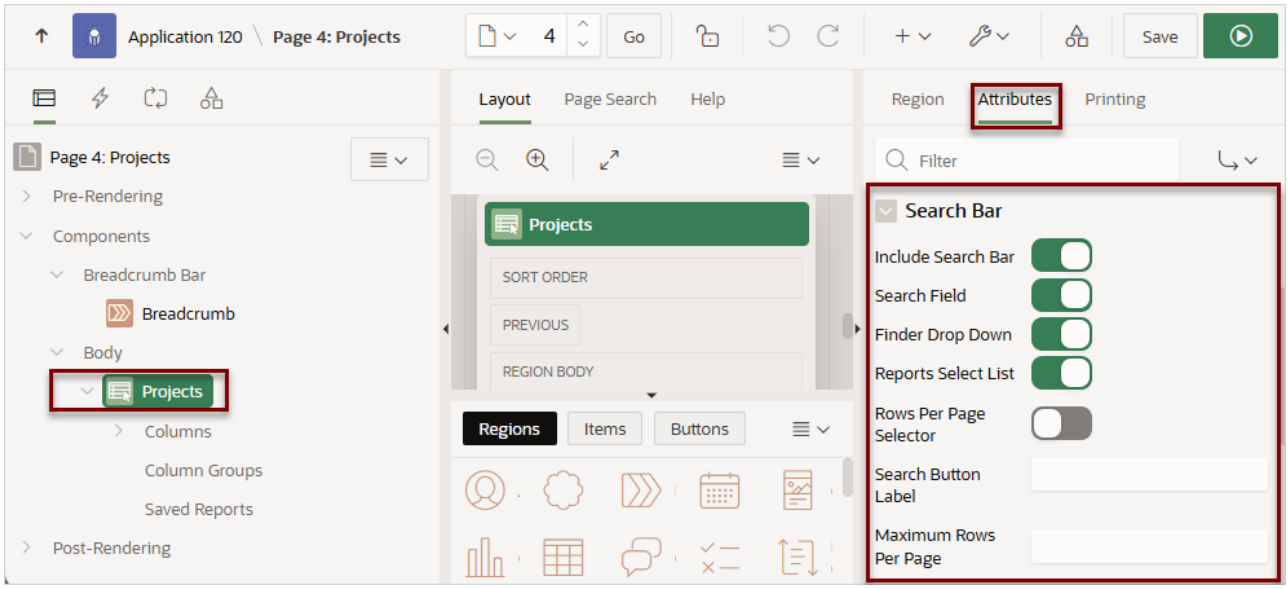
To customize the search bar:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor, find the **Search Bar** group.



5. Under **Search Bar**, determine which options to include by selecting **On** or **Off**:
   - **Include Search Bar** - Select **On** to include a search bar above the report. If you include the search bar, you can also fully customize which functions to display. Select **Off** to not include the Search bar.
   - **Search Field** - Specify whether the search field is included within the search bar.
   - **Finder Drop Down** - Specify whether the Finder drop down is included at the beginning of the Search field within the search bar. The Finder drop down is a list of columns that enables users to limit their search to a specific column. If a column is selected then the search is only performed against the values in the selected column. Within a running interactive report, this option is identified as **Select columns to search**.
   - **Reports Select List** - Specify whether the Reports select list displays in the search bar. The Reports select list only displays if a report, other than the primary report, is accessible to the user.
   - **Rows Per Page Selector** - Specify whether a separate drop down row selector is included within the search bar. Do not select both this attribute and **Rows Per Page**.
   - **Search Button Label** - Enter the text for the search button label.
   - **Maximum Rows Per Page** - Enter the maximum number to display in the Rows Per Page selector.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Saving Interactive Reports

8.9.6.3.3 Removing the Interactive Report Search Bar

Disable an interactive report search bar.

To remove the search bar from an interactive report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Turn the Search Bar off:
   a. In the Property Editor, find the **Search Bar** group.
   b. **Include Search Bar**- Select **Off**.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.4 Customizing the Interactive Report Actions Menu

Customize or disable an interactive report Actions menu.

The Actions menu appears to the right of the Go button on the Search bar in an interactive report. Users use the Actions menu to customize how report information displays. Developers control what options display on the Actions menu in an interactive report.

> **Tip:**
>
> The structure of the interactive report Actions menu changed in release 5.1 to be consistent with interactive grids. To alter an interactive report Actions menu to use the previous (or legacy) structure, go to **Shared Components**, **Component Settings**, and configure the **Interactive Report** component. See Configuring Interactive Report Action Menu Structure.

To customize the Actions menu:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. In the Property Editor, find the **Actions Menu** group.
5. Under Actions Menu, enable and disable options by selecting **On** or **Of**
   - **Include Actions Menu** - Specify whether the Actions menu is included within the Search Bar.
   - **Filter** - Specify whether end users can define filters on this report.

- **Select Columns** - Specify whether end users can use Select Columns in the Actions menu to change the columns displayed in this report. User can select (or show), deselect (or hide) and reorder multiple columns using this option.
- **Rows Per Page** - Specify whether end users can change the number of rows displayed for this report.
- **Sort** - Specify whether end users can define sort columns for this report.
- **Control Break** - Specify whether end users can define control breaks for this report.
- **Highlight** - Specify whether end users can define highlights for this report.
- **Compute** - Specify whether end users can define computational columns for this report.
- **Aggregate** - Specify whether end users can define aggregates for this report.
- **Chart** - Specify whether end users can define charts for this report.
- **Group By** - Specify whether end users can define group by break-downs for this report.
- **Pivot** - Specify whether end users can define pivot break-downs for this report.
- **Flashback** - Specify whether end users can utilize flashback queries for this report.
- **Save Report** - Specify whether end users can save their current report definition as a Private report. **Private reports** are displayed in a select list of available reports in the Search Bar, and are only displayed to the user who saved the report, providing they are authenticated. The option on the Actions menu is only available to authenticated users.
- **Save Public Report** - Specify whether end users can save their current report definition as a Public report. **Public reports** are displayed in a select list of available reports in the Search Bar, and are displayed to everyone. The option on the Actions menu is only available to authenticated users and is further restricted if you also populate the Save Public Report Authorization.
- **Save Public Report Authorization** - Select an authorization scheme to restrict who can save public report definitions. To enable an end user to save a public report, this authorization scheme must evaluate to TRUE. If no authorization scheme is provided then reports may be saved by any user.
- **Reset** - Specify whether end users can reset their reports from the Actions menu. Resetting the report reverts all customizations made by the user, or specified by the calling URL, and displays the Primary Report.
- **Help** - Specify whether end users can invoke the built-in help modal dialog for interactive reports. The help dialog only outlines those options available to the current user.
- **Subscription** - Specify whether end users can define subscription settings for this report.
- **Download** - Specify whether end users can download the report to one of the download formats specified.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.5 Configuring Interactive Report Download Options

Configure interactive report download options available on the Actions menu..

The Download option on the Actions menu enables users to download an interactive report as a comma-delimited file (CSV) format, HTML, Microsoft Excel (XLSX) format, Adobe Portable Document Format (PDF), or as one of the formats attached to an email.

> **WARNING:**
>
> Downloads may also take considerable time to complete for large data sets.

> **Note:**
>
> The Pivot view does not support the PDF download format even if it is enabled and report server is configured.

- Configuring Interactive Report Download Formats
  Configure interactive report download options.
- Enabling Interactive Report CSV Download
  Enable the Enable CSV download in an interactive report.
- Enabling Interactive Report Email Download
  Enable interactive report email download.

8.9.6.5.1 Configuring Interactive Report Download Formats

Configure interactive report download options.

> **Tip:**
>
> Email is only available to authenticated users, and requires the SMTP Server to be configured within Oracle APEX Administration Services application (Administration Services), by an Instance Administrator.

To configure download formats:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. **Search Bar**, **Include Search Bar** - Select **On**.
5. **Actions Menu**, **Download** - Select **On**.

> **Tip:**
>
> Setting **Download** to **Off** prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.

6. Under **Download**:
   a. **Download**, **Authorization** - Select an authorization scheme to restrict who can download. To enable an end user to download, this authorization scheme must evaluate to TRUE for that user.

      If no authorization scheme is provided then any user may download.

   b. **Download**, **Download Formats** - Select the download formats users can utilize to download the currently displayed columns. Options include:
      - **CSV**
      - **HTML**
      - **Excel**
      - **PDF**

      > **Tip:**
      >
      > The Download attributes only appear on the Actions menu if you select a file format under **Download Formats**. If a report server has not been defined within Instance Administration then certain options do not display. To learn more, see Configuring Report Printing in a Full Development Environment in *Oracle APEX Administration Guide*.

   c. **Send as Email** - Specify whether end users can send an email, with the report download included as an attachment.

      Email is only available to authenticated users, and requires the SMTP to be configured within APEX Administration Services application by an Instance Administrator.

   d. **CSV Separator** - Enter a column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.

   e. **CSV Enclosed By** - Enter a delimiter character to delineate the starting and ending boundary of a data value. The default delimiter is double quotation marks.

   f. **Filename** - Enter the filename for the document when downloaded. Do not specify a file extension. The extension is added automatically. If you leave this attribute blank, the region name is used as the file name.

   g. **Supplemental Text** - Enter text that appears at the top of the download files. You can enter data classification, legal or warning messages about the data in the file.

      > **WARNING:**
      >
      > Do not use the Supplemental Text attribute with CSV download. If you enable **CSV** download, the defined text will invalidate the CSV file format.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> About Printing Reports to PDF and Excel

8.9.6.5.2 Enabling Interactive Report CSV Download

Enable the Enable CSV download in an interactive report.

To enable support for CSV download, you must first enable **Automatic CSV Encoding** on the Edit Globalization Attributes page. See Accessing the Globalization Page and Automatic CSV Encoding.

To enable the Enable CSV download in an interactive report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. **Search Bar**, **Include Search Bar** - Select **On**.
5. **Actions Menu**, **Download** - Select **On**.

   > **Tip:**
   >
   > Setting Download to **Off** prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.
6. Under **Download**:
   a. **Download Formats** - Select **CSV**.

      > **Tip:**
      >
      > The Download option only appears on the Actions menu if a file format is selected.

   b. **CSV Separator** - (CSV only) Define the column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.

   c. **CSV Enclosed By** - (CSV only) Enter a delimiter character. This character delineates the starting and ending boundary of a data value. Default delimiter is double quotation marks (" ").

   d. **Filename** - Enter a name for the downloaded document. Do not specify a file extension, the extension is added automatically. If you leave this blank, the region name is used as the file name.

> **WARNING:**
>
> Do not use the Supplemental Text attribute with CSV download. If you enable **CSV** download, the defined text will invalidate the CSV file format.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.5.3 Enabling Interactive Report Email Download

Enable interactive report email download.

> **Tip:**
>
> Email is only available to authenticated users, and requires the SMTP to be configured within APEX Administration Services application (Administration Services), by an Instance Administrator.

Interactive reports support email of all formats except Word (RTF). Note this only applies to Download dialog, not the Subscription dialog.

> **Tip:**
>
> Email download is only supported on authenticated pages. If you enable Email download on a public page, the Email download option does not display in Download dialog.

To enable support for email download, an APEX administrator must configure email at the Instance level. See *Configuring Mail* in *Oracle APEX Administration Guide*.

To enable email download:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.
4. **Search Bar**, **Include Search Bar** - Select **On**.
5. **Actions Menu**, **Download** - Select **On**.

   > **Tip:**
   >
   > Setting Download to **Off** prevents the Download menu from displaying on the Actions menu. To fully prevent users from downloading data, deselect all Download Formats described in the next step.
6. Under **Download**:

a. **Send as Email** - Specifies whether end users can send an email, with the report download included as an attachment. Enable **Send as Email**.

> **Tip:**
>
> The Download attributes only appear on the Actions menu if you select a file format under **Download Formats**.

b. **Filename** - Enter a name for the downloaded document. Do not specify a file extension, the extension is added automatically. If you leave this blank, the region name is used as the file name.

c. **Supplemental Text** - Enter text that displays at the top of the download files. You can enter data classification, legal, or warning messages about the data in the file.

> **WARNING:**
>
> Do not use the Supplemental Text attribute with CSV download. If you enable **CSV** download, the defined text will invalidate the CSV file format.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Note:**
>
> Email sent from an Email Download contains a system generated email signature identifying who sent the email. This signature cannot be removed.

> **See Also:**
>
> About Emailing from an Interactive Report

8.9.6.6 Controlling Interactive Report Pagination

Control interactive report pagination.

Pagination provides the end user with information about the number of rows and the current position within the result set. You control how pagination displays by making selections from Pagination attributes on the Attributes page in the Property Editor.

To edit interactive report pagination:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups

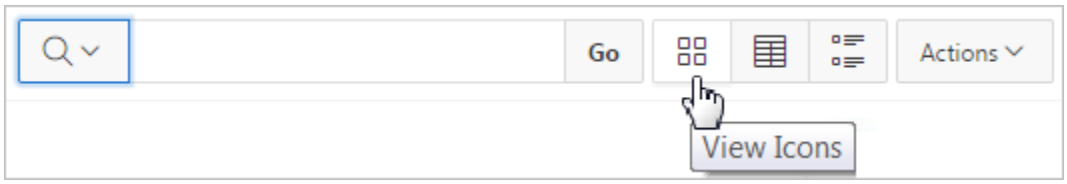
4. Configure Pagination attributes:
   a. Find **Pagination**.
   b. **Type** - Select the pagination type for this report. Often only a certain number of rows of a report display on a page. In order to include additional rows, the application end user must to navigate to the next page of the report.
   c. **Display Position** - Select where the report pagination displays.

      Pagination can be shown above or below the report (or both) and can be positioned on the left or right.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.7 Enabling Icon View

Learn about interactive report Icon View.

By default, most interactive reports display as a report. You can optionally display columns as icons. When configured to display columns as icons, a View Icons icon displays on the Search bar.



To use Icon view, you must identify the columns used to identify the icon, the label, and the target (that is, the link). As a best practice, set these columns to display as hidden since they are typically not useful for end users.

- Adding a View Icon Button
  Add Icon view to an interactive report.
- Adding a Custom Icon View Link
  Add a custom Icon View link.

8.9.6.7.1 Adding a View Icon Button

Add Icon view to an interactive report.

To add Icon view to an interactive report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find **Icon View**. Configure the following attributes:
   a. **Show** - Select **On** to add a View Icons icon to the Search bar.
   b. **Custom** - Specify whether you wish to define a custom link for the Icon View. This option provides greater flexibility in defining the content of the Icon View, including the ability to utilize more than one column. Select **On** or **Off**.

c. **Columns Per Row** - Select the number of icons to display per row in Icon View. A value of 5 displays 5 icons per row of the report.

d. **Link Column** - Select the column that returns the link target of the icon.

e. **Image Source Column** - Identify the column that returns the image source.

f. **Label Column** - Select the column that returns the image label.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Adding a Custom Icon View Link

8.9.6.7.2 Adding a Custom Icon View Link

Add a custom Icon View link.

A custom Icon View link provides greater flexibility in defining the content of the Icon View, including the ability to utilize more than one column.

To add a custom Icon View link:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups

4. Find **Icon View**. Configure the following attributes:

   a. **Show** - Select **On**.

   b. **Custom** - Select **On**.

   c. **Columns Per Page** - Select the number of icons to display per row in Icon View. A value of 5 displays 5 icons per row of the report.

   d. **Custom Link** - Enter the column link for the Icon view. Use substitution strings for column names, for example:

   ```
   <a href="#EMP_LINK#"><img src="#EMP_IMG#" title="#ENAME#"
   alt=""><br>#ENAME#</a>
   ```

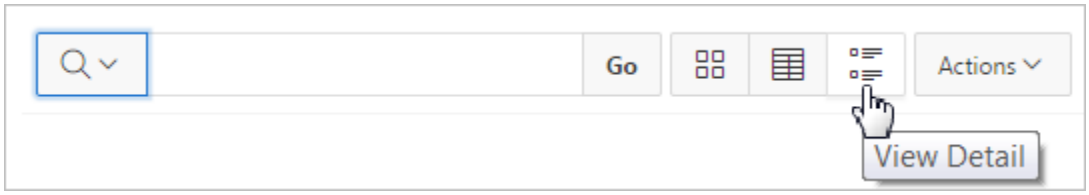
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Adding a Custom Icon View Link

8.9.6.8 Enabling Detail View

Add Details view to an interactive report.

Be default, most interactive reports display as a report. You can optionally display details about each column. When configured, a View Details icon displays on the Search bar.



To add Details view to an interactive report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find **Detail View**. Configure the following attributes:
   a. **Show** - Select **On**.
   b. **Before Rows** - Enter the HTML to be displayed before report rows. For example:

      ```
      <table>
      ```
   c. **For Each Row** - Enter the body for report rows in the Detail View. Use `#COLUMN_NAME#` substitution strings for column names and column labels. For example:

      ```
      <tr><td class="u-textRight">#ENAME_LABEL#:</td><td>#ENAME#</td></tr>
      <tr><td class="u-textRight">#JOB_LABEL#:</td><td>#JOB#</td></tr>
      ```
   d. **After Rows** - Enter the HTML to be displayed after report rows. For example:

      ```
      </table>
      ```
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.6.9 Configuring Advanced Attributes for Interactive Reports

Configure advanced interactive report attributes to define the region alias, specify an item whose value stores the saved report ID, and define a from address in an interactive report subscription or email download

- About Emailing from an Interactive Report
  Learn about emailing from an interactive report.
- Configuring Advanced Attributes
  Configure Advanced interactive report Attributes.

8.9.6.9.1 About Emailing from an Interactive Report

Learn about emailing from an interactive report.

You can define the from address for emails sent from an interactive report in two ways:

- Application-level — Edit the **Application Email From Address** attribute on the Edit Application Definition page.
- Report-level — Access the report Attributes. Under **Advanced**, edit the **Email from Address** attribute.

In either case, the value can be a literal string containing a valid email or an application substitution defined at the application-level using substitution syntax or `&APP_EMAIL.` substitution which returns the Application Email From Address defined at the application-level. Examples:

```
john.doe@abc.com
&MY_APP_EMAIL_FROM.
&APP_EMAIL.
```

> **Note:**
>
> Oracle does not recommend using an item substitution at the application or page-level since it only works in email download, but not for subscriptions.

> **See Also:**
>
> - Accessing the Application Definition Page
> - Configuring Advanced Attributes for Interactive Reports
> - Enabling Interactive Report Email Download

8.9.6.9.2 Configuring Advanced Attributes

Configure Advanced interactive report Attributes.

To configure Advanced Attributes:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups
4. Find **Advanced** and configure the appropriate options.
   a. **Region Alias** - Enter an alphanumeric alias for this interactive report region.

      The alias must be unique within the application. You can use an alias to reference an interactive report region in your custom API call. For example, you can write a function

to return the `INTERACTIVE_REPORT_ID` by querying the ALIAS column in the `APEX_APPLICATION_PAGE_IR` view. Since the `INTERACTIVE_REPORT_ID` value can change when you export or import the application to different instance, referencing interactive report regions using an alias may be preferable.

b. **Report ID Item** - Select a page or application item whose value stores the saved report ID.

   Interactive report uses this item value to link to a saved report. For example, you can define list of saved reports with links with defined item name and saved report ID as its value to link to a saved default reports, public or private reports. The saved report IDs can be retrieved from the `APEX_APPLICATION_PAGE_IR_RPT` view.

c. **Email From Address** - Enter a valid email address to use as the From address when sending email from this report.

   Emails are sent from this report for Downloads of type Email and for Subscriptions. The value can be a literal string containing a valid email address or a static substitution reference defined in the application using substitution syntax.

   Oracle does not recommend using an item substitution at the application or page-level since it only works in an email download, but not for subscriptions.

   Examples:

   ```
   john.doe@abc.com
   &MY_APP_EMAIL_FROM.
   &APP_EMAIL.
   ```

d. **Oracle Text Index Column** - Select a column where the column is indexed by an Oracle Text index.

   Row Search returns results based on the Oracle Text operator entered, providing the corresponding Oracle Text index has been created for the column. This option should only be enabled if the necessary index exist, and the end users have been trained in how to utilize the Oracle Text search operators. To simplify Oracle Text syntax for end users, you can specify a function for the Oracle Text PL/SQL Function Name Component Setting of Interactive Grids in Shared Components.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.7 Managing Interactive Report Column Attributes

Edit column attributes to control the display, features, and behavior of interactive report columns.

- Editing Interactive Report Column Attributes
  Edit Column attributes to precisely control the report layout.
- Creating a Column Link in an Interactive Report
  Configure Column Attributes to create a link from a report to another page in your application or to a URL
- Defining a Column as a List of Values in an Interactive Report
  Add a list of values in an interactive report to improve the speed of built-in filter tools.
- Formatting Columns in an Interactive Report
  Customize column display with HTML expressions.
- About Filtering on Conditional Links in Interactive Report Columns
  Learn about about filtering on conditional links in interactive report columns.

> **See Also:**
>
> Editing Pages in Page Designer

8.9.7.1 Editing Interactive Report Column Attributes

Edit Column attributes to precisely control the report layout.

Use these attributes to alter a column heading, change column positioning, hide a column, or control how users can manipulate a column.

To edit column attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. Edit the appropriate attributes.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. Identification, Type - Select the relevant column display type to define how column values displaye. To learn more about a column display type, select the **Type** attribute in the Property Editor and click the **Help** tab in the central pane.
6. Heading - Available attributes include:
   a. Heading , Heading - Enter the report column heading text.
   b. Heading, Alternative Label - Enter the alternative label to use in dialogs and in various column heading placements. Use an alternative label when the heading contains extra formatting, such as HTML tags, which do not display properly.
   c. Heading, Alignment - Select the heading alignment. Note that your alignment selection must be supported by the selected report template.

      Available options include:
      - **start**
      - **center**
      - **end**
7. Enable Users To - Control how users can manipulate a column. For each attribute select **On** or **Off**

To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> About Filtering on Conditional Links in Interactive Report Columns and Right Pane of Page Designer (Property Editor)

8.9.7.2 Creating a Column Link in an Interactive Report

Configure Column Attributes to create a link from a report to another page in your application or to a URL

To create a column link to another page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. Find **Identification**. From Type, select **Link**
5. Find **Link** and click **No Link Defined**.

   The Link Builder - Target dialog appears.
6. To create a link to another page, in the Link Builder - Target dialog:
   a. Type - Select **Page in this Application**.
   b. Page - Specify the target page number.
   c. Set Items - Select a Name and Value to specify session state for an item.
   d. Clear/Reset, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple page, enter a comma-delimited list of page numbers.
   e. Clear/Reset, Action - Select one of the following:
      - **None** - No clear or reset action when page loads.
      - **Clear Regions** - Clears all region settings on the page.
      - **Reset Regions** - Resets all regions on the page back to the default state, including sort orders, hidden columns and so on.
      - **Reset Pagination** - This option enables backwards compatibility.
   f. Advanced, Request - Specify the request to be used.

g. Click **OK**.

7. To create a link to a URL, in the Link Builder - Target dialog:
   a. Type - Select **URL**.
   b. URL - Enter the URL address.
   c. Click **OK**.
8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.7.3 Defining a Column as a List of Values in an Interactive Report

Add a list of values in an interactive report to improve the speed of built-in filter tools.

To define a report column as a list of values:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region containing the report.
3. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
4. Select a column to edit.
5. Configure Column Filter:
   a. Find **Column Filter**.
   b. For Type, select how to derive the filters provided in the report column drop down list. Options include:
      - **None** - No column filter is provided for this report.
      - **Default Based on Column Type** - `STRING` and `NUMBER` columns derive the list of values from distinct column values. `DATE` columns derive the list of values from predefined date ranges.
      - **Use Defined List of Values to Filter Exact Match** - Derive the list of values from a defined list of values SQL query to create an equal filter.
      - **Use Defined List of Values to Filter Word Contains** - Derive the list of values from a defined list of values SQL query to create a contains filter.
      - **Use Named List of Values to Filter Exact Match** - Derive the list of values from the named list of values to create an equal filter.
      - **Use Named List of Values to Filter Word Contains** - Derive the list of values from the named list of values to create a contains filter.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.7.4 Formatting Columns in an Interactive Report

Customize column display with HTML expressions.

To define column formatting:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.
4. Configure the column format:
   a. Find **Column Formatting**.
   b. In HTML Expression, enter HTML expressions to be shown in this column. Use `#COLUMN#` syntax to show column values in HTML.
      - **Example 1: Change the background and text colors**

        ```
        <span style="background-color: yellow; color:
        red;">#EMPLOYEE_STATUS#</span>
        ```

      - **Example 2: Use Heading 2**

        ```
        <h2>#EMPLOYEE_NAME#</h2>
        ```

      - **Example 3: Use a dynamic class from the SQL query**

        ```
        <span class="#COLUMN_CLASS#">#EMPLOYEE_STATUS#</span>
        ```

        The `#COLUMN_CLASS#` and `#EMPLOYEE_STATUS#` substitutions in this example are columns selected in the report SQL query.
      - **Example 4: Show an image, and a tooltip on hover**

        ```
        <img src="#IMAGE_NAME#" alt="#EMPLOYEE_NAME#"
        title="#EMPLOYEE_NAME#">
        ```

        The next two examples demonstrate are examples that use template directives. To learn more, see Using Template Directives.
      - **Example 5: Loop over a comma separated list of values**

        ```
        <ul>{loop "," TAGS/}<li>&APEX$ITEM.</li>{endloop/}</ul>
        ```

      - **Example 6: Use an "if" directive for a column with a quoted identifier**

        ```
        {if Mixed Case Name/}<strong>#"Mixed Case Name"#</strong>{endif/}
        ```
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.7.5 About Filtering on Conditional Links in Interactive Report Columns

Learn about about filtering on conditional links in interactive report columns.

If an interactive report has a column value that is a link and the link markup is constructed in the report query, attempting to define a filter on this value from the column header menu fails. Oracle APEX is not be able to render the link correctly in the column header menu because the column values are themselves rendered as links, such that they can be selected to apply the filter. If the developer wanted to conditionally render the link based on query logic, the link may have been defined in the report query as opposed to using the declarative column link support.

For filtering to work on conditional links in columns, developers have two options:

- **Option 1: Use an HTML Expression column:**
  1. Add another column to your query that selects the equivalent of the link text from the link and define a column alias of `my_link_display`. This column alias will be the value used in the column header menu.
  2. Change the `my_link` column to be of type **Hidden**.
  3. In the `my_link_display` column, set the column heading to be the same as the `my_link` column and set **HTML Expression** to be `#MY_LINK#`. Keep the default **Escape Special Characters** as **Yes**.
  4. Run the page.

     If needed, reorder the columns so that the new column in the same position as the old one. Select the **Actions** menu and then **Select Columns**.
  5. Save a new default report. Select the **Actions** menu, **Save Report**, and then **Save as Default Report Settings**.

  The link still renders fine in the report and the column header menu now renders just the `my_link_display` value which sets the filter correctly.
- **Option 2: Use a custom Column Filter for the link column, where you could query for the link text.**

> **See Also:**
>
> Editing Interactive Report Column Attributes

8.9.8 Customizing Interactive Reports in a Running Application

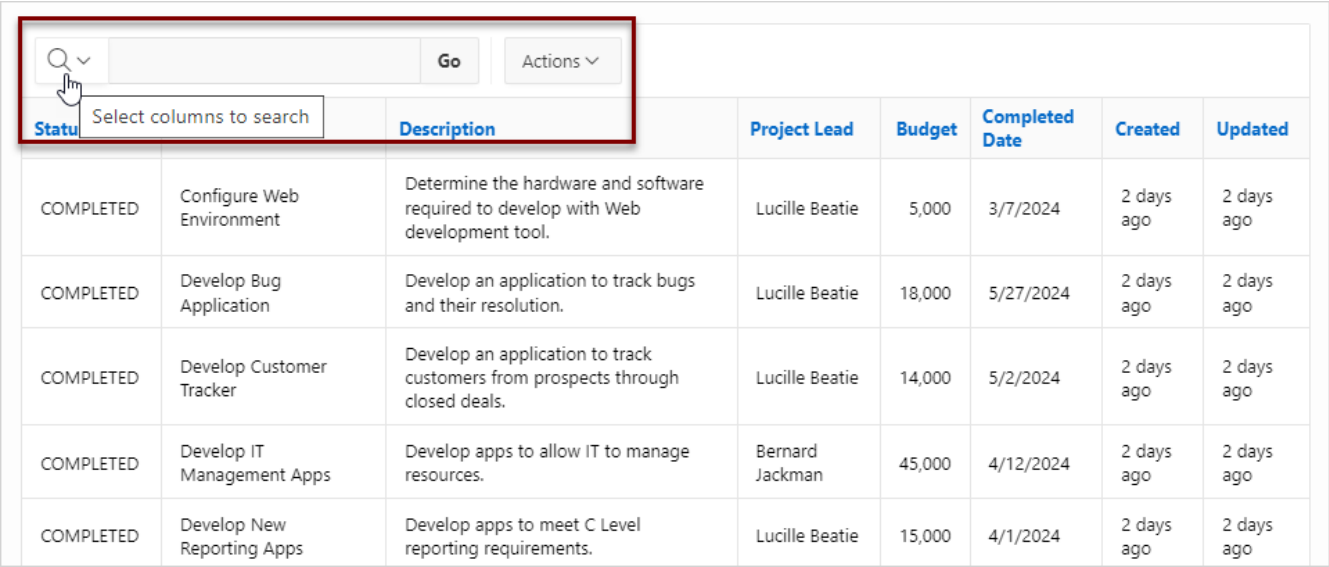
When running an application, end users and developers can alter the layout of interactive reports data by choosing the columns they are interested in, applying filters, highlighting, and sorting. They can also define breaks, aggregations, charts, group bys, and add their own computations.

- About the Search Bar
  Learn about the interactive report search bar.
- About the Actions Menu
  Learn about the interactive report Actions menu.
- Using Interactive Report Filters
  Learn about using interactive report filters.
- Saving Interactive Reports
  Learn about saving private and public interactive reports.

> **See Also:**
>
> Customizing an Interactive Report Using the Actions Menu in *Oracle APEX End User's Guide*

8.9.8.1 About the Search Bar

Learn about the interactive report search bar.



A search bar displays at the top of every interactive report and may include the following features:

- **Select Columns to Search** - This icon resembles a magnifying glass. Click this icon to narrow your search to specific columns. To search all columns, select **All Columns**.
- **Text Area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go Button** - Executes a search.
- **Reports** - Displays alternate default and saved private, or public reports. See Saving Interactive Reports.
- **View Icons** - Switches between an View Icon, View Report and View Detail of the default report (if enabled). May also include Chart and Group By View (if defined).
- **Actions menu** - Use the Actions menu to customize an interactive report. See About the Actions Menu.

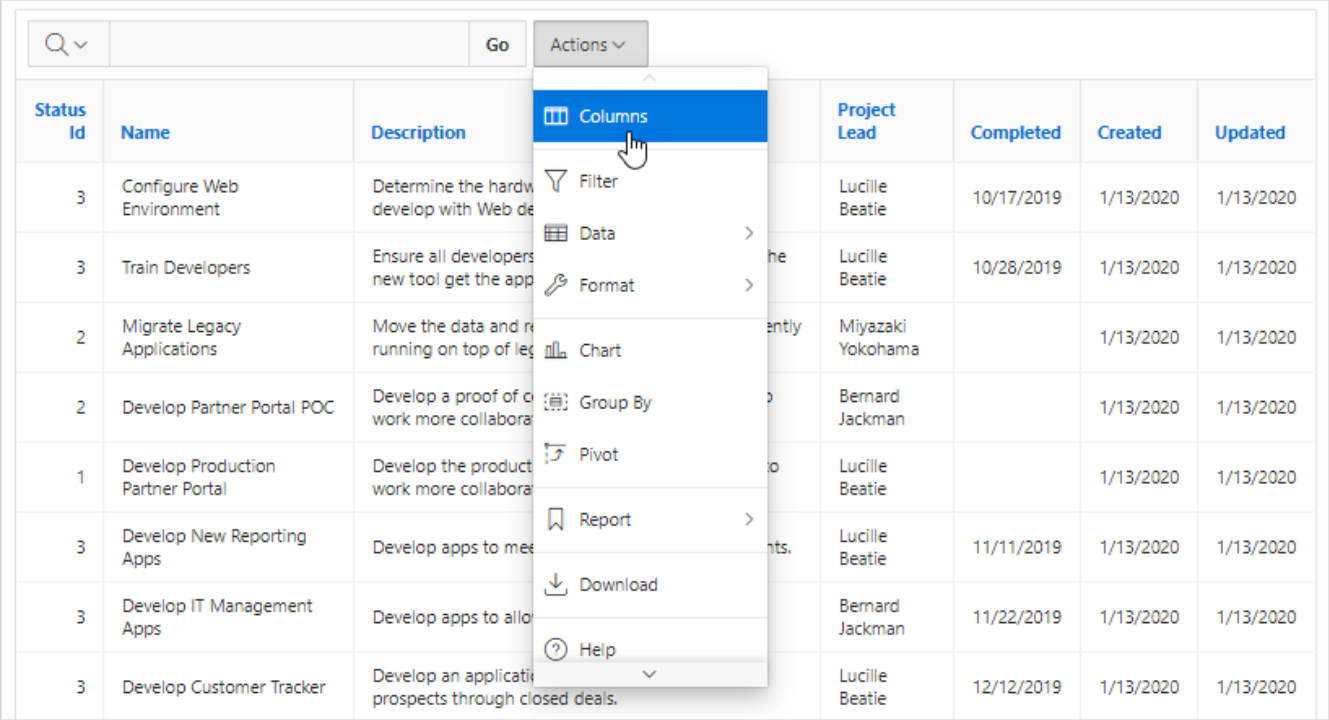
> **See Also:**
>
> Customizing an Interactive Report Search Bar

8.9.8.2 About the Actions Menu

Learn about the interactive report Actions menu.

> **Note:**
>
> The structure of the interactive report Actions menu has changed to be consistent with interactive grids. To alter an interactive report Actions menu to use the previous (or legacy) structure, go to **Shared Components**, **Component Settings**, and configure the **Interactive Report** component. See Configuring Interactive Report Action Menu Structure.

The Actions menu appears to the right of the Go button on the Search bar. End users use the Actions menu to customize an interactive report.

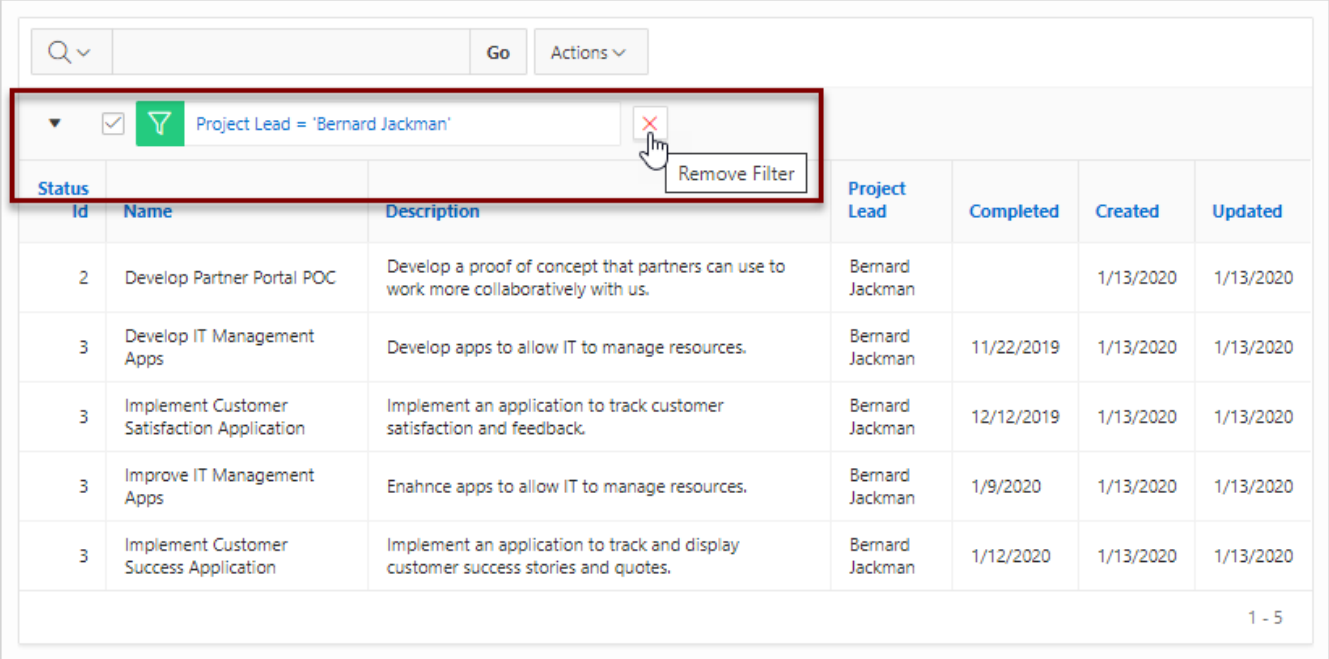


> **See Also:**
>
> Customizing an Interactive Report Using the Actions Menu in *Oracle APEX End User's Guide*

8.9.8.3 Using Interactive Report Filters

Learn about using interactive report filters.

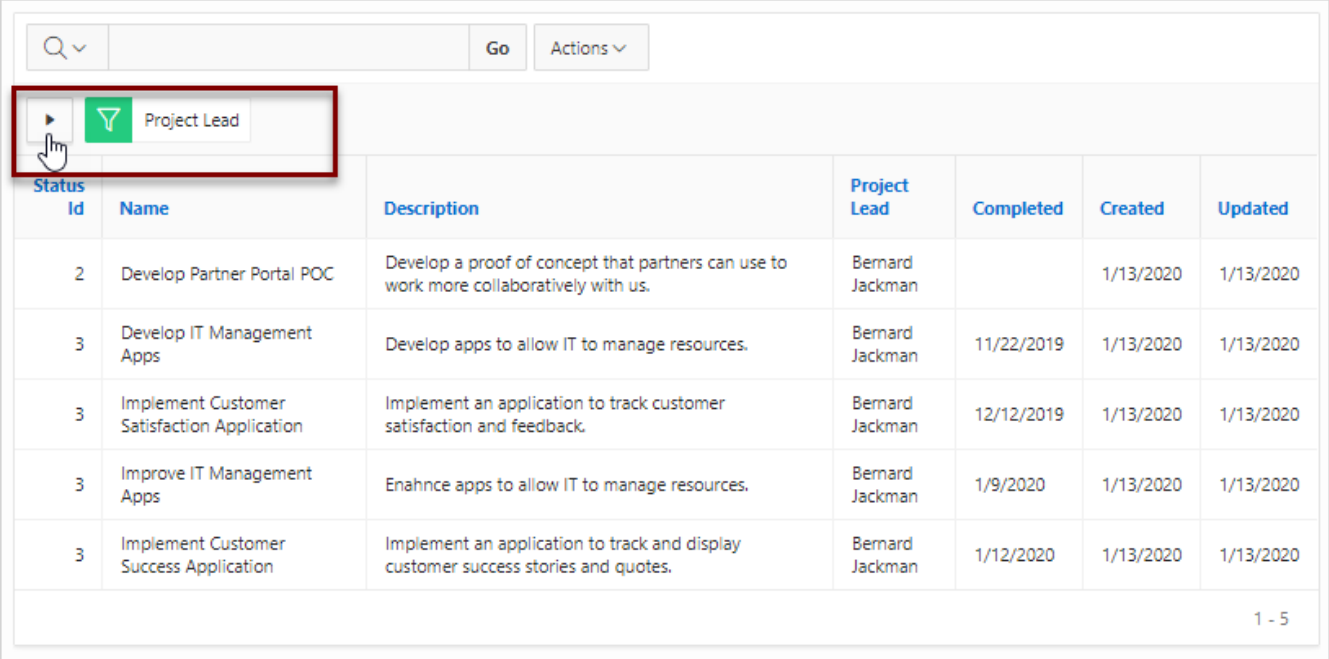
When you customize an interactive report, a filter displays between the search bar and the report. The following illustration shows an interactive report with one filter, `Project lead = 'Bernard Jackman'`.



Other keys features of interactive report filters include:

- **Enable/Disable** - To enable and disable a filter, select and deselect the checkbox to the left of the filter name.
- **Filter Edit** - To edit a filter, click the filter name.
- **Remove Filter** - To remove a filter, select the **Remove Filter** icon to the right of the filter name.

By default, multiple filters display horizontally stacked on top of one another. To minimize the amount of information that displays and hide filter details, click the arrow to the left of the filter name. The following illustration shows the same report with the filter details hidden.



8.9.8.4 Saving Interactive Reports

Learn about saving private and public interactive reports.

All users can save a private or public interactive report. However, only the user who creates a private report can view, save, rename, or delete it. Developers have additional save capabilities in that they can save the report that initially displays, called the Primary Default, or create an Alternative Default report.

- About the User Type and Available Save Options
  Learn how end users and developers can save interactive reports.
- About Configuration Dependencies When Saving Interactive Reports
  Developers control how users can save an interactive report by editing report Attributes.
- Saving a Public or Private Interactive Report
  End users can save public or private interactive reports.
- Saving a Default Interactive Report
  Developers can save primary and alternative interactive reports.
- Renaming a Public or Private Interactive Report
  Rename a public or private interactive report.
- Deleting a Public or Private Interactive Report
  Delete a public or private interactive report
- About Exporting Interactive Reports
  Learn about exporting interactive reports.

> **See Also:**
>
> Managing Saved Interactive Reports in *Oracle APEX Administration Guide*

8.9.8.4.1 About the User Type and Available Save Options

Learn how end users and developers can save interactive reports.

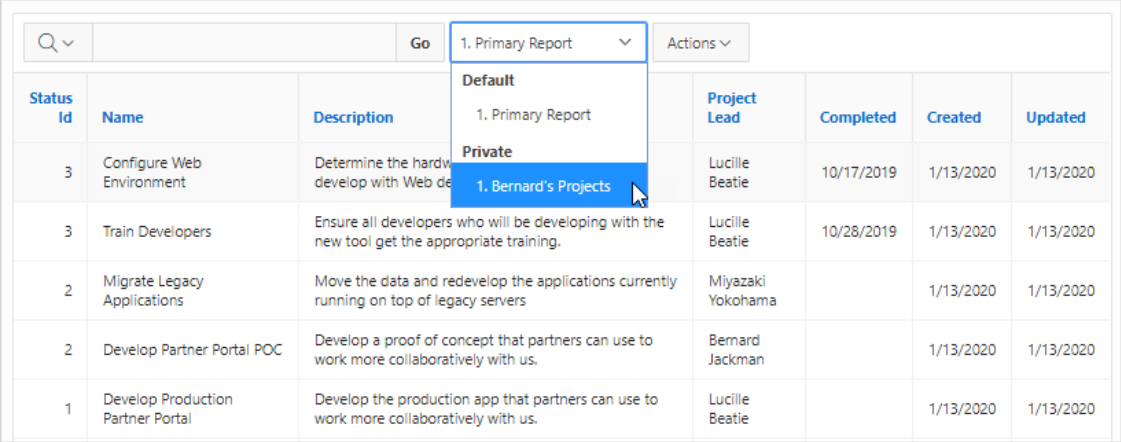
An end user can save an interactive report in two ways:

- **Public Report** (End user and developer). This report is viewable by all users. However, only the user who creates a public report can save, rename, or delete it. Although all users can view a public report, they can only save it under a new report name.
- **Private Report** (End user and developer). Only the user who creates the private report can view, save, rename, or delete it.

A developer can additionally save an interactive report in the following ways:

- **Primary Default** (Developer only). The Primary Default is the report that initially displays. Primary Default reports cannot be renamed or deleted.
- **Alternative Report** (Developer only). Enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative Report.

The following illustration shows the Reports list on the Search bar of an interactive report.



This example shows two reports:

- **Default - Primary Report**. This is the initial report created by the application developer. Default, Primary reports cannot be renamed or deleted.
- **Private - 1. Bernard's Projects**. This is a Private report. Only the user who creates a private report can view, save, rename, or delete it.

> **Tip:**
>
> You can view saved report activity from the Workspace Administration page. On the Workspace home page, click the **Administration** menu and then select **Administration**, **Monitor Activity**, and then under **Page Views** select **By Interactive Report**.

8.9.8.4.2 About Configuration Dependencies When Saving Interactive Reports

Developers control how users can save an interactive report by editing report Attributes.

To fully enable users to save private reports, a developer must:

1. Define an authentication scheme for the current application.
2. Edit the following report Attributes:
   - **Search Bar**, **Reports Select List** — Set this attribute to **On** to include a Reports select list in the Search bar.
   - **Actions Menu**, **Save Report** — Set this attribute to **On** to enable end users to save their current report definition as a *Private* report.
   - **Actions Menu**, **Save Public Report** — Set this attribute to **On** to enable end users to save their current report definition as a *Public* report.
   - **Actions Menu**, **Save Public Report Authorization** — Select an authorization scheme to restrict who can save public report definitions. To enable an end user to save a public report, this authorization scheme must evaluate to TRUE. If no authorization scheme is provided then reports may be saved by any user.

See Also:

- Establishing User Identity Through Authentication
- Customizing the Interactive Report Actions Menu
- Customizing an Interactive Report Search Bar

8.9.8.4.3 Saving a Public or Private Interactive Report

End users can save public or private interactive reports.

End users can save an interactive report and classify it as being either public or private. Public reports can be viewed by all users. Private reports can only be viewed by the user who saves the report.

Tip:

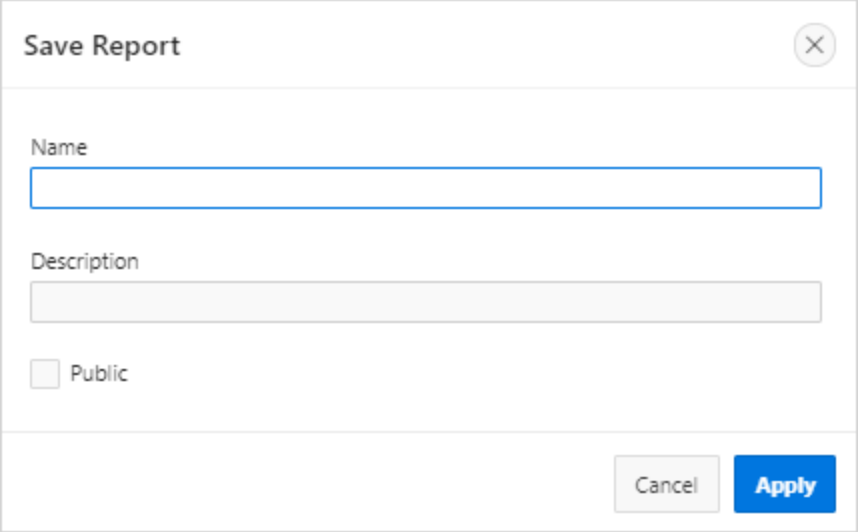
To save a public report, the developer must edit the report Attributes. In the Property Editor, locate **Actions Menu** and set **Save Public Report** to **On**. Developers can also apply an authorization scheme to enable or disable **Save Public Report**. See About Configuration Dependencies When Saving Interactive Reports.

To save a public or private interactive report:

1. Go to the page containing the interactive report you want to save.
2. Customize the report (for example, hide columns, add filters, and so on).
3. Click the **Actions** menu and select **Report** and then **Save Report**.

The Save Report dialog appears.

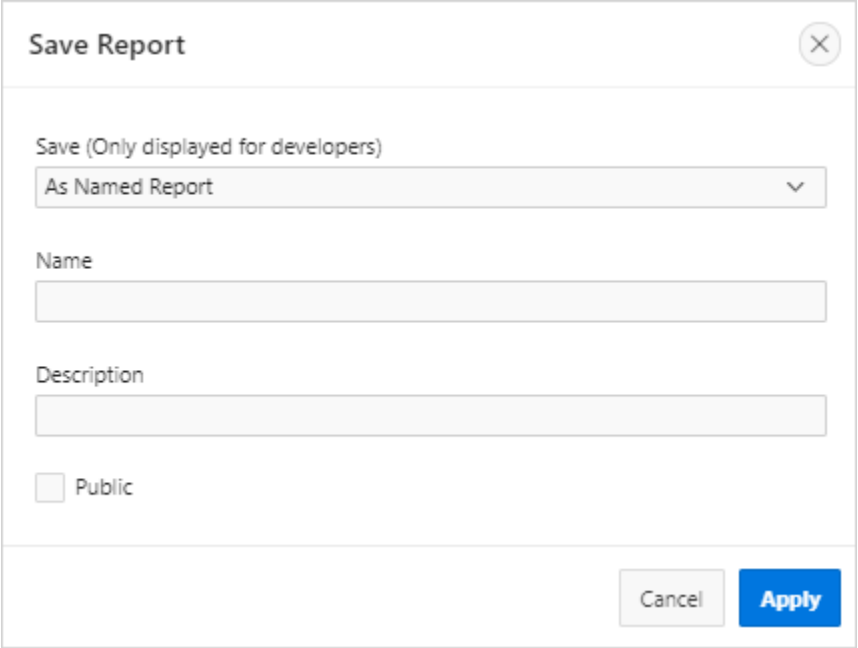
4. For end users, the following Save Report dialog appears.



In Save Report:

a. **Name** - Enter a name for the report.
b. **Description** - Enter an optional description.
c. **Public** - Select this option to make the report viewable to all users. Deselect this option to make the report private.
d. Click **Apply**.

5. For developers, the following Save Report dialog appears.



In Save Report:

a. **Save** - Select **As Named Report**.

b. **Name** - Enter a name for the report.

c. **Description** - Enter an optional description.

d. **Public**- Select this option to make the report viewable to all users. Deselect this option to make the report private.

> **Tip:**
>
> The Public checkbox only displays if you have enabled the **Save Public Report** report attribute.

e. Click **Apply**.

8.9.8.4.4 Saving a Default Interactive Report

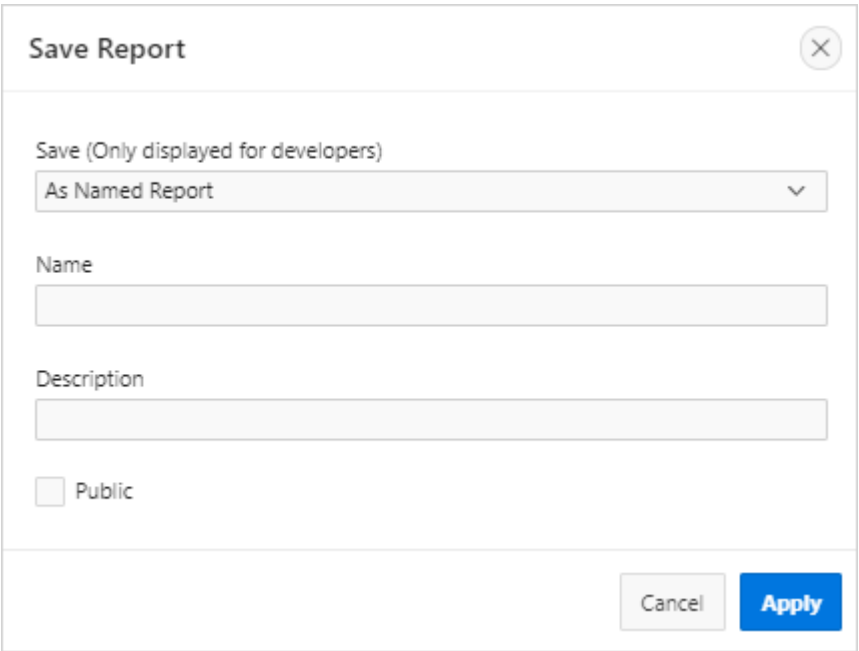
Developers can save primary and alternative interactive reports.

Developers can save two types of default interactive report: **primary** and **alternative**. Both reports display on the Report list on the search bar. However, primary default reports cannot be renamed or deleted.

To save a default interactive report:

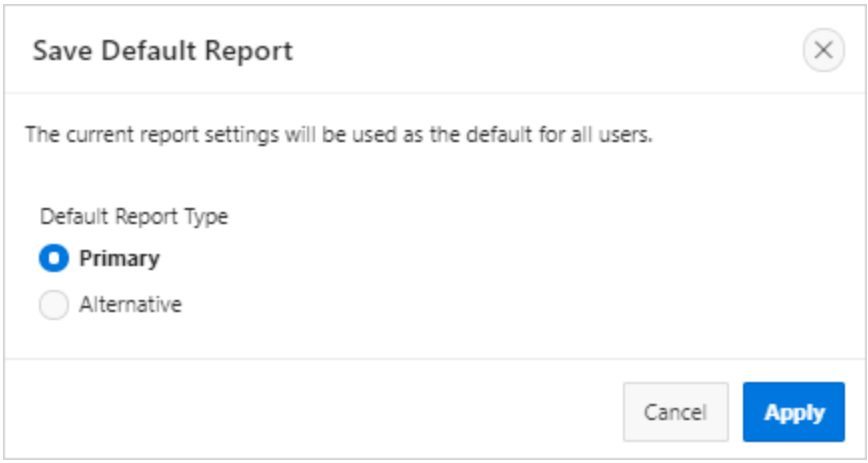
1. Run the report as a developer.
2. Customize the report (for example, hide columns, add filters, and so on).
3. Click the **Actions** menu and select **Report** and then **Save Report**.

   The Save Report dialog appears.



4. From Save, select **As Default Report Settings**.

The Save Default Report dialog appears.



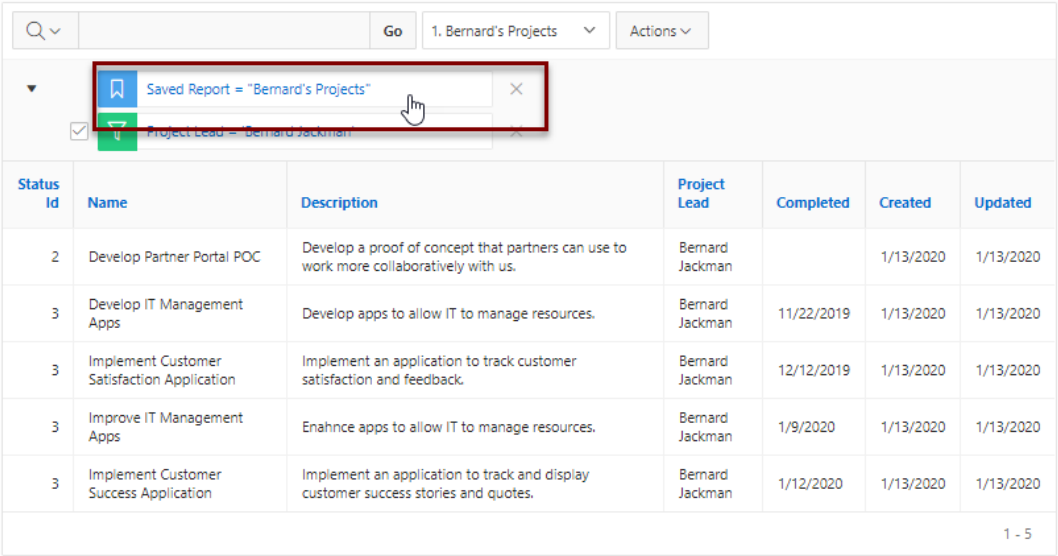
5. For Default Report Type, select either:
   - **Primary** - This report displays as the primary report when the page runs.
   - **Alternative** - This report displays as an alternative option on the Reports list on the Search bar. If you select **Alternative**, enter a name.
6. Click **Apply**.

8.9.8.4.5 Renaming a Public or Private Interactive Report

Rename a public or private interactive report.

To rename a public or private interactive report:

1. Run the report as a developer.
2. Select the public or private interactive report to rename.
3. Click the saved report link.



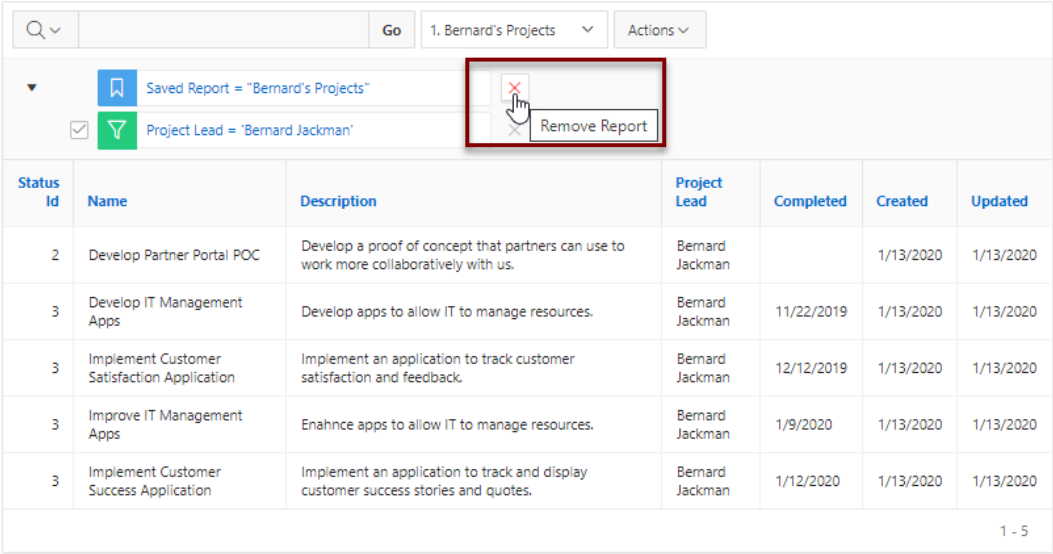
4. In the Rename Report dialog:
   a. Edit the attributes. For example, enter a new name in Name.
   b. Click **Apply** to save your changes.

8.9.8.4.6 Deleting a Public or Private Interactive Report

Delete a public or private interactive report

To delete a public or private interactive report:

1. Run the report as a developer.
2. Click the **Remove Report** icon next to the report name link.



3. Click **Apply**.

8.9.8.4.7 About Exporting Interactive Reports

Learn about exporting interactive reports.

When you export an application, the Primary Default, Alternative Default, and public reports are included by default. Configure the **Export Public Reports** and **Export Private Reports** attributes to control the type of reports to include in your export.

> **See Also:**
>
> Exporting an Application

8.9.9 Linking to Interactive Reports

Link to interactive reports using the syntax in the `Request` value and `ItemNames` section of the URL.

Data in an interactive report is driven by declarative filters. Developer can set report parameters by modifying interactive report settings for an end user. However, these settings

only affect the end user's working report and do not affect any saved report settings. Developers can use the syntax described in this topic to link to interactive reports.

> **Note:**
>
> When you add additional interactive report regions, be sure to change the old single interactive report link syntax. Otherwise, the link generates an error. A link to a page with more than one interactive report regions, the request or name syntax needs to fully qualify with region static ID. To open a link to an interactive report from outside of an application (for example, from an email), you may need to enable the Session Management, Deep Linking attribute in your application. See Session Management.

- Interactive Report Request Syntax
  Link to a specific saved report using `Request` syntax.
- Interactive Report Name and Value Syntax
  Create a filter using `itemNames` and `itemValues` syntax.
- Interactive Report ClearCache Syntax
  Use the `ClearCache` section of the URL to reset, clear, or reset pagination the interactive report.
- Restrictions When Linking to Interactive Reports
  Learn about restrictions when linking to interactive reports.
- Valid Interactive Report Linking Examples
  View valid interactive report linking examples using Friendly URL Syntax.
- Invalid Interactive Report Linking Examples
  View invalid interactive report linking examples.
- Linking to Shared Interactive Reports
  Learn about linking to shared interactive reports.

> **See Also:**
>
> Understanding URL Syntax

8.9.9.1 Interactive Report Request Syntax

Link to a specific saved report using `Request` syntax.

To link to a specific saved report, use the following `Request` syntax:

```
IR[region static ID]_<report_alias>
```

Consider the following example:

```
IR[EMP]_dept20
```

> **Note:**
>
> The use of `Request` in the URL is optional. To link to multiple saved reports, separate each Request with comma.

8.9.9.2 Interactive Report Name and Value Syntax

Create a filter using `itemNames` and `itemValues` syntax.

To create a filter, use the following `itemNames` and `itemValues` syntax:

```
IR[region static ID]<operator>_<target column alias>
```

Consider the following example:

```
IR[EMP]C_ENAME:KING
```

> **Note:**
>
> If there is only one interactive report region on the page, the `itemName` does not need to be fully qualified with the region static ID. If an operator is not defined, a filter with equal operator is added.

To create a filter on an interactive report in a link, use the `itemNames` syntax described previously and add the filter value in the corresponding location in the item names section of the URL.

To create row text which contains a filter on an interactive report, use `ROWFILTER` as the target column alias in the item name syntax and add a filter value in the item `itemValues` of the URL. The link will add a Row Text Contains filter which searches all columns displayed in the report with type `STRING` or `NUMBER`.

Consider the following example:

```
IR[EMP]_ROWFILTER:SALES
```

Valid operators include:

- `C` = Contains
- `EQ` = Equals (this is the default)
- `GTE` = Greater than or equal to
- `GT` = Greater Than
- `LIKE` = SQL Like operator
- `LT` = Less than
- `LTE` = Less than or equal to
- `N` = Null
- `NC` = Not Contains
- `NEQ` = Not Equals

- `NLIKE` = Not Like
- `NN` = Not Null
- `NIN` = Not In (escape the comma separated values with a leading and trailing backslash, \)
- `IN` = In (escape the comma separated values with a leading and trailing backslash, \)
- `ROWFILTER` = Row Text Contains (this searches all columns displayed in the report with type STRING or NUMBER)

8.9.9.3 Interactive Report ClearCache Syntax

Use the `ClearCache` section of the URL to reset, clear, or reset pagination the interactive report.

> **Tip:**
>
> Using `ClearCache` without specifying a saved report alias in the `Request`, clears the primary default report. To clear a specific interactive report region on a page with more than one interactive reports, specify the region static ID along with saved report alias in the `Request`.

Use the following syntax in the `ClearCache` section of the URL to reset, clear, or reset pagination the interactive report:

- `RR` - Resets an interactive report and pagination. This is equivalent to the end user choosing the Reset option from the interactive report Actions menu on the target page. The report is returned to the default report settings specified by the developer or saved by the user.
- `CR` - Clears interactive report and resets pagination. This clears all of the session report settings such as control break, aggregate, flashback, chart, number of rows to display, filter, highlight, computation, group by, and pivot.
- `RP` - Resets interactive report pagination.

8.9.9.4 Restrictions When Linking to Interactive Reports

Learn about restrictions when linking to interactive reports.

When linking to interactive reports, remember the following restrictions:

1. If a `Request` value is not defined in the URL, all of the primary default report settings will change.
2. If a `Request` value is not defined and `itemNames` section does not include a fully qualified filter column name with region static ID, an error occurs if there are more than one interactive report region in the target page.
3. If a `Request` value is not defined and `itemNames` section do not include a fully qualified filter column name with region static ID, it will create a filter on the primary default report if there is only one interactive report region in the target page.

8.9.9.5 Valid Interactive Report Linking Examples

View valid interactive report linking examples using Friendly URL Syntax.

Valid Linking Example 1

This example includes a link with no `Request` value, but includes a qualified filter column with region static ID in the `itemNames` section of the URL. This example creates a filter on the primary default report in the referenced region item name.

- f?p URL Syntax:

  ```
  https://hostname:port/ords/f?p=&APP_ID.:2:&SESSION.::::IR[EMP]_ENAME:KING
  ```

- Friendly URL Syntax:

  ```
  https://hostname:port/ords/r/mycompany/hr-app/simple-ir?
  session=15332852263343&IR_JOB=CLERK
  ```

Valid Linking Example 2

To support backwards compatibility with previous releases, this example uses older syntax and works on a page with one interactive report region.

- f?p URL Syntax:

  ```
  https://hostname:port/ords/f?
  p=&APP_ID.:1:&SESSION.:IR_REPORT_12345::RIR,CIR::RIR,CIR:IR_ENAME:KING
  ```

- Friendly URL Syntax:

  ```
  https://hostname:port/ords/r/mycompany/hr-app/simple-ir?IRC_JOB=ERK
  ```

Valid Linking Example 3

In this example the link has no `REQUEST` value. The clear cache section `RR` resets the report for all interactive report regions for page 1.

f?p URL Syntax:

```
https://hostname:port/ords/f?p=&APP_ID.:1:&SESSION.:::RR
```

> **See Also:**
>
> Understanding Friendly URL Syntax

8.9.9.6 Invalid Interactive Report Linking Examples

View invalid interactive report linking examples.

Invalid Linking Example 1

This link has no `Request` value and does not include a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error if there is more than one interactive report region in the target page.

f?p URL Syntax:

```
https://hostname:port/ords/f?p=&APP_ID.:2:&SESSION.::::ENAME:KING
```

Invalid Linking Example 2

This example has a link with two request values separated by commas and does not contain a fully qualified filter column with region static ID in the item name section of the URL. This example generates an error since it is not clear where the filter needs to be created.

f?p URL Syntax:

```
https://hostname:port/ords/f?
p=&APP_ID.:2:&SESSION.:IR[test]_employee,IR[test2]_dept10:::ENAME:KING
```

Invalid Linking Example 3

This example uses old request value syntax used on a page with multiple interactive report regions. This example will generate an error. The old syntax only works if there is one interactive report region.

f?p URL Syntax:

```
https://hostname:port/ords/f?p=&APP_ID.:2:&SESSION.:IR_REPORT_myrpt:::ENAME:KING
```

8.9.9.7 Linking to Shared Interactive Reports

Learn about linking to shared interactive reports.

Link to saved primary default, alternative default, and public reports using `IR[region static ID]_<report_alias>` in the `Request` value of the URL.

- Editing the Interactive Report Saved Report Alias
  Edit the Alias for a saved reports to create a more meaningful URL link to the saved report.
- Link Examples to a Primary Report
  View examples that demonstrate how to link to a primary report (`report_alias=PRIMARY`).
- Link Examples to Saved Reports
  View examples that demonstrate how to link to a saved report (`report_alias=REPORT1`).

See Also:

- About Using f?p Syntax to Link Pages
- Saving a Default Interactive Report
- Deleting Saved Interactive Reports
- Managing Saved Interactive Reports in *Oracle APEX Administration Guide*

8.9.9.7.1 Editing the Interactive Report Saved Report Alias

Edit the Alias for a saved reports to create a more meaningful URL link to the saved report.

To edit the Alias for saved reports:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, find the region containing the report.
3. Under the region, expand **Saved Reports** and select the saved report.

   The Property Editor displays the attributes for the Saved Report.
4. **Identification**, **Alias** - Enter a new alias for the report. Use this alias to define a more meaningful URL link to this saved report. This link can be used to link directly from elsewhere in the application.
5. **Link**, **Link Example** - View the URL link for this saved report. This attribute is view only and cannot be updated. Use this attribute to quickly copy the displayed URL. This URL can be used to link directly to this particular saved report from anywhere else in the application.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.9.9.7.2 Link Examples to a Primary Report

View examples that demonstrate how to link to a primary report (`report_alias=PRIMARY`).

This example links to a primary report:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY
```

This example links, resets, and clears primary report settings:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY::RR,CR:
```

This example links, resets, and clears primary report settings. It additionally creates a `ENAME = 'KING'` filter on the primary report:

```
f?p=100:1:&APP_SESSION.:IR_PRIMARY::RR,CR:IR_ENAME:KING
```

8.9.9.7.3 Link Examples to Saved Reports

View examples that demonstrate how to link to a saved report (`report_alias=REPORT1`).

This example links to a saved report:

```
f?p=100:1:&SESSION.:IR_REPORT1
```

This example links, resets, and clears settings for saved report `REPORT1`:

```
f?p=100:1:&APP_SESSION.:IR_REPORT1::RR,CR:
```

This example links, resets, and clears saved report `REPORT1` settings. It additionally creates a `ENAME = 'KING'` filter on saved report REPORT1.

```
f?p=100:1:&APP_SESSION.:IR_REPORT1::RR,CR:IR_ENAME:KING
```

8.10 Managing Classic Reports

A classic report features a simple report page based on the formatted result of a SQL query.

Developers control how a classic report works by editing the following attributes in Page Designer: region attributes, report Attributes, and Column attributes.

- Example Classic Report
  View a classic report example.
- Supported Columns in Classic Reports
  Learn about column types supported by classic reports.
- Creating a Classic Report Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains one or multiple classic report pages.
- Creating a Classic Report Using the Create Page Wizard
  Run the Create Page Wizard to add a new classic report page to an existing application.
- Editing Classic Report Region Attributes
  Edit Region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.
- Managing Classic Report Attributes
  Edit report Attributes control how a classic report works.
- Managing Classic Report Column Attributes
  Edit column attributes to control the display, features, and column behavior.

8.10.1 Example Classic Report

View a classic report example.

A classic report is a formatted result of a SQL query. You choose a table on which to build a report, or provide a custom SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement. With the exception of sorting and simple filtering, end users cannot customize a classic report. Both the Create Application Wizard and Create Page Wizard support the creation of classic reports.

The following is an example of a classic report.

Status Id	Name	Description	Project Lead	Completed	Created	Updated
3	Configure Web Environment	Determine the hardware and software required to develop with Web development tool.	Lucille Beatie	10/17/2019	1/13/2020	1/13/2020
3	Train Developers	Ensure all developers who will be developing with the new tool get the appropriate training.	Lucille Beatie	10/28/2019	1/13/2020	1/13/2020
2	Migrate Legacy Applications	Move the data and redevelop the applications currently running on top of legacy servers	Miyazaki Yokohama		1/13/2020	1/13/2020
2	Develop Partner Portal POC	Develop a proof of concept that partners can use to work more collaboratively with us.	Bernard Jackman		1/13/2020	1/13/2020
1	Develop Production Partner Portal	Develop the production app that partners can use to work more collaboratively with us.	Lucille Beatie		1/13/2020	1/13/2020
3	Develop New Reporting Apps	Develop apps to meet C Level reporting requirements.	Lucille Beatie	11/11/2019	1/13/2020	1/13/2020

To create a classic report, developers either select a table or provide a SQL statement. Classic reports support general keyword search capability, the ability to specify the number of rows that display, and basic column sorting. You also have the option to include a form page for creating or updating records.

Tip:

To view an example, install the sample app, *Sample Reporting*. To learn more, see Installing Apps from the Gallery.

Note:

Developers can create highly customized reports using report template (in particular, named column templates). See Report Templates.

8.10.2 Supported Columns in Classic Reports

Learn about column types supported by classic reports.

Classic reports support the following column types. To view the column type in Page Designer, select the column and find the **Identification**, **Type** attribute in the Property Editor.

Note:

You can also view this information in Page Designer Help. View the page containing the report in Page Designer. Select a column and then **Identification**, **Type** attribute. Click the **Help** tab in the central pane.

Column Type	Description
Plain Text	Displays the column value.
Plain Text (based on List of Values)	Displays the value returned from the list of values specified.
Rich Text	Renders the column using HTML or Markdown.

Column Type	Description
Link	Displays the column as a link.
Display Image	Displays the column as an image.
Download BLOB	Displays the column as a link to download the file.
Percent Graph	Displays the column as a percentage graph. The value retrieved must be between 0 and 100.
Hidden	Does not display the column, but it is referenceable.

8.10.3 Creating a Classic Report Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains one or multiple classic report pages.

To create a classic report using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. To add a classic report:
   a. Click **Add Page**.
   b. Expand **Additional Pages** and select **Classic Report**.
8. On Add Report Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Select the source for the report:
      - **Table or View**
      - **SQL Query**

      What you select determines what displays next. Follow the on-screen instructions. To learn more about an attribute, click the Help icon in the lower left corner.
   d. Include Form - Select **Include Form** to include a form page for creating or updating records.
   e. Lookup Columns - Expand to see attributes for defining a lookup to another table. Use Lookup Columns to replace identifiers with a display column, such as showing the department name instead of the department number. Specify the following:
      - Lookup Key - Select the column to change to output the display column.
      - Display Column - Select the column (from the lookup table) to be displayed.

> **Tip:**
>
> **Lookup Columns** only appear if the current page is based on a table and has foreign key constraints to another table.

f. Advanced - Expand **Advanced** to configure these options:
   - Set as Home Page - Enable this to make this page the home page for the application.
   - Set as Administration Page - Enable this to have this page display under Administration.
   - Page Help - Enter text to be displayed when the user selects Page Help.

g. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

> **See Also:**
>
> Creating a Report Using the Create Page Wizard

8.10.4 Creating a Classic Report Using the Create Page Wizard

Run the Create Page Wizard to add a new classic report page to an existing application.

To create an classic report page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Classic Report**.
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

> **Tip:**
>
> This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

c. Page Mode - Identify the page mode. To learn more, see field-level Help.

d. Include Form Page - Enable this flag to create a form page along with the report page.

   Form pages are always created as Modal Dialog pages. Enabling this flag displays additional attributes. To learn more about an attribute, see field-level Help.

6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table / View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

9. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

8.10.5 Editing Classic Report Region Attributes

Edit Region attributes to alter the SQL source, change the region layout and appearance, define a region displays selector, and create region conditions.

To edit classic report Region attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

   In the Property Editor, the **Region** tab displays the attributes. Attributes are organized in groups.

   > **Tip:**
   >
   > To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. Edit the appropriate attributes.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

8.10.6 Managing Classic Report Attributes

Edit report Attributes control how a classic report works.

- Editing Classic Report Attributes
  Customize a classic report by editing report Attributes.
- Updating Classic Report Appearance Attributes
  Change the appearance of a classic report by editing Appearance attributes.
- Editing Classic Report Pagination
  Edit classic report pagination.
- Adding Support for XML File or a CSV File Export
  Add support for exporting a report as an XML or CSV by selecting a report template.
- Enabling Classic Report CSV Output Option
  Enable classic report CSV output option.
- Controlling Classic Report Column Breaks
  Control classic report column breaks.

> **See Also:**
>
> Editing Pages in Page Designer

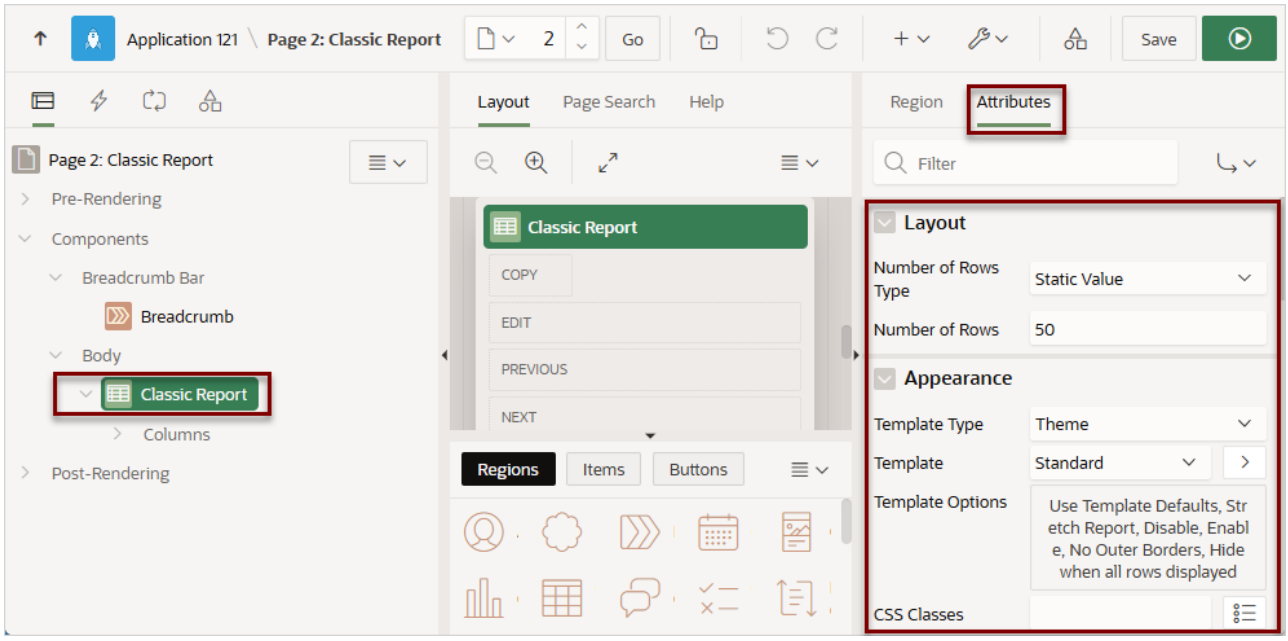
8.10.6.1 Editing Classic Report Attributes

Customize a classic report by editing report Attributes.

To edit report Attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.



The Property Editor displays the report Attributes. Attributes are organized in groups.

To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Edit the appropriate attributes.
5. To view Help, select the attribute and click the Help tab. Once you activate the Help pane, the content that displays changes every time you select another attribute.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.6.2 Updating Classic Report Appearance Attributes

Change the appearance of a classic report by editing Appearance attributes.

To change the appearance of a classic report:

1. View the page in Page Designer:

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.
4. Under **Appearance**, edit the following:
   a. **Template Type** - Select template type o this report.
   b. **Template** - Select a report template to define the appearance and layout of this classic report.

      Report templates are defined in the application theme. When a classic report is created, the template is automatically set to the default report template defined in the current theme. Many themes contain classic report templates for displaying data in a table, key value pairs, search results, cards, and so forth.

Note that some report templates, such as cards and search results, are "row based" and expect your SQL query to return a particular set of column aliases that are substituted in the template. The Universal Theme provides the following report templates:

- **Alerts** - Displays any number of horizontal alerts or notifications on your page.
- **Badge List** - Displays key value pairs in a dashboard-like user interface.
- **Cards** - Displays report data in cards, or small blocks and is useful for presenting a variety of information. Cards can be heavily customized with Template Options.
- **Comments** - Displays comments and other messages.
- **Content Row** - Displays report data in a set of rows, with support for a selection column, icons, and actions. Suitable for most types of content.
- **Contextual Info** - Displays simple key-value pairs one after the next. It uses the column headers as the labels and displays the associated column value next to each label.
- **Media List** - Displays report content in a media list, with an icon, heading, description, and badge.
- **Search Results** - Displays search-engine style search results.
- **Standard** - This is the default report template for displaying data in a table and can be heavily customized with Template Options.
- **Timeline** - Displays a timeline of actions. It is well suited for home pages, change logs, and other areas where an activity feed of recent actions is needed.
- **Value Attribute Pairs - Column** - Displays key-value pairs for column-based queries. It uses the column header as the label for each item.
- **Value Attribute Pairs - Row** - Displays key-value pairs for row-based queries. It uses the first column for labels, and second column for values.

c. **Template Options** - Select template options to customize the user interface of this component.

   You can also modify the template options when running the application using **Live Template Options**. Live Template Options enable you to customize your application in real time.

   To learn more see Page Designer Help and Managing Template Options.

d. **CSS Classes** - Enter additional space delimited CSS classes you want to add to your report. Your report template must include the `#COMPONENT_CSS_CLASSES#` substitution string.

e. **Show Null Values as** - Enter the text you want to display for null columns.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.6.3 Editing Classic Report Pagination

Edit classic report pagination.

To control classic report pagination:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.

4. Under **Pagination**, edit the following:

   a. **Type** - Select the pagination type for this report. Generally only a subset of the results are currently displayed in the report. Pagination provides the user with information about the number of rows and the current position within the result set. Pagination also defines the style of links or buttons that are used to navigate to the next or previous page.

   b. **Display Position** - Select where the report pagination displays.

   c. **Partial Page Refresh** - Specify whether to use Partial Page Refresh (PPR) to update the report display.

      If the end user performs an action, such as a pagination request or selection of a new sort column, the current report rows must be refreshed. Setting this attribute to **On** refreshes just the report region. Setting this attribute to **No** refreshes the entire page.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.6.4 Adding Support for XML File or a CSV File Export

Add support for exporting a report as an XML or CSV by selecting a report template.

To export a report as a file:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.

4. Under **Appearance**, edit the following:

   a. **Template Type** - Select **Predefined**.

   b. **Template** - Select **XML** or **CSV**.

      Selecting **XML** prevents the APEX engine from rendering the page and dumps the content to an XML file.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.6.5 Enabling Classic Report CSV Output Option

Enable classic report CSV output option.

To use the enable the CSV output option:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.

4. Under **Download**, edit the following:

a. **CSV Export Enabled** - Specify whether end users can download the report contents to a CSV file. To enable CSV Downloads, you must use a report template with a `#CSV_LINK#` substitution string and set this option to **On**.

b. **CSV Separator** - Enter a column separator. If no value is entered, a comma or semicolon is used depending on your current NLS settings.

c. **CSV Enclosed By** - Enter a delimiter character to delineate the starting and ending boundary of a data value. The default delimiter is double quotation marks.

d. **Link Text** - Specify the text for the link to invoke the CSV download.

e. **Filename** - Specify a name for the CSV download file. If no name is specified, the region name is used followed by the extension `*.csv`.

f. **Supplemental Text** - Enter text that displays at the top of the download file. Use this attribute to for data classification, legal, or warning messages about the data in the file.

> **WARNING:**
>
> Do not use the Supplemental Text attribute with CSV download. If you enable **CSV** download, the defined text will invalidate the CSV file format.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.6.6 Controlling Classic Report Column Breaks

Control classic report column breaks.

You can control if a specific column repeats and how column breaks appear when printed using **Break Formatting** attributes. For example, suppose your report displays employee information by department number. If multiple employees are members of the same department, you can increase the readability by specifying that the department number only appears once.

To control column breaks:

1. View the page in Page Designer:

   Page Designer appears.

2. In the Rendering tab, select the region containing the report.

3. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups.

4. Under **Break Formatting**edit the following:

   a. **Report Sum Label** - Enter the text that displays adjacent to report sum values.

   b. **Break Columns** - Select how many report columns are incorporated into the break formatting. The columns must be consecutive columns, starting from the first column displayed in the report.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.7 Managing Classic Report Column Attributes

Edit column attributes to control the display, features, and column behavior.

- Altering Classic Report Column Layout
  Edit Column attributes to precisely control the report layout.
- Enabling Column Sorting in a Classic Report
  Learn how to enable column sorting.
- Creating a Column Link in a Classic Report
  Use Column Link attributes to create a link from a report to another page in your application or to a URL.
- Defining an Updatable Column in a Classic Report
  Make a column updatable by editing column attributes.
- Defining a Column as a List of Values in a Classic Report
  Render a report column as a list of values.
- Controlling When Classic Report Columns Display
  Control when a column displays using an Authorization Scheme and Server-side Condition.
- Adding a Download Link to a Classic Report
  Create a link within a report to enable users to export the report as a comma-delimited file (.csv) file.

> **See Also:**
>
> Editing Pages in Page Designer

8.10.7.1 Altering Classic Report Column Layout

Edit Column attributes to precisely control the report layout.

For example, you can use these attributes to alter column heading text, change column positioning, hide a column, create a sum of a column, or select a sort sequence.

To edit the Column attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.
4. Edit the appropriate attributes in the Property Editor. Attributes are organized in groups.

> **Tip:**
>
> To view Help, select the attribute and click the Help tab. Once you activate the Help pane, the content that displays changes every time you select another attribute.

The following table describes common ways to edit report columns.

Table 8-1 Common Report Column Edits for Classic Reports

Description	Developer Action
Hide a column.	Find **Identification**. For **Type**, select **Hidden**.
Edit the column heading text.	Find **Heading**. Edit the **Heading** and **Alignment** attributes.
Change the column heading text alignment.	Find **Heading**. Under Heading, make a new selection from **Alignment**.
Alter the column display sequence.	Find **Layout**. Edit the **Sequence** attribute.
Enable a unique sort sequence.	Find **Sorting**. For **Sortable**, select **On** and select a **Default Sequence**. Any number of columns can be sort enabled. However, at least one column must have a Sort Sequence defined.
Enable the sum of a column.	Find **Advanced**. For **Compute Sum**, select **On**.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.7.2 Enabling Column Sorting in a Classic Report

Learn how to enable column sorting.

To enable column sorting:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. Find **Sorting**.
5. Under Sorting, edit the following:
   a. Default Sequence - Select the sort sequence for this column. Each column should have a unique sequence to correctly order the report output.
   b. Sortable -To enable sorting, select **On**. Selecting **On** enables the end user to sort this column in the report.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Tip:**
>
> Column sorting must be enabled if you want columns with null values to display at the top or end of the report. To learn more about the Sort Nulls attribute, see Editing Classic Report Pagination.

8.10.7.3 Creating a Column Link in a Classic Report

Use Column Link attributes to create a link from a report to another page in your application or to a URL.

To create a column link to another page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. Find **Identification**. From Type, select **Link**.
5. Find **Link** and click **No Link Defined**.

   The Link Builder - Target dialog appears.
6. To create a link to another page, in the Link Builder - Target dialog:
   a. Type - Select **Page in this application**.
   b. Page - Specify the target page number.
   c. Set Items - Select a Name and Value to specify session state for an item.
   d. Clear/Reset, Clear Cache - Specify the page numbers on which to clear cache. To specify multiple pages, enter a comma-delimited list of page numbers.
   e. Clear/Reset, Action - Select one of the following:
      - **None** - No clear or reset action when page loads.
      - **Clear Regions** - Clears all region settings on the page.
      - **Reset Regions** - Resets all regions on the page back to the default state, including sort orders, hidden columns and so on.
      - **Reset Pagination** - This option enables backwards compatibility.
   f. Advanced, Request - Specify the request to be used.
   g. Click **OK**.
7. To create a link to a URL, in the Link Builder - Target dialog:
   a. Type - Select **URL**.

   b. URL - Enter the URL address.
   c. Click **OK**.
8. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.7.4 Defining an Updatable Column in a Classic Report

Make a column updatable by editing column attributes.

Note that the Oracle APEX engine can only perform updates if:

- A multirow update is defined.
- A PL/SQL process is implemented to process updated data.
- When using the built-in tabular form elements and display types, then the report has to be defined using the type **SQL Query (updatable report)**.

To define updatable column attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.
4. In the Property Editor, find **Identification** and edit the Type attribute.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.7.5 Defining a Column as a List of Values in a Classic Report

Render a report column as a list of values.

A column can be rendered as Radio Group or a static, named, or query-based LOV. For example, a column can be rendered using a select list or a popup list of values. Or, a column can be rendered as read-only text based on a list of values.

This last approach is an effective strategy when creating display lookup values and is particularly useful in regular, nonupdatable reports. This approach enables you to display the value of a column without having to write a SQL JOIN statement.

To render a report column as a list of values:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.

   The columns appear in the Rendering tab.

3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.

4. Find **Identification**. From **Type**, select **Plain Text (based on List of Values)**.
5. Find **List of Values**. Select a **Type**:
   - **Shared Component** - Creates a list of values based on a predefined list of values defined in Shared Components.
   - **SQL Query** - Creates a dynamic list of values based on the SQL Query you provide.
   - **Static Values** - Creates a static list of values is based on the text value you provide.
   - **PL/SQL Function Body returning SQL Query** - Creates a dynamic list of values is based on the SQL Query returned by the entered PL/SQL Function Body you provide.
6. The UI changes based on the **List of Values**, **Type** you select. Fill in all required attributes (identified in red).
7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

8.10.7.6 Controlling When Classic Report Columns Display

Control when a column displays using an Authorization Scheme and Server-side Condition.

Use Authorization Scheme to control access to resources (such as a report column) based on predefined user privileges. For example, you could create an authorization scheme in which only managers can view a specific report column. Before you can select an authorization scheme, you must first create it. Use Server-side Condition

A condition is a small unit of logic that enables you to control the display of a column based on a predefined condition type. The condition evaluates to true or false based on the values you enter in the Expressions fields.

To specify Authorization Scheme and Server-side Condition attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

2. In the Rendering tab, locate the region and expand the Columns node.
3. Select a column to edit.

   The Property Editor displays the attributes for that column. Attributes are organized in groups.

4. If controlling a column with an authorization scheme, select an authorization scheme:
   a. Find **Security**.
   b. Authorization Scheme - Make a selection.
5. If controlling the column with a condition define a condition, define a condition:
   a. Find **Server-side Condition**.

b. Type - Make a selection.

c. Depending upon your selection, enter an expression or value in the fields provided.

If the authorization is successful and the condition type display evaluates to true, the column displays.

6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> - Providing Security Through Authorization
> - Understanding Conditional Rendering and Processing
> - Available Conditions

8.10.7.7 Adding a Download Link to a Classic Report

Create a link within a report to enable users to export the report as a comma-delimited file (.csv) file.

To add a CSV link to a report, you must enable **Automatic CSV Encoding**. When using Automatic CSV Encoding, the report template is not important. You can include a CSV link with any report template that has the CSV export substitution string defined.

> **See Also:**
>
> Accessing the Globalization Page and Automatic CSV Encoding

8.11 Managing Sorting with Order By Item

Create an inline static LOV by configuring region-level Order By Item attributes.

- Order By Item Sort Example
  View an Order By Item sort example.
- Configuring Sorting with Order By Item
  Configure region Order By attributes to create an inline static LOV.

8.11.1 Order By Item Sort Example

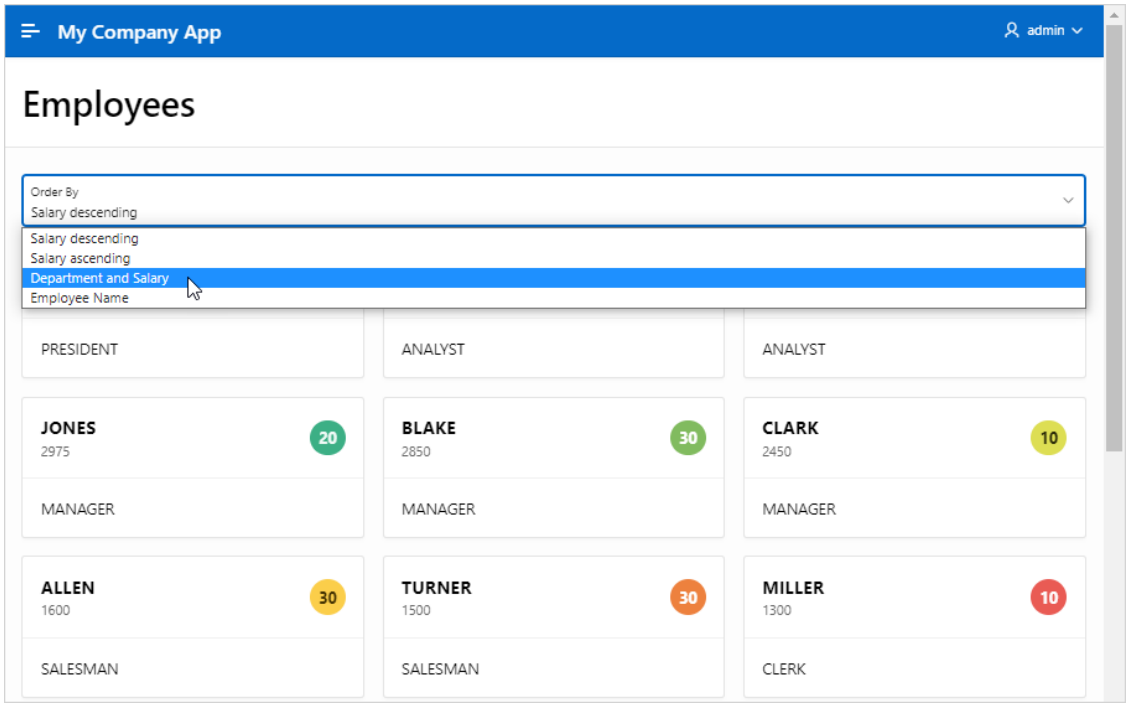
View an Order By Item sort example.

Configuring an Order By Item enables end users to choose region ordering from a set of predefined options. Including an Order by Item is particularly useful for region types which do not support ordering on their own, such as a Cards region. Order By Item is supported by all region types except map regions, tree regions, interactive reports, and interactive grids.

An Order By Item configuration consists of predefined order by clauses which are maintained at the region level and a page item which is available to end users at runtime. While the default

page item is a Select List, other item types, such as radio groups, plug-in based types or even hidden items, are also supported.

The following example shows a cards region. An **Order By** select list displays above the cards and contains the options: Salary descending, Salary ascending, Department and Salary, and Employee Name.



8.11.2 Configuring Sorting with Order By Item

Configure region Order By attributes to create an inline static LOV.

The following example configures an Order By Item for a Cards region built on the `EMP` table. The `EMP` table is available in the sample dataset, EMP / DEPT.

> **Tip:**
>
> In the following task, steps 1 to 5 creates a cards region. If you already have a cards region, skip to step 6.

To configure an order by item for a cards region:

Create a cards region:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create a new cards region:
   a. Click the **Layout** tab in the central pane.

b. In the Gallery, click the **Regions** tab. Right-click **Cards** and select **Add To**, and select the appropriate location.

> **Tip:**
>
> You can also select the Cards region in the Gallery and drag it to the appropriate location in the Layout tab.

c. Page Designer indicates what to do next. If there are errors, the **Show Messages** icon displays in the Page Developer toolbar.

- Click the **Show Messages** icon on the Page Designer toolbar.

  The Messages dialog displays errors that must be addressed.
- Select an error to highlight the associated attribute in the Property Editor.

3. Edit the **Region** attributes:

   a. In the Property Editor, select the **Region** tab (if not already selected).

   b. **Identification**, **Title** - Enter a region title. For example, enter `Employees`.

   c. Under **Source**:

   - Location - Accept the default value, **Local Database**.
   - Type - Accept the default value, **Table/View**.
   - Table Name - Enter `EMP`. You can also type in the name or pick from the list.

   d. Click **Save**.

4. Edit the cards region Attributes:

   a. In the Property Editor, click the **Attributes** tab.

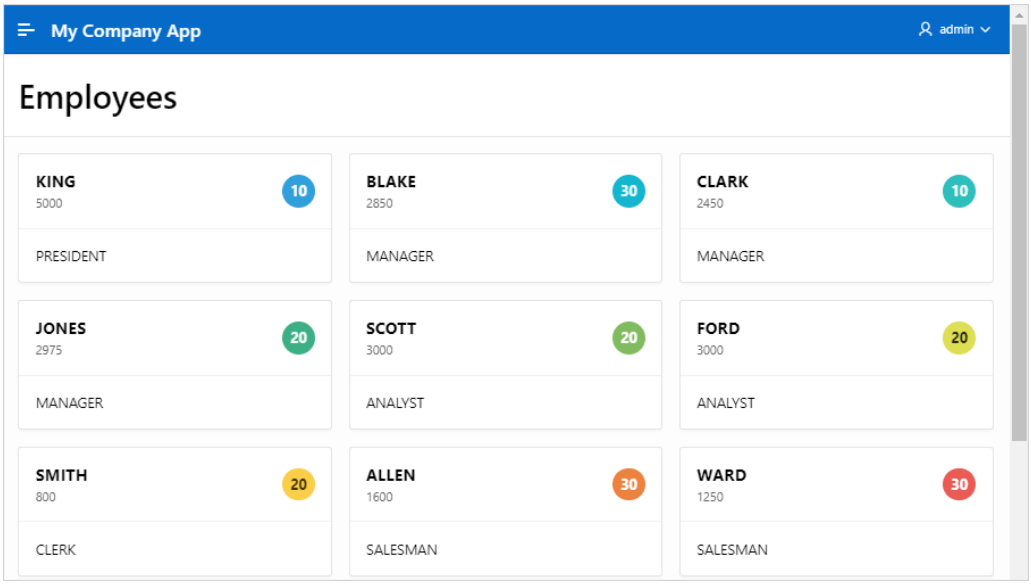
   b. Edit the following cards region Attributes:

   - Title - Select the column to be used as title. Select **ENAME**
   - Subtitle - Select the column to be used as subtitle. Select **SAL**
   - Body - Select the column to be used as body. Enter Select **JOB**
   - Icon and Badge:
     - Icon Source - Select **Initials**
     - Icon Column - Select **DEPTNO**
     - Icon Position - Select **End**

5. Preview the page.

   a. Click **Save or Save and Run Page**.

   The cards page loads, displaying a cards for each employee. Each card includes the employee name, salary, job title, and an icon with the department number.



b. Return to Page Designer. Click Page *XX* on the Runtime Developer toolbar where *XX* is the page number.

Configure the Order By Item:

6. Enable Order By Items to create an inline static LOV:
   a. In the Rendering tab, verify that cards region, **Employees**, is selected.
   b. In the Property Editor, click the **Region** tab find **Order By**.
   c. Order By, Type - Select **Item**.
   d. Order By, Iten - Click the **No Order By Item** button.

   The Order by Item dialog appears.
7. In the Order By Item dialog:
   a. Order By Clauses - Enter multiple Order By Clauses from which users can select at runtime. Each can contain one or more comma-separated column names, including directives, to define the order in which records are retrieved. See Page Designer Help for more details on directives and examples.

      The following creates a LOV named Order By which contains the options: Salary descending, Salary ascending, Department and Salary, and Employee Name.

      - Row 1

        Clause - Enter `SAL DESC`

        Key - Enter `SAL1`

        Display - Enter `Salary descending`
      - Row 2

        Clause - Enter `SAL ASC`

        Key - Enter `SAL2`

        Display - Enter `Salary ascending`
      - Row 3

        Clause - Enter `DEPTNO ASC, SAL DESC`

        Key - Enter `DEPT1`

Display - Enter `Department and Salary`

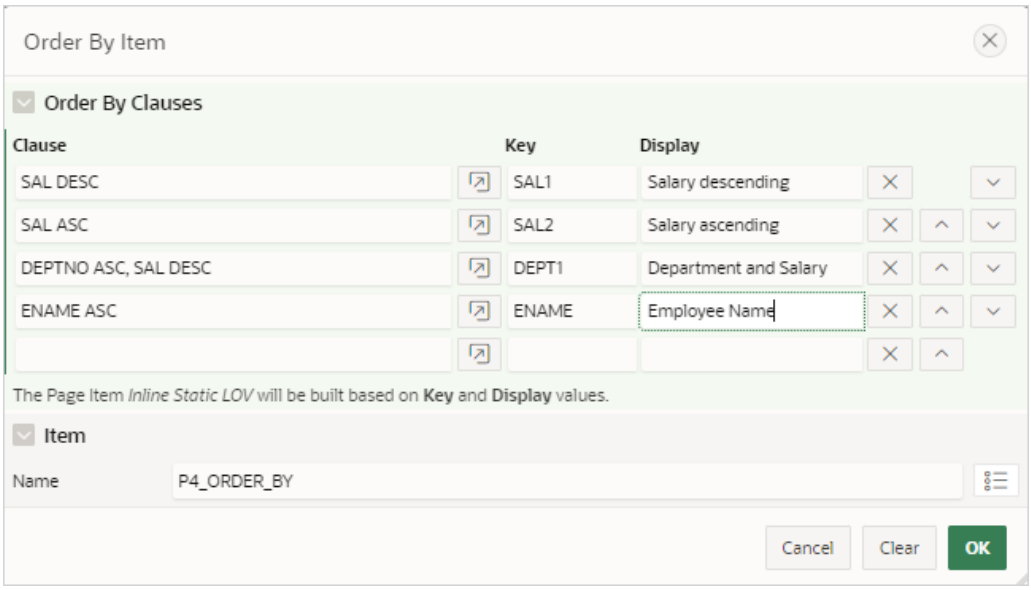
- Row 4

  Clause - Enter `ENAME ASC`

  Key - Enter `ENAME`

  Display - Enter `Employee Name`

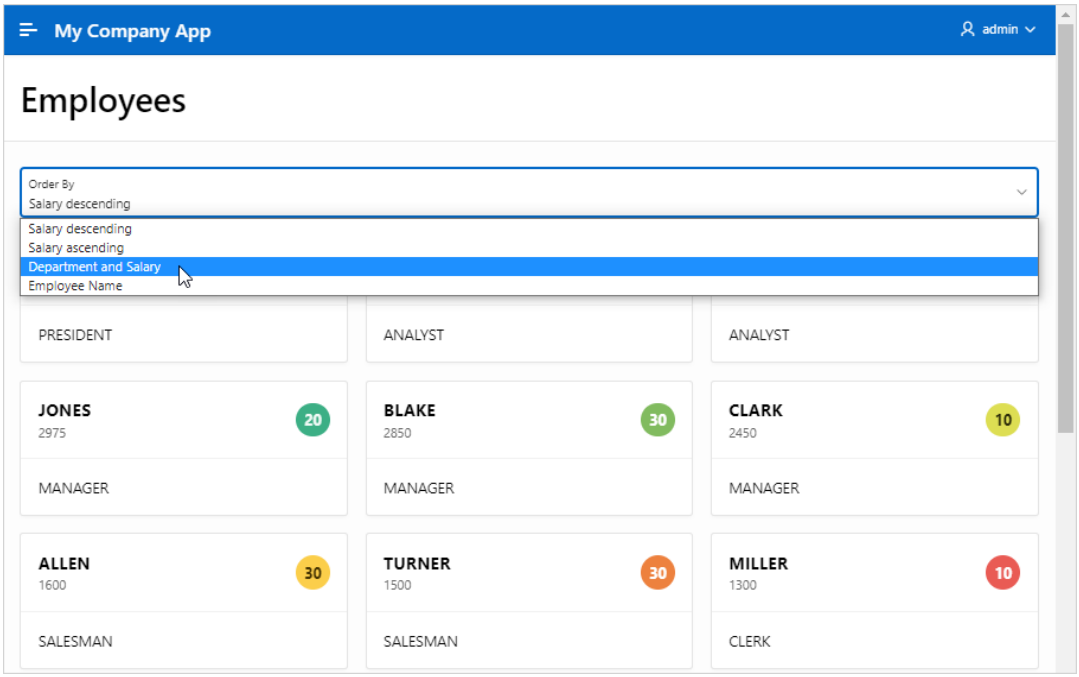
b. Item - Displays the page item to be built based on the Key and Display values defined under Order By Clauses. You can accept the Item Name or to make a new selection by clicking the **List of Values: Name** icon.



c. Click **OK**.

8. Click **Save or Save and Run Page**.

   The following example shows the revised cards region. An **Order By** select list now displays above the cards. Note, however, you can position the item anywhere on the page in Page Designer.



Select options from the **Order By** select list to alter the sort order.

8.12 Printing Report Regions

Configure a report region to print by exporting it to several different formats.

> **Tip:**
>
> You must enable network services to use report printing if you configured an external print server. See Enabling Network Services in Oracle Database.

- About Report Printing
  Learn about region report printing.
- About Printing Reports to PDF and Excel
  Learn about printing reports to PDF and Microsoft Excel.
- Configuring Report Printing at the Application Level
  Configuring report printing at the application level.
- About Classic Report Printing Methods
  Learn about printing classic report regions.
- Configuring Classic Report Printing Attributes
  Learn about printing a report region by configurng Printing Attributes.
- About Report Queries
  Learn about report queries.
- About Report Layouts
  Learn about report layouts.

See Also:

- Configuring Interactive Report Download Options
- Configuring Interactive Grid Download Formats

8.12.1 About Report Printing

Learn about region report printing.

By taking advantage of region report printing, your application users can view and print reports that have a predefined orientation, page size, column headings, and page header and footer. Classic reports, interactive reports and interactive grids also have the ability to export to PDF, Microsoft Excel, HTML, and Comma Separated Values (CSV). Additional export formats require the use of an external print server.

Oracle APEX provides the following report printing configuration options:

- **None (Native)** - Select this option if you do not use an external print server. This option enables you to use the basic printing functionality, which includes CSV, HTML, PDF, and XLSX.
- **Oracle Document Generator Pre-built Function** - Select this option if you are using Oracle Document Generator Pre-built Function. This option enables you to take report query results and convert them from JSON to PDF/Microsoft Office format using Oracle Document Generator Pre-built Function. Select this option to upload your own Microsoft Office templates for printing reports within APEX.
- **Oracle BI Publisher** - This option requires a valid license for Oracle BI Publisher (also known as Oracle XML Publisher). This option enables you to take report query results and convert them from XML to RTF format using Oracle BI Publisher. Select this option to upload your own customized RTF or XSL-FO templates for printing reports within APEX.

  Oracle BI Publisher supports Adobe Portable Document Format (PDF), Microsoft Word Rich Text Format (RTF), Microsoft Excel format (XLS), or Extensible Markup Language (XML). If you choose to use other third-party rendering engines, you can also configure other output formats.
- **APEX Office Print** - Select this option if you are using APEX Office Print. This option enables you to take report query results and convert them from JSON to PDF/MS Office formats using APEX Office Print. Select this option to upload your own MS Office templates for printing reports within APEX.
- **External (Apache FOP)** - Select this option if you are using Apache FOP on an external J2EE server. This option enables you to use the basic printing functionality, which includes creating report queries and printing report regions using the default templates provided in APEX and using your own customized XSL-FO templates.

Note:

Your APEX instance administrator can change the report printing configuration for your instance. See Configuring Report Printing for an Instance in *Oracle APEX Administration Guide*

8.12.2 About Printing Reports to PDF and Excel

Learn about printing reports to PDF and Microsoft Excel.

Oracle APEX includes native support for printing basic reports to PDF and Microsoft Excel. If you want full control and advanced layouts for your reports, Oracle recommends using a report server.

Limitations for native PDF and Microsoft Excel:

- A report row in PDF may span multiple lines and may have a single page break. The text is truncated if a row exceeds this limit.
- The maximum text size of a cell in Microsoft Excel is 32k. The text is truncated If a cell exceeds this limit.

8.12.3 Configuring Report Printing at the Application Level

Configuring report printing at the application level.

Configure the print server type for an application by editing the **Print Server Type** attribute on the Application Definition page. The application-level **Print Server Type** overrides similar settings in Instance Administration.

To configure report printing at the application level:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the application home page, you can access the Definition page in two ways:
   - Click the **Edit Application Definition** button.
   - From Shared Components:
     - Click **Shared Components**.
     - Under Application Logic, click **Application Definition**.

   The Definition page appears.
4. Report Printing:
   a. Print Server Type - Print Server Type controls the print server of the application. This setting overrides similar settings in Administration Services. Options include:
      - **Native Printing** - Use native printing. No external print server required. Custom report layouts are not supported.
      - **Remote Print Server** - Choose a remote server of the type "Print Server" that is configured in the Workspace Utilities.
      - **Use Instance Settings** - Use the Report Printing settings defined at the instance level in Administration Services.

   b. Click **Apply Changes**.

> **Note:**
>
> Document Generator cannot be selected or added as a print server at the application level. It must be configured in Administration Services. To use Document Generator once it is configured, set Print Server Type to **Use Instance Settings**.

5. (Remote Print Servers only) Authentication Required - Specify whether the remote print server requires authentication.
6. (Remote Print Servers only) Credentials - Select a credential defined in Workspace Utilities.

To add a new Remote Print Server:

7. Click **Add Remote Print Server**.

   The Add Remote Print Server wizard appears.
8. Print Server Type - Select the Print Server type:
   - **Oracle Analytics Publisher** - Select this option if you are using a licensed version of Oracle Analytics Publisher. This option enables you to take report query results and convert them from XML to RTF format using Oracle Analytics Publisher. Select this option if you want to upload your own customized RTF or XSL-FO templates for printing reports within APEX.
   - **APEX Office Print** - Select this option if you are using APEX Office Print.
9. Specify the following:
   a. Name - Enter a name for the remote server object here. Remote Server names must be unique within the workspace.
   b. Static Identifier - Use the Static ID to reference the Remote Server in API calls.
   c. Endpoint URL - Enter the Endpoint URL of the external print server.
   d. HTTPS Host Name - Enter the host name to be matched against the common name (CN) of the remote server's certificate for an HTTPS request. If null, the host name in the given URL is used.
   e. Click **Next**.
10. Add the Remote Print Server Credentials:
    a. Authentication Required - Determines whether to use authentication for the remote print server. See field-level help for more details.
    b. Name - Enter a descriptive name for the Web Credential. This name must be unique within the workspace.
    c. Static Identifier - Use the Static ID to reference the Web Credential in API calls.
    d. (Oracle Analytics Publisher only) Username - Enter the Username for Basic Authentication. APEX does not store this information encrypted.
    e. (Oracle Analytics Publisher only) Password - Enter the Password for Basic Authentication. APEX stores this information encrypted, so it cannot be retrieved back in clear text.
    f. (APEX Office Print only) API Key - Enter the API key for authentication. APEX stores this information encrypted, so it cannot be retrieved back in clear text.
11. Click **Apply Changes**.

See Also:

- Creating a Remote Server
- Configuring Report Printing in a Full Development Environment in *Oracle APEX Administration Guide*

8.12.4 About Classic Report Printing Methods

Learn about printing classic report regions.

There are two ways to print classic report regions:

- **Configure Printing Attributes for a Report Region.** You can print a report region by configuring the Printing Attributes for the region. See Configuring Classic Report Printing Attributes,
- **Create a Report Query**. You can print a report by defining a report query as a Shared Component. See Printing a Report Region by Defining a Report Query

Both report regions and report queries can be downloaded in the following formats:

- PDF - Adobe Portable Document Format.
- RTF - Microsoft Word Rich Text Format.
- Excel - Microsoft Excel format. Note that when BI Publisher is configured as print server and a custom report layout is specified, the export is not a true `.xls` file because the content is HTML-based.
- HTML
- XML - Extensible Markup Language.

To format either a report region or report query, you associate it with a report layout.

Tip:

You can also have the output format specified by an item that determines the output format at runtime. Select the item that holds the format value. Valid values are PDF, RTF (to open the document in Microsoft Word), XLS (to open the document in Microsoft Excel) and HTML (to download the document as an HTML file).

See Also:

Formatting a Report Region or Report Query Using Report Layouts

8.12.5 Configuring Classic Report Printing Attributes

Learn about printing a report region by configurng Printing Attributes.

One approach to printing a report region is to configure Print Attributes. Once configured, these attributes apply only to the current region and cannot be used outside the context of the region.

To configure a classic report region for printing:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region containing the report.
3. Enable the Attribute, **Printing**:
   a. In the Property Editor, select the **Attributes** tab.
   b. Under **Printing**, set **Enabled** to **On**.
4. Configure **Printing** attributes:
   a. In the Property Editor, select the **Printing** tab.
   b. Edit the appropriate Printing attributes.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> - About Classic Report Printing Methods
> - Configuring Report Printing for an Instance in *Oracle APEX Administration Guide*

8.12.6 About Report Queries

Learn about report queries.

You can print a report region by defining a report query as a Shared Component. A report query uses a table, a SQL query, or a REST Data Source to identify the data to be extracted. Note that report queries must be SQL statements, not functions returning SQL statements.

> **See Also:**
>
> Printing a Report Region by Defining a Report Query

8.12.7 About Report Layouts

Learn about report layouts.

To format a classic report region, interactive report region, or report query, you associate it with a report layout. Using report layouts renders the data in a printer-friendly format. If you do not select a report layout, a default XSL-FO layout is used.

> **See Also:**
>
> Formatting a Report Region or Report Query Using Report Layouts

8.13 Understanding BLOB Support in Forms and Reports

Oracle APEX includes declarative `BLOB` support to enable developers to declaratively upload files in forms, and download or display files in reports.

`BLOB` display and download can also be authored procedurally using PL/SQL. This section describes how to upload, download and display files, and how to manage additional file attributes such as `MIME` type and file name that are important for proper management of files stored in `BLOB` columns. Using this functionality, you can easily extend your APEX applications to manage files including images, documents, videos, and so on.

- About BLOB Support in Reports
  Oracle APEX includes `BLOB` support for both classic and interactive reports.
- Including a Download Link
  Include a download link in a report by adding columns to your report query.
- Displaying a BLOB Image in a Report
  Display a `BLOB` image in a report by adding columns to your report query.
- About Working With BLOBs Procedurally
  As an alternative to using the built-in methods of providing a download link, use the `APEX_UTIL.GET_BLOB_FILE_SRC` function.
- About Other Declarative Download Options
  Learn about other declarative options to download files.

8.13.1 About BLOB Support in Reports

Oracle APEX includes `BLOB` support for both classic and interactive reports.

If you use a wizard to create a report and include a column of type `BLOB`, this column will not be included in the report definition. To include a `BLOB` download link, additional attributes have to be configured after generation to provide the download capability.

8.13.2 Including a Download Link

Include a download link in a report by adding columns to your report query.

By default, `BLOB` columns do not display in a report. Reports that include a `BLOB` column will display the text `[unsupported data type]` for the column. To include a download link in a report, you need to add a number column to your report query and select the display type **Download BLOB**.

To include a download link:

1. View the page containing the report in Page Designer.
2. If your report's **Source** is based on a **Table / View**, change this attribute to **SQL Query**.

   The SQL Query attribute automatically generates the corresponding query.
3. Add a new numeric/integer column expression to your query, typically the size of the file referenced by the download link. If the length is 0, the `BLOB` is `NULL`, and no download link is displayed.

   To calculate the file size use an expression such as:

   ```
   ...
      col_n,
      sys.dbms_lob.getlength([blob_column]) as download
   from ...
   ```
4. Find **Identification** of your download column. From **Type**, select **Download BLOB**.
5. Find **BLOB Attributes**. Configure the following:
   a. **Table Name** - Enter the table name, which stores the `BLOB` content you want to display. You can type in the table name or pick from the list of available tables.
   b. **BLOB Column** - Select the `BLOB` column from the table, which stores the `BLOB` content you want to display.
   c. **Primary Key Column(s)** - Select the primary key column(s) from the table, which stores the `BLOB` content you want to display. If the primary key of this table has more than two columns, you must select the ROWID column.
   d. **Mime Type Column** – Optionally, select the mime type column from the table, which stores the `BLOB` content you want to display.
   e. **Filename Column** – Optionally, select the filename column from the table, which stores the `BLOB` content you want to display.
   f. **Last Updated Column** – Optionally, select the last updated column from the table, which stores the `BLOB` content you want to display. Ideally, this column represents the update of the file, not other column's values.
   g. **Character Set Column** – Optionally, select the character set column from the table, which stores the `BLOB` content you want to display.
6. Find **Appearance**:
   a. **Download Text** - Enter the text to use for the download link on the page.
   b. **Content Disposition** – Choose between **Attachment** and **Inline**.

Attachment forces a download of the file, while **Inline** will try to render the file in the browser, if possible. The inline behavior is depending on browser capabilities and configuration with respect to the file's mime type.

7. Click **Save** or **Save and Run Page**.

8.13.3 Displaying a BLOB Image in a Report

Display a `BLOB` image in a report by adding columns to your report query.

If the `BLOB` you are working with is an image, you can display it in a report using the display type **Display Image**.

To display a BLOB that is an image in a report:

1. View the page containing the report in Page Designer.
2. If your report's **Source** is based on a **Table / View**, change this attribute to **SQL Query**.

   The SQL Query attribute automatically generates the corresponding query.
3. Add a new numeric/integer column expression to your query, typically the size of the image file. If the length is 0, the **BLOB** is **NULL** and no image is displayed.

   To calculate the file size use expression such as:

   ```
   ...
      col_n,
      sys.dbms_lob.getlength([blob_column]) as image
   from ...
   ```

4. Identification, Type - Select **Display Image**.
5. Configure BLOB Attributes:
   a. BLOB Attributes, Table Name - Enter the table name, which stores the `BLOB` content you want to display. You can type in the table name or pick from the list of available tables.
   b. BLOB Attributes, BLOB Column - Select the `BLOB` column from the table, which stores the `BLOB` content you want to display.
   c. BLOB Attributes, Primary Key Column(s) - Select the primary key column(s) from the table, which stores the `BLOB` content you want to display. If the primary key of this table has more than two columns, you must select the ROWID column.
   d. BLOB Attributes, Mime Type Column – Optionally, select the mime type column from the table, which stores the `BLOB` content you want to display.
   e. BLOB Attributes, Filename Column – Optionally, select the filename column from the table, which stores the `BLOB` content you want to display.
   f. BLOB Attributes, Last Updated Column – Optionally, select the last updated column from the table, which stores the `BLOB` content you want to display. Ideally, this column represents the update of the file, not other column's values.

   If the `BLOB` column contains a file other than an image and depending on how the browser handles the file type, the content will not display.

   The image will be rendered as-is, meaning the image displays with its original height and width in pixels.
6. Accessibility, Image Description - Provide text or a reference to a column containing the text to describe the image. This text will be used as alternative text for the image and will

be rendered with the `<alt>` tag. If no Image Description is provided, then the tag with double quote will be rendered. To learn more, click the Help tab in the central pane.

7. To add some custom CSS to your page so that the images fits in your report:
   a. Advanced, Static ID - Enter the static ID for the column, for example, `my_report_image_column`. To learn more, click the Help tab in the central pane.
   b. CSS, Inline Attribute - Add CSS code to the **Inline Attribute**. For example, to display the image with a height of 25 pixels you would add:

```
td[headers="my_report_image_column"] img {
    height: 25px;
}
```

8. Click **Save** or **Save and Run Page**.

8.13.4 About Working With BLOBs Procedurally

As an alternative to using the built-in methods of providing a download link, use the `APEX_UTIL.GET_BLOB_FILE_SRC` function.

One advantage of using the `APEX_UTIL.GET_BLOB_FILE_SRC` function is the ability to more specifically format the display of the image (with height and width tags). Note that this approach is only valid if called from a valid Oracle APEX session. Also, this method requires that the parameters that describe the `BLOB` to be listed as the format of a valid item within the application. That item is then referenced by the function.

> **See Also:**
>
> GET_BLOB_FILE_SRC Function in *Oracle APEX API Reference*

8.13.5 About Other Declarative Download Options

Learn about other declarative options to download files.

Oracle APEX also offers other declarative options to download files, such as a built-in processes or dynamic action plug-ins. Each of these solutions offer simple downloads based on a SQL query you provide.

> **See Also:**
>
> Managing Dynamic Actions and Understanding Page Processes

9
Developing Forms

Developers can create a variety of different application forms either manually or by using wizards. Some forms enable users to update a single row in a table, while other forms enable users to update multiple rows in a table.

> **Tip:**
>
> Run the **Upgrade Application** utility to automatically upgrade legacy forms to the new form region. See Running Application Upgrade.

- Understanding Form Types
  When creating an application, developers can create four basic form types: editable interactive grid, form, report with form, or master detail.
- Creating a Form
  Developers can create forms using the Create Application Wizard and Create Page Wizard.
- Configuring Lost Update Detection
  Prevent form users from overwriting one another by configuring the Lost Update Detection.
- About Making an Interactive Grid Editable
  Developers determine whether the underlying data is read-only or editable by users. An interactive grid presents users a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.
- Creating Master Detail Forms
  Developers can create a master detail form with either the Create Application Wizard or the Create Page Wizard.
- Validating User Input in Forms
  Developers can create validation and an associated error message to check the data a user enters before processing.
- Understanding BLOB Support in Forms
  Oracle APEX includes BLOB support to enable developers to declaratively upload and download files in forms.

> **See Also:**
>
> - Managing Legacy Tabular Forms
> - Developing Reports

9.1 Understanding Form Types

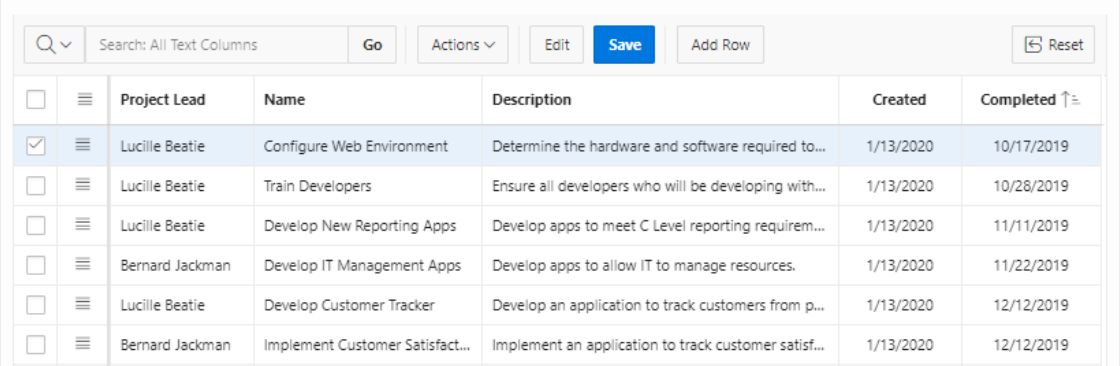
When creating an application, developers can create four basic form types: editable interactive grid, form, report with form, or master detail.

- Editable Interactive Grid
  View an example of an editable interactive grid.
- Form
  View an example form.
- Report with Form
  View an example report with form.
- Master Detail Forms
  View an example master detail form.

9.1.1 Editable Interactive Grid

View an example of an editable interactive grid.

An interactive grid presents users with a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Functionally, an interactive grid includes most customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse. The following is an example of an editable interactive grid.



Project Lead	Name	Description	Created	Completed
Lucille Beatie	Configure Web Environment	Determine the hardware and software required to...	1/13/2020	10/17/2019
Lucille Beatie	Train Developers	Ensure all developers who will be developing with...	1/13/2020	10/28/2019
Lucille Beatie	Develop New Reporting Apps	Develop apps to meet C Level reporting requirem...	1/13/2020	11/11/2019
Bernard Jackman	Develop IT Management Apps	Develop apps to allow IT to manage resources.	1/13/2020	11/22/2019
Lucille Beatie	Develop Customer Tracker	Develop an application to track customers from p...	1/13/2020	12/12/2019
Bernard Jackman	Implement Customer Satisfact...	Implement an application to track customer satisf...	1/13/2020	12/12/2019

You can create an editable interactive grid with either the Create Application Wizard or Create Page Wizard by enabling the **Editing Enabled** flag when creating the report. In an existing interactive grid, you can make it editable by enabling the **Edit**, **Enabled** under Attributes in Page Designer.

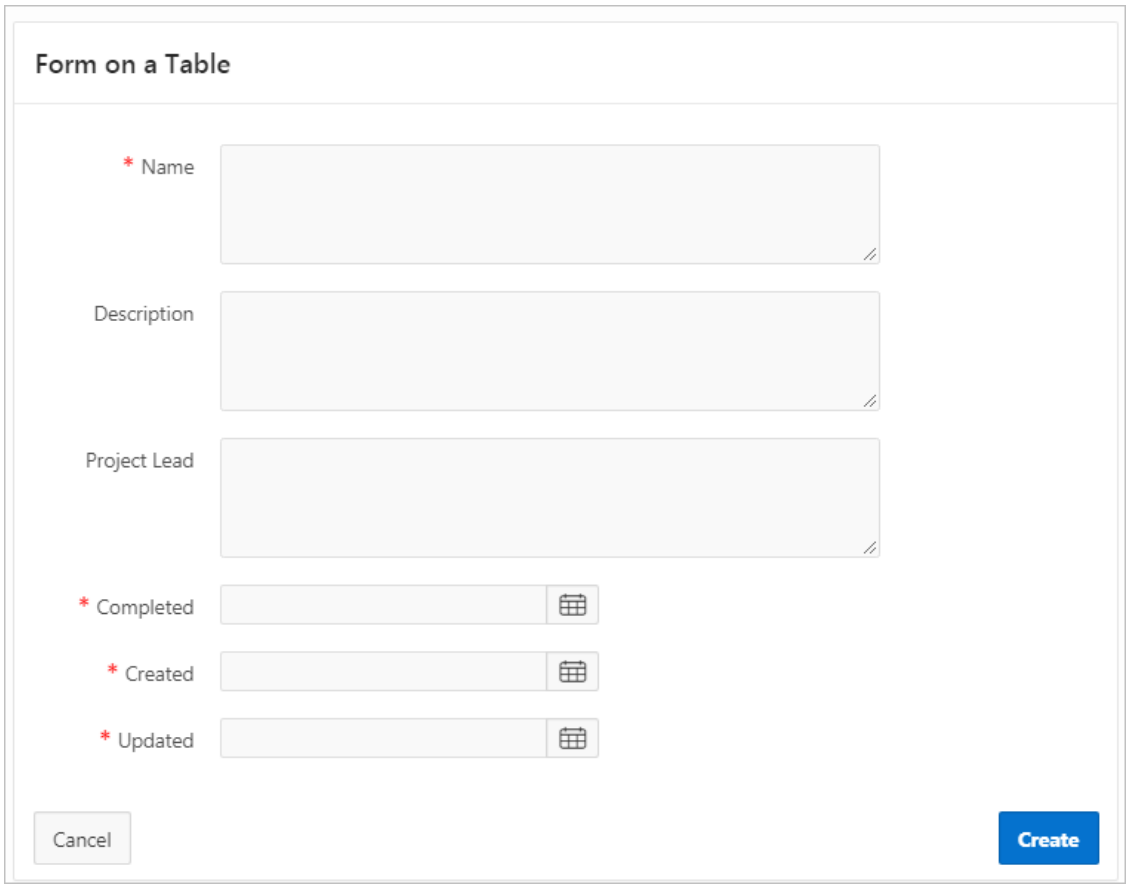
See Also:

- Managing Interactive Grids
- Using an Editable Interactive Grid in *Oracle APEX End User's Guide*

9.1.2 Form

View an example form.

Form creates a form that enables users to update a single row in a database table. The following is an example of a form on table.



Both the Create Application Wizard and Create Page Wizard support the creation of a simple form. The main difference between these two wizards is that the Create Page Wizard offers more customization and data source options. Using the Create Page Wizard, the developer has the option to select a data source (that is, Local Database, REST Enabled SQL Service, or REST Data Source). Additionally, the developer can also specify whether to include and configure breadcrumbs or a navigation menu and select the columns and the order in which they display.

> **See Also:**
>
> Creating a Form Using the Create Page Wizard

9.1.3 Report with Form

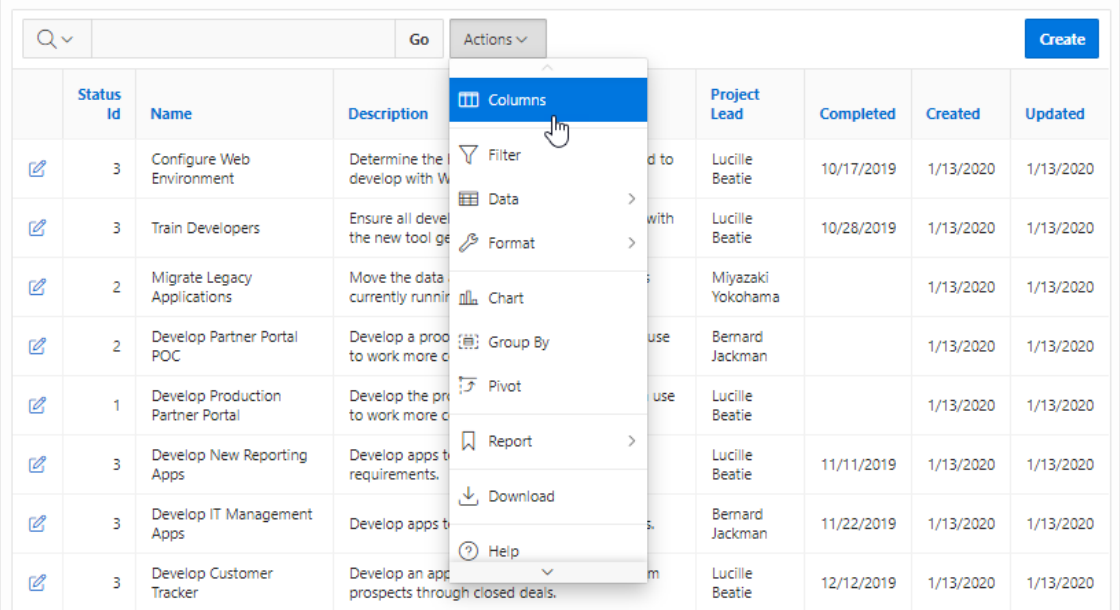
View an example report with form.

Both the Create Application Wizard and Create Page Wizard support the creation of a interactive report or classic report and form combination. In both wizards, you create the report and then enable a flag to add the form.

In the Create Application Wizard, you click **Add Page** and then select either **Interactive Report** or a **Classic Report** and then select the **Include Form** option.

In the Create Page Wizard, you select either **Interactive Report** or a **Classic Report** and then enable **Include Form Page**. The Create Page Wizard offers more customization and data source options. Using the Create Page Wizard, the developer select a data source (that is, Local Database, REST Enabled SQL Service, or REST Data Source) and also specify whether to include and configure breadcrumbs or a navigation menu.

The following is an example of an interactive report with form.



See Also:

- Managing Interactive Reports
- Managing Classic Reports

9.1.4 Master Detail Forms

View an example master detail form.

A master detail form reflects a one-to-many relationship between two tables in a database. Master detail forms enable users to insert, update, and delete values from two tables or views. Typically, a master detail form displays a master row and multiple detail rows within a single

HTML form. Developers can create a single page or two page master detail. You choose the tables on which to build the master and detail regions. Master Detail form options include:

- **Stacked** - Creates a single page master detail with editable interactive grids.
- **Side by Side** - Creates a single page (or Side by Side) master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report.
- **Drill Down** - Creates a two page (or Drill Down) master detail. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grid for the detail.

> **See Also:**
>
> - Creating Master Detail Forms
> - About Master Detail Forms

9.2 Creating a Form

Developers can create forms using the Create Application Wizard and Create Page Wizard.

> **Note:**
>
> Both the Create Application Wizard and Create Page Wizard support the creation of a interactive report or classic report and form combination. In both wizards, you create the report and then enable a flag to add the form.

- Creating a Form Using the Create Application Wizard
  Use the Create Application Wizard to create a form.
- Creating a Form Using the Create Page Wizard
  Use the Create Page Wizard to add a form page to an existing application.

> **See Also:**
>
> - Managing Interactive Reports
> - Managing Classic Reports
> - Managing Legacy Tabular Forms

9.2.1 Creating a Form Using the Create Application Wizard

Use the Create Application Wizard to create a form.

To create a form using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Form**.
8. On Create Form Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Table - Select the table for the form.
   d. Include Report - Select **Include Report** to create a report and form. The first page is an interactive report. Each row provides a link to a form where users can update a record.
   e. Advanced - Expand **Advanced**:
      - Page Help - Enter text to be displayed when the user selects Page Help.

      > **Tip:**
      >
      > This setting requires you select the **About Page** feature. If you select the **About Page** feature, a help icon is generated in the navigation bar with an entry for page help.

   f. Click **Add Page**.
9. Under Features, select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.

   > **Tip:**
   >
   > Click the **Check All** button to select all features.
10. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

> **See Also:**
>
> Creating Applications

9.2.2 Creating a Form Using the Create Page Wizard

Use the Create Page Wizard to add a form page to an existing application.

> **Note:**
>
> Using the Create Page Wizard creates a complex form region which includes multiple form items and processes. Although you can create a region directly in the Rendering tree or drag a region from the Gallery, these approaches will not create these additional components.

To create a form using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Form**.
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

      This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
7. Source Type - Specify the source of the new page. Options include:
   - **Table**:

  - Table/View Owner - Select the owner of the table on which you are building the page.
  - Table/View Name - Select the table or view on which the page will be based.
- **SQL Query**:
  - Enter a SQL SELECT statement - Enter the SQL query.

8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Primary Key Column - Select a primary key column, or accept the default.
11. Branch Pages:
    - Branch Here on Submit - Select a page to branch to after the page processing is complete.
    - Cancel and Go To Page - Select a page to which application shall redirect, when the Cancel button is clicked.
12. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

9.3 Configuring Lost Update Detection

Prevent form users from overwriting one another by configuring the Lost Update Detection.

Lost updates occur when two or more users try to update the same data, and the last update overwrites updates made by the previous user. To prevent this, updates can check the existing column value(s) in the database with the value(s) retrieved when the record was originally queried.

Configuring Lost Update Detection, you configure the Form region Attribute, **Lost Update Type**, and enable the **Prevent Lost Updates** attribute for the corresponding Form - Automatic Row Processing (DML) process.

To enable Lost Update Detection:

1. Create a form region using the Create Page Wizard.
2. View the page containing the form region in Page Designer.

3. In the Rendering tab, select the form region.
4. In the Property Editor, select the **Attributes** tab.

   The Property Editor displays the attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. In the Property Editor, find the **Edit** group and configure the **Lost Update Type** attribute. Options include:
   - **Row Values** - A checksum value is calculated for each row, when initially querying the data. The checksum is calculated by concatenating all of the updateable columns into a string and then generating a unique value. When committing the updated record, this checksum is compared to the checksum value.
   - **Row Version Column** - If your database table includes a column which is incremented every time a record is updated, preferably by a database trigger, then this column can be used instead of calculating checksums. Oracle does not recommend this option if your form updates data in multiple tables.

   Next, update the corresponding Form - Automatic Row Processing (DML) process.
6. In the left pane, click the Processing tab.
7. Find and select the corresponding Form - Automatic Row Processing (DML) process.
8. In the Property Editor, find the **Settings**.
9. For **Prevent Lost Updates**, select **On**.
10. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Editing Pages in Page Designer

9.4 About Making an Interactive Grid Editable

Developers determine whether the underlying data is read-only or editable by users. An interactive grid presents users a set of data in a searchable, customizable report. In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page.

As described in Using Interactive Grids in *Oracle APEX End User's Guide*, users can lock, hide, filter, freeze, highlight, and sort individual columns. Advanced users can also define breaks, aggregations, and computations against columns. Users can also directly customize the appearance of an interactive grid. Users can use the mouse to resize the width of a column and drag and drop columns into different places in the grid.

In an editable interactive grid, users can also add to, modify, and refresh the data set directly on the page. Editable interactive grids expand the functionality of regular interactive grids to enable users to more directly update a grid's structure and contents. Users can add and delete rows, edit cell contents, and refresh the grid with the latest updates.

> **See Also:**
>
> Making an Existing Interactive Grid Editable and Managing Interactive Grids

9.5 Creating Master Detail Forms

Developers can create a master detail form with either the Create Application Wizard or the Create Page Wizard.

- About Master Detail Forms
  A master detail form enables users to query, insert, update, and delete values from two related tables or views. You choose the tables on which to build the master and detail regions.
- Creating a Master Detail Form Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains a Stacked or Side by Side master detail forms.
- Creating a Stacked Master Detail Using the Create Page Wizard
  Use the Create Page Wizard to add a stacked master detail to an existing application.
- Creating a Side by Side Master Detail Using the Create Page Wizard
  Run the Create Page Wizard to add a new page with a Side by Side master detail.
- Creating a Drill Down Master Detail Using the Create Page Wizard
  Run the Create Page Wizard to add a new page with a Drill Down master detail.

> **See Also:**
>
> Creating Master Detail from an Existing Interactive Grid

9.5.1 About Master Detail Forms

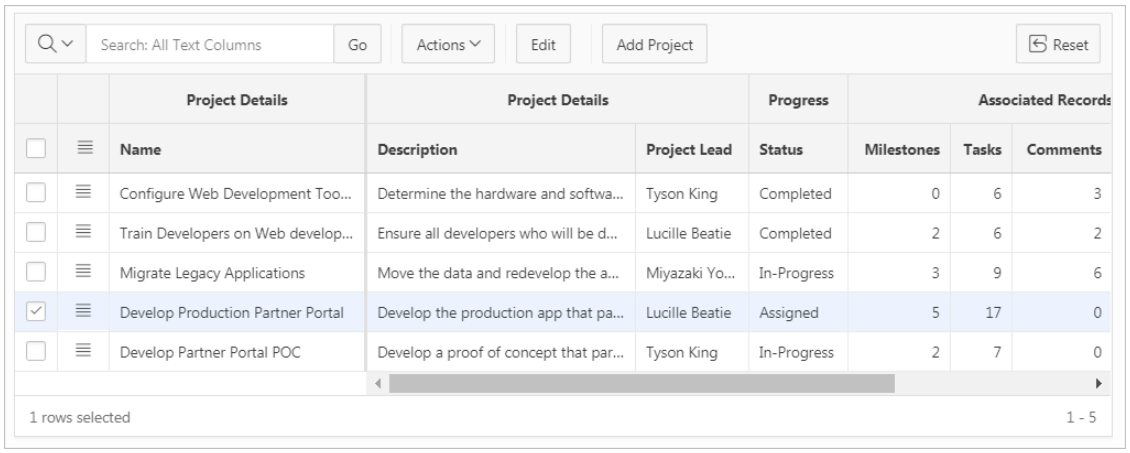
A master detail form enables users to query, insert, update, and delete values from two related tables or views. You choose the tables on which to build the master and detail regions.

A master detail form reflects a one-to-many relationship between tables in a database. Typically, a master detail form displays a master row and multiple detail rows. When you create a master detail form using the Create Page or Create Application Wizards, you choose the tables on which to build the master form and the detail form. **App Builder** supports three types of master detail form options: **Stacked**; **Drill Down**; and **Side by Side**. While the Create Page Wizard supports all master detail form options, the Create Application Wizard supports only Stacked and Side by Side.

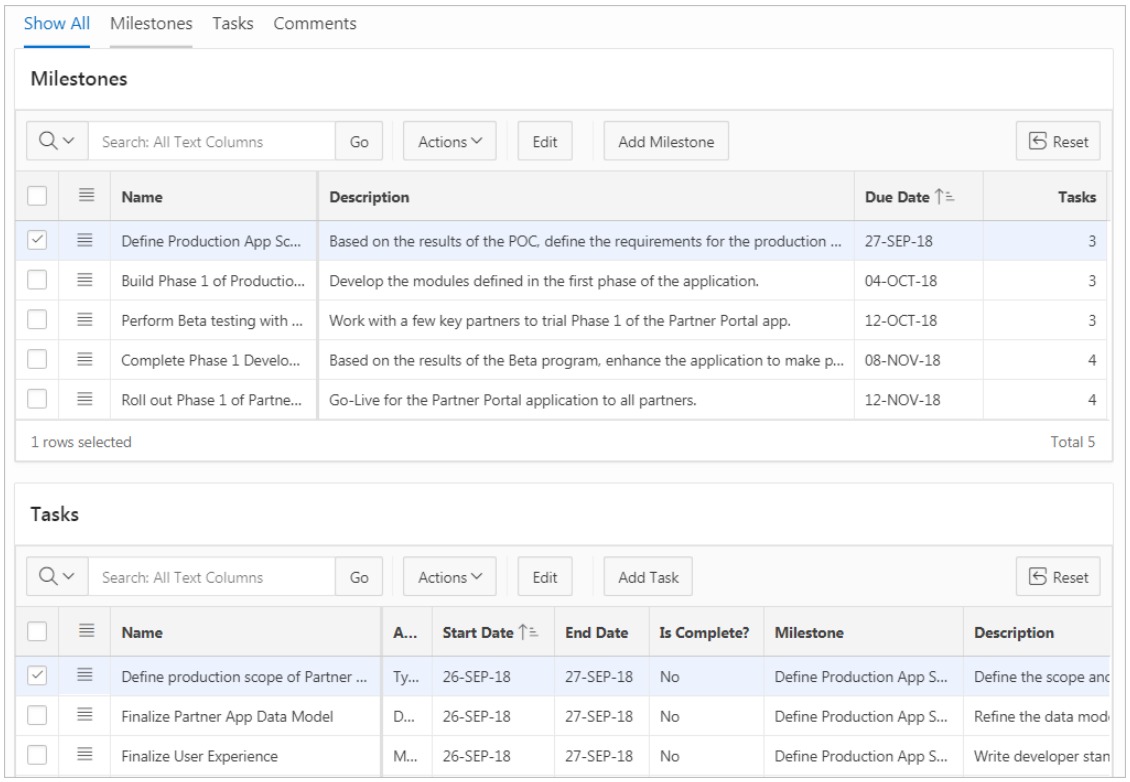
Stacked Master Detail

A **Stacked** master detail form contains a single page with editable interactive grids based on two related tables or views. Users select a row in the master grid to update the detail grids. Developers can create a stacked master detail with either the Create Application Wizard or the Create Page Wizard.

The following example displays master detail interactive grids on the same page. The user has option of interacting with either the master or the detail without leaving the page.

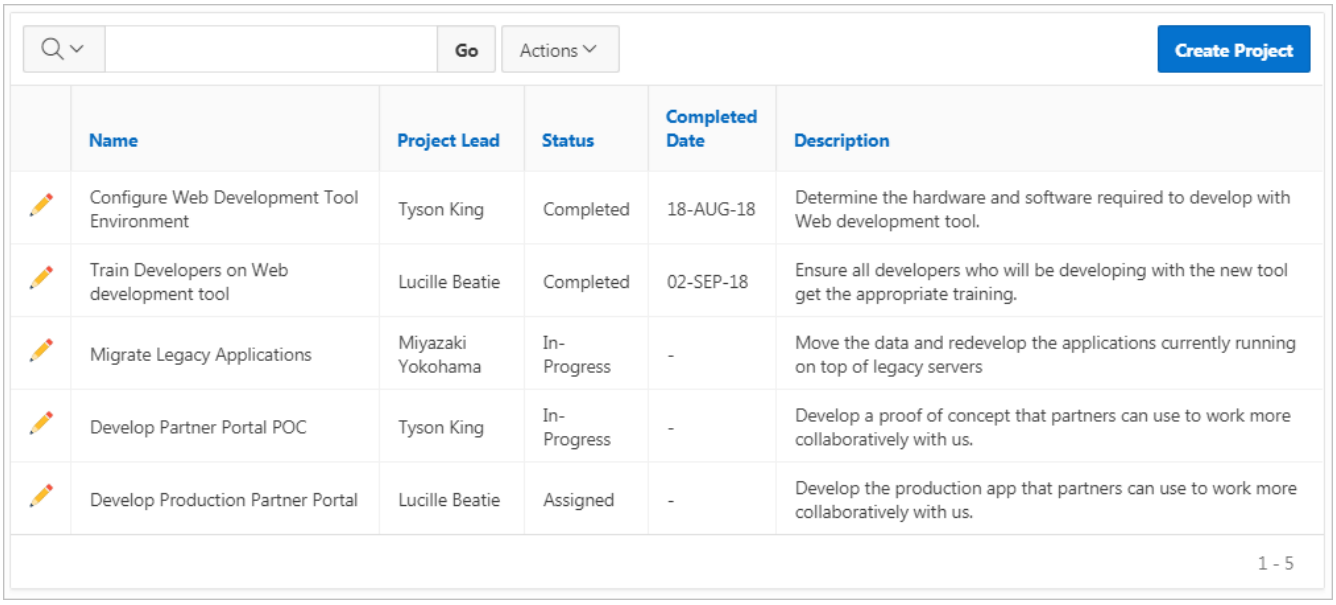


The previous example shows the master interactive grid with the project, *Develop Production Partner Portal* selected. The next example shows detail interactive grids for *Milestones* and *Tasks*.

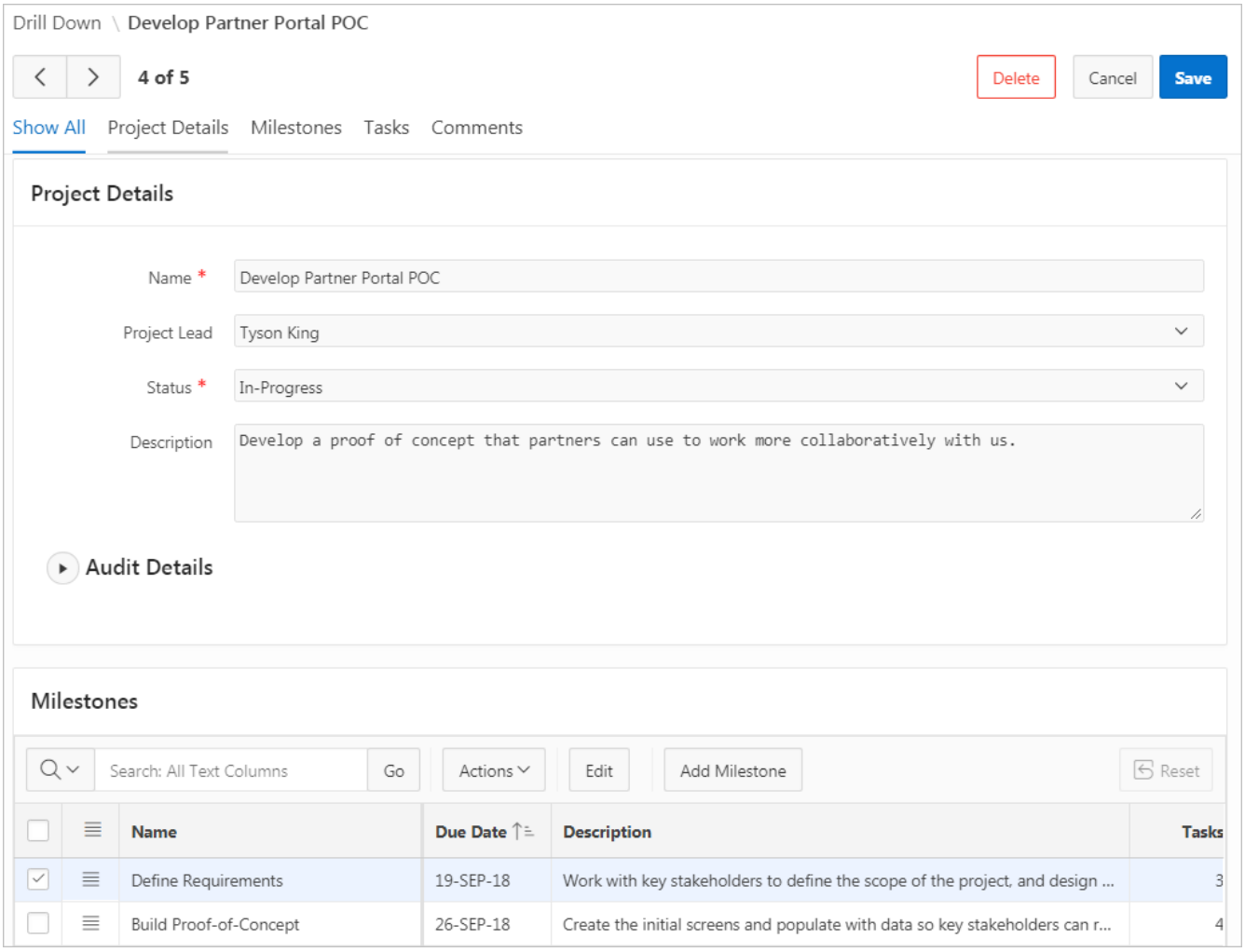


Drill Down Master Detail

A **Drill Down** master detail contains two pages based on two related tables or views. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grids for the detail. Developers can create a Drill Down master detail with the Create Page Wizard.



In this example, users click the Edit icon to view project details. The second page includes both a standard form (for the master) and editable interactive grids.

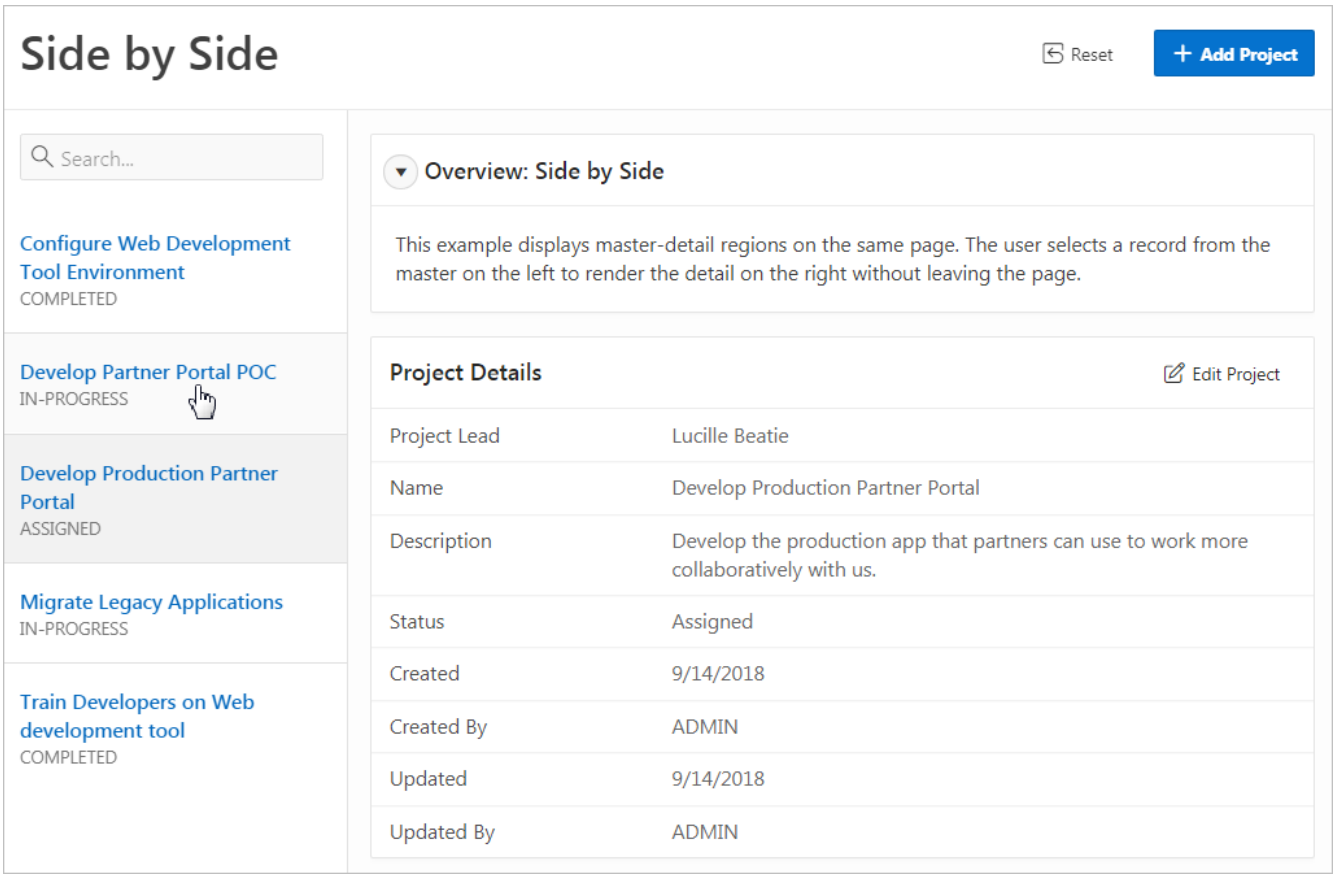


For numeric and date fields, you can also pre-define date and number format masks, or apply those format masks after generating the initial form. Wizard generated master detail forms also automatically create validations for some columns. Validations are created for columns that are set to `NOT NULL` in the underlying table and columns of type `NUMBER`, `DATE` or `TIMESTAMP`. Note that validations are not created for columns if the column is set to read-only, either based on the user's selection, or defined UI Defaults.

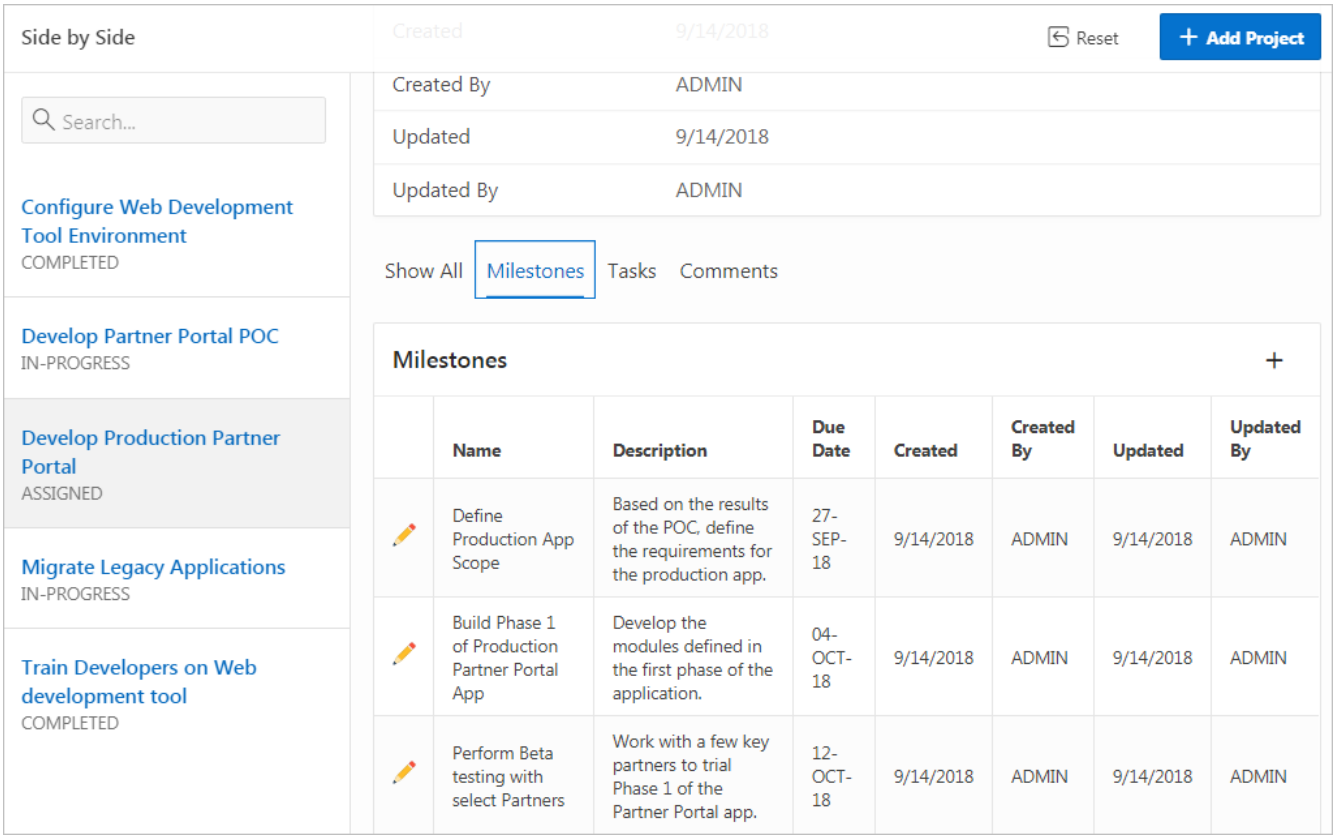
> **Tip:**
>
> You can control the appearance of a specific column by editing column attributes.

Side by Side Master Detail

A **Side by Side** master detail form contains a single page master detail utilizing a side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report(s). Developers can create Side by Side master detail with either Create Application Wizard or Create Page Wizard.



The example that follows shows the Milestones detail interactive report. Users can Milestones by clicking an Edit icon.



9.5.2 Creating a Master Detail Form Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains a Stacked or Side by Side master detail forms.

> **Tip:**
>
> The Create Application Wizard uses general best practices to generate the application and does not include several advanced options included in the Create Page Wizard.

To create a master detail form using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.

6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Master Detail**.
8. On Add Master Detail Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Page Type - Select one of the following:
      - **Side by Side** - Creates a single page (or Side by Side) master detail with a master table and detail table. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report.
      - **Stacked** - Creates single page master detail with editable interactive grids.

      If Page Type is **Side by Side**:
      - Table - Select the master table for the page.
      - Primary Display Column - Select the column to display in the list of master records.
      - Secondary Display Column - Select the descriptive column to display in the list of master records, under the Primary Display Column.
      - Detail Table - Select the detail table.

      If Page Type is **Stacked**:
      - Table - Select the master table for the editable interactive grid.
      - Detail Table - Select the detail table for the page.
   d. Advanced - Expand **Advanced**:
      - Page Help - Enter text to be displayed when the user selects Page Help.

      > **Tip:**
      >
      > This setting requires you select the **Help Pages** feature. If the **Help Pages** feature is not selected, the Help icon is not be generated and end users cannot access page help.

   e. Click **Add Page**.
9. Under Features, select features to include with the application. Features provide application-level functionality and are typically added once per application. To learn more, click the Help icon adjacent to **Features**.

   > **Tip:**
   >
   > Click the **Check All** button to select all features.

10. Under Settings, specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

> **See Also:**
>
> Creating Applications

9.5.3 Creating a Stacked Master Detail Using the Create Page Wizard

Use the Create Page Wizard to add a stacked master detail to an existing application.

A **Stacked** master detail form contains a single page with editable interactive grids based on two related tables or views. Users select a row in the master grid to update the detail grids.

To create a Stacked master detail on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Master Detail**.
5. Master Detail Style - Select **Stacked** and click **Next**.
6. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

      > **Tip:**
      >
      > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
7. Master Data Source:
   - Data Source - Select a Data Source:
     - **Local Database** - Data is sourced from a local database.
     - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
     - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
     - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
     - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

- Table /View Owner - Select the owner of the table on which you are building the page.
- Table/View Name - Select the table or view on which the page will be based.

The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Primary Key Column:
    - Primary Key Column 1 - Select a primary key column.
    - Primary Key Column 2 - Select a primary key column.
11. Click **Next**.
12. For Detail Data Source:
    a. Detail Region Title - Enter a region title for the detail region.
    b. Data Source - Select a Data Source for the page:
       - **Local Database** - Data is sourced from the local database.
       - **REST Enabled SQL Service** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL**.
       - **REST Data Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.

       The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
    c. Show Only Related Tables - Specify whether to restrict the list of detail tables to only those tables related to the master table.
    d. Table /View Owner - Select the owner of the table on which you are building the page.
    e. Table/View Name - Select the table or view on which the page will be based.
    f. Click **Next**.
13. Detail Primary Key:
    - Primary Key Column 1 - Select primary key column.
    - Primary Key Column 2 - Select primary key column.
14. Master Detail Foreign Key - Select the foreign key that relates between the master and detail tables.
15. Click **Create Page**.
16. To view the new master detail, click **Save and Run Page**.

    The rendered page appears.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

9.5.4 Creating a Side by Side Master Detail Using the Create Page Wizard

Run the Create Page Wizard to add a new page with a Side by Side master detail.

A **Side by Side** master detail form contains a single page master detail utilizing a side by side layout and report regions with modal edit windows. The left side contains a master list to navigate to the master record. The right side contains the selected master record and the associated detail report(s).

To create a Side by Side master detail on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Master Detail**.
5. Master Detail Style - Select **Side by Side**.
6. Page Definition:
   a. Starting Page Number - Enter an unused starting page number. The wizard will create multiple pages starting with the number specified.The default is the first available page number within this application.
   b. Master Detail Page Name - Enter the name of the master detail page which contains side by side master detail regions. The left side contains master list to navigate to different master record. The right side contains selected master record and associated detail report.
7. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
8. Click **Next**.
9. Define the Master and Details tables:
   a. Table /View Owner - Select the owner of the table on which you are building the page.

b. Master Table - Select the master table or view. When entering manually, note that the table name is case sensitive.

c. Primary Display Column - Select the column to display as primary column in the report to navigate to a different master record.

d. Secondary Display Column - Select the column to display as secondary column in the report to navigate to a different master record (optional).

e. Detail Table - Select the detail table or view.

   Add additional detail tables as needed.

f. Detail Table - Select the detail table or view.

> **Tip:**
>
> The previous steps assume the Data Source for this master detail is a Local Database. To create a master detail on an external database, such a REST Enabled SQL Service or REST Data Source, click Advanced and select the appropriate Data Source type.

10. Click **Create Page**.
11. To view the new master detail, click **Save and Run Page**.

    The rendered page appears.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

9.5.5 Creating a Drill Down Master Detail Using the Create Page Wizard

Run the Create Page Wizard to add a new page with a Drill Down master detail.

A **Drill Down** master detail contains two pages based on two related tables or views. The first page contains an interactive report for the master table. The second page features a standard form for the master and interactive grids for the detail.

To create a two page Drill Down master detail on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

4. Under Component, select **Drill Down**.
5. Master Page Definition:
   a. Page Number - Enter an unused starting page number. The wizard will create multiple pages starting with the number specified. The default is the first available page number within this application.
   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

6. Master Data Source:
   - Data Source - Select a Data Source for the page:
     - **Local Database** - Data is sourced from a local database.
     - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
     - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
     - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
     - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.
   - Table /View Owner - Select the owner of the table on which you are building the page.
   - Table/View Name - Select the table or view on which the page will be based.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
7. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
8. Click **Next**.
9. Primary Key:
   - Primary Key Column 1 - Select a primary key column.
   - Primary Key Column 2 - Select primary key column (optional).
   - Form Navigation Order - Select the navigation order column used by the previous and next buttons on the Detail Page, which navigates to a different master record.
10. Click **Next**.

11. Detail Page Definition:
    a. Page Number - The page number is an integer value that identifies a page within an application.
    b. Name - Specify a text name for this page.

    > **Tip:**
    >
    > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

12. Detail Data Source:
    - Data Source - Select a Data Source:
      - **Local Database** - Data is sourced from the local database.
      - **REST Enabled SQL Service** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL**.
      - **REST Data Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
    - Show Only Related Tables - Specify whether to restrict the list of detail tables to only those tables related to the master table.
    - Table /View Owner - Select the owner of the table on which you are building the page.
    - Table/View Name - Select the table or view on which the page will be based.

    The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
13. Click **Next**.
14. Detail Primary Key:
    - Primary Key Column 1 - Select a primary key column.
    - Primary Key Column 2 - Select a primary key column (optional).
15. Master Detail Foreign Key - Select the foreign key that relates between the master and detail tables.
16. Click **Create Page**.
17. To view the new master detail, click **Save and Run Page**.

    The rendered page appears.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

> **See Also:**
>
> Understanding Page Types in the Create Page Wizard

9.6 Validating User Input in Forms

Developers can create validation and an associated error message to check the data a user enters before processing.

- About Creating Validations and Error Messages
  Learn about creating validations and error messages.
- Associating a Validation with a Specific Item
  Associate an item with a validation and specify error message text.
- About the Error Message Display Location
  Learn about error message display location.

9.6.1 About Creating Validations and Error Messages

Learn about creating validations and error messages.

You can use validations to check data a user enters before processing. Once you create a validation and the associated error message, you can associate it with a specific item. You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

Creating an inline error message involves these steps:

- Step 1 - Create a validation and specify error message text. To learn more, see Understanding Validations.
- Step 2 - Associate the validation with a specific item.

9.6.2 Associating a Validation with a Specific Item

Associate an item with a validation and specify error message text.

To associate an item with a validation and specify error message text:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In Page Designer, select the Processing tab in the left pane.
3. Under Validating, select the validation you want to associate.

   The Property Editor displays Validation attributes.
4. Under Validation, edit the following attributes:
   - Error, Error Message - Enter the text to be displayed in the event that the validation does not pass.

- Error, Display Location - Select where the error message displays for this validation. Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the associated item label or in a notification area defined as part of the page template.
- Error, Associated Item - Select the item where this validation error message displays. .

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Click **Save**.

9.6.3 About the Error Message Display Location

Learn about error message display location.

The error message display location identifies where a validation error message displays. Validation error messages can display on an error page or inline within the existing page. Inline error messages can display in a notification area (defined as part of the page template) or within the field label.

To create a hard error that stops processes, including any remaining validations, you must display the error on an error page.

9.7 Understanding BLOB Support in Forms

Oracle APEX includes BLOB support to enable developers to declaratively upload and download files in forms.

- About BLOB Support in Forms
  Learn about BLOB support in forms.
- About Uploading and Downloading Files into a BLOB
  Learn about uploading and downloading files into a BLOB.
- About Displaying the BLOB
  Learn about displaying the BLOB.
- About Removing the Image Reference
  Learn about removing the image reference.

9.7.1 About BLOB Support in Forms

Learn about BLOB support in forms.

If you create a form using the Create Application Wizard, create a page of type of **Form** or **Report and Form**, create a region of type **Form**, or add an item to an existing form, any item whose source is a database column of type `BLOB` results in an item of type **File Browse**. When the form is called for `INSERT`, the file selected by the user is loaded into the `BLOB` column. When the form is called for update, a download link is displayed to the right of the Browse button. Users can use this link to download the file.

9.7.2 About Uploading and Downloading Files into a BLOB

Learn about uploading and downloading files into a BLOB.

The defaulted `BLOB` support does not give you all the information a typical application needs to effectively manage a `BLOB`. In addition to knowing that the column is a `BLOB`, more information about the file provides a better experience for the end-user. The File Browse page item has additional settings to facilitate managing this additional information completely declaratively.

> **Tip:**
>
> For more information on File Browse settings, see **File Browse** in About Item Types.

There are two different types of storage types available within the File Browse item type:

- `BLOB` column specific in Item Source Attribute - Completely declarative approach that supports configuration of the additional settings discussed here. This references a `BLOB` in your own database table.
- Table `WWV_FLOW_FILES` - Available for backwards compatibility. Oracle does not recommend using this in new applications.
- Table `APEX_APPLICATION_TEMP_FILES` - Store the uploaded file in a temporary location that you can access with the view `APEX_APPLICATION_TEMP_FILES`. Oracle APEX automatically deletes the file at the end of the session or at the end of the upload request, depending on what you choose for Purge File At.

To provide this additional information, it is recommended that you add additional columns to your base table to store and track the MIME type, file name, last updated date and character set settings. You can accomplish this by extending your table. For example:

```
ALTER TABLE emp ADD
   (ATTACH_MIMETYPE     VARCHAR2(255),
    ATTACH_FILENAME     VARCHAR2(255),
    ATTACH_LAST_UPDATE  DATE,
    ATTACH_CHARSET      VARCHAR2(128));
```

> **Note:**
>
> The character set of the `BLOB` is not automatically set on upload. To store the character set value for your `BLOB`, you must provide an additional page item on your page which is bound to the column you use to store the character set, and where the user will be able to specify the character set for the document they are uploading.

If you manually create a form on a custom table, you can still take advantage of this feature. To do so, use the File Browse item type with a Storage Type setting of `BLOB` column specified in Item Source Attribute, on a page with a DML Process type of `DML_PROCESS_ROW`. This process determines the table name and primary key columns.

9.7.3 About Displaying the BLOB

Learn about displaying the BLOB.

If the BLOB you are working with is an image, you can display it in a form. To handle it declaratively, use the Display Image item type as described in Display Image. To handle it procedurally, see Understanding BLOB Support in Forms and Reports.

9.7.4 About Removing the Image Reference

Learn about removing the image reference.

Because there is no set to `NULL` when using File Browse, if you need to provide a mechanism to remove an image reference, you must include a special Remove Image button to nullify the necessary columns. Consider the following example:

```
UPDATE demo_product_info
   SET product_image = NULL,
       MIMETYPE = NULL,
       FILENAME = NULL,
       IMAGE_LAST_UPDATE = NULL,
       CHARSET = NULL
 WHERE product_id = :P6_PRODUCT_ID;
```

10
Managing Application Components

Oracle APEX supports the creation the following application components: calendars, maps, charts, and application Help.

- Understanding Oracle JET Integration with Oracle APEX
  Oracle APEX includes the Oracle JET (JavaScript Extension Toolkit) library. This section provides information about JET and how it is integrated with APEX.
- Creating Calendars
  App Builder includes support for creating a calendar with monthly, weekly, daily, and list views. Once you specify the table on which the calendar is based, you can create drill-down links to information stored in specific columns and enable drag and drop capability.
- Creating Charts
  Create charts in App Builder by running a built-in wizard or by creating them manually in Page Designer.
- Creating Maps
  Create maps that visualize location data (real coordinates) based on data in a table.
- Creating Help for Your Application
  Help created in App Builder displays on a dedicated Help page. To access Help, users click a link that takes them to a dedicated Help page. This Help page displays page and field-level Help topics specific to the page they are viewing.

> **See Also:**
>
> Controlling Page Layout and Adding Navigation

10.1 Understanding Oracle JET Integration with Oracle APEX

Oracle APEX includes the Oracle JET (JavaScript Extension Toolkit) library. This section provides information about JET and how it is integrated with APEX.

- About Oracle JET
  Learn about Oracle JET.
- How Oracle JET is integrated with Oracle APEX
  Learn about how Oracle JET is integrated with Oracle APEX.

10.1.1 About Oracle JET

Learn about Oracle JET.

Oracle JET (JavaScript Extension Toolkit) is a collection of open source JavaScript libraries with a set of Oracle contributed JavaScript libraries. Oracle JET is targeted at intermediate and advanced JavaScript developers and provides a rich set of UI components that can be easily

integrated into APEX. Oracle JET is a modular toolkit which enables developers to use as much or as little of the features that they desire.

> **See Also:**
>
> Oracle JET Homepage to learn more about Oracle JET.

10.1.2 How Oracle JET is integrated with Oracle APEX

Learn about how Oracle JET is integrated with Oracle APEX.

Currently APEX integrates some parts of Oracle JET, primarily with Chart region types.

> **Note:**
>
> Although Oracle JET provides two-way data binding (using `knockout.js`), currently APEX does not natively utilize this aspect of the toolkit.

Oracle JET uses a module loader (`RequireJS`) to only load modules that are required for specific functionality to work. This means that you do not have to load the entire Oracle JET library for certain functionality and also enables each module to define its own dependencies. At runtime, when a module is requested, the module loader looks at the module being requested for any dependencies to other modules. If there are any dependencies, these will also be loaded.

Module loader can however result in a large number of additional resources being individually loaded at runtime, which can have a negative performance impact. For this reason, APEX makes use of the `RequireJS Optimization Tool`, which:

- provides a way to determine all the dependencies for modules either at design time or as part of a standard build process
- bundles them up into one combined, minified file

`RequireJS Optimization Tool` is loaded at runtime to avoid the need for all the additional requests.

Find the list of the bundles used for `Chart` native type:

- `jetCommonBundle.min.js`
- `chartBundle.min.js`

> **Note:**
>
> You can utilize the Common bundle `jetCommonBundle.min.js` in your plug-ins. This Common Bundle gets included on any APEX page as soon as a file is included with your plug-in, which defines the `[require jet]` prefix. This means that any module your plug-in uses that is already included in the common bundle does not need to be refetched by `requireJS`.

jetCommonBundle.min.js

- **Directory**: `/images/libraries/apex/minified/`
- **Included when**: When a JavaScript file is requested that defines itself as requiring JET, by using the `[require jet]` prefix in APEX, and the page is not in Debug mode
  When debugging, the bundle is not loaded and `requireJS` loads all the individual resources separately.

chartBundle.min.js

- **Directory**: `/images/libraries/apex/minified/`
- **Included when**: A JET chart is included on a page (for example, when a `Chart` region, or `Interactive Grid` region is added to a page), and the page is not in debug mode.

  When debugging, the bundle is not loaded and `requireJS` loads all the individual resources separately.

> **See Also:**
>
> - requireJS homepage to learn more about RequireJS
> - RequireJS Optimization Tool to learn more about RequireJS Optimization Tool

10.2 Creating Calendars

App Builder includes support for creating a calendar with monthly, weekly, daily, and list views. Once you specify the table on which the calendar is based, you can create drill-down links to information stored in specific columns and enable drag and drop capability.

- About Supported Calendars
  App Builder supports two types of calendars: Calendar and Legacy Calendar.
- Sample Calendar
  View a sample calendar in the *Sample Calendar* sample app.
- Creating a Calendar Using the Create Application Wizard
  Create a new application that contains a calendar.
- Creating a Calendar Using the Create Page Wizard
  Use the Create Page Wizard to add a new page with a calendar.
- Creating a Calendar in Page Designer
  Manually create a calendar in Page Designer by providing the SQL query.
- Managing Calendar Attributes
  Once you create a calendar, you can alter the display by editing attributes in the Property Editor.
- About Dynamic Action Support for Calendar
  Use the Dynamic Action Create Wizard to specify an action that is performed when a defined set of conditions occur. You can specify which elements are affected by the action and when and how they are affected.

> **See Also:**
>
> Managing Legacy Calendars

10.2.1 About Supported Calendars

App Builder supports two types of calendars: Calendar and Legacy Calendar.

Calendar is based on the FullCalendar jQuery library and can only be customized through CSS. **Legacy Calendar** is template-based and may be deprecated in the future release. Although Oracle still supports legacy components, the creation of new calendars is based on the FullCalendar jQuery library.

> **Tip:**
>
> In previous releases **Legacy Calendar** was referred to as **Calendar (Template)**, **Classic calendar**, **Easy calendar**, and **SQL calendar**).

You create a calendar based on a table or SQL query you provide. During the creation process, you are prompted to select a date column and display column. Because it offers a better user interface and numerous additional features, Oracle recommends converting all legacy calendars to the latest calendar.

Calendar

The Calendar option supports:

- Rendering calendar events on multiple views (Month, Week, Day, or List).
- Rendering of duration and non-duration based events.
- Ability to modify the start and end dates by dragging and dropping events on different dates. Drag and drop is only supported for local data sources, that is, database objects in the referenced database schema and not on external data sources such as a Google calendar.
- Ability to change the duration by resizing the length of the events, by changing the end date.
- Ability to edit or add new events on calendar using forms by clicking either on events or empty calendar cell.
- Rendering events from external sources using web service calls or Google Calendar feeds.
- Ability to render events titles on single or multiple lines.
- Using different CSS classes, developer can choose different styles for different type of events.
- Ability to download events in PDF printable format
- Support of multiple formats for sharing events (iCal, CSV, XML).
- Ability to add tooltip on Month/Week/Day views to make it easier for users to have a quick look at details of each event.

Legacy Calendar

Legacy Calendars option supports:

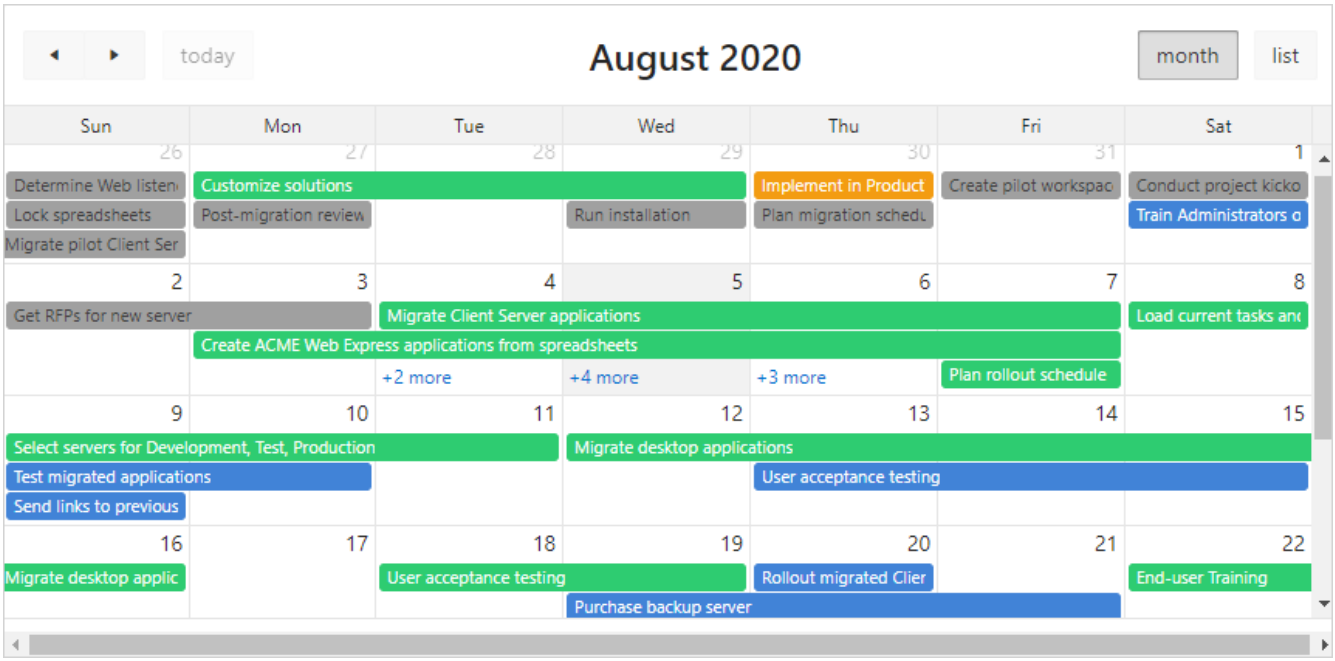
- Rendering of start date only.
- Look and feel defined by templates.
- Modification of start date by dragging and dropping events.
- Calendars that include daily, weekly, and monthly views.
- The date column determines the days on the calendar which contains entries.
- The display column defines a specific row which displays the calendar.

> **See Also:**
>
> Managing Legacy Calendars

10.2.2 Sample Calendar

View a sample calendar in the *Sample Calendar* sample app.

Sample apps are designed to highlight specific functionality and are intended to serve as a developer guide on how to make use of a particular feature. The following is **Monthly Calendar: Projects** from the *Sample Calendar* sample app.



Features of this calendar include:

- Previous - Navigates to the previous month.

- Next - Navigates to the next month.
- Month - Displays a monthly view of all orders.
- List - Displays a list of all orders.

> **Note:**
>
> To view calendar examples, install the sample app, *Sample Calendar*. To learn more, see Installing Apps from the Gallery.

10.2.3 Creating a Calendar Using the Create Application Wizard

Create a new application that contains a calendar.

> **Tip:**
>
> The Create Application Wizard uses general best practices to generate the application and does not include several advanced options included in the Create Page Wizard.

To create a calendar using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Calendar**.
8. On Add Calendar Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Table or View - Select the database object for the calendar.
   d. Display Column - Select the column from the table or view to display in the calendar events.
   e. Start Date Column - Select the column which holds the start date for events displayed on this calendar.
   f. End Date Column - Select the date column from the table or view to use as the end date for calendar events. If **End Date** is specified then duration based events will be displayed.

g. Show Time - Select **Yes** to show the time portion for events and also enable the Weekly and Daily calendar views. Select **No** to only show the dates for events.

h. Advanced - Expand this region to edit the following:

- Set as Home Page - Make this page the home page for the application. Any page previously defined as the home page will be updated.
- Set as Administration Page - Add this page under Administration, rather than as a normal page in the application. Users access this page by clicking Administration in the main menu and then selecting it the Application Administration list.

> **Note:**
>
> The same page cannot be both a Home page and a page under Administration.

i. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to Features.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to Settings.
11. Click **Create Application**.

10.2.4 Creating a Calendar Using the Create Page Wizard

Use the Create Page Wizard to add a new page with a calendar.

To create a calendar using the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Calendar**.
5. Page Definition:

   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
6. Data Source - Select a Data Source for the page:

   - **Local Database** - Data is sourced from a local database.

- **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     - Table/View Owner - Select the owner of the table on which you are building the page.
     - Table/View Name - Select the table or view on which the page will be based. When entering manually, note that the table name is case sensitive.
   - **SQL Query**:
     - Enter a SQL SELECT statement - Enter the SQL query.
8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Edit the calendar attributes:
    a. Display Column - Select the column which holds the text displayed for events on this calendar.
    b. Start Date Column - Select the column which holds the start date for events displayed on this calendar.
    c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If **End Date** is specified, then the calendar displays duration based events.
    d. Show Time - Select **Yes** to show the time portion for events and also enable the Weekly and Daily calendar views. Select **No** to only show the dates for events.
11. Click **Create Page**.
12. To view page, click **Save and Run Page**.

    The rendered page appears.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

10.2.5 Creating a Calendar in Page Designer

Manually create a calendar in Page Designer by providing the SQL query.

> **Tip:**
>
> To create a calendar in Page Designer, you must provide a SQL Query. To create a calendar by selecting a table, use the Create Page Wizard. See Creating a Calendar Using the Create Application Wizard.

To create a calendar in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. If necessary, click the Layout tab in the central pane.
3. In the Gallery, click the **Regions** tab.
4. Right-click **Calendar**, click **Add To**, and select the appropriate location.

   > **Tip:**
   >
   > You can also select **Calendar** with the mouse and drag it to the appropriate location in the Layout tab.
5. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
6. In the Property Editor, edit the appropriate Region attributes:
   - Identification, Title - Enter a region title.
   - Source, Location - Select the location of the database which contains column which holds the CSS Class to style the events displayed on this calendar. Options include:
     – **Local Database** - Data is sourced from a local database.
     – **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST

Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.

- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

What displays next depends upon Source, Location you select.

> **Tip:**
>
> To learn more, select the attribute in the Property Editor and click the **Help** tab in the central pane.

7. Edit the calendar Attributes.
   - In the Rendering tab, locate the Calendar region and click **Attributes**.

     The Property Editor displays the calendar Attributes.
   - Edit the calendar Attributes:
     - Settings, Display Column - Select the column which holds the text displayed for events on this calendar.
     - Settings, Start Date - Select the column which holds the start date for events displayed on this calendar.
8. Click **Save.**

10.2.6 Managing Calendar Attributes

Once you create a calendar, you can alter the display by editing attributes in the Property Editor.

Each calendar has two sets of editable attributes: **Region attributes** and **calendar Attributes**.

- Editing Calendar Attributes in the Property Editor
  Access calendar attributes in the Property Editor.
- Editing an Existing Calendar to Include Add and Edit Functionality
  Edit and existing calendar to include add and edit functionality.
- Adding Calendar Initialization JavaScript Code
  Add a JavaScript function to override the standard jQuery FullCalendar initialization attributes by editing the Initialization JavaScript Code attribute.
- Enabling the Dragging and Dropping of Data in an Existing Calendar
  The Calendar Region supports moving or changing calendar events with drag and drop.

10.2.6.1 Editing Calendar Attributes in the Property Editor

Access calendar attributes in the Property Editor.

Region attributes define the area of the page which functions as a container for the calendar. **Calendar Attributes** specify the template, date columns, and general calendar formatting.

To edit calendar Attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate and select the calendar.

   The Property Editor displays two tabs: **Region** and **Attributes**.
3. In the Property Editor, select the appropriate tab:
   - **Region** - **Region attributes** define the area of the page which functions as a container for the calendar.
   - **Attributes** - **Calendar Attributes** specify the template, date columns, and general calendar formatting.
4. Edit the calendar attributes.

   To view help for an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

10.2.6.2 Editing an Existing Calendar to Include Add and Edit Functionality

Edit and existing calendar to include add and edit functionality.

> **Tip:**
>
> When defining calendar create and edit links for Desktop applications, Oracle recommends defining two separate pages. To create an **Edit Link**, select one of the table or SQL query columns as the Primary Key Column in the Calendar attributes.

To edit an existing calendar to include add and edit functionality:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, locate and select the calendar.
3. In the Property Editor, click the **Attributes** tab.

The Property Editor displays the calendar Attributes.

4. Edit **Settings**, **Create Link** to specify a target page to call when the user clicks an empty cell or an existing calendar entry.
   a. **Settings**, **Create Link** - Click **No Link Defined**.

      The Link Builder dialog appears.
   b. **Type** - Select the type of link target. The UI changes depending upon the Type you select.
   c. **Application** - Only displays if **Type** is **Page in a different application**. Enter the application ID.
   d. **Page** - Specify the target page number.
   e. **Set Items** - Configure the values to be passed from the calendar to the target page. The **Create Link** is called after a date or date range has been selected in the calendar. Therefore, the selected start and end dates must be passed to the Create Page.
      - To pass the selected date or the start date of a selected date range:
        - **Name** - Specify the name of the target page item containing the start date of the event. If a form, this should be a date picker element.
        - **Value** - `&APEX$NEW_START_DATE.`
      - When a date range has been selected, pass the end date as follows:
        - **Name** - Specify the name of the target page item containing the end date of the event. For a form, this should be a date picker element.
        - **Value** - `&APEX$NEW_END_DATE.`

          You can pass additional values in the same way as any other link to an application page.
        - Click **OK**.

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Edit **Settings**, **View/Edit Link** to define a target page to be called when the user clicks an existing entry.

> **Tip:**
>
> To create an Edit Link, you must pass a primary key value to the target page in order for the form to look up the table row.

   a. Settings, Edit Link - Click **No Link Defined**.

      The Link Builder dialog appears.
   b. **Type** - Select the type of link target. The UI changes depending upon the Type you select.

c. **Application** - Only displays if **Type** is **Page in a different application**. Enter the application ID.

d. Page - Specify the target page number.

e. Set Items - Configure the values to be passed from the calendar to the target page. The Create Link is called after a date or date range has been selected in the calendar. Therefore, the selected start and end dates must be passed to the Create Page.

   - Name - Specify the name of the target page item containing the primary key value; typically a hidden item, for example:

     `PX_ID`.

   - Value - Specify the table or SQL query column containing the primary key value, for example:

     `&ID.`

   - Click **OK**.

6. Click **Save**.

10.2.6.3 Adding Calendar Initialization JavaScript Code

Add a JavaScript function to override the standard jQuery FullCalendar initialization attributes by editing the Initialization JavaScript Code attribute.

Note that the function must return a JavaScript Object containing the calendar initialization attributes as parameters.

To add initialization JavaScript code:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. In the Rendering tab, locate and select the calendar.

   The Property Editor displays Region attributes.

3. In the Property Editor, click the **Attributes** tab.

   The Property Editor displays the calendar Attributes.

4. Find **Advanced**.

   > **Tip:**
   >
   > To search for a group or an attribute, enter a keyword in the Filter Properties field. The groups and attributes containing the term appear.

5. In **Advanced**, **Initialization JavaScript Code**, enter a JavaScript function to override the standard jQuery FullCalendar initialization attributes.

   Example: Calendar based FullCalendar v5:

```
function ( pOptions) {
    pOptions.slotMinTime            =
```

```
"07:00:00";                                                 // hide slots
before minTime
    pOptions.slotMaxTime            =
"21:00:00";                                                 // hide slots
after maxTime
    pOptions.dayHeaderFormat        = { weekday: 'short', month:
'numeric', day: 'numeric' };   // week view column headings
    pOptions.slotDuration           =
"00:15:00";                                                 // custom slot
duration
    pOptions.weekNumbers            =
true;                                                       // show week
numbers
    pOptions.weekText               =
"CW";                                                       // heading for
week numbers
    pOptions.weekNumberCalculation  =
"ISO";                                                      // use "ISO"
week numbers
    pOptions.displayEventTime       =
true;                                                       // show event
time ...
    pOptions.displayEventEnd        =
false;                                                      // ... but not
the end time
    pOptions.disableKeyboardSupport =
true;                                                       // disable
builtin keyboard navigation
    pOptions.windowResize           =
null;                                                       // suppress
automatic switch to list view on small screens
    return pOptions;
}
```

Example: Calendar based FullCalendar v3 (Deprecated):

```
function ( pOptions) {
    pOptions.titleFormat      = "[Conference
Schedule]";                  // custom title
    pOptions.minTime          =
"07:00:00";                              // hide slots before minTime
    pOptions.maxTime          =
"18:00:00";                              // hide slots after maxTime
    pOptions.columnFormat     = {month: '', week: 'dddd', day:
'dddd'};   // week view column headings
    pOptions.slotDuration     =
"00:15:00";                              // custom slot duration
    pOptions.weekNumbers      =
true;                                    // show week numbers
    pOptions.weekNumberTitle  = "CW";
    return pOptions;
}
```

> **Tip:**
>
> Calendars based on FullCalendar v3 have been deprecated and will be desupported in a future release.

6. Click **Save**.

10.2.6.4 Enabling the Dragging and Dropping of Data in an Existing Calendar

The Calendar Region supports moving or changing calendar events with drag and drop.

To enable support for drag and drop for a calendar region:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Property Editor, click the **Attributes** tab.

   The Property Editor displays the calendar Attributes.
3. Find **Settings** and edit the following attributes:
   a. **Settings**, **Primary Key Column** - Select a table or SQL Query column.

      The Drag and Drop field appears in the Property Editor, below Additional Calendar Views.
   b. **Settings**, **Drag and Drop** - Select **On**.

      Drag and Drop PL/SQL Code appears.
   c. **Settings**, **Drag and Drop PL/SQL Code** - Enter the PL/SQL code to execute when the end user finishes a drag and drop for an event.

      Within the PL/SQL code, you can reference the ID of the dragged event, the new start, and end end dates as:

      `:APEX$PK_VALUE, :APEX$NEW_START_DATE amd :APEX$_NEW_END_DATE`

      For example, the following code changes the event row in the `PROJECTS` table after the end user has dragged the event to a new date.

      ```
      begin
           update projects
              set start_date = to_date(:APEX$NEW_START_DATE,
      'YYYYMMDDHH24MISS'),
                  end_date = to_date(:APEX$NEW_END_DATE, 'YYYYMMDDHH24MISS')
            where id = :APEX$PK_VALUE;
       end;
      ```

      To learn more, click the **Help** tab in the central pane.
4. Click **Save**.

10.2.7 About Dynamic Action Support for Calendar

Use the Dynamic Action Create Wizard to specify an action that is performed when a defined set of conditions occur. You can specify which elements are affected by the action and when and how they are affected.

Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. When working with dynamic actions, you should be mindful of the fact that the more dynamic actions you add to a page, the greater your overall page size. This is because the dynamic action framework emits additional code to the client for each dynamic action defined, which then also must be downloaded and executed by the framework in the client.

Dynamic actions contain a condition that can be specified to control the action. APEX components such as reports or charts can react to specific calendar events. For example, if you view a calendar and click a button to view a subsequent month, you can create a dynamic action to automatically refresh an associated report.

To use this feature, the developer creates a new dynamic action on the Calendar page in Page Designer. Select the new dynamic action in the Dynamic Action tab in the left pane of Page Designer. Once created, the new dynamic action displays in red with the label **New**. Then the developer edits the dynamic action attributes in the Property Editor. In Property Editor, locate and expand **When** attribute. Under the **When** attribute, for **Event**, select one of the following events:

- **Date Selected [Calendar]**

  This event fires when the user selects an empty date or date range either with the mouse or with the keyboard. If the developer provides a "create link" in the calendar attributes, no event fires.

  Information about the current view is being passed as the "data" object:

  ```
  { "newStartDate"::: "newEndDate"::: }

  { "newStartDate":{first day of selection} (YYYYMMDDHH24MISS),
   "newEndDate": {last day of selection} YYYYMMDDHH24MISS }
  ```

- **Event Selected [Calendar]**

  This event fires when the user clicks an existing event with the mouse or selects it with the keyboard. When the developer provided an "edit link" in the Calendar attributes, no event is being fired. The "data" object contains the jQuery FullCalendar JSON representation of the event.

  > **See Also:**
  >
  > Full Calendar documentation for Event Object.

- **View Changed [Calendar]**

  This event fires when either the view type changes (day, week, month, list) or when the view changes to its previous or next view page. The "data" object contains the following information:

```
{ "viewType": {view type, day, week, month or list},
 "startDate": {first day of view} (YYYYMMDDHH24MISS),
 "endDate": {last day of view} (YYYYMMDDHH24MISS) }
```

> **See Also:**
>
> Managing Dynamic Actions and Debugging Dynamic Actions for information on how to debug problems

10.3 Creating Charts

Create charts in App Builder by running a built-in wizard or by creating them manually in Page Designer.

- About Supported Charts
  App Builder supports charts based on the Oracle JavaScript Extension Toolkit (Oracle JET) Data Visualizations.
- Sample Charts
  View chart examples in the *Sample Charts* sample app.
- Creating a Chart Using the Create Application Wizard
  Create a new application that contains a chart page.
- Creating a Dashboard Page Using the Create Application Wizard
  Create a new application that contains a Dashboard page with up to four charts.
- Adding a Chart Using the Create Page Wizard
  Run the Create Page Wizard to add a new chart to an existing application.
- Adding a Chart in Page Designer
  Add a chart to a page by creating it manually in Page Designer.
- Editing Charts
  You can alter how a chart displays by editing chart attributes in Page Designer.

10.3.1 About Supported Charts

App Builder supports charts based on the Oracle JavaScript Extension Toolkit (Oracle JET) Data Visualizations.

Oracle JET empowers developers by providing a modular open source toolkit based on modern JavaScript, CSS3, and HTML5 design and development principles. The Oracle JET data visualization components include customizable charts, gauges, and other components that you can use to present flat or hierarchical data in a graphical display for data analysis. Each Oracle JET visualization supports animation, accessibility, responsive layout, internationalization, test automation, and a range of inter activity features. Charts provide dozens of different ways to visualize a data set.

> **See Also:**
>
> Work with Data Visualizations in *Developing Oracle JET Apps Using MVVM Architecture*.

Chart Creation Methods

You can create charts by running the Create Application Wizard, Create Page Wizard, or by adding a new chart manually to a page in Page Designer. The Create Page Wizard offers the most supported chart types. The following table summarizes various methods for creating charts.

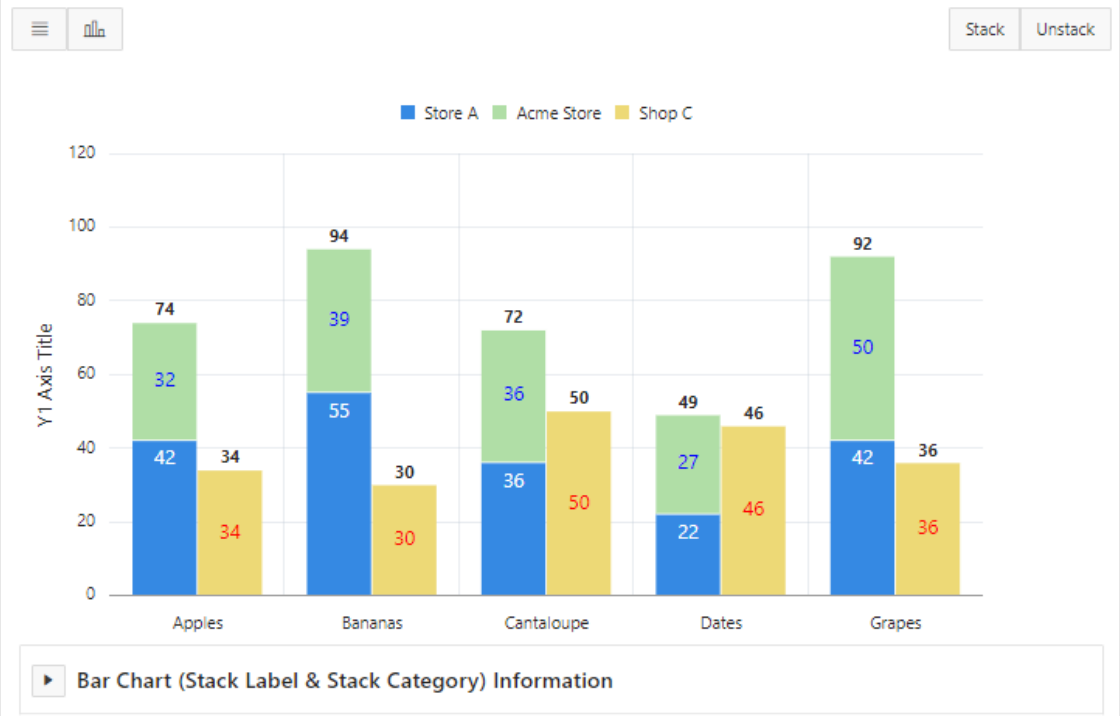
Method	Description	To Learn More
Create Application Wizard, Chart	Creates a page containing a chart. Supported chart types include: • Area • Bar • Line • Pie	Creating a Chart Using the Create Application Wizard
Create Application Wizard, Dashboard	Creates a dashboard page with up to four charts. Supported chart types include: • Area • Bar • Line • Pie	Creating a Dashboard Page Using the Create Application Wizard
Create Page Wizard, Chart	Creates a new page containing a chart. Supported chart types include: • Area • Bar • Box Plot • Bubble • Combination • Status Meter Gauge • Donut • Funnel • Gantt • Line • Line with Area • Pie • Polar • Pyramid • Radar • Scatter • Stock	See Adding a Chart Using the Create Page Wizard
Create Page Wizard, Dashboard	Creates dashboard page. Choose from four different chart layouts which you can easily customize in Page Designer. Available chart layouts include: • Primary and 2 secondary charts • Grid of 4 charts • Primary and 3 secondary charts • Grid of 6 charts	Adding a New Page to an Application. See online Help when creating this page type.
Page Designer	Add a new chart to an existing page in Page Designer.	Adding a Chart in Page Designer

> **See Also:**
>
> Editing Charts

10.3.2 Sample Charts

View chart examples in the *Sample Charts* sample app.

Sample apps are designed to highlight specific functionality and are intended to serve as a developer guide on how to make use of a particular feature. The following is a sample bar chart from *Sample Charts* sample app.



Click **Horizontal** and **Vertical** to toggle between a horizontal or vertical display. Click **Stack** and **Unstack** to view two series of data or three series of data.

This chart displays two series of data, *Store A* and *Acme Store*, stacked and associated with the Y axis, and the third series of data, *Shop C*, associated with an extra Y axis and rendered as a separate chart below the other two series of data.

Expand the **Bar Chart (Stack Label & Stack Category) Information** region at the bottom of the page to view information on the chart.

> **Tip:**
>
> To view chart examples, install the sample app, *Sample Charts*. To learn more, see Installing Apps from the Gallery.

10.3.3 Creating a Chart Using the Create Application Wizard

Create a new application that contains a chart page.

Run the Create Application Wizard and create a new page containing one of the following chart types: Area, Bar, Line, or Pie.

> **Note:**
>
> Create up to four charts at once by adding a Dashboard page.

To add a chart page using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Chart**.
8. On Add Chart Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Select a chart type: **Area**, **Bar**, **Line**, or **Pie**.
   d. Table or View - Click the LOV icon (bullet points and lines) and select a table or view on which to build the chart.
   e. Label Column - Select the column name used to define the label(s) of the x-axis on the chart.
   f. Value Aggregation - Select how to aggregate the chart Value Column (**Column Value**, **Sum**, or **Count**.

      For example, selecting **Count** bases the aggregation on the selected **Label Column**. For all other aggregation types, if no **Value Column** is selected, the aggregation is based on the select Label Column.
   g. Value Column - Select the column name to be used for defining the value on this chart.
   h. Click **Add Page**.
9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to Features.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to Settings.

11. Click **Create Application**.

10.3.4 Creating a Dashboard Page Using the Create Application Wizard

Create a new application that contains a Dashboard page with up to four charts.

A Dashboard page can contain up to four charts. To create a chart, select the chart (Chart 1, Chart 2, Chart 3, or Chart 4), the Chart Type, and then define the remaining attributes.

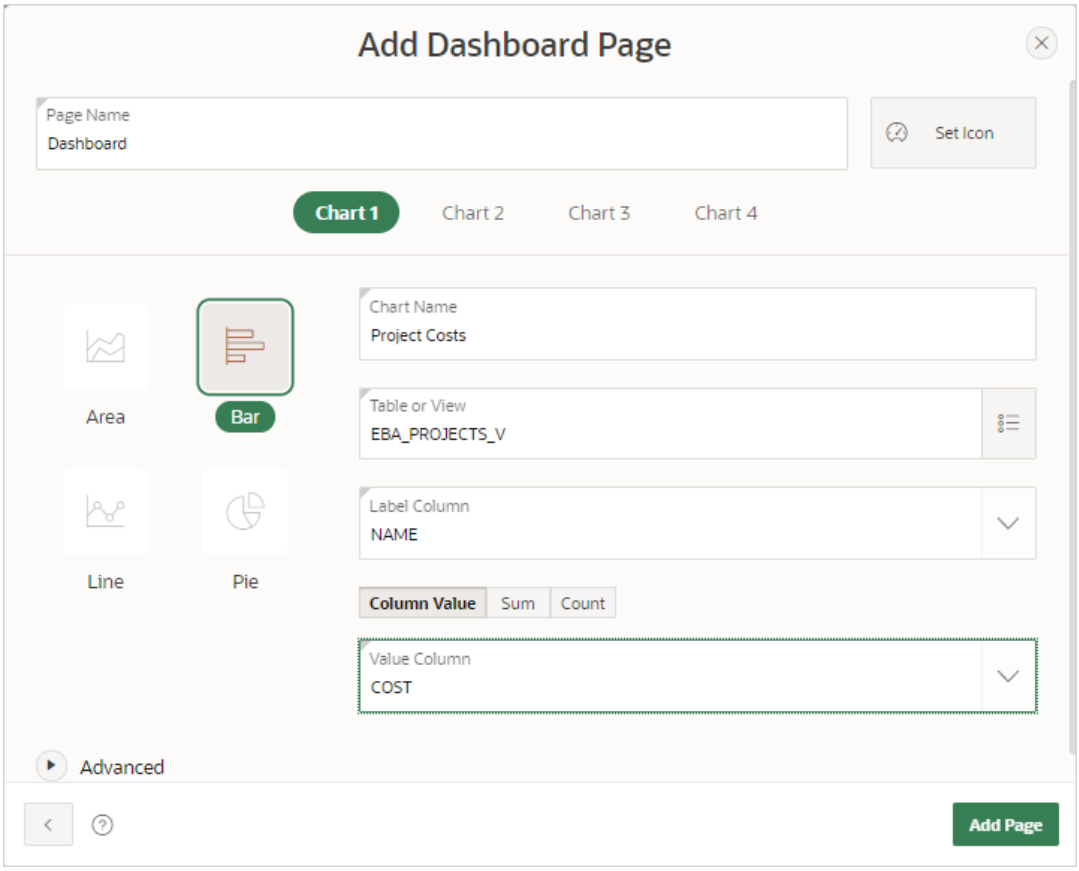
To add a Dashboard page using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default or click **Set Appearance** to change it.
7. Click **Add Page** and select **Dashboard**.
8. Add Cashboard Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
9. Create **Chart 1**:
   a. Chart Type - Select **Area**, **Bar**, **Line**, or **Pie**.
   b. Chart Name - Enter the name for the chart.
   c. Table or View - Click the LOV icon (bullet points and lines) and select a table or view on which to build the chart.
   d. Label Column - Select the column from the table or view to use as the label for the chart.
   e. Value Aggregation - Select how to aggregate the chart Value Column (**Column Value**, **Sum**, or **Count**.

      For example, selecting **Count** bases the aggregation on the selected **Label Column**. For all other aggregation types, if no **Value Column** is selected, the aggregation is based on the select Label Column.
   f. Value Column - Select the column name to be used for defining the value on this chart.



10. Create **Chart 2**:
    a. Chart Type - Select **Area**, **Bar**, **Line**, or **Pie**.
    b. Chart Name - Enter the name for the chart.
    c. Table or View - Click the LOV icon (bullet points and lines) and select a table or view on which to build the chart.
    d. Label Column - Select the column from the table or view to use as the label for the chart.
    e. Value Aggregation - Select how to aggregate the chart Value Column (**Column Value**, **Sum**, or **Count**.

       For example, selecting **Count** bases the aggregation on the selected **Label Column**. For all other aggregation types, if no **Value Column** is selected, the aggregation is based on the select Label Column.
    f. Value Column - Select the column name to be used for defining the value on this chart.
11. Repeat the previous step for Chart 3 and Chart 4. When you are done creating charts, click **Add Page**.
12. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to Features.
13. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to Settings.
14. Click **Create Application**.

10.3.5 Adding a Chart Using the Create Page Wizard

Run the Create Page Wizard to add a new chart to an existing application.

To create a chart by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Chart**.
5. Select a chart type.
6. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

      > **Tip:**
      >
      > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
7. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.
8. Source Type - Specify the source of the new page. Options include:
   - **Table**:
     – Table /View Owner - Select the owner of the table on which you are building the page.
     – Table/View Name - Select the table or view on which the page will be based.

- **SQL Query**:
  - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

9. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
10. Click **Next**.
11. Edit the chart attributes. The options that display vary depending upon the chart type selected. To learn more about an attribute, click the field-level Help.
12. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

10.3.6 Adding a Chart in Page Designer

Add a chart to a page by creating it manually in Page Designer.

To add a chart to an existing page in Page Designer:

1. View the page to contain the chart in Page Designer.
2. In the Gallery at the bottom of the central pane, click **Regions** and locate **Chart**.

   The Gallery lists all controls or components you can add to a page. Passing the cursor over a control or component displays a tooltip that describes it.
3. From the Gallery,
   a. Right-click **Chart** to view the context menu.
   b. Select **Add To** and then the desired location.

   > **Tip:**
   >
   > You can also click **Chart** in the Gallery and drag it to the appropriate location in the Layout tab.
4. Page Designer indicates what to do next. The Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
5. Edit the Series:
   a. In the Rendering tab under the chart, expand the **Series** node and select the new series.

b. Under **Series** node, select a series.

c. **Source**, **Location** - Select the location of the database you want to use. Options include:

- **Local Database** - Data is sourced from a local database.
- **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

d. **Column Mapping**, **Label** - Select the column name to be used for defining the label(s) of the x-axis on the chart. For a Stock chart, the column should be of data type Date/Timestamp, to represent the stock information on a time axis

e. **Column Mapping**, **Value** - Select the column name to be used for defining the value on this chart. For a Box Plot chart, this column will be used for defining the minimum, quartiles, median, and maximum values of the numerical data set.

> **Note:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Expand the **Axes** node:

> **Tip:**
>
> The Axes node only displays for chart types that support axes (such as area, bar, bubble, combination, line, line with area, polar, radar, range, scatter, and stock).

a. In the Rendering tab under the chart, expand the **Axes** node.

b. Under **Axes** node, select an Axis.

c. Edit the required axis attributes and define any data formatting to be applied to the axis values.

7. Edit the **Region** attributes.

a. In the left pane (or Rendering tab), select the chart.

The Property Editor, **Region** tab displays the region attributes

b. Review and edit the region attributes in the Property Editor.

8. Edit the chart **Attributes**:

a. In the Property Editor, select the **Attributes** tab.

b. Edit the required chart attributes.

9. Click **Save** or **Save and Run Page**.

10.3.7 Editing Charts

You can alter how a chart displays by editing chart attributes in Page Designer.

- Editing Chart Attributes
  Learn about editing chart Region attributes, chart Attributes, and Series attributes.
- Switching Chart Type
  Change the chart type by editing chart Attributes.
- Enabling Automatic Refresh
  Monitor chart information by enabling the Automatic Refresh, Interval attribute, or by using a dynamic action with the Refresh action.
- Using Custom JavaScript with Charts
  Control the look and feel of a chart by adding custom JavaScript.

> **See Also:**
>
> Viewing a Page in Page Designer

10.3.7.1 Editing Chart Attributes

Learn about editing chart Region attributes, chart Attributes, and Series attributes.

Charts have three types of attributes: **Region attributes**, **chart Attributes**, and **Series attributes**. When you select a chart in the Rendering tab, the Property Editor displays two tabs: **Region** (which contains Region attributes) and **Attributes** (which contains chart Attributes). **Series** attributes display in the Rendering tab under the chart.

To edit chart attributes:

1. View the page containing the calendar in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

   The Property Editor in the right pane displays displays two tabs: **Region** and **Attributes**. Attributes are organized in groups. To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

2. To edit the Region attributes.

   a. In the left pane (or Rendering tab), select the chart.

      The Property Editor, **Region** tab displays the region attributes

   b. Review and edit the region attributes in the Property Editor.

3. To edit the chart Attributes.

   a. In the left pane (or Rendering tab), select the chart.
   b. In the Property Editor, select the **Attributes** tab.
   c. Review and edit the chart Attributes in the Property Editor.
4. To edit the Series:
   a. In the Rendering tab under the chart, select the **Series**.
   b. Review and edit the Series attributes in the Property Editor.
5. Click **Save**.

10.3.7.2 Switching Chart Type

Change the chart type by editing chart Attributes.

To switch a chart type:

1. View the page containing the chart in Page Designer.
2. In the left pane (or Rendering tab), select the chart.

   The Property Editor (right pane) displays two tabs.
3. Edit the chart Attributes and select a new chart type:
   a. In the Property Editor, select the **Attributes** tab.

      The Property Editor displays chart Attributes. Attributes are organized in groups. To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
   b. **Chart**, **Type** - Select the chart type you want to switch to. For example, **Combination**.
4. Edit the Series:
   a. In the Rendering tab under the chart, select the **Series**.

      The **Series** tab appears in the Property Editor.
   b. **Column Mapping** - Ensure all required Column Mapping columns have been defined.

      > **Note:**
      >
      > Column mappings may differ for different chart types, so you must ensure all required column mappings are defined for the selected chart type.

   c. **Type** - Select the chart series type. For example, **Bar Range**.

      Only certain chart types support combining different types of series on the same data plot. The Series Type attribute only displays for chart types **Combination**, **Polar**, **Radar**, and **Range**.
5. Click **Save** or **Save and Run Page**.

10.3.7.3 Enabling Automatic Refresh

Monitor chart information by enabling the Automatic Refresh, Interval attribute, or by using a dynamic action with the Refresh action.

- Enabling the Automatic Refresh Attribute
  Enabling the Automatic Refresh attribute updates the chart to reflect changes in the underlying data within a specified time interval.
- Creating a Refresh Dynamic Action
  Learn about creating a Refresh dynamic action.

10.3.7.3.1 Enabling the Automatic Refresh Attribute

Enabling the Automatic Refresh attribute updates the chart to reflect changes in the underlying data within a specified time interval.

To enable automatic refresh updates:

1. View the page containing the chart in Page Designer.
2. In the left pane (or Rendering tab), select the chart.
3. In the Property Editor, select the **Attributes** tab.
4. Find **Automatic Refresh**.
5. In **Automatic Refresh**, **Interval** - Enter the interval in seconds between chart updates.

   Very small updates intervals, such as 2 seconds, are discouraged since they may cause serious database performance issues.
6. Click **Save**.

10.3.7.3.2 Creating a Refresh Dynamic Action

Learn about creating a Refresh dynamic action.

To create a Refresh dynamic action:

1. View the page containing the chart in Page Designer.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, right-click **Events** and select **Create Dynamic Action**.
4. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
5. In the Property Editor, edit the following Dynamic Action attributes:
   a. **Identification**, **Name** - Enter a name of the dynamic action.
   b. **Execution**, **Sequence** - Specify the sequence for this component. The sequence determines the order of evaluation.
   c. **Execution**, **Event Scope** - Review the scope of the event. Scope determines when the event is evaluated a second or successive time.
   d. **Execution**, **Type** - Choose the type of execution method for the Dynamic Action. Options include **Immediate** (default), **Debounce**, and **Throttle**.
   e. **When**, **Event** - Specify the event that causes the dynamic action to fire.
   f. **When**, **Selection Type** - Select the type of page element or construct to be used to trigger the event. What displays next, depends upon selected element or construct.

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

6. In the Rendering tab, select the first action under **True** and edit the attributes:
   a. **Identification**, **Name** - Enter the name of the action to identify what the action does (for example, select **Refresh Chart**).
   b. **Identification**, **Action** - Specify which action you want to perform (for example, select **Refresh**).
   c. **Affected Elements**, **Selection Type** - Select the type of page components to be affected when this action executes (for example, select **Region**).
   d. **Affected Elements**, **Region** - Select the region containing the chart.
   e. **Execution**, **Sequence** - Review the sequence for this component. The sequence determines the order of execution.
   f. **Execution**, **Event** - Review the selected dynamic action event (for example, **Refresh Chart**).
   g. **Execution**, **Fire When Event Result Is** - Specifies the selected dynamic action event (for example, **Refresh Chart**).
   h. **Execution**, **Fire on Initialization** - Enable this option. For a refresh dynamic action, this option causes the dynamic action to fire when the page loads.
7. Click **Save**.

> **See Also:**
>
> Managing Dynamic Actions

10.3.7.4 Using Custom JavaScript with Charts

Control the look and feel of a chart by adding custom JavaScript.

Control the look and feel of a chart by adding custom JavaScript.

To use custom JavaScript:

1. View the page in Page Designer:

   Page Designer appears.
2. In the left pane (or Rendering tab), select the chart.
3. In the Property Editor, select the **Attributes** tab.
4. Find **Advanced**.
5. In **Advanced**, **JavaScript Initialization Code**, enter the code to customize your chart.

   For example:

```
function( options ){
    // Setup a callback function which gets called
```

```
    // when data is retrieved, it allows to manipulate the series
     options.dataFilter = function( data ) {

         // e.g Set the first series of chart to the colour red
         data.series[ 0 ].color = "red";
         return data;
     };
     // Set chart initialization options
     // e.g. Set chart type to Line
     options.type = "line";
     return options;
}
```

6. Click **Save**.

> **Tip:**
>
> For more information on supported chart options, see ojChart in JSDoc pages for the Oracle JavaScript Extension Toolkit (JET).

> **Note:**
>
> The *Sample Charts* sample app contains a number of examples of using custom JavaScript with chart. To learn more, see Sample Apps and Installing Apps from the Gallery.

10.4 Creating Maps

Create maps that visualize location data (real coordinates) based on data in a table.

- About Supported Maps
  The map component consists of a background map and spatial geometry objects sourced from data in a table or from a REST Data Source.
- Sample Maps
  View sample map pages.
- Creating a Map Using the Create Application Wizard
  Run the Create Application Wizard to create a new application that contains a map page.
- Adding a Map Region in Page Designer
  Manually create a map region in Page Designer by providing the SQL query.
- Adding a Map Using the Create Page Wizard
  Run the Create Page Wizard to create a new page with a map.
- Editing Maps
  Edit map region components in Page Designer.
- Creating Custom Map Backgrounds
  Create custom map backgrounds for use in map regions, Display Map items, and Geocoded Address items.

10.4.1 About Supported Maps

The map component consists of a background map and spatial geometry objects sourced from data in a table or from a REST Data Source.

About Background Maps

The background map is static and comes from Oracle Elocation Service. These background maps are ready to use and do not require any API keys. However, in order to display the background map, the end user's browser must have an internet connection and access to https://maps.oracle.com.

Some background maps are also available as Vector Tile Layers which include the layers `OpenStreetMap Position`, `OpenStreetMap Bright`, and `OpenStreetMap DarkMatter`. Vector maps provide superior resolution over raster maps for all map zoom levels. For new applications, Vector Tile Layer usage is enabled by default. For applications created before Oracle APEX release 22.2, you must enable Vector Tile Layers in Shared Components, Component Settings. See Configuring Vector Tile Layers for Maps

Custom Background Maps

Developers can add custom map backgrounds under **Shared Components**, **Other Components**. Supported map background types include Raster, Vector, and OGC WMS. To learn more, see Creating Custom Map Backgrounds.

About Spatial Geometry Objects

Spatial geometry objects are drawn on the background map based on data in a table. That table can be sourced from local table or a SQL query. Maps created using the Create Page Wizard or in Page Designer, support both local database tables, REST Enabled SQL, and from REST Data Sources. Maps support a rich set of built-in marker icons and shapes. Supported spatial geometry objects include:

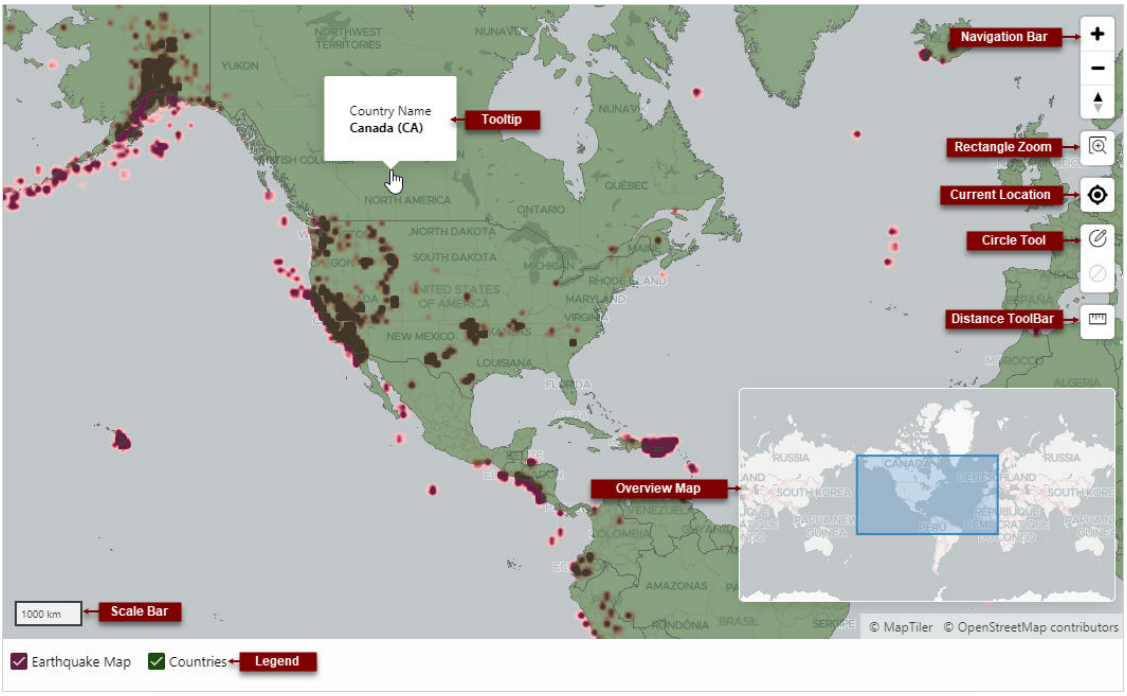
- **Points** - Points (for example, customer or supplier locations) display as markers.
- **Lines** - Lines represent features like roads or paths.
- **Polygons** - Polygons represent areas like parcels, states or countries.
- **Heat Map** - Heat Maps are used to visualize the point density. The more points that are clustered together, the more intense the color becomes. Use this option to visualize the spatial distribution of population or incidents.
- **Extruded Polygons** - Display as three-dimensional, extruded, objects. The height of the 3D object visualizes a column value.

You can source spatial geometry objects from either:

- **Geometry Column** - Supported datatypes include `SDO_GEOMETRY`, `VARCHAR2`, or `CLOB`. `VARCHAR2` and `CLOB` columns must contain geometry information in GeoJSON format.
- **Two Numeric Columns** - These columns must contain longitude and latitude values. This option only applies to Point and Heat Map objects.

About Map Controls

From a user's perspective, map regions include a number of controls that enable users to change the display at runtime. Not all of these controls are enabled by default. You can add or remove controls by editing the map region Attributes in Page Designer.



The previous map includes the following controls:

- Navigation Bar - Options include **Zoom**, **Zoom Out** and **Reset Bearing to North**.
- Rectangle Zoom - Zoom the map to an area by drawing a rectangle onto the map.
- Current Location - Displays the user's current location from the browser.
- Circle Tool - Enables end users to draw a circle onto the map. When finished, an event is triggered, so that the circle can be processed by Dynamic Actions or JavaScript code.
- Distance Tool - Zooms the map to an area by drawing a rectangle on the map.
- Tooltip - Displays a tooltip. The above example identifies the country name.
- Overview Map - Display an overview map in the lower right.
- Scale Bar - Displays a scale bar on the map.
- Legend - Map legend.

> **See Also:**
>
> Viewing and Editing a Map in Page Designer

10.4.2 Sample Maps

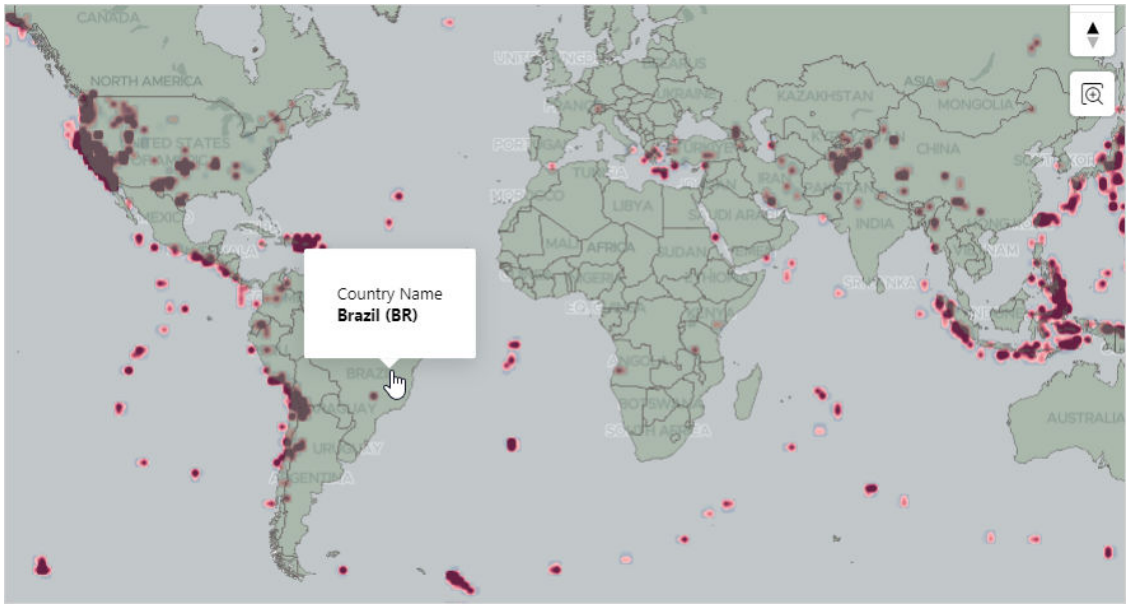
View sample map pages.

Sample Map with One Layer

The following sample map is built on a local table called `COUNTRIES_TABLE` and includes a Polygon object layer mapped to the `GEOMETRY` column and tooltips pulled from the `COUNTRY_NAME` column.

Sample Map with Two Layers

The following sample map is built on a local table called `EARTHQUAKE_TABLE` and features two object layers. The first layer is a Heat Map based on the `SDO_GEOMETRY` column . The second layer features Polygons based on the `COUNTRIES_TABLE` column and mapped to the `GEOMETRY` column.



> **Tip:**
>
> To view more samples, install the sample app, *Sample Maps*. See Installing Apps from the Gallery.

> **See Also:**
>
> Viewing and Editing a Map in Page Designer

10.4.3 Creating a Map Using the Create Application Wizard

Run the Create Application Wizard to create a new application that contains a map page.

> **Tip:**
>
> The Create Application Wizard only supports maps created a local database. To create maps based on an external data source, use the Create Page Wizard or add a map region in Page Designer. See Adding a Map Using the Create Page Wizard and Adding a Map Region in Page Designer.

To create a map using the Create Application Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Click **Create**.

   The wizard prompts you to select a method of creating an application.
3. Click **Use Create App Wizard**.

   The Create an Application page appears.
4. Set Icon - Click **Set Icon** to select an application icon. The icon you select is used as the Favicon, the PWA icon, the Apple touch icon, and the App Builder icon.
5. Name - Enter the name used to identify the application to developers.
6. Appearance - Accept the default Theme Style and menu layout (Vita, Side Menu), or change it by clicking the **Set Appearance** icon adjacent to the Appearance field.
7. Click **Add Page** and select **Map**.
8. On Create Map Page:
   a. Page Name - Enter a name for this page.
   b. Set Icon - Select an icon to display in the navigation menu for this page.
   c. Table - Select a table on which to build the map.
   d. Select the spatial geometry layer type. Options include:
      - **Points** - Points (for example, customer or supplier locations) are display as markers. The marker shape and color is configurable.

- **Lines** - Lines represent features such as roads or paths. The stroke style and color is configurable.
- **Polygons** - Polygons represent areas like parcels, states, or countries. The style and color are configurable for stroke and fill.
- **Heat Map** - Heat Maps are used to visualize the point density. The more points that are clustered together, the more intense the color becomes. Use this option to visualize the spatial distribution of population or incidents.
- **Extruded Polygons** - Display as three-dimensional, extruded, objects. The height of the 3D object visualizes a column value.

e. Geometry Column Type - Select how spatial objects are sourced. Options include:

- **Geometry Column** - Supported datatypes include `SDO_GEOMETRY`, `VARCHAR2`, or `CLOB`.

  `VARCHAR2` and `CLOB` columns must contain geometry information in GeoJSON format.
- **Two Numeric Columns** - These columns must contain longitude and latitude values. This option only applies to Point and Heat Map layers.

f. What appears next depends upon the spatial objects (or geometries) and Geometry Type you select. Select the appropriate columns.

g. Advanced - Expand **Advanced** to configure these options:

- Set as Home Page - Enable this to make this page the home page for the application.
- Set as Administration Page - Enable this to have this page display under Administration.
- Page Help - Enter text to be displayed when the user selects Page Help.

h. Click **Add Page**.

9. Features - Select features to include with the application. Features provide application-level functionality and can only be added once per application. To learn more, click the Help icon adjacent to **Features**.
10. Settings - Specify settings used to generate this application. To learn more about an attribute, click the Help icon adjacent to **Settings**.
11. Click **Create Application**.

    The Application home page appears.
12. Click **Run Application** to view the map.

10.4.4 Adding a Map Region in Page Designer

Manually create a map region in Page Designer by providing the SQL query.

Adding a map in Page Designer supports spatial geometry objects (or layers) based on a local table, SQL query, REST Enabled SQL Service, or REST Data Sources.

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. If necessary, click the Layout tab in the central pane.
3. In the Gallery, click the **Regions** tab.
4. Right-click **Maps**, click **Add To**, and select the appropriate location.

> **Tip:**
>
> You can also select **Map** with the mouse and drag it to the appropriate location in the Layout tab.

5. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
6. Source - Identify the source of the map:
   a. Source, Location - Select the location of the data. Options include:
      - **Local Database** - Data is sourced from a local database.
      - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
      - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
      - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
      - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

      What displays next depends upon the Source, Location you select. This task assumes the Source, Location is **Local Database**.
   b. Source, Type - Select how the data is queried. Options include:
      - **Table/View** - The data is retrieve based on a table or view.
      - **SQL Query** - The data is retrieved based on a SQL Query.
      - **Function Body returning SQL Query** - The data is retrieved based on the SQL Query returned by executing a Function Body.

   What displays next depends upon the Source, Type you select.
7. Column Mapping - Identify how and where spatial objects are sourced:
   - Geometry Column Data Type - Identify the data type of the geometry column. Options include:
     – **SDO_GEOMETRY** - Oracle `SDO_GEOMETRY` type.

       Local databases and REST Data Sources support the `SDO_GEOMETRY` column type. However, REST Enabled SQL data sources do not support the `SDO_GEOMETRY` column type.

  - **GeoJSON** - `GeoJSON`, as `VARCHAR2` or `CLOB` types.
  - **Longitude/Latitude** - Longitude and Latitude, as two `NUMBER` columns.

  What appears next depends upon the Geometry Column Data Type you select. Select the appropriate columns.

- Primary Key Column - Choose the column containing the unique key to identify a row. Use this to provide the primary key value to custom JavaScript code or map event handlers.

8. Layer attributes:
   a. In Rendering tab (left pane), select the **New** layer.
   b. Identification, Name - Enter a descriptive name for this layer.
   c. Identification, Layer Type - Select the spatial geometry layer type. Options include:
      - **Points** - Points (for example, customer or supplier locations) are display as markers. The marker shape and color is configurable.
      - **Lines** - Lines represent features such as roads or paths. The stroke style and color is configurable.
      - **Polygons** - Polygons represent areas like parcels, states, or countries. The style and color are configurable for stroke and fill.
      - **Heat Map** - Heat Maps are used to visualize the point density. The more points that are clustered together, the more intense the color becomes. Use this option to visualize the spatial distribution of population or incidents.
      - **Extruded Polygons** - Display as three-dimensional, extruded, objects. The height of the 3D object visualizes a column value.
9. Click **Save**.
10. Click **Save and Run Page** to view the map

10.4.5 Adding a Map Using the Create Page Wizard

Run the Create Page Wizard to create a new page with a map.

The Create Page Wizard supports spatial geometry objects based on a local table, SQL query, REST Enabled SQL Service, or REST Data Sources.

To create a map page by running the Create Page Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Map**.
5. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

> **Tip:**
>
> This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

c. Page Mode - Identify the page mode. To learn more, see field-level Help.

6. Data Source - Select a Data Source for the page:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

> **Tip:**
>
> The steps that follow differ depending upon the selected Data Source. This task assumes the Data Source is **Local Database**.

7. Source Type - Specify the source of the new page. Options include:
   - Select a Source Type. Options include:
     - **Table**:
       - Table / View Owner - Select the owner of the table on which you are building the page.
       - Table/View Name - Select the table or view on which the page will be based.
     - **SQL Query**:
       - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.
8. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
9. Click **Next**.
10. Map Style - Select the type of spatial geometry layer. Options include:
    - **Points** - Points (for example, customer or supplier locations) are display as markers. The marker shape and color is configurable.

- **Lines** - Lines represent features such as roads or paths. The stroke style and color is configurable.
- **Polygons** - Polygons represent areas like parcels, states, or countries. The style and color are configurable for stroke and fill.
- **Heat Map** - Heat Maps are used to visualize the point density. The more points that are clustered together, the more intense the color becomes. Use this option to visualize the spatial distribution of population or incidents.
- **Extruded Polygons** - Display as three-dimensional, extruded, objects. The height of the 3D object visualizes a column value.

11. Map Attributes:
    a. Geometry Column Type - Select how spatial objects are sourced. Options include:
        - **Geometry Column** - Supported data types include `SDO_GEOMETRY`, `VARCHAR2`, or `CLOB`.

          Local databases and REST Data Sources support the `SDO_GEOMETRY` column type. However, REST Enabled SQL data sources do not support the `SDO_GEOMETRY` column type.

          `VARCHAR2` and `CLOB` columns must contain geometry information in GeoJSON format.
        - **Two Numeric Columns** - This option only displays for **Point** and **Heat Map** layers. These columns must contain longitude and latitude values.
    b. What appears next depends upon the spatial objects (or geometries) and Geometry Type you select. Select the appropriate columns.
    c. Create a Faceted Search Page - Enable this option to create the map page as a Faceted Search page. The Faceted Search region will contain only a Search facet, other facets can be added later in Page Designer.
12. Click **Create Page**.

> **Tip:**
>
> To learn more about external data sources, see Managing REST Enabled SQL References and Managing REST Data Sources.

10.4.6 Editing Maps

Edit map region components in Page Designer.

- Viewing and Editing a Map in Page Designer
  View and edit the components that comprise a map by viewing attributes in Page Designer.
- Adding a New Layer to an Existing Map
  Edit the map region defintion to add more spatial geometry objects.
- Controlling Layer Visibility
  Control layer visibility by editing attributes in Page Designer.
- Connecting a Map Region with a Report Region
  Connect a map region with a classic report region.

10.4.6.1 Viewing and Editing a Map in Page Designer

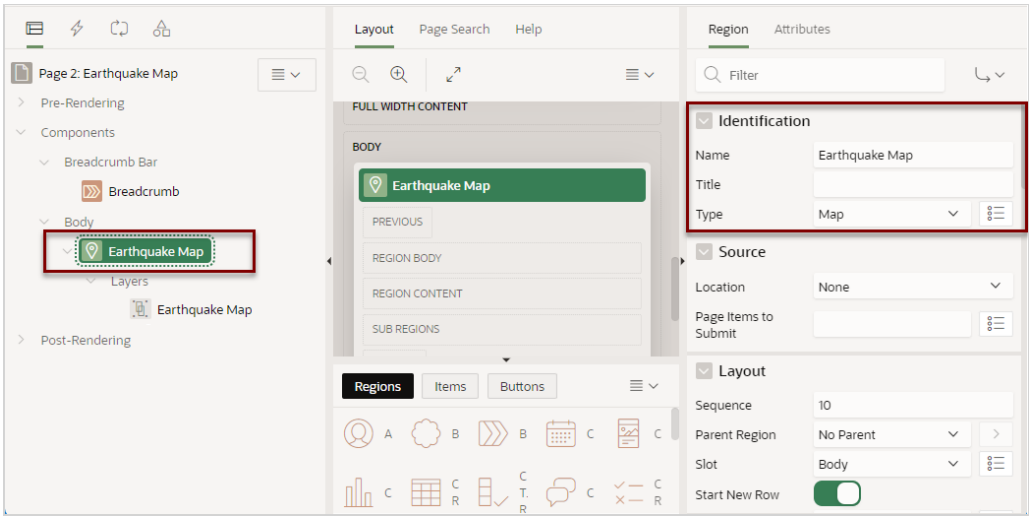
View and edit the components that comprise a map by viewing attributes in Page Designer.

A map features a map region which contains spatial geometry layers. This topic explores an earthquake map built on a local table named `EARTHQUAKE_TABLE` which features a Heat Map spatial geometry layer.

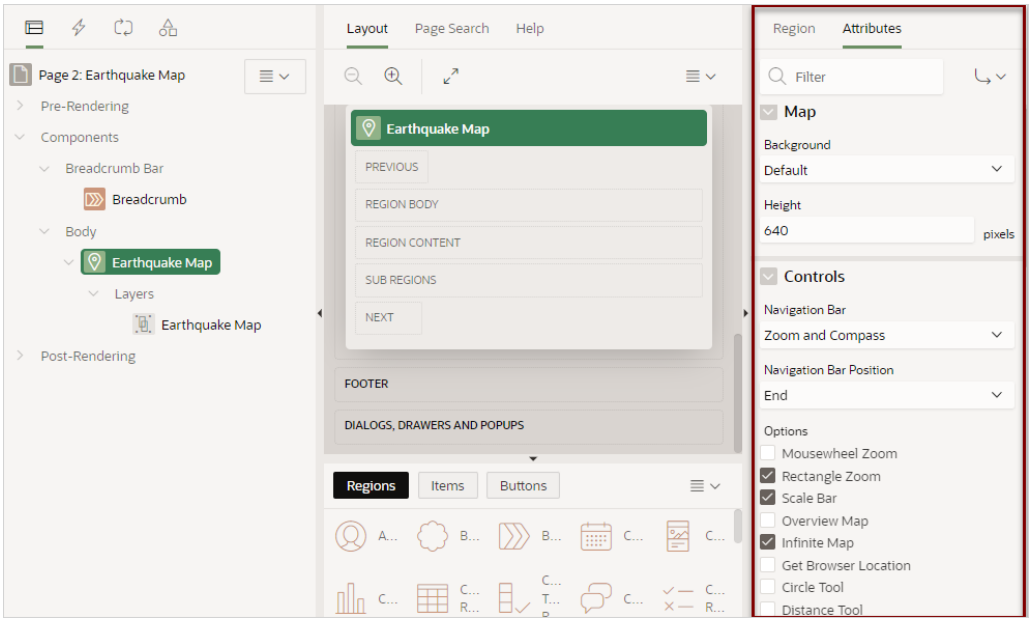
To view or edit a map in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears displaying Page attributes in the Property Editor (right pane).
2. Review the map region.
   a. In the Rendering tab select the Map region. The Property Editor (right pane) displays two tabs: **Region** and **Attributes**.



   In the previous example, note that Identification, Name indicates the map is named **Earthquake Map** and the Type is **Map**.
   b. In the Property Editor, click the **Attributes** tab.



Map Attributes determine basic map characteristics. In the previous example, note the attributes under **Controls**. Controls determine the various ways users can change the map display at runtime, for example:

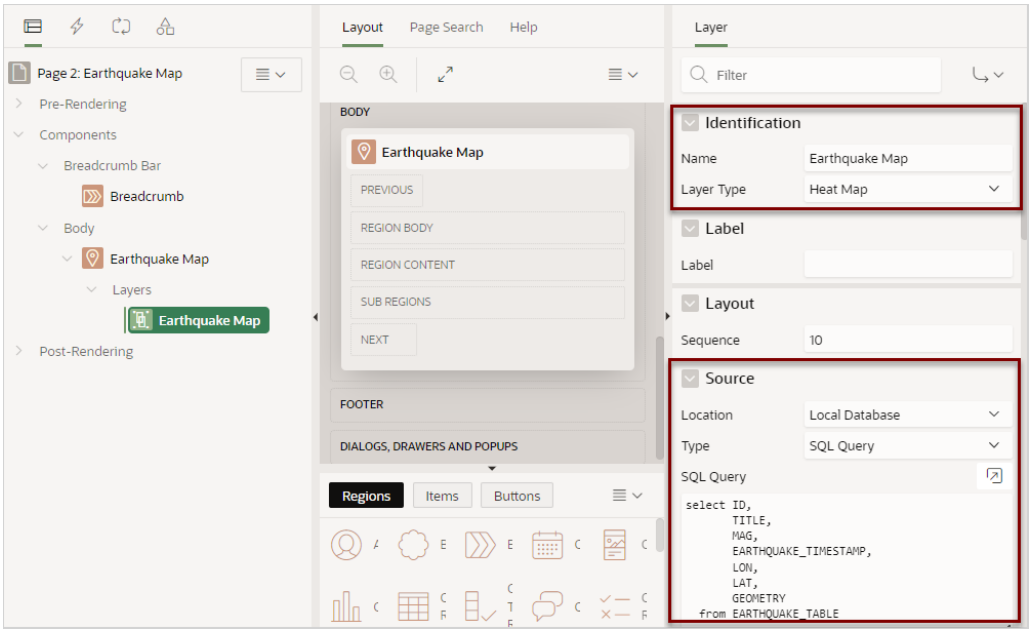
- Navigation Bar - Options include **None**, **Zoom Only** and **Zoom and Compass**.
- Navigation Bar Position - **Start** positions the navigation bar on the starting position (typically to the left side) of the map. **End** positions the navigation bar on the ending position (typically to the right side) of the map.
- Mousewheel Zoom - Zoom in or out by using the mouse wheel.
- Rectangle Zoom - Zoom the map to an area by drawing a rectangle onto the map.
- Scale Bar - Display a scale bar on the map.
- Overview Map - Display an overview map in the lower right.
- Infinite Map - Enables infinite map scrolling.
- Get Browser - Displays the user's current location from their browser.
- Circle Tool - Enables end users to draw a circle onto the map. When finished, an event is triggered, so that the circle can be processed by Dynamic Actions or JavaScript code.
- Distance Tool - Enables end users to measure distances between points on the map.

Note that in the previous example only some of these controls are enabled.

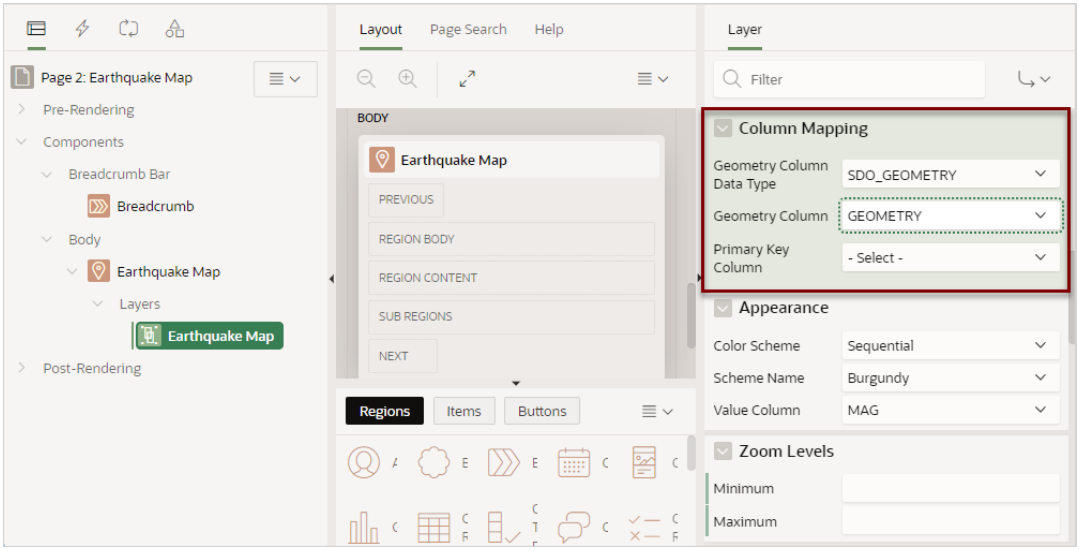
3. Review the map **Layers**. Maps have spatial geometry objects (or layers) which are drawn on the map based on data in a table.

   - In the Rendering tab, select the Layer, **Earthquake Map**.

     Note that the following example has a Heat Map layer which is sourced from a local database using a SQL Query.



4. Review the Column Mappings. In the Property Editor, scroll down to Column Mappings.



In the previous example, note that the Column Mapping attributes indicate the Geometry Column Data Type is **SDO_GEOMETRY** and the Geometry Column is **GEOMETRY**.

10.4.6.2 Adding a New Layer to an Existing Map

Edit the map region defintion to add more spatial geometry objects.

Maps have spatial geometry objects which are drawn on the map based on data in a table. This topic demonstates how to add a polygon object layer to an existing map that identifies the county. This topic includes an example which adds an object layer named Countries sourced from a local table named `COUNTRIES_TABLE` and mapped to the `GEOMETRY` column.

To add a new object layer to an existing map:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

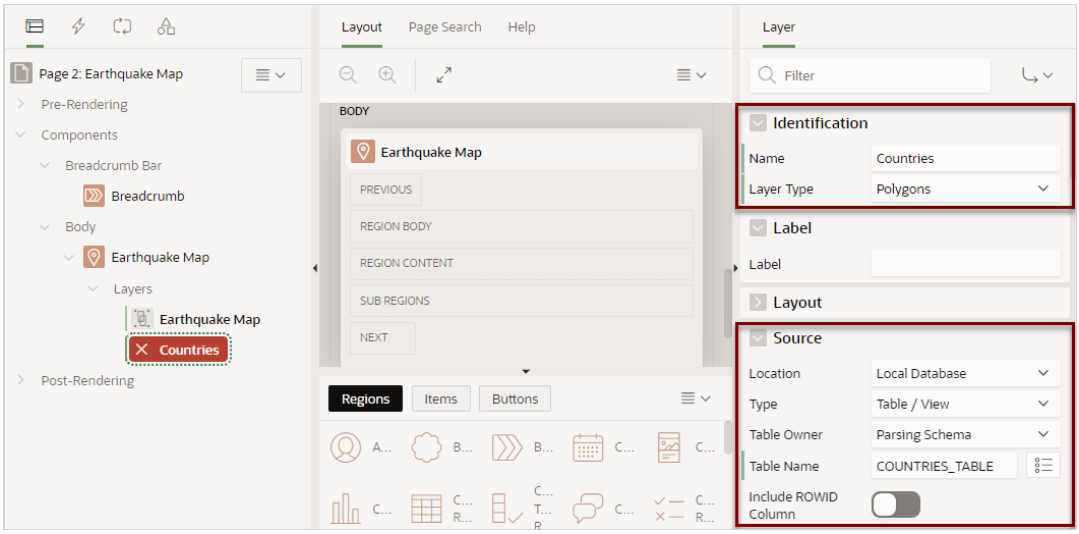
c. Select a page.

Page Designer appears displaying Page attributes in the Property Editor (right pane).

2. Add a spatial geometry layer. In the Rendering tab, right-click **Layers** and select **Create Layer**.
3. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

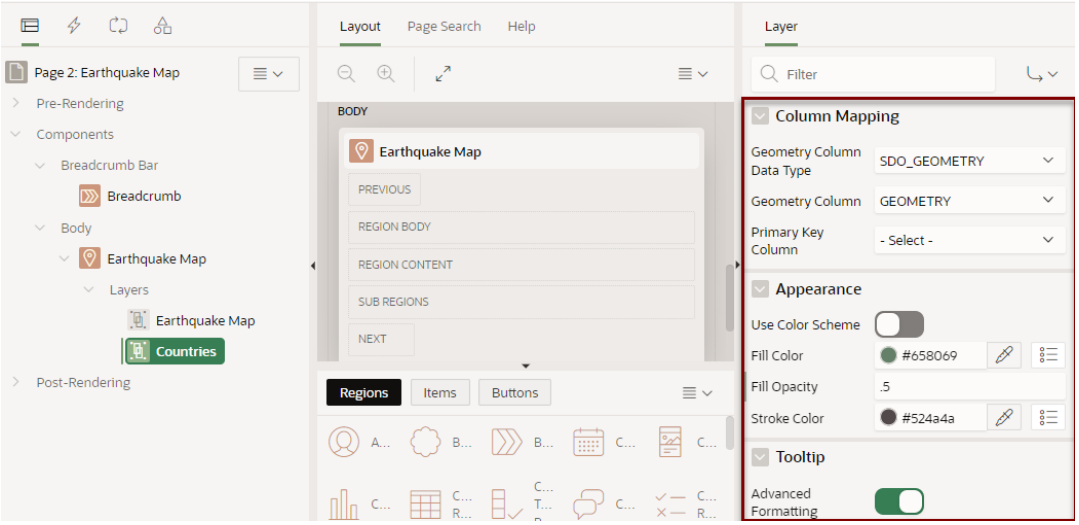
      The Messages dialog displays errors that must be addressed.

   b. Select an error to highlight the associated attribute in the Property Editor.
4. Edit the Identification attributes:
   a. Identification, Name - Enter a descriptive name (for example, **Countries**).
   b. Identification, Layer Type - Select spatial geometry type (for example, **Polygons**).
5. Edit the Source attributes.
   a. Source, Location - Determines the location of the date (for example, **Local Database**.
   b. Source, Type - Select how the data is queried (for example, **Table / View**.
   c. Source, Table Name - Select the table (for example, `COUNTRIES_TABLE`).



6. Edit the Column Mapping attributes:
   a. Column Mapping, Geometry Column Data Type - Select the data type of the geometry column (for example, **SDO_GEOMETRY**).
   b. Column Mapping, Geometry Column - Select the data type of the geometry column (for example, **GEOMETRY**).
7. Edit the Appearance attributes:
   a. Appearance, Use Color Scheme - Specify whether to use a Color Scheme for this layer. A color scheme is a list of colors. Based on a value and the minimum and maximum values, the map widget will pick the actual display color for this object from the list. In the example that follows, this option is disabled.
   b. Appearance, Fill Color - Specify the fill color.
   c. Appearance, Fill Opacity - Specify fill opacity. Ranges from zero (fully transparent) to 1.0 (opaque).

d. Appearance, Stroke Color - Specify the stroke color.



8. Configure the Tool Tip. The Tooltip displays when the user's mouse hovers over a polygon. The following example uses an HTML expression with column substitutions. However, you can also use the Template Directives, as with the Cards Region type.

   a. Tooltip, Advanced Formatting - Specify whether the tooltip displays using a column or advanced formatting using HTML expressions and supported substitution strings. This example enables **Advanced Formatting**.

   b. Appearance, HTML Expression - Enter HTML expressions to be shown as tooltip when hovering over an object on the map. Use `&COLUMN.` syntax to show column values in HTML. For example:

```
Country Name<br>
<b>&COUNTRY_NAME. (&ISO_A2.)</b>
```

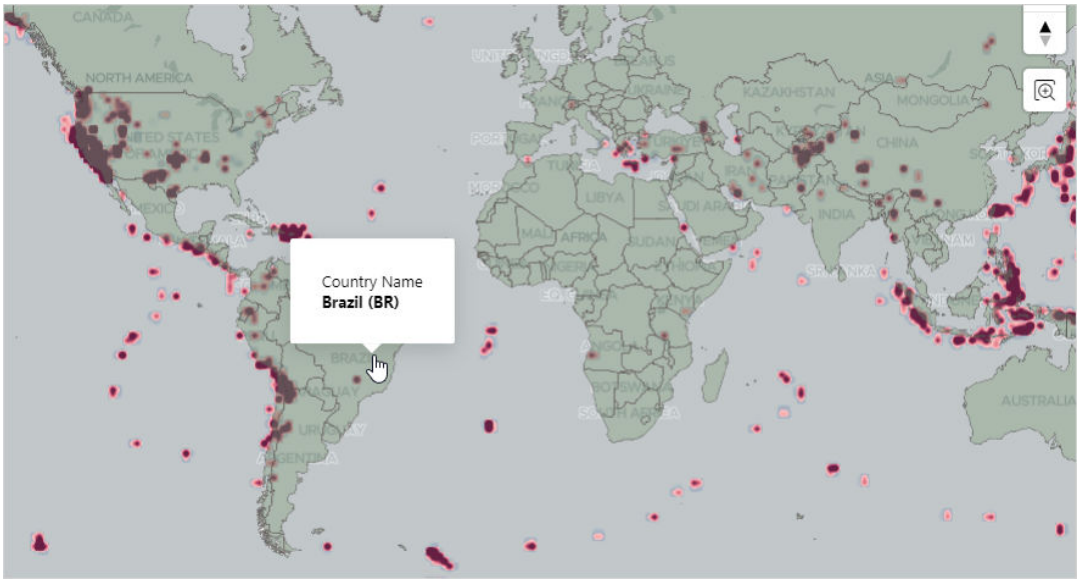
> **Tip:**
>
> For more information on the supported Template Directives, see applyTemplate in *Oracle APEX JavaScript API Reference*.

9. Legend, Enable to Hide - Determines whether layer visibility can be toggled by clicking layer name in the legend. Prevent the legend from hiding the country polygons. Disable **Enable to Hide**.

10. Click **Save**.

11. Click **Save and Run Page**.

    The map appears showing two layers: Earthquake Heat Map and Countries.



10.4.6.3 Controlling Layer Visibility

Control layer visibility by editing attributes in Page Designer.

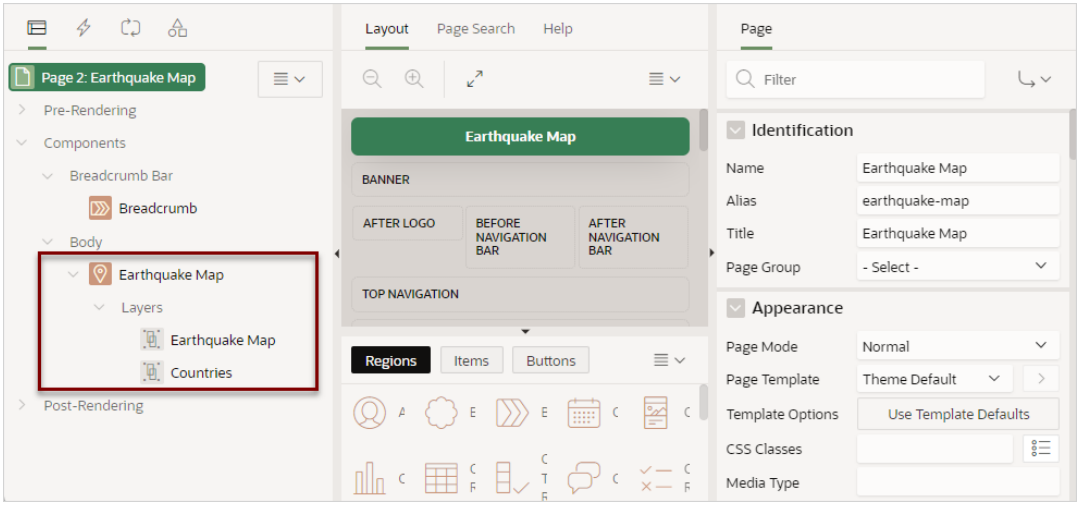
This topic assumes you have a map that has two spatial geometry objects layers: Countries and Earthquake Map. The Countries layer is based on a local table named `COUNTRIES_TABLE`. Earthquake Map is sourced from a local table named `EARTHQUAKE_TABLE` and features a Heat Map spatial geometry layer.

This task demonstrates how to attributes in Page Designer and control the visibility of the Earthquake Map object layer. In this topic you specify zoom levels and add a server-side condition that specifies that only earthquakes with a magnitude of greater than two display.

To control layer visibility:

1. View the map page in Page Designer:

   Page Designer appears. In the following example, the map has two spatial geometry objects layers: Countries and Earthquake Map.



2. In the Rendering tab (left pane), under **Layers**, select **Earthquake Map**.

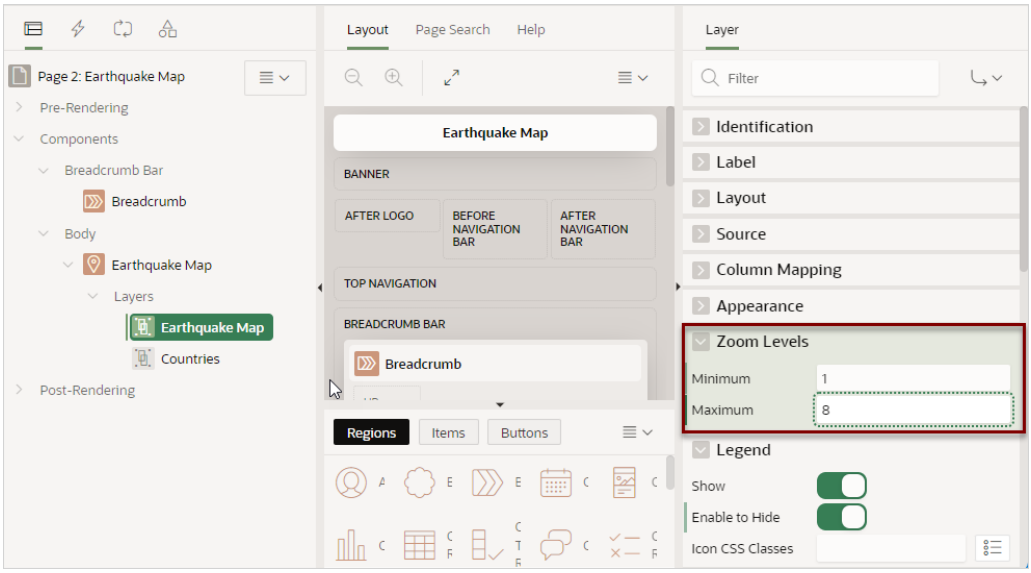
   Layer attributes display in the Property Editor.

3. In the Property Editor, edit Zoom Levels attributes:

   Zoom Levels must have a Minimum value of 1 and a Maximum value of 18.

   a. Zoom Levels, Minimum - Specify the Minimum zoom level for this layer to become visible (for example, enter `1`).

   b. Zoom Levels, Maximum - Specify the Maximum zoom level for this layer to become visible (for example, enter `8`).



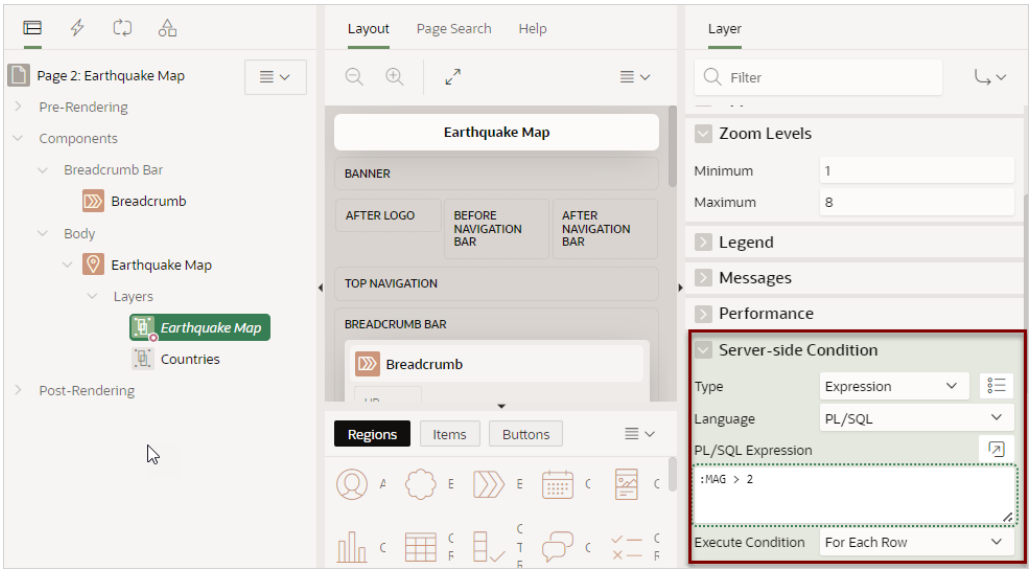
   These settings decrease the available zoom levels from the default (1 to 18) to 1 to 8. When the user zooms outside of specified Zoom Levels, the layer disappears.

4. Create a server-side condition:

   a. Server-side Condition, Type - Select **Expression**.

   b. Server-side Condition, PL/SQL Expression - Enter `:MAG > 2`

   c. Server-side Condition, Execute Condition - Select **For Each Row**.



5. Click **Save and Run Page** to view the application runtime. and only eight zoom levels are available.

The map appears. Only eight zoom levels are available and only earthquakes with a magnitude greater than 2 display.

10.4.6.4 Connecting a Map Region with a Report Region

Connect a map region with a classic report region.

This topic explains how to connect a map region with a classic report so that whenever the map region changes (that is, moves or zooms in or out), the report region refreshes to reflect the change. In order to connect a map region with a report region, this example creates a page which includes the following:

- A map containing an object layer named `Earthquake Map` sourced from a local table named `EARTHQUAKE_TABLE`.
- A classic report based on a SQL query also based on `EARTHQUAKE_TABLE`.
- To sync the map with the report, a hidden page item used to maintain the current map status (also known as a **bounding box**). A bounding box (usually shortened to **bbox**) is an area defined by two longitudes and two latitudes.
- A dynamic action to refresh the report as the map display changes.

> **Note:**
>
> Because of a restrictions within Oracle Spatial, this example only works if the map shows less than half of the earth. Therefore, the report will not refresh for zoom levels 1 and 2 of the map. See Coordinate Systems (Spatial Reference Systems) in *Oracle Spatial Developer's Guide*.

To connect map with a report region:

1. Run the Create Application Wizard and create an application.
2. Run the Create Page Wizard and create a new map page:
   a. On the Application home page, click **Create Page**.

      Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

   b. Under Component, select **Map**.
   c. Page Definition:
      - Name - Specify a text name for this page (for example, `Earthquake Map`).
      - Page Mode - Accept the default **Normal**.
   d. Data Source:
      - Data Source - Select a Data Source for the page (for example, **Local Database**).
      - Source Type - Specify the source of the new page (for example, **Table**).
      - Table/View Name - Select a table on which to build the map (for example, **EARTHQUAKE_TABLE**).
   e. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   f. Click **Next**.

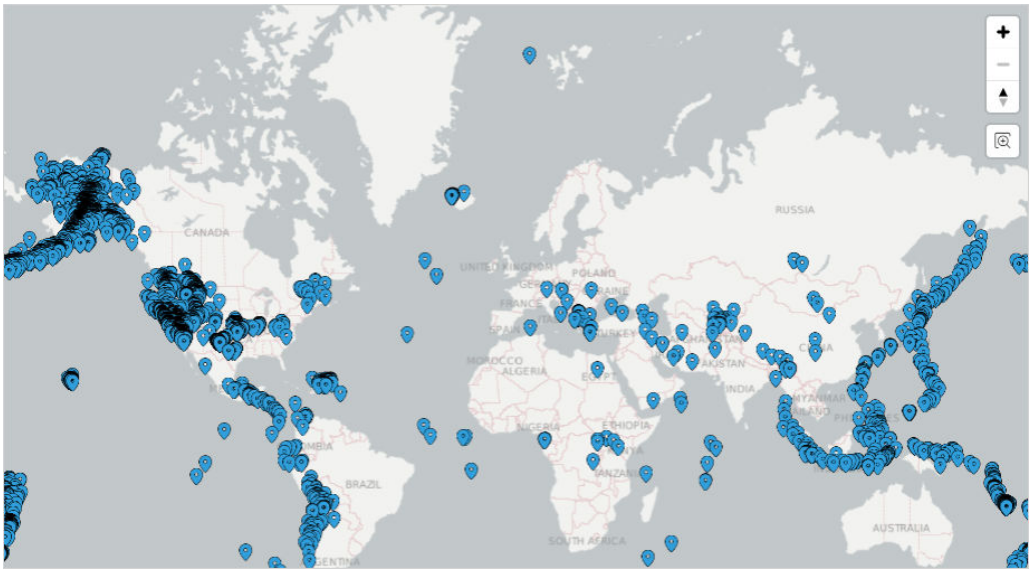
g. Map Style - Select the spatial geometry layer type (for example, **Points**).

h. Map Attributes:

- Geometry Column Type - Select how spatial objects are sourced. (for example, **Geometry Column**).
- Geometry Column - Pick the spatial geometry column (for example, **GEOMETRY**).
- Tooltip Column - Select the column to be used as tooltip (for example, **MAG**).

i. Click **Create Page**.

j. If you run the page, an earthquake map appears.



k. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar, where *X* is the page number.

Page Designer appears.

3. Add a classic report adjacent to the map:

a. In the Rendering tab, select **Earthquake Map** region.

b. In the Gallery, Regions tab at the bottom of the central pane, locate **Classic Report**.

c. Right-click **Classic Report** and select **Add To**, **Body**, **Earthquake Map**, and then **Column after**.

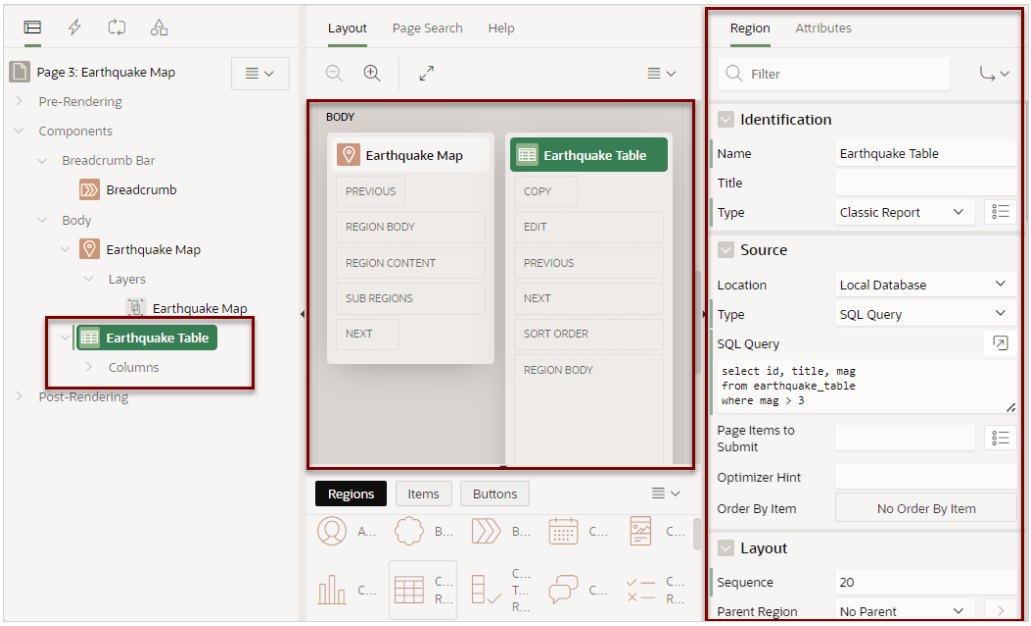
A New Classic Report region appears next to the Earthquake Map in the Layout tab.

d. In the Property Editor, edit the following Region attributes:

- Identification, Title - Enter `Earthquake Table`
- Source, Type - Select **SQL Query**.
- Source, SQL Query - Enter:

```
select id, title, mag
from earthquake_table
where mag > 3
```

- Layout, Sequence - Enter `20`



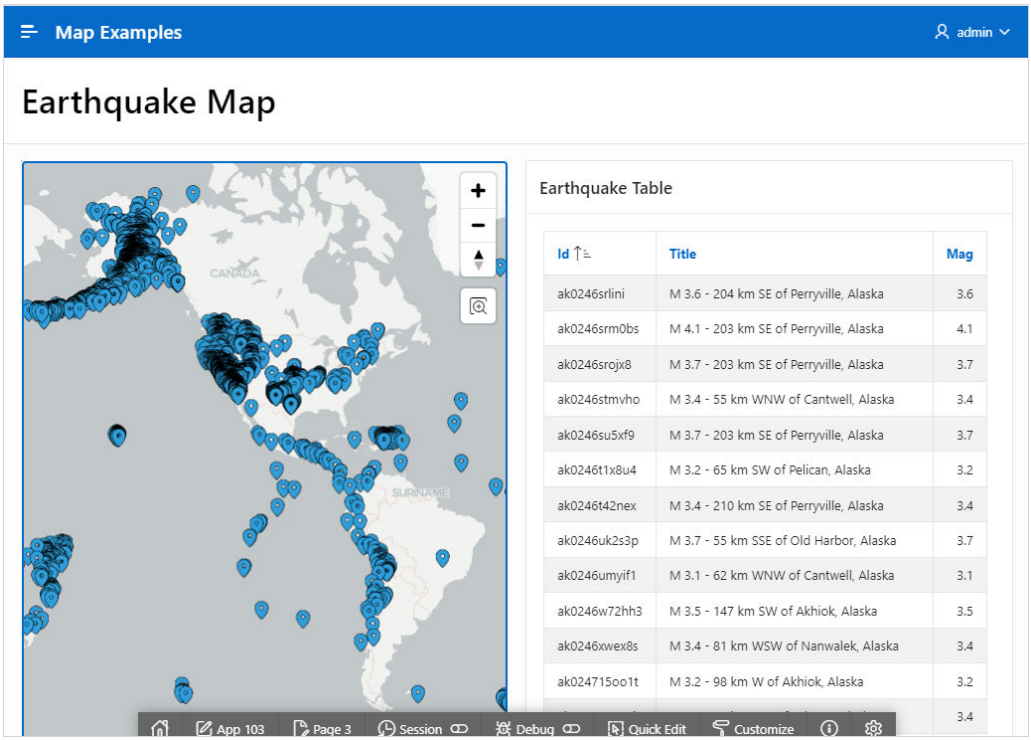
e. Edit the Classic Report region Attributes:

- Click the **Attributes** tab.
- Messages, When No Data Found - Enter:

  ```
  No features on the map. If you're zoomed out to the whole earth, zoom in to see data.
  ```

f. Click **Save and Run Page**.

The new classic report displays to the right of the earthquake map.

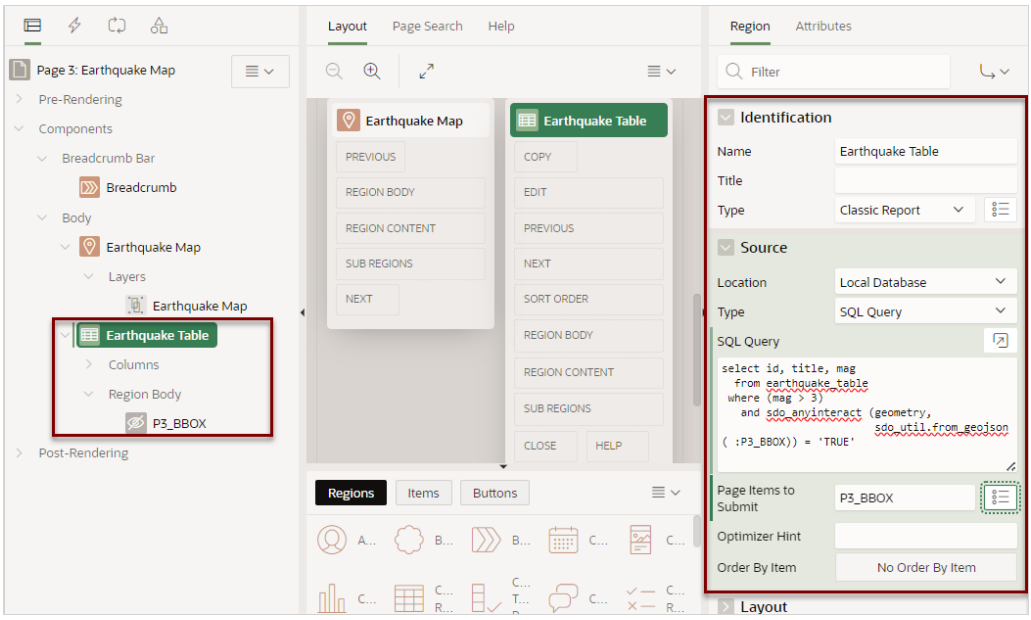


g. Return to Page Designer. Click **Page X** on the Runtime Developer toolbar, where *X* is the page number.

4. Add a hidden page item to maintain the current map status (also known as a **bounding box**).
   a. In the Rendering tab, right-click **Earthquake Table** and select **Create Page Item Below**.
   b. Edit the following Page Item attributes in the Property Editor:
      - Identification, Name - Enter `PX_BBOX`, where *X* is the page number.
      - Identification, Type - Select **Hidden**.
      - Settings, Value Protected - Disable this option.
5. Update the Earthquake Map region:
   a. In the Rendering tab, select the **Earthquake Map** region.
   b. In the Property Editor, click **Attributes**.
   c. Advanced, Bounding Box Item - Select **PX_BBOX** , where *X* is the page number.
   d. Click **Save**.
6. Update the SQL query that defines the classic report to include the spatial filter:
   a. In the Rendering tab, select the **Earthquake Table** region.
   b. In the Property Editor, update the following:
      - Source, SQL Query - Enter:

```
select id, title, mag
  from earthquake_table
 where (mag > 3)
   and sdo_anyinteract (geometry,
                        sdo_util.from_geojson ( :PX_BBOX)) = 'TRUE'
```

        Where *X* in **PX_BBOX** is the page number.
      - Source, Page Items to Submit - Select **PX_BBOX** , where *X* is the page number.



7. Add a dynamic action to refresh the report as the map display changes:

a. In the Rendering tab, right-click the map region, **Earthquake Map** and select **Create Dynamic Action**.

b. In the Property Editor, edit the following Dynamic Action attributes:

- Identification, Name - Enter a name for this dynamic action (for example, `Map Change`).
- When, Event - Select **Map Changed [Map]**.
- Client-side Condition, Type - Select **Item is not null**.
- Client-side Condition, Items - Select **PX_BBOX** , where *X* is the page number.

Next, add TRUE actions.

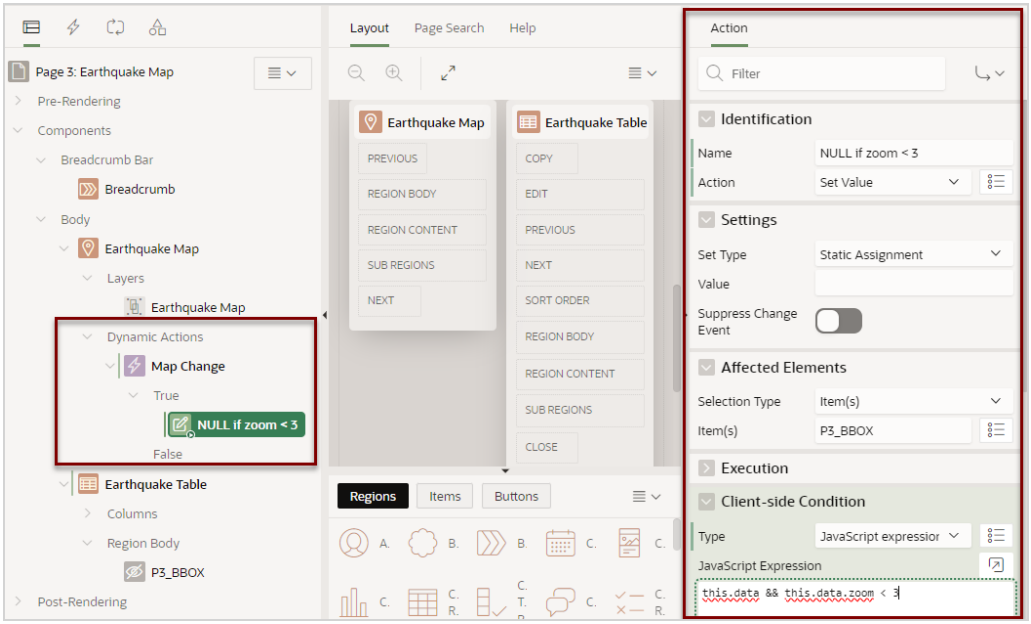
8. Add a Set Value Action which resets the P*X*_BBOX item (which is set by the map region) to NULL, if the current zoom level is below 3.

a. In the Rendering tab, select the existing True action, **Show**.

b. Edit the following attributes in the Property Editor:

- Identification, Name - Enter the name of the action (for example, `NULL if zoom < 3`).
- Identification, Action - Select **Set Value**.
- Settings, Set Type - Select **Static Assignment**.
- Affected Elements, Item(s) - Select **PX_BBOX** , where *X* is the page number.
- Client-side Condition, Type - Select **JavaScript expression**.
- Client-side Condition, JavaScript expression - Enter:

```
this.data && this.data.zoom < 3
```



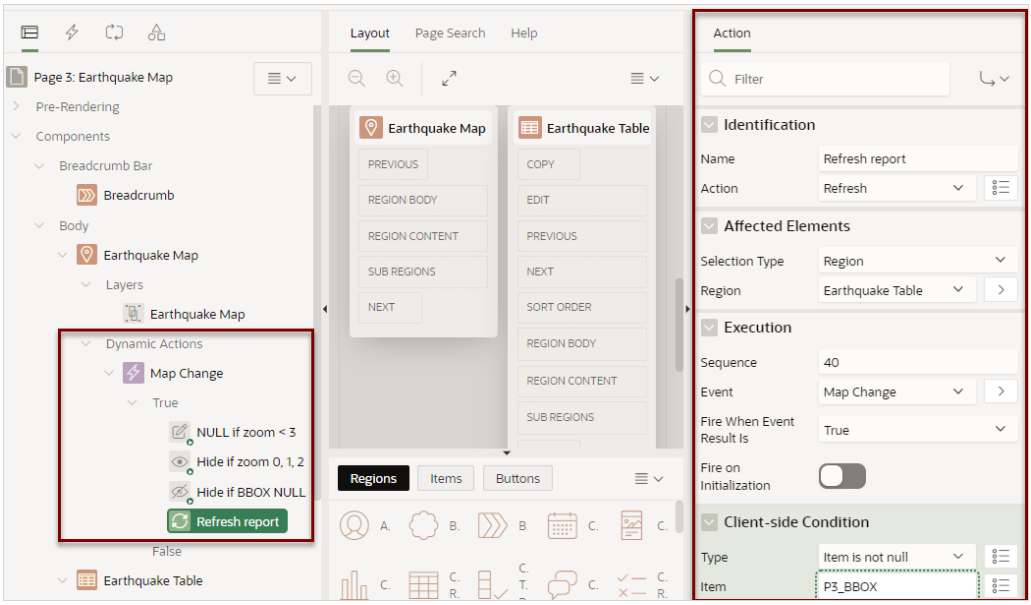
9. Add a Show Action to show the report region if P*X*_BBOX is not NULL. This action in combination with the Set Value action will hide the report for zoom levels 0, 1 and 2.

a. In the Rendering tab, select the **Map Change** dynamic action and select **Create TRUE Action**.

b. Edit the following attributes in the Property Editor:

- Identification, Name - Enter the name of the action (for example, **Hide if zoom 0, 1, 2**).
- Identification, Action - Select **Show**.
- Affected Elements, Selection Type - Select **Region**.
- Affected Elements, Region - Select **Earthquake Table**.
- Client-side Condition, Type - Select **Item is not null**.
- Client-side Condition, Item - Select **PX_BBOX** , where *X* is the page number.

10. Add Hide Action to hide the classic report if P*X*_BBOX is NULL.
    a. In the Rendering tab, right-click the **Map Change** dynamic action and select **Create TRUE Action**.
    b. Edit the following attributes in the Property Editor:
       - Identification, Name - Enter the name of the action (for example, **Hide if BBOX NULL**).
       - Identification, Action - Select **Hide**.
       - Affected Elements, Selection Type - Select **Region**.
       - Affected Elements, Region - Select **Earthquake Table**.
       - Client-side Condition, Type - Select **Item is null**.
       - Client-side Condition, Item - Select **PX_BBOX** , where *X* is the page number.
       - Click **Save**.
11. Add Refresh Action to refresh the classic report if it is visible (that is, if P*X*_BBOX is not NULL).
    a. In the Rendering tab, right-click the **Map Change** dynamic action and select **Create TRUE Action**.
    b. Edit the following attributes in the Property Editor:
       - Identification, Name - Enter the name of the action (for example, **Refresh report**).
       - Identification, Action - Select **Refresh**.
       - Affected Elements, Selection Type - Select **Region**.
       - Affected Elements, Region - Select **Earthquake Table**.
       - Client-side Condition, Type - Select **Item is not null**.
       - Client-side Condition, Item - Select **PX_BBOX** , where *X* is the page number.
       - Click **Save**.

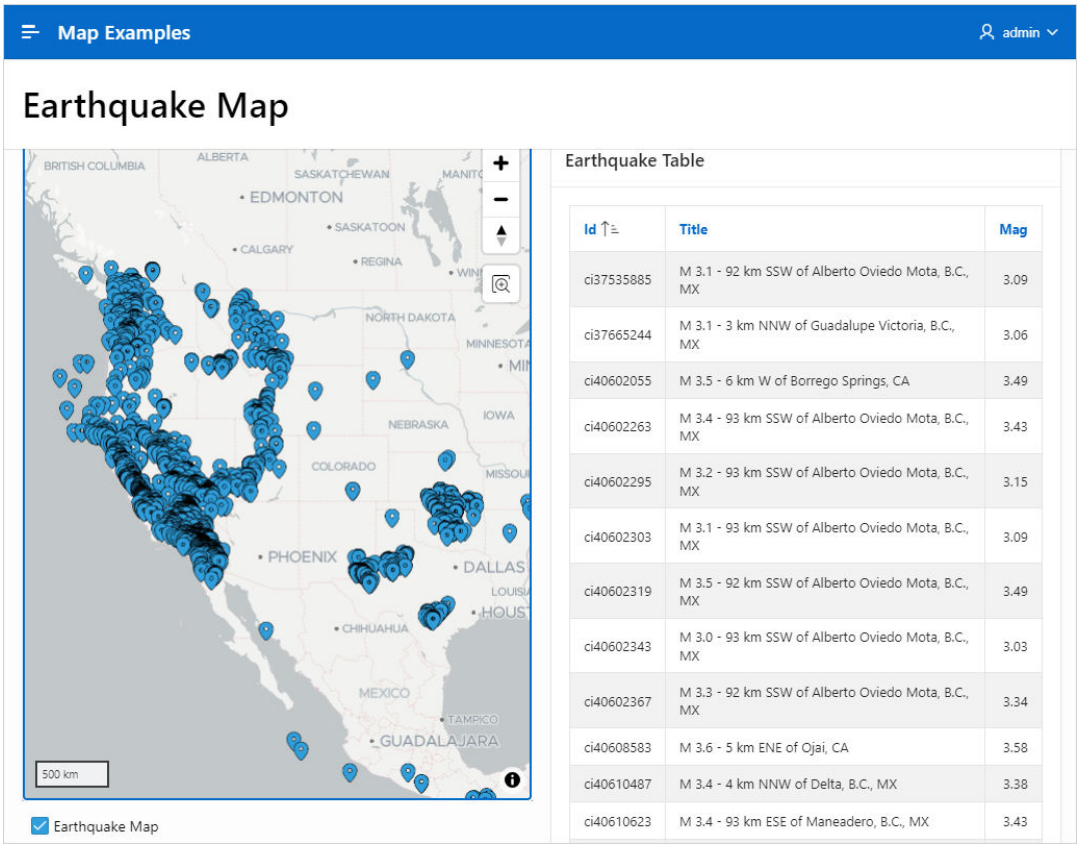


12. Click **Save and Run Page**.

    Initially, the report does not appear.

13. Reposition the map over North America and click **Zoom** (+) twice.

    The Earthquake Table appears.



If you reposition the map, Earthquake Table updates as well.

10.4.7 Creating Custom Map Backgrounds

Create custom map backgrounds for use in map regions, Display Map items, and Geocoded Address items.

- About Supported Map Background Types
  Learn about supported map background types.
- Creating a Custom Map Background
  Create a custom map background by selecting the application and navigating to Shared Components, Other Components, Map Backgrounds.
- Editing a Custom Map Background
  Edit a custom map background on the Map Backgrounds page in Shared Components.
- Deleting a Custom Map Background
  Delete a custom map background on the Map Background page in Shared Components.
- Copying a Custom Map Background
  Copy a custom map background on the Map Backgrounds page in Shared Components.
- Accessing Map Background Reports
  View the map background Utilization and History reports by clicking the appropriate tab on the Map Backgrounds page.
- Using Custom Map Backgrounds
  Use custom map backgrounds in a map regions, Display Map items, or Geocoded Address items.

10.4.7.1 About Supported Map Background Types

Learn about supported map background types.

App Builder supports the following map background types:

- **Vector Tile Layer** - Requires an URL that points to the *style.json* file describing the map layer.
- **Raster XYZ Tile Layer** - Requires an URL to fetch the raster tile layer where values for x, y, and z can be specified.
- **OGC WMS** - Points to an URL from the Web Map Service, from the Open Geospatial Consortium.

10.4.7.2 Creating a Custom Map Background

Create a custom map background by selecting the application and navigating to Shared Components, Other Components, Map Backgrounds.

To create a new map background, you need the following information from your map service provider:

- Map Background Type - Supported background map types include **Raster**, **Vector**, and **OGC WMS**.
- API Key - Only required if supplied by the map service provider.
- URL - URL to the map layer. This attribute that displays in the wizard changes based on the map background Type.

To create a custom map background:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
>
>
>) on the Page Designer toolbar.

2. Under User Interface, click **Map Backgrounds**.
3. Click **Create**.

   The Create Map Background Wizard appears.

> **Tip:**
>
> You can also copy a map background from another application. Click **Copy from another app** and follow the on-screen instructions.

4. Name - Enter a name for this Map Background.
5. Type - Select the type of the map tile layer retrieved from the URL. Options include:
   - **Vector Tile Layer** - Retrieves a map tile layer described in a `style.json` file from an URL.
   - **Raster XYZ Tile Layer** - Renders a raster tile that is retrieved from an URL passing x, y, and z values.
   - **OGC WMS** - Retrieves a layer from the Web Map Service from the Open Geospatial Consortium.
6. API Key - If required by the provider, enter the API Key value that can be set in the URL.
7. The **URL to Map** layer attribute displays next. The attributes name changes depending upon the selected map **Type**.

> **Note:**
>
> For all options, you can add your API key (if provided) and using the placeholder `{api-key}` have its value replaced in the URL (for example, `http://example.com/path/{api-key}`).

   - If **Type** equals **Vector Tile Layer** - **URL to Map Stylesheet** - Retrieves a map tile layer described in a `style.json` file from an URL.

- If **Type** equals **Raster XYZ Tile Layer** - **Raster Tiles (XYZ) URL**- URL used to fetch a raster tile passing x, y, and z values.
- If **Type** equals **OGC WMS** - **WMS URL**- URL used to fetch a map tile layer from the Web Map Service - Open Geospatial Consortium. Pass in the URL with all the required WMS parameters, except the following ones, which are added automatically by the Map Region:.
  - BBOX
  - WIDTH
  - HEIGHT
  - REQUEST
  - FORMAT
  - SRS / CRS

Examples:

```
https://wms.example.com/wms?
service=WMS&version=1.3.0&layers=maplayer&styles=default
```

```
https://wms.example.com/wms?service=WMS&version=1.1.1&layers=countries,rivers
```

8. Click **Create Map Background**.

10.4.7.3 Editing a Custom Map Background

Edit a custom map background on the Map Backgrounds page in Shared Components.

To edit a custom map background:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components**
> (
>
>) on the Page Designer toolbar.

2. Under User Interface, click **Map Backgrounds**.
3. On Map Backgrounds, click a background Name.

   The Map Background appears.
4. Edit the appropriate settings.

   The attibutes that appear differ depending upon the map background type. For example, Raster XYZ or OGC WMS map tile layers, include an **Attribution** setting. You can use this setting to include text, such as a copyright notice, on the lower right corner of map.

To learn more about an attribute, see field level Help.

5. Edit the appropriate attributes and click **Apply Changes**.

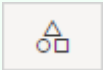
10.4.7.4 Deleting a Custom Map Background

Delete a custom map background on the Map Background page in Shared Components.

To delete a custom map background:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
> 
>
>) on the Page Designer toolbar.

2. Under User Interface, click **Map Backgrounds**.
3. On Map Backgrounds, click a background Name.

   The Map Background appears.
4. Click **Delete**.

10.4.7.5 Copying a Custom Map Background

Copy a custom map background on the Map Backgrounds page in Shared Components.

> **Tip:**
>
> You can also copy a map background from another application while running the Create Map Background wizard. See Creating a Custom Map Background.

To copy a custom map background:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
>
>
>) on the Page Designer toolbar.

2. Under User Interface, click **Map Backgrounds**.

   The Map Backgrounds page appears.
3. If the page does not display as a report, click **View Report** on the navigation bar.
4. Locate the Map Background to copy and click the **Copy** icon in the Copy column.

   The Copy Map Background dialog appears.
5. On Copy Map Background:
   a. New Map Background Name - Identify the name of the map background you would like to create. Map background name must be unique within the application.
   b. Click **Copy Map Background**.

10.4.7.6 Accessing Map Background Reports

View the map background Utilization and History reports by clicking the appropriate tab on the Map Backgrounds page.

To view map background reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, click **Map Backgrounds**.

   The Map Backgrounds page appears.
3. Click the appropriate tab:
   - **Utilization** - Displays where map backgrounds are being used within the application.
   - **History** - Displays recent modifications made to custom map backgrounds in this application.

10.4.7.7 Using Custom Map Backgrounds

Use custom map backgrounds in a map regions, Display Map items, or Geocoded Address items.

- Applying a Map Background to a Map Region
  Apply a map background to a map region in Page Designer.
- Applying a Map Background to a Display Map Item
  Apply a map background to a Display Map item in Page Designer.

- Configuring a Map Background for Geocoded Address Items
  Configure a custom map background for Geocoded Address items in Component Settings, Geocoded Address.

10.4.7.7.1 Applying a Map Background to a Map Region

Apply a map background to a map region in Page Designer.

> **Note:**
>
> This task assumes you have already created an application with a Display Map item and a custom map background as described in Creating a Custom Map Background.

To apply a map background to a map region:

1. Create a custom map background as described in Creating a Custom Map Background.
2. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears displaying Page attributes in the Property Editor (right pane).
3. In the Rendering tab (left pane), select the Map region.

   The Property Editor (right pane) displays two tabs: **Region** and **Attributes**.
4. In the Property Editor, click the **Attributes** tab.
5. In the **Map** group, edit the following attributes:
   a. Map, Background - Select the background map to use.

      Options include:

      - **Default** - Use default background map settings.
      - **Built-In** - Pick the map tile layers to use for the background map.
      - **Shared Component** - Select the custom map tile layers from the shared components to use as a background map.

      Note that the UI changes based on your selection. For this example, select **Built-In**.
   b. Map, Standard - Choose the tile layer to use as the background map.
   c. Map, Dark Mode - Choose the tile layer to use as the background map when dark mode is enabled.
   d. Review and edit any other attributes as appropriate.

      To learn more about an attributes, select the attribute in Page Designer and select the **Help** tab.
6. Click **Save** or **Save and Run Page**.

10.4.7.7.2 Applying a Map Background to a Display Map Item

Apply a map background to a Display Map item in Page Designer.

A Display Map item a map based on coordinates in GeoJSON point format. See Managing Page Items and Display Map.

> **Note:**
>
> This task assumes you have already created an application with a Display Map item and a custom map background as described in Creating a Custom Map Background.

To apply a map background to a Display Map item:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears displaying Page attributes in the Property Editor (right pane).
2. In the Rendering tab, select the Page Map item.
3. In the Property Editor, Page Item tab:
   a. Settings, Background - Select **Shared Component**.

      New attributes appear.
   b. Settings, Custom Background - Select a map background defined in Shared Components.
   c. Review and edit any other attributes as appropriate.

      To learn more about an attributes, select the attribute in Page Designer and select the Help tab.
4. Click **Save** or **Save and Run Page**.

10.4.7.7.3 Configuring a Map Background for Geocoded Address Items

Configure a custom map background for Geocoded Address items in Component Settings, Geocoded Address.

A Geocoded Address item provides geocoding functionality that turns a postal address to a coordinate. To learn more, see Managing Page Items and Geocoded Address.

> **Note:**
>
> This task assumes you have already created an application with a Display Map item and a custom map background as described in Creating a Custom Map Background.

To configure a custom map background for Geocoded Address: items:

1. Navigate to the Shared Components page:

a. On the Workspace home page, click **App Builder**.
b. Select an application.
c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

Page Designer appears displaying Page attributes in the Property Editor (right pane).

2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Geocoded Address**.
4. **Background** - Select Shared Component.
5. **Custom Background** - Only displays if **Background** is set to **Shared Component**. Select a background map to use.
6. Click **Apply Changes**.

> **See Also:**
>
> Configuring Geocoded Address

10.5 Creating Help for Your Application

Help created in App Builder displays on a dedicated Help page. To access Help, users click a link that takes them to a dedicated Help page. This Help page displays page and field-level Help topics specific to the page they are viewing.

- Creating Help
  Developers can create both a dedicated Help page and item Help text for an application.
- Using the Edit Item Help Report
  Edit all Help topics within an application using the Edit Item Help report
- Seeding Item Help Topics
  For applications that do not yet contain Help, you can perform a mass update (or seed) of default Help text.
- Creating a Help Navigation Bar Entry
  Once you have created your Help, the next step is to create a navigation bar entry so users can link to it.

10.5.1 Creating Help

Developers can create both a dedicated Help page and item Help text for an application.

- About Creating Help
  Learn about built-in attributes to create Help for your application.
- Creating a Help Page and Region
  Learn about creating a help page and region.
- Defining Page Help
  Learn defining page Help text.
- Defining Item Help
  Define item Help in Page Designer.

10.5.1.1 About Creating Help

Learn about built-in attributes to create Help for your application.

App Builder includes built-in attributes to create Help for your application. Creating Help for your application involves the following steps:

- **Step 1:** Create a dedicated Help page and Help region. See Creating a Help Page and Region.
- **Step 2:** Define page Help text. See Defining Page Help.
- **Step 3:** Define item Help text. See Defining Item Help.
- **Step 4:** Create a navigation bar icon to link to your Help page. See Creating a Help Navigation Bar Entry.

10.5.1.2 Creating a Help Page and Region

Learn about creating a help page and region.

To create a Help page and region in Page Designer:

1. Create a blank page for your Help.
2. View the new page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
3. In the Gallery, right-click **Help Text** region, select **Add To**, and select the appropriate location.

   > **Tip:**
   >
   > You can also drag and drop from the Gallery. From the Gallery select the **Help Text** region and drag it to the appropriate location in the Layout tab.

4. In the Property Editor, edit the following attributes:
   a. Identification, Title - Enter a title for the region.
   b. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
   c. Layout, Parent Region - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region.
   d. Layout, Position - Specify the Sequence, Parent Region, and Position.
   e. Appearance, Template - Select a region template to define the appearance and layout of this region. Region templates are defined in the application theme. When a region is created, the template is automatically set to the default region template defined in the current theme.

f. Appearance, Render Components - Select if the region components display above or below the main region content.

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

5. Click **Save**.

> **Tip:**
>
> For more information on supported chart options, see ojChart in JSDoc pages for the Oracle JavaScript Extension Toolkit (JET).

> **See Also:**
>
> Adding a New Page to an Application

10.5.1.3 Defining Page Help

Learn defining page Help text.

To define page Help text:

1. View the new page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, click the page name.
3. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In Property Editor - Page, find **Help** and enter text in **Help Text** attribute.
5. Click **Save**.

10.5.1.4 Defining Item Help

Define item Help in Page Designer.

To define item Help:

1. View the new page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, click item for which you want to define Help.
3. To find a group or attribute:
   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. In the Property Editor, find**Help** and enter text in **Help** attribute.
5. Click **Save**.
6. Repeat the previous steps for each item requiring Help text.

10.5.2 Using the Edit Item Help Report

Edit all Help topics within an application using the Edit Item Help report

Use the Edit Item Help report to view the item Label and Name, access the item, or link to the page containing the item to the associate page

To view the Edit Item Help report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. Under Page Specific Utilities, select **Item Utilities**.
5. Click **Edit Item Help Text**.
6. On the Edit Item Help age, you can:
   - Edit the Help text that appears and click **Apply Changes**.
   - Link to the page containing the item by clicking the page number.
   - Link to the page item by clicking the item name.

> **See Also:**
>
> Seeding Item Help Topics

10.5.3 Seeding Item Help Topics

For applications that do not yet contain Help, you can perform a mass update (or seed) of default Help text.

To create help text for all items that currently do not have help text

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. Under Page Specific Utilities, select **Item Utilities**.
5. Click **Edit Item Help Text**.

   The Edit Item Help page appears. See Using the Edit Item Help Report .
6. Click **Seed Item Help Text**.
7. Click **Apply Changes**.

10.5.4 Creating a Help Navigation Bar Entry

Once you have created your Help, the next step is to create a navigation bar entry so users can link to it.

To create a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

   The Create Navigation Bar Entry Wizard appears.
3. For Method, select **From Scratch** and click **Next**.
4. For Type, select **Navigation to URL** and click **Next**.
5. For Attributes, specify the following:
   a. Sequence - Specify the order of evaluation for this component.
   b. Entry Label - Enter display text for this navigation bar entry.
   c. Icon Image Name - Enter an image name. For naming conventions, see field-level Help.
   d. Image ALT - Enter ALT text for navigation icons that are images. If you do not specify an image name, then this text displays.
   e. Image Height - Defines the height of the image.
   f. Width - Defines the width of the image.
   g. Click **Next**.

6. For Target:
   a. Target is a - Select **Page in this application**.
   b. Page - Specify the page number of the help page you created in Creating Help .
   c. Request - Enter the following:

      ```
      &APP_PAGE_ID.
      ```

      By specifying substitution string *&APP_PAGE_ID* as the Request, you are instructing the Oracle APEX engine to display Help text for the current page when the user clicks this icon.
   d. Click **Next**.
7. To set a condition for displaying the navigation bar entry, select a condition type.
8. Click **Create**.

11

Using Themes and Theme Styles

Developers can alter an application's user interface and page layout through themes and theme styles.

A **theme** is collections of templates that enable developers to define the layout and style of an entire application. A **theme style** is a CSS that is added to the base CSS.

- Using Themes
  Learn about the Universal theme and how to control the layout of your application using theme styles, Theme Roller, and template options.
- Managing Themes and Subscriptions
  Developers can manage themes directly or through the use of theme subscriptions.
- Creating Custom Themes
  Create a custom theme by modifying existing templates.
- Using Custom Cascading Style Sheets
  Control the style of a web page by uploading and referencing a cascading style sheet.

> **See Also:**
>
> How Oracle APEX Uses Templates

11.1 Using Themes

Learn about the Universal theme and how to control the layout of your application using theme styles, Theme Roller, and template options.

- About Themes
  Themes are collections of templates that enable developers to define the layout and style of an entire application.
- Understanding the Universal Theme
  Universal Theme - 42 (Universal Theme) features a responsive design and enables developers to create web applications without extensive knowledge of HTML, CSS, or JavaScript.
- Using Theme Styles
  A theme style is a CSS that is added to the base CSS. Using the Theme Roller utility, developers can change the appearance of an application. Universal Theme - 42 includes theme styles.
- Using Theme Roller
  Use Theme Roller to quickly transform the appearance of an application without editing any code.
- Understanding Template Options
  Template options provide developers with a declarative approach to applying different styles to components on an Oracle APEX page.

> **See Also:**
>
> Managing Themes and Subscriptions

11.1.1 About Themes

Themes are collections of templates that enable developers to define the layout and style of an entire application.

Themes provide developers with a complete set of templates that accommodate every UI pattern that may be needed in an application. Templates are organized first by template type and then by template class. Template types include page, region, report, list, button, label, and popup list of values (LOV).

Each template type has a number of template classes. A template class defines the purpose of a template within a template type. For example, a region template can be classified as a form region template, a report region template, and so on. These classifications enable Oracle APEX to map templates among themes, making it easy to quickly change the entire look and feel of an application.

Administrators can add themes to the theme repository as follows:

- **Workspace Themes** - Workspace administrators can create themes that are available to all developers within the workspace. See Managing Workspace Themes.
- **Public Themes** - Instance administrators can create public themes by adding them to the Oracle APEX Administration Services. Once added, these public themes are available to all developers across all workspaces in an instance. See Managing Public Themes in *Oracle APEX Administration Guide*.

11.1.2 Understanding the Universal Theme

Universal Theme - 42 (Universal Theme) features a responsive design and enables developers to create web applications without extensive knowledge of HTML, CSS, or JavaScript.

> **Tip:**
>
> To view more details and best practices when using the Universal Theme, run the Universal Theme Reference app. See Accessing the Universal Theme Reference App.

- About the Universal Theme
  The Universal Theme features a responsive design, versatile UI components, and can be easily customized.
- About Updating or Migrating to the Universal Theme
  Update or migrate existing apps to the newest Universal Theme to take advantage of new features and updates.
- Refreshing the Universal Theme
  Update an application to use the latest Universal Theme to take advantage of new features, enhancements, and bug fixes as well as styling for new APEX components.

11.1.2.1 About the Universal Theme

The Universal Theme features a responsive design, versatile UI components, and can be easily customized.

Applications you create with the Create Application Wizard use the Universal Theme. Advantages of the Universal Theme include:

- **Responsive Design** - Designed to work just as well on small screen devices (such as smartphones and tablets) as it does on larger screen devices (including laptops and desktops). The UI components in Universal Theme work across varying screen resolutions while maintaining the same or similar functionality. In addition, Universal Theme takes full advantage of ultra high screen resolutions by utilizing vector graphics where possible, and relying upon CSS3 features for UI styling.
- **Versatile User Interface** - Provides all the components and building blocks necessary to build practically any type of business application user interface.

  > **Tip:**
  >
  > To browse all of the components provided with Universal Theme, run the Universal Theme Reference app. From the **Main Navigation** menu, select **Components**.

- **Easy Customization** - Effortlessly customize and fully control the look and feel of your applications without becoming an expert in UI design, HTML, CSS, or JavaScript. Using Theme Roller and Template Options, you can easily customize your application to fit your company's brand and customize the look and feel of various components using Template Options.
- **Support for Theme Styles** - The Universal Theme includes support for theme styles. A theme style is a CSS style sheet that is added to the base CSS. Developers can change the appearance of an application by altering the theme style using the Theme Roller utility.

About Responsive Design

The Universal Theme is an example of a responsive user interface theme.

Responsive design enables you to design web pages so that the layout fits the available space regardless of the device on which page displays (for example, a desktop computer, laptop computer, tablet, or smartphone).

By implementing a responsive design, the user gets the same full experience as they would on larger screens. On smartphones and tablets, the layout adjusts to the size of the specific device. During this resizing process, elements shift position, re-size, or become hidden. The goal of responsive design is to present all essential content in a user friendly way for all possible screen sizes. Keep in mind, that responsive design is not just a matter of picking the correct set of templates. As the application developer, you are responsible for using the templates and the available components to design a page that is truly responsive.

See Also:

- Accessing the Universal Theme Reference App
- Understanding Template Options
- Using Theme Styles

11.1.2.2 About Updating or Migrating to the Universal Theme

Update or migrate existing apps to the newest Universal Theme to take advantage of new features and updates.

Compared to themes in previous releases of Oracle APEX, the Universal Template features cleaner templates, improved grid support, and responsive behavior.

Universal Theme migration paths include:

- **Refresh the Universal Theme** - If your app already uses the Universal Theme, you can easily update it to take advantages of new features and enhancements.
- **Migrate from Other Themes** - Update your application to use the latest version of Universal Theme and keep current with the latest features, enhancements and bug fixes.

See Also:

Migration Guide page in the Universal Theme Reference app.

11.1.2.3 Refreshing the Universal Theme

Update an application to use the latest Universal Theme to take advantage of new features, enhancements, and bug fixes as well as styling for new APEX components.

Tip:

Before starting this procedure, Oracle recommends backing up your application by either exporting it or creating a copy. See Exporting an Application and Copying an Application.

To update an application to use the latest Universal Theme

1. Navigate to the App Builder home page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

      If the Universal Theme is not up-to-date, the App Builder home page displays a banner at the top of the page that reads:

      ```
      There is a new version of Universal Theme available! You can refresh your
      theme to take advantage of new features, enhancements, and bug fixes.
      ```

2. Click **Refresh Theme**.
3. Run and test your application.

> **See Also:**
>
> Migration Guide page in the Universal Theme Reference app.

11.1.3 Using Theme Styles

A theme style is a CSS that is added to the base CSS. Using the Theme Roller utility, developers can change the appearance of an application. Universal Theme - 42 includes theme styles.

- About Theme Styles
  A theme style is a CSS style sheet that is added to the base CSS to alter the look and feel of an application.
- Creating a Theme Style
  Create a theme style by selecting an existing style and selecting Add Style.
- Editing a Theme Style
  Edit a theme style from the Themes page.
- Selecting a Theme Style on the User Interface Page
  Select a theme style on the application User Interface page.
- Enabling Users to Select a Theme Style
  Enable users to select a theme style in a running application.

11.1.3.1 About Theme Styles

A theme style is a CSS style sheet that is added to the base CSS to alter the look and feel of an application.

Newer themes such as *Universal Theme - 42* can have a base CSS file plus the theme style CSS file. The theme style CSS file are referenced in the page template using the `#THEME_STYLE_CSS#` substitution string. The Oracle APEX engine replaces this substitution string with the CSS file references defined in the theme style attributes. Use theme styles to customize themes, to switch to a different color scheme, apply a flat look, or make a theme responsive. A theme can have multiple theme styles with one style set as active. You can modify a theme style CSS file using Theme Roller.

Selecting a Theme Style

Once defined, developers can select a theme style by:

- Editing the Theme Styles attribute when running the Create Application Wizard. See Understanding Page Types, Features, and Settings.
- Accessing the Create/Edit Theme page. See Editing a Theme Style.
- Select a new theme style on the application User Interface page. See Selecting a Theme Style on the User Interface Page.

11.1.3.2 Creating a Theme Style

Create a theme style by selecting an existing style and selecting Add Style.

To create a theme style:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.
2. Select a theme and click the **Styles** tab.

   The **Current Theme Style** list displays the current Theme Style.
3. Click **Add Style**.

   The Theme Styles page appears.
4. For Settings:
   a. Name - Provide a short descriptive name for the theme style.
   b. Is Public - Select whether this style can be chosen by end users.
   c. Accessibility Tested - Specify if this Theme Style has been tested to the applicable accessibility guidelines. For Theme Styles created by Oracle, please see our product VPAT for the accessibility guidelines we follow..
   d. File URLs - Enter Cascading Style Sheet file URLs to be loaded on every page if the theme style is current. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL. File URLs you enter here will replace the `#THEME_STYLE_CSS#` substitution string in the page template. See field-level Help to view examples.
   e. CSS Classes - Enter CSS classes to be appended to the`#PAGE_CSS_CLASSES#` substitution string and applied to every page of the application if this Theme Style is currently active.
5. For Theme Roller Attributes:
   a. Read Only - Select **On** to prevent any updates to the theme style. Select **Off** to allow the theme style to be edited and overwritten using Theme Roller.
   b. Input Parameter File URLs - Enter the LESS file URLs that will be used to generate this theme style when using Theme Roller.
   c. Output CSS File URL- Enter the URL for the CSS file that will be generated by Theme Roller for this theme style. .
   d. Theme Roller JSON Configuration, JSON Configuration - The JSON configuration is generated by Theme Roller when the theme style is saved.

      Oracle does not recommend manually updating the configuration. However, you can copy a JSON configuration from another theme style, such as a theme style in another application, to manually update the configuration.
6. For Comments - Enter any comments or notes here. These comments will never be displayed when running the application.

7. Click **Create**.

11.1.3.3 Editing a Theme Style

Edit a theme style from the Themes page.

To edit a theme style:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.
2. Click the theme Name.
3. On Create / Edit Theme, click the **Styles** tab.

   The Styles tab displays the available theme styles.
4. Current Theme Style - Displays the current theme style. To change the current theme style, select another theme style from the list.
5. To edit a theme style:
   a. On the Styles tab, click the theme style Name.
   b. On the Theme Styles page, edit the appropriate attributes.

      To learn more about an attribute, see field-level help.
   c. To save your changes, click **Apply Changes**.
6. To delete a theme style, click **Delete**.
   a. On the Styles tab, click the theme style Name.
   b. On the Edit Theme Styles page, edit the appropriate attributes.

      To learn more about an attribute, see field-level help.
   c. Click **Delete**.

11.1.3.4 Selecting a Theme Style on the User Interface Page

Select a theme style on the application User Interface page.

To select a theme style from the User Interface page:

1. Navigate to the User Interface page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, click **User Interface Attributes**.

      The User Interface page appears.
2. Attributes, Theme Style - Select a new theme style.

   The Theme Styles page appears.
3. Click **Apply Changes**.

11.1.3.5 Enabling Users to Select a Theme Style

Enable users to select a theme style in a running application.

Once enabled, a Customize link appears in the running application, typically in the lower left corner. When the user clicks the Customize link a pop-up displays listing available theme styles. When a user selects a theme style, the theme style is stored persistently as a user preference and will be retained during future sessions.

To enable users to select a theme style:

1. Create an application which has at least two theme styles.
2. Go to the application User Interface page:
   a. Click **Shared Components**.
   b. Under User Interface, click **User Interface Attributes**.

      The User Interface Attributes page appears..
3. Under Attributes, enable **Enable End Users to choose Theme Style**.
4. Click **Apply Changes**.

11.1.4 Using Theme Roller

Use Theme Roller to quickly transform the appearance of an application without editing any code.

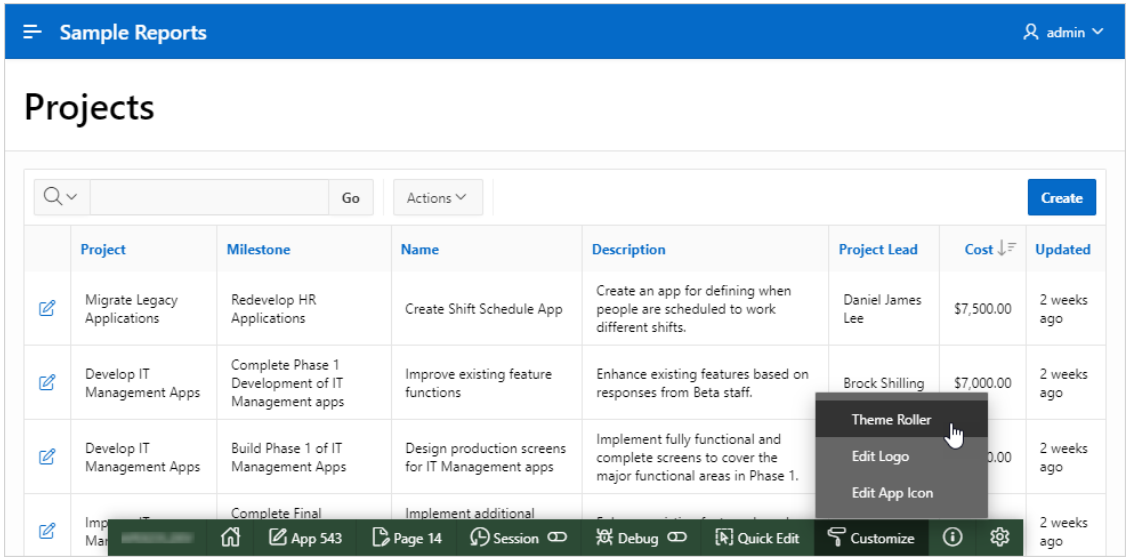
- About Theme Roller
  Learn about the Theme Roller option available on the Customize menu on the Runtime Developer toolbar.
- Using the Theme Roller Dialog
  Access the Theme Roller dialog by clicking Theme Roller from the Customize menu on the Runtime Developer toolbar.
- About the Redwood Light Theme Style
  The Redwood Light theme style features a number of unique class options in Theme Roller.
- Importing and Exporting Theme Styles
  Import and export theme styles by selecting Import and Export from the Theme Roller Additional Options menu.
- Style Compatibility
  Learn about Vita theme and Redwood Light theme compatibility issues.
- About Developer Console Integration
  Use the exposed function `apex.utr.config();` in your web browser developer's console to get and set the custom CSS code and LESS compilation configuration

11.1.4.1 About Theme Roller

Learn about the Theme Roller option available on the Customize menu on the Runtime Developer toolbar.

Theme Roller is a live CSS editor that enables you to quickly change the theme style, colors, rounded corners and other application attributes.

When you run an application from App Builder and view a page, the Runtime Developer toolbar displays at the bottom of any editable running page. To access the Theme Roller dialog, click **Theme Roller** from the Customize menu.



In order for Theme Roller to appear on the Runtime Developer toolbar:

- Theme styles must be defined.
- At least one theme styles must have the **Input Parameter File URLs** attribute defined.

> **See Also:**
>
> - Creating a Theme Style
> - Editing a Theme Style
> - Using the Theme Roller Dialog

11.1.4.2 Using the Theme Roller Dialog

Access the Theme Roller dialog by clicking Theme Roller from the Customize menu on the Runtime Developer toolbar.

> **Note:**
>
> Concurrent use of the Live Template Options, Theme Roller, and Logo Editor is not supported.

Theme Roller is a live CSS editor that enables developers to quickly change the colors, rounded corners and other attributes of their applications without touching a line of code. Out-of-the-box themes cannot be overwritten but can be saved as a new theme and modified accordingly.

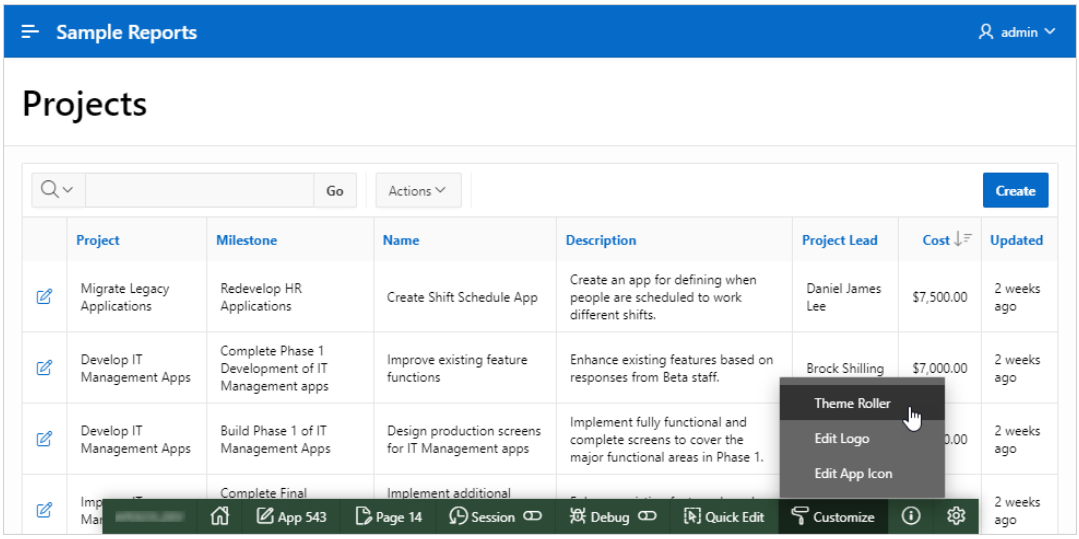
> **Tip:**
>
> Theme Roller displays on the Runtime Developer toolbar if theme styles have been defined and at least one theme styles has the **Input Parameter File URLs** attribute defined.

To use the Theme Roller dialog:

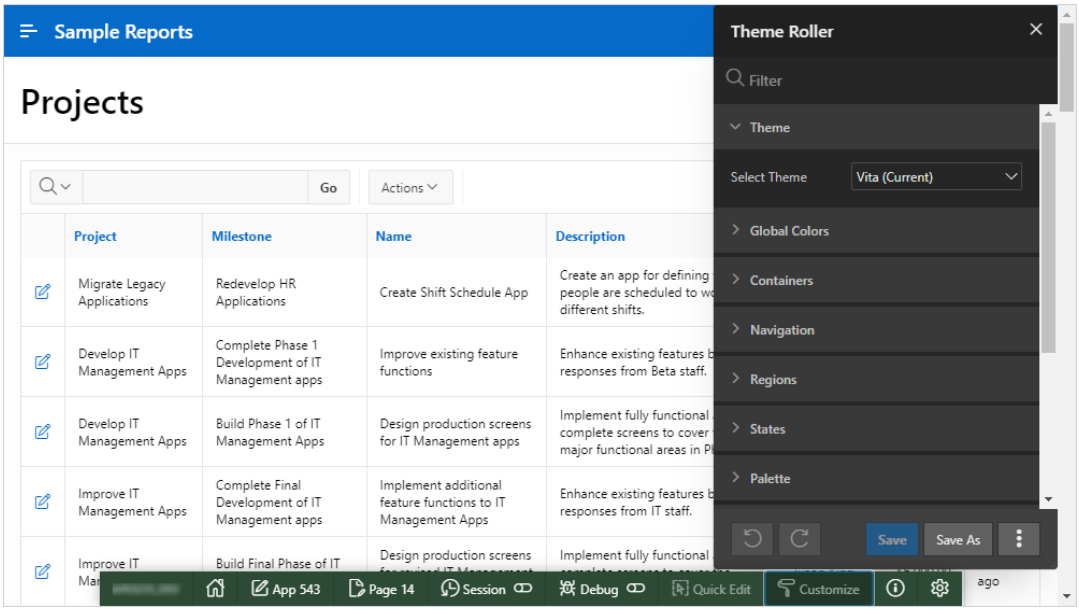
1. Preview the page by running it.

   The Runtime Developer toolbar displays at the bottom of any editable running page.

2. On the Runtime Developer Toolbar, click **Customize** and then select **Theme Roller**.



Theme Roller fetches the styles for your application and loads them in the editor. The Theme Roller dialog appears.

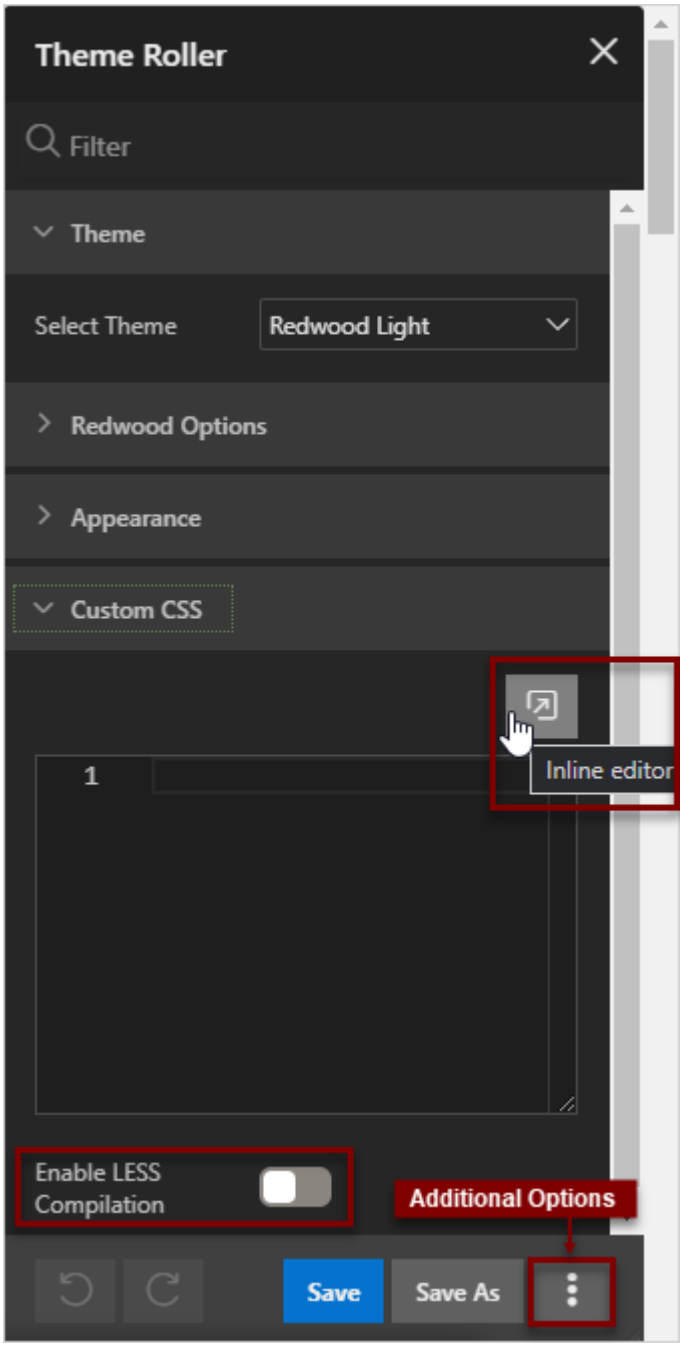


3. To search for an attribute, enter search terms in the Search field at the top of dialog. Only attributes that contain the search term appear.

> **Tip:**
>
> To navigate the Theme Roller dialog using keyboard shortcuts, press Ctrl+Fn+M and then release. Then, press Tab to navigate.

4. To select or modify a theme:
   a. Theme, Select Theme - Select a theme from the drop-down list. In the next illustration, the Theme is set to Redwood Light.
   b. Expand or collapse sections by clicking on the section name.
   c. Modify the attributes.

      Undo and Redo - When you modify an attribute, the **Undo** and **Redo** buttons at the bottom of the dialog become active.
   d. Reset a Theme Style - To revert any unsaved theme style changes, select **Reset** from the **Additional Options** menu . The **Additional Options** menu is located next to the **Save As** button at the bottom of the dialog.
5. Custom CSS - Enter CSS or or LESS-compatible code in the Code Editor. If you are including LESS-compatible code, you must enable the **Enable LESS Compilation** switch.



6. Export or Import - Export or import theme style changes directly from the Theme Roller dialog. Select **Export** or **Import** from the Additional Options menu located next to the Save As button at the bottom of the dialog.
7. Save the theme by clicking one of the following:
   - **Save** - If the theme is editable, **Save** saves any style change and sets the theme as the application's current style.
   - **Save As** - Click **Save As** to save the theme as a new style and sets it as the application's current style.

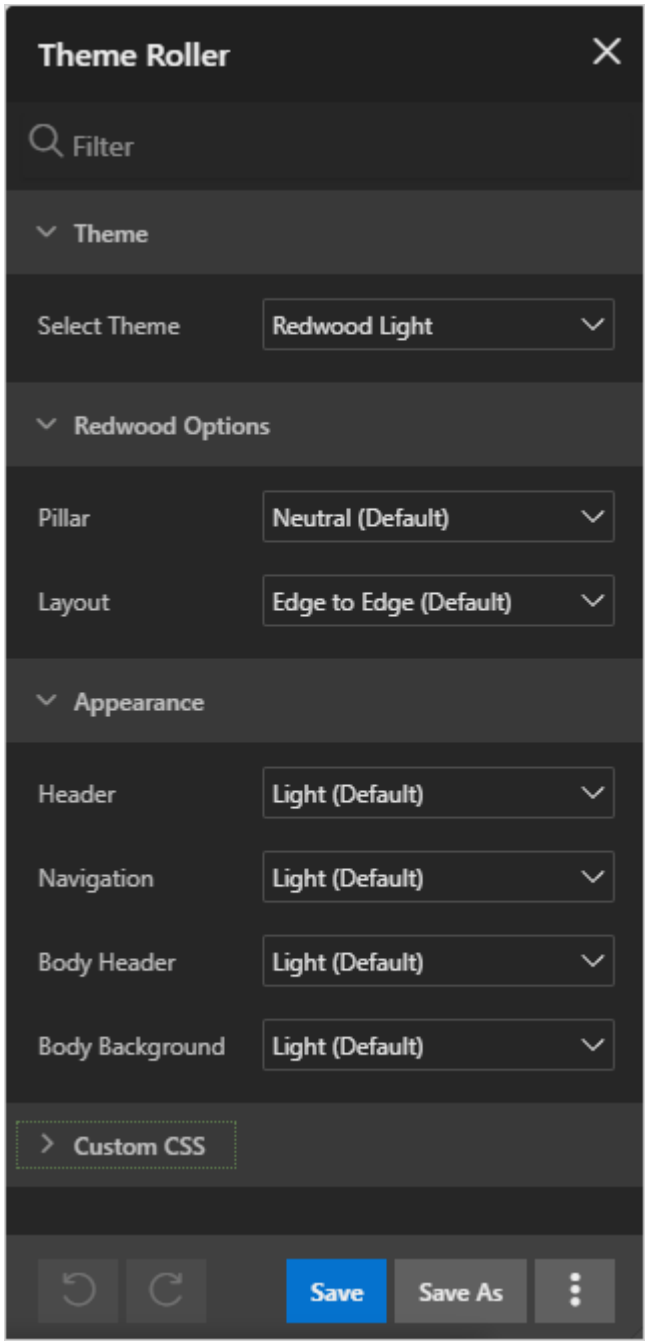
11.1.4.3 About the Redwood Light Theme Style

The Redwood Light theme style features a number of unique class options in Theme Roller.

As shown in the following image, selecting the Redwood Light theme style includes a number of unique class options:

- **Pillar** - The Pillar class enables users to change the background and header images.
- **Layout** - The Layout class enables users to switch the following layout options: **Edge to Edge**, **Floating**, or **Contained**.
- **Appearance** - Attributes under Appearance include Header, Navigation, Body Header, and Background.

The following image shows Theme Roller with the Redwood Light theme style selected.



11.1.4.4 Importing and Exporting Theme Styles

Import and export theme styles by selecting Import and Export from the Theme Roller Additional Options menu.

You can export and import style changes directly from Theme Roller by selecting options from the Additional Options menu: Import, Export, and Reset. When importing and exporting, the supported file format is JSON.

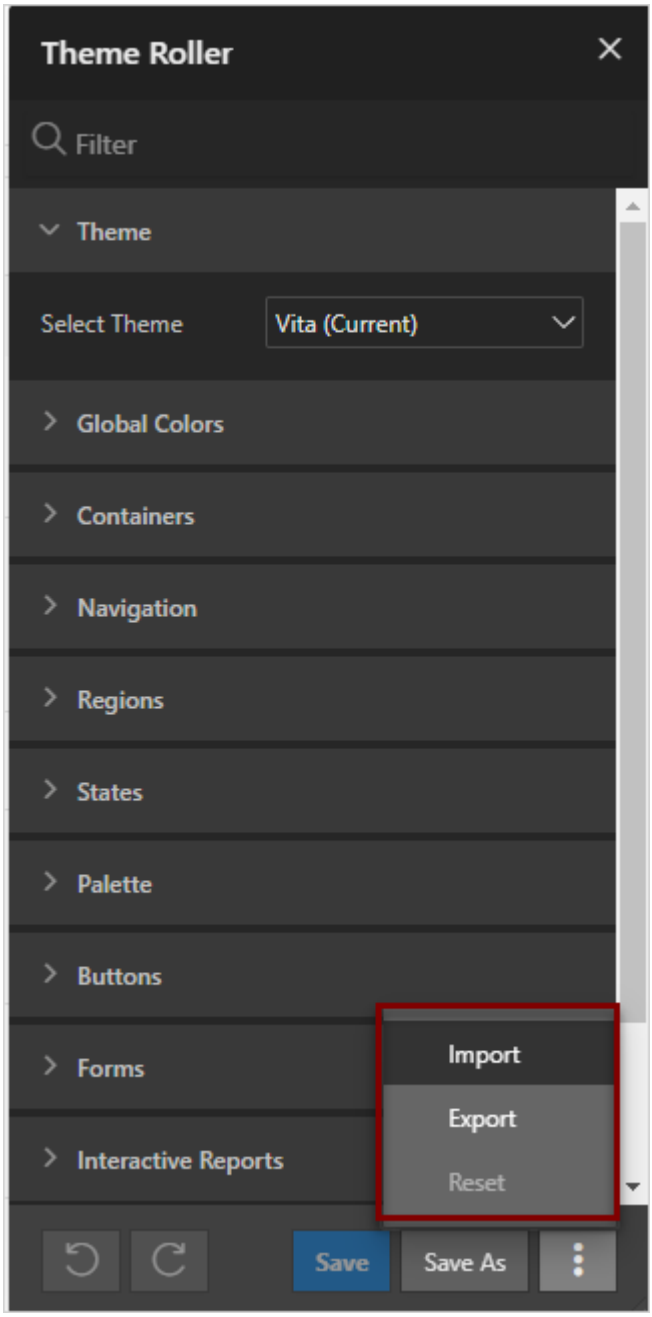
To import and export Theme Styles:

1. Preview the page by running it.

   The Runtime Developer toolbar displays at the bottom of any editable running page.

2. On the Runtime Developer Toolbar, click **Customize** and then select **Theme Roller**.

   The Theme Roller dialog appears displaying the styles for your application. You can export and import style changes directly from Theme Roller by selecting the following options from the Additional Options menu: **Import**, **Export**, and **Reset**.



> **Tip:**
>
> To navigate the Theme Roller dialog using keyboard shortcuts, press Ctrl+Fn+M and then release. Then, press Tab to navigate.

3. To import theme styles:
   a. From the Additional Options menu, select **Import**.

      A File Browser appears.
   b. Navigate to the appropriate file to set to set the custom CSS code, variable style changes, add classes and LESS compilation configuration to be applied to the current theme style.

When importing a file, the following JSON key-values are required:

```
{
    "classes": [],
    "vars": {},
    "customCSS":"",
    "useCustomLess":""
}
```

The expected values are described in the following table.

Key	Type	Value(s)
`classes`	JSON array	List of comma-separated class names that are added to the page.
`vars`	JSON object	Theme style variables with values.
`customCSS`	String	Custom CSS code.
`useCustomLess`	String	"Y" if the theme style should be compiled using LESS or "N" for no.

4. To export theme styles:
   - From the Additional Options menu, select **Export**.

     This option exports the current style configuration into a JSON file.
5. To reset theme styles:
   - From the Additional Options menu, select **Reset**.

     This option reverts any unsaved theme style changes.

11.1.4.5 Style Compatibility

Learn about Vita theme and Redwood Light theme compatibility issues.

The Vita theme and Redwood Light theme are not currently compatible because their variables are not present in each other's theme files.

When an export from the Vita theme is imported into Redwood Light, no changes display because the Vita theme variables are not present in the Redwood Light theme. The same is true for the reverse scenario. When an export from Redwood Light is imported into the Vita theme, no changes display because the Redwood Light variables are not present in the Vita theme.

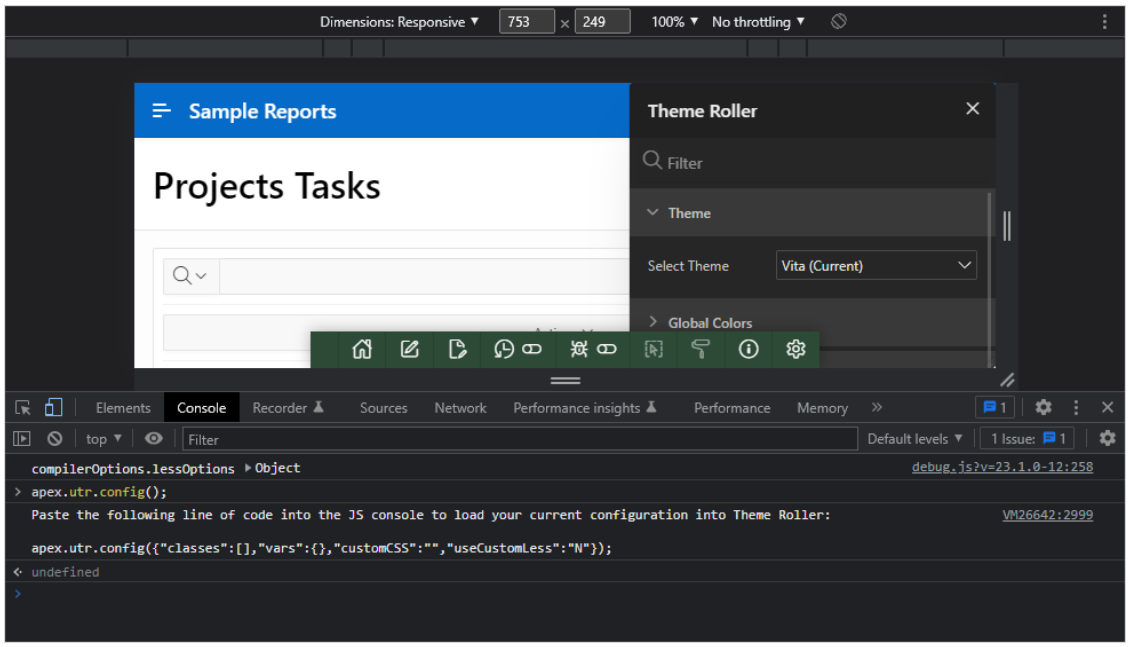
11.1.4.6 About Developer Console Integration

Use the exposed function `apex.utr.config();` in your web browser developer's console to get and set the custom CSS code and LESS compilation configuration

Use the exposed function `apex.utr.config();` in your web browser developer's console to get and set the custom CSS code and LESS compilation configuration.

To get styles, write `apex.utr.config()` without parameters into the console:

```
apex.utr.config();
Paste the following line of code into the JS console to load your current
configuration into Theme Roller:
  apex.utr.config({"classes":[],"vars":
{},"customCSS":"","useCustomLess":"Y"});
```



To set styles, add less or css code in the customCSS object of the JSON configuration into the console:

```
apex.utr.config({"classes":[],"vars":{},"customCSS":"h1 { color: #111; font-
family: 'Helvetica Neue', sans-serif; ... }", "useCustomLess":"Y"});
```

The custom CSS will be added to the theme roller Code Editor and applied to the page automatically.

11.1.5 Understanding Template Options

Template options provide developers with a declarative approach to applying different styles to components on an Oracle APEX page.

- About Template Options
  Use template options to declaratively apply CSS modifiers to the templates you have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons.
- About Default Template Options
  Developers can select default template options that are not part of a template option group.
- About Template Option Groups and Presets
  Developers can define the purpose of related template options by creating template option groups.

> **See Also:**
>
> Using Template Options and Using Quick Edit to Modify Live Template Options

11.1.5.1 About Template Options

Use template options to declaratively apply CSS modifiers to the templates you have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons.

A **CSS modifier** is a reference to a CSS class defined in a CSS style sheet. CSS modifiers enable a developer to use the same HTML markup but present it in a variety of different styles simply by applying a CSS. Template options reduce the need to have a large number of nearly identical templates defined in a theme in order to achieve different styles for button colors, region widths, font settings, item label markup, and so on.

Examples of how developers can use template options include:

- Applying different colors or accents
- Applying different spacing and padding
- Rendering buttons in different styles, with and without icons
- Displaying form fields with different alignments.

Without template options, applying these variations would require a large number of nearly identical templates, or in-depth knowledge of CSS. Components that support template options include: pages; regions; classic reports; breadcrumbs; lists; items and labels; and buttons.

11.1.5.2 About Default Template Options

Developers can select default template options that are not part of a template option group.

App Builder automatically applies default template options by rendering the component that references the template. Default template options are not written to an actual component. Instead, a `#DEFAULT#` substitution string is defined for the component, thus enabling developers to centrally modify the template option defaults.

Developers can choose not to use the template option defaults, which removes the `#DEFAULT#` substitution string from the component. If template option defaults are disabled, then developers can selectively apply available template options to a component. All template options that are selected directly when editing the component are written to the component's metadata, represented by their corresponding CSS classes string. At runtime, the selected template options and default template option are part of the default are combined with the component's CSS classes string and applied to the component.

11.1.5.3 About Template Option Groups and Presets

Developers can define the purpose of related template options by creating template option groups.

Examples of template option groups include Button Size, Button Style, List Examples, List Style, and Form Label Position. When developers create a template option group, they can specify a template option preset at the template-level. For example, suppose you have a button template that has the template options *Large* and *Small*. If you define *Small* as the preset, then any new button referencing the template would automatically have the template

option *Small* applied to it when is it created. Template options that are part of a group are optional unless a present is defined at the template-level.

11.2 Managing Themes and Subscriptions

Developers can manage themes directly or through the use of theme subscriptions.

- Managing Theme Subscriptions
  Developers can manage themes using of theme subscriptions.
- Accessing New Themes
  Only themes currently associated with an application display on the Themes page. To access other themes, you run the Create Theme Wizard.
- Editing Themes
  Review or edit a theme by updating theme attributes.
- Switching Themes
  Migrate an application to a new theme by running the Switch Theme Wizard.
- Changing a Theme Identification Number
  Use the Change Theme ID utility to change a theme ID to another identification number.
- Copying a Theme
  Copying a theme is useful when experimenting with editing a theme or to export a theme with a different ID.
- Deleting a Theme
  Delete inactive themes.
- Creating a Master Theme Application
  Create a master theme application.
- Managing Workspace Themes
  Workspace administrators manage the Theme repository for a workspace.
- Managing Instance Themes
  Instance administrators manage the Theme repository for an entire Oracle APEX instance. Only an Instance administrators can add or delete a theme from the repository.
- About Exporting and Importing Themes
  Export a theme in the same way you export any related application files.
- Viewing Theme Reports
  Access the following reports to better manage themes and templates.

11.2.1 Managing Theme Subscriptions

Developers can manage themes using of theme subscriptions.

- About Theme Subscriptions
  Subscribing to a master theme enables the theme to be upgraded during future Oracle APEX releases.
- About Modifying a Theme Subscription
  Modify a template without breaking the subscription to the master theme by creating a local copy of the template in your workspace.
- Verifying if a Theme Refresh Is Needed
  Use the Verify Theme Subscription Wizard to review all theme attributes, templates, and template options in the subscribing theme and view errors for those that are not in sync with the master theme.

- Refreshing a Subscribed Theme
  Refresh a subscribed theme on the Create/Edit Theme page.
- Unsubscribing to a Theme
  Unsubscribe from a theme to remove the subscription from all templates, styles, files and so on that are part of the theme.
- Re-subscribing to a Built-in Theme
  If a theme is currently unsubscribed, you can return to a subscribed version of this theme.

11.2.1.1 About Theme Subscriptions

Subscribing to a master theme enables the theme to be upgraded during future Oracle APEX releases.

Developers subscribe to a theme when:

- **Running the Create Application Wizard**.

  When a developer runs the Create Application Wizard and selects a built-in theme, the theme is automatically subscribed to a master theme. As an alternative to using a built-in theme, the Create Application Wizard provides the option to copy a theme from an existing application. When copying a theme from an existing applications, developers can choose if they wish to have this theme be subscribed to. See Understanding Page Types, Features, and Settings.
- **Creating a new theme from the Theme repository**.

  When you create a new theme the associated application automatically subscribes to it. See Accessing New Themes.

Once an application subscribes to a master theme only the default templates can be changed. All theme attributes, subscribed template options, and subscribed templates are set to read-only. In other words, only the default templates can be changed. The advantage of subscribing to a master theme is that the theme can be upgraded during future APEX releases. When a theme is subscribed to, developers cannot modify the theme or template metadata when they access it from the theme or template edit pages.

11.2.1.2 About Modifying a Theme Subscription

Modify a template without breaking the subscription to the master theme by creating a local copy of the template in your workspace.

Local template copies are not subscribed to and are therefore editable. You can take the same approach with theme styles. You can create a local copy of a theme style and then add it to a subscribed theme. Theme styles created locally are editable but will be excluded from theme refreshes.

> **See Also:**
>
> Copying a Theme

11.2.1.3 Verifying if a Theme Refresh Is Needed

Use the Verify Theme Subscription Wizard to review all theme attributes, templates, and template options in the subscribing theme and view errors for those that are not in sync with the master theme.

This wizard also highlights any templates that are defined in the master theme but not yet published to the subscribing theme.

To run the Verify Theme Subscription Wizard:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. Select the theme.

   The Create/Edit Theme page appears.
3. Locate Theme Subscription and click **Verify**.
4. Follow the on-screen instructions.

   To publish changes from the master theme to the subscribing theme, click **Refresh** at the end of the wizard.

11.2.1.4 Refreshing a Subscribed Theme

Refresh a subscribed theme on the Create/Edit Theme page.

To refresh a subscribed theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. Select the theme.

   The Create/Edit Theme page appears.
3. Locate Theme Subscription and click **Refresh Theme**.

> **Tip:**
>
> If the theme is not subscribed, the Refresh Theme button does not appear. See Re-subscribing to a Built-in Theme.

4. Click **Apply Changes**.

11.2.1.5 Unsubscribing to a Theme

Unsubscribe from a theme to remove the subscription from all templates, styles, files and so on that are part of the theme.

> **Note:**
>
> Unsubscribing from a theme prevents future theme updates. Be aware that re-subscribing to a theme is multiple step process that involves copying and switching themes. To learn more, see Re-subscribing to a Built-in Theme.

To unsubscribe from a theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. Select the theme.

   The Create/Edit Theme page appears.
3. Under Theme Subscription, click **Unsubscribe**.

   A warning dialog displays.
4. Click **Unsubscribe Theme**.

11.2.1.6 Re-subscribing to a Built-in Theme

If a theme is currently unsubscribed, you can return to a subscribed version of this theme.

Re-subscribing to a theme is a multiple step process. First, you change the theme identification number. Second, you run the Create Theme Wizard and create the theme again. Third, you switch the current application to the use the new theme. Fourth, you delete the original theme.

To re-subscribe to a theme.

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. From the Tasks list, select **Change Identification Number** and change the theme to a new identification number. See Changing a Theme Identification Number.

3. Run the Create Theme Wizard and create the theme again. See Accessing New Themes.
4. Switch the current application to the use the new theme. See Switching Themes.
5. Delete the original theme from which you created a new identification number. See Deleting a Theme.

11.2.2 Accessing New Themes

Only themes currently associated with an application display on the Themes page. To access other themes, you run the Create Theme Wizard.

Once you create a theme, you can access it on the Themes page. You can create a theme from scratch or select an existing theme from the repository.

> **Tip:**
>
> When you create a new theme, the associated application automatically subscribes to it. See Managing Themes and Subscriptions.

To access a new theme by running the Create Theme Wizard:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. Click **Create**.
3. Select a creation method. Options include:
   - **From the Repository** - View and select a theme from the repository.
   - **As a copy from another application** - Select the application and then the theme.
   - **From Scratch** - Create a theme from scratch.
   - **From Export** - Create a theme to import from the export repository.
4. Follow the on-screen instructions.

   To learn more about an attribute, see field-level Help.

> **See Also:**
>
> Editing Themes

11.2.3 Editing Themes

Review or edit a theme by updating theme attributes.

- Editing a Theme
  Review or edit a theme by selecting the theme name on Themes page in Shared Components.
- Theme Attributes
  Learn about Theme attributes available on the Create/Edit Theme Attributes page.

> **See Also:**
>
> Creating Custom Themes

11.2.3.1 Editing a Theme

Review or edit a theme by selecting the theme name on Themes page in Shared Components.

To review or edit a theme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Themes**.

   The Themes page appears.
5. On the Themes page, click the **View Report** icon.

   The Themes page appears as a report. A check mark in the Is Current column indicates which theme is selected.
6. Click the theme name.

   The Create/Edit Theme page appears and is divided into sections.
7. Edit the appropriate Theme attributes as described in Theme Attributes.
8. To save any changes, click **Apply Changes**.

11.2.3.2 Theme Attributes

Learn about Theme attributes available on the Create/Edit Theme Attributes page.

To learn more about an attribute and view examples, see field-level Help.

- Name
- Theme Subscription
- JavaScript and Cascading Style Sheets
- Component Defaults
- Region Defaults
- Dialog Defaults

- Global Templates Options
- Icons
- Styles
- Files

11.2.3.2.1 Name

Table 11-1 describes the attributes under Name.

Table 11-1 Name

Attribute	Description
Application	Identifies the current application ID. **See Also**: Editing the Application Definition
Theme Number	Identifies the number of a theme. A theme is a collection of templates which define the user interface of an application.
Name	Enter a short descriptive name for the theme.
Identifier	Specifies the internal theme identifier.
Navigation Type	Identifies the selected Navigation Type. Valid options include: Tabs or List. **See Also**: Managing Tabs and Creating Lists
Navigation Bar Implementation	Identifies the selected Navigation Bar Implementation. Valid options include: Classic or List. **See Also**: Navigation Bar
Description	Enter a description for the theme.

11.2.3.2.2 Theme Subscription

Displays themes to which this application subscribes. When an application subscribes to a theme, all theme attributes, subscribed template options, and subscribed templates are set to read-only.

> **See Also:**
>
> Managing Theme Subscriptions

11.2.3.2.3 JavaScript and Cascading Style Sheets

Table 11-2 describes the attributes under JavaScript and Cascading Style Sheets.

Table 11-2 JavaScript and Cascading Style Sheets

Attribute	Description
JavaScript File URLs	Enter JavaScript file URLs for code to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file, you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode.JavaScript file URLs you enter here replaces the `#THEME_JAVASCRIPT#` substitution string in the page template To view examples, see field-level Help.
CSS File URLs	Enter Cascading Style Sheet file URLs to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.File URLs you enter here will replace the `#THEME_CSS#` substitution string in the page template. To view examples, see field-level Help.

11.2.3.2.4 Component Defaults

Table 11-3 describes the default templates by component type.

Table 11-3 Component Default Templates

Attribute	Description
Page	Identifies the default template for displaying pages. If a developer does not explicitly choose a template, then the Oracle APEX engine uses the template specified here.
Navigation Bar List	Specifies the default navigation bar list template used when you define the navigation bar type as List in your application. If a developer does not explicitly choose a template, then the APEX engine uses the template specified here.
Navigation Menu List Position	Specifies the default navigation menu list position.
Navigation Menu List (Top)	Specifies the default navigation menu list template used when you create a new list to be displayed on the top of the page.
Navigation Menu List (Side)	Specifies the default navigation menu list template used when you create a new list to be displayed on the side of the page.
Login Page	Specifies the default template for displaying pages. If a developer does not explicitly choose a template then the APEX engine uses the template specified here.
Error Page	Optional. Specifies a page template to use for errors that display on a separate page as opposed to those that display inline. Leave this attribute blank if you do not wish to use a template designed to display errors. This setting only applies to Normal pages.

Table 11-3 (Cont.) Component Default Templates

Attribute	Description
Printer Friendly Page	Identifies the template to be used when the APEX engine is in printer friendly mode. When calling the APEX engine to render a page, you have the option to identify a printer friendly attribute with values of **Yes** or **No**. If you select **Yes**, then the page displays using a printer friendly template. The APEX engine displays all text within HTML form fields as text. The printer friendly template does not need to have the `#FORM_OPEN#` or `#FORM_CLOSE#` tags. The objective is to be able to display information with few tables and in a format suitable for printing.
Breadcrumb	Identifies the default breadcrumb template used when you create a breadcrumb.
Button	Identifies the default button template used when you create a button.
Legacy Calendar	Specifies the default calendar template used when you create a calendar.
Default Label	Specifies the default label template used when you create new label
Optional Label	Specifies the default optional label template used when you create new label.
Required Label	Specifies the default required label template used when you create new label.
List	Specifies the default list template used when you create a list.
Region	Specifies the default region template used when you create a region.
Classic Report	Identifies the default region template used when you create a report.
Header Toolbar	Identifies the default header toolbar region template used when you create a new page that includes a header or footer tool bars.
Footer Toolbar	Identifies the default header toolbar region template used when you create a new page that includes a header or footer tool bars.

11.2.3.2.5 Region Defaults

Table 11-4 describes the default templates available under Region Defaults.

Table 11-4 Region Defaults

Attribute	Description
Breadcrumb	Identifies the default breadcrumb template used when you create a breadcrumb.
Charts	Specifies the default chart template used when you create a chart.
Forms	Specifies the default form template used when you create a form.
Lists	Specifies the default region template used when you create a list.
Reports	Specifies the default region template used when you create a report.
Wizards	Specifies the default region template used when you create a new wizard component.
Interactive Reports	Specifies the default region template used when you create an interactive report

11.2.3.2.6 Dialog Defaults

Table 11-5 describes the default templates available under Dialog Defaults.

Table 11-5 Dialog Defaults

Attribute	Description
Dialog Content Region	Specifies the default region template used when you create a content region on a Dialog using the Create Page Wizard and Create Application Wizard.
Dialog Button Region	Specifies the default region template used when you create a button region on a Dialog using the Create Page Wizard and Create Application Wizard.
Dialog Page	Specifies the default template for displaying modal or non-modal dialog pages. If a developer does not explicitly choose a template then the APEX engine uses the template specified here. This template will also be used for errors that display on a separate page as opposed to those that display inline.

11.2.3.2.7 Global Templates Options

Global template options are defined at the theme-level and are available for all components of a given type.

> **See Also:**
>
> Understanding Template Options and Managing Global Template Options

11.2.3.2.8 Icons

Table 11-6 describes attributes under Icons.

Table 11-6 Icons

Attribute	Description
Library	Select the icon library that is loaded when APEX displays a page. The icons within the library are also be listed in the picker for Region and Button Icon CSS Classes, implemented within Page Designer.
Custom Library File URLs	Enter Cascading Style Sheet file URLs for custom icons to be loaded on every page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.File URLs you enter here will replace the `#THEME_CSS#` substitution string in the page template. To view examples, see field-level Help.
Custom Classes	Specify a comma delimited list of CSS class names which is listed in the picker for Region and Button Icon CSS Classes, implemented within Page Designer.
Custom Prefix Class	Specify a CSS class that will at runtime prefix the Icon CSS Classes defined for Regions, Buttons and Lists, if they are not using a class of the Icon Library.
Date Picker Icon Name	Enter the name and location of the image to be used when items of type Date Picker are displayed. For example: `#APP_FILES#new_cal.bmp`

Table 11-6 (Cont.) Icons

Attribute	Description
Date Picker Icon Attributes	Enter the image attributes for the Calendar Icon. For example: `width="16" height="16" alt="Display Calendar"`

11.2.3.2.9 Styles

Displays theme styles available to the current theme. Theme Styles define the CSS for your application's theme and control the user interface of your application. Theme styles are referenced in the page template using the `#THEME_STYLE_CSS#` substitution string. The **Current Theme Style** list displays the current Theme Style.

> **See Also:**
>
> Using Theme Styles

11.2.3.2.10 Files

The **File Prefix** attribute displays the virtual path the Web server uses to point to the files of the theme. Do not specify anything to reference files which are stored with your theme definition in the database. For performance reasons you can also store your theme files on your Web Server. Use `#APEX_FILES#`or any valid URL to reference them.

Do not specify anything to reference files which are stored with your theme definition in the database. If you specify File Prefix and still wish to reference files stored with your theme in the database, use `#THEME_DB_FILES#` rather than `#THEME_FILES#` in your templates.

Examples:

- `/myFiles/`
- `https://example.com/myFiles/`
- `#APEX_FILES#myFiles/`

To upload a file:

Click **Create File**. On Create File:

1. Click **Create File**.
2. On Create File:
   a. Directory - Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`. If no directory is specified, the file is stored in the root directory.
   b. File Name - If creating a blank file (for example, `script.js`), enter the name of the file.
   c. Content - If uploading a file, do one of the following:
      - Drag and drop the file to the Content region.
      - Click the Content region and select the file.

d. Click **Create** or **Create and Create Another**.

> **See Also:**
>
> Managing Static Application Files and Managing Static Workspace Files

11.2.4 Switching Themes

Migrate an application to a new theme by running the Switch Theme Wizard.

- About Switching the Active Theme
  Switching to another theme, assigns all components with assigned templates to corresponding templates in the new theme.
- Switching an Active Theme
  Run the Switch Theme Wizard to migrate an application to a new theme.

11.2.4.1 About Switching the Active Theme

Switching to another theme, assigns all components with assigned templates to corresponding templates in the new theme.

App Builder accomplishes template mapping through the assignment of template class identifiers.

When switching an active theme, remember:

- You can only switch to a theme if the theme exists. For example, before you can switch to a theme available in the repository, you must first create it. See Accessing New Themes.
- If the wizard locates multiple matches (that is, when several templates share the same template type and templates class), the wizard defaults to a template matching type, class, and name. If no match is found, you must select another template with the same type and class.
- If no matching template is found (that is, no template with the same type and class exists) then you are prompted to select a template with the same type but with a different class. Then, a warning displays explaining this issue might lead to problems with page functionality, or issues with the page display. When no template of the same template type exists, then the theme cannot be switched. You are blocked from switching themes.

11.2.4.2 Switching an Active Theme

Run the Switch Theme Wizard to migrate an application to a new theme.

To complete this procedure, you must have at least two themes installed. If you do not have at least two themes installed, on the Themes page click **Create Theme**. You can create a theme from scratch or simply select an existing theme from the repository. See Accessing New Themes.

> **Tip:**
>
> Before starting this procedure, Oracle recommends creating back up your application by either exporting the application or creating a copy.

To switch an active theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears. A check mark in the Is Current column indicates which theme is selected.
2. Click **Switch Theme**.

   The Switch Theme page appears.
3. For Identify Theme:
   a. Currently Active Theme - Select the current theme for the application. A theme identifies a collection of templates which define the look and feel of the application.
   b. Switch to Theme - Select the new theme for the application.
   c. Click **Next**.

   The Verify Compatibility page appears.
4. Review the Status column to identify problematic mappings:
   - A **check** indicates the mapping was successful.
   - A **warning** indicates there are more than one template in the theme you are switching to with the identified class. The warning provides a select list from which to choose the appropriate template.
   - An **error** indicates that App Builder was unable to map the class among the themes. Ensure that a class is identified for the templates in both themes.

   > **Note:**
   >
   > Failure to resolve warnings may result in a loss of functionality and applications that do not display well.
5. Click **Next** to continue.
6. Click **Switch Theme**.

See Also:

- Refreshing the Universal Theme
- Exporting an Application
- Copying an Application

11.2.5 Changing a Theme Identification Number

Use the Change Theme ID utility to change a theme ID to another identification number.

Each theme has an identification number (ID). Changing a theme ID is useful when you want to export a theme with a different number and then import it into another application.

To change a theme identification number:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. On the Tasks list, click **Change Identification Number**.
3. For Theme Number:
   a. Identify Theme - Select a theme.
   b. New Theme Identification Number - Specify an identification number.
   c. Click **Next**.
4. Confirm your changes and click **Change Theme ID**.

11.2.6 Copying a Theme

Copying a theme is useful when experimenting with editing a theme or to export a theme with a different ID.

Each theme is identified by a numeric identification number (ID). When you copy a theme, you specify a theme ID.

To copy a theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. On the Tasks list, click **Copy Theme**.

3. For Theme Number:
   a. Application - Identifies the current application.
   b. Copy From Theme - Select the theme you want to copy.
   c. Copy to this Theme ID - Enter a new ID for the theme.
   d. Subscribe Theme - Select **On** or **Off**.
   e. Click **Next**.
4. Confirm your changes and click **Copy Theme**.

> **See Also:**
>
> Managing Theme Subscriptions

11.2.7 Deleting a Theme

Delete inactive themes.

When you delete a theme, App Builder only removes inactive templates.

> **Tip:**
>
> Themes are tied to an application. If you modify a theme, those changes only apply to the application being edited. If you delete a theme, any template modifications are lost. To keep template modifications, you must export the theme. See Exporting Themes.

To delete an active theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. On the Tasks list, click **Delete Theme**.
3. From Delete Theme, select the theme you want to delete and click **Next**.
4. Click **Delete Theme**.

11.2.8 Creating a Master Theme Application

Create a master theme application.

Oracle recommends starting with an existing Sample App or Starter App that uses the *Universal Theme - 42*.

- Creating a Master Theme Application by Creating a New Application
  Create a master theme application by creating a new application.
- Creating a Master Theme Application from an Existing Application
  Create a master theme application by copying an existing theme and subscribing to it and then switching to the new theme.

> **See Also:**
>
> Using the App Gallery

11.2.8.1 Creating a Master Theme Application by Creating a New Application

Create a master theme application by creating a new application.

To create a master theme application:

1. On the Workspace home page, click the **App Builder** icon.
2. Click the **Create** button.

   The Create Application Wizard appears.
3. Click **Copy Existing App**.
4. For Identify Copy Target:
   a. Copy From - Select an application that uses *Universal Theme - 42*.
   b. Copy To - Enter a short descriptive name for the application to distinguish the new application in select lists and reports.
   c. New Application ID - Specify an integer value that identifies the application. The application ID must be unique within your Oracle APEX instance. Application IDs in the range of 3000 to 9000 are reserved for internal use by APEX
   d. Copy Supporting Object Definitions - You can copy supporting object definitions, configuration options, and scripts with your application. Select **Yes** to include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file. Select **No** to not include supporting object definitions.
   e. Click **Next**.
5. Confirm your selection and click **Create Application**.

11.2.8.2 Creating a Master Theme Application from an Existing Application

Create a master theme application by copying an existing theme and subscribing to it and then switching to the new theme.

To create a master theme application from an existing application:

1. Copy the theme from an application that uses Universal Theme - 42 (for example, *Universal Theme Reference*) and subscribe to it. See Copying a Theme.
2. Switch to the new theme. See Switching Themes.
3. Follow the on-screen instructions.

11.2.9 Managing Workspace Themes

Workspace administrators manage the Theme repository for a workspace.

- Adding a Workspace Theme to the Theme Repository
  Workspace administrators can add a theme to the Theme repository and make it available to all developers within a workspace.
- Viewing an Existing Workspace Theme
  View existing Workspace themes on the Workspace Themes page.
- Deleting a Workspace Theme
  Workspace administrators can delete workspace themes from the Theme repository .
- Modifying a Workspace Theme
  Workspace administrators can modify a workspace theme by creating an application using the theme, modifying it, and then manually adding it to the workspace Theme repository
- Exporting a Workspace Theme
  Export a theme in the same way you export any related application files.

> **See Also:**
>
> Managing Public Themes in *Oracle APEX Administration Guide*

11.2.9.1 Adding a Workspace Theme to the Theme Repository

Workspace administrators can add a theme to the Theme repository and make it available to all developers within a workspace.

To add a theme to the Theme repository:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. On the Tasks list, click **Manage Workspace Themes**.
3. Select **Create Workspace Theme** and click **Next**.

   > **Tip:**
   >
   > If no workspace themes exist, the wizard selects **Create** for you.
4. For Create Workspace Theme:
   a. Application - Select the application.
   b. Application Theme to Copy - Select the theme to copy to the workspace theme repository.

c. Subscribe Theme - Select whether to subscribe the workspace theme to the selected application theme. This option enables you to refresh the workspace theme with changes made to the selected application theme.

d. Click **Next**.

5.
   a. Theme Number - Enter a number with a value greater than 100. This number uniquely identifies this theme within a workspace.

   b. Theme Name - Enter a theme name. This name can be the same or different from the application theme name.

   c. Description - Enter a description of the theme.

   d. Click **Next**.

6. Click **Create Workspace Theme**.

11.2.9.2 Viewing an Existing Workspace Theme

View existing Workspace themes on the Workspace Themes page.

To view an existing workspace theme:

1. Navigate to the Themes page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Click **Shared Components**.

   d. Under User Interface, select **Themes**.

   The Themes page appears.

2. From the Tasks List, click **Manage Workspace Themes**.

   The Workspace Themes page appears.

> **Tip:**
>
> You cannot edit a workspace theme directly. To modify a workspace theme, you must create an application using the theme, modify it, and then manually add it to the workspace theme repository. See Modifying a Workspace Theme.

11.2.9.3 Deleting a Workspace Theme

Workspace administrators can delete workspace themes from the Theme repository .

To delete a workspace theme:

1. Navigate to the Themes page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Click **Shared Components**.

   d. Under User Interface, select **Themes**.

   The Themes page appears.

2. From the Tasks List, click **Manage Workspace Themes**.

   The Workspace Themes page appears.
3. Select the theme name.
4. Click **Delete**.
5. Confirm your delete request and click **OK**.

11.2.9.4 Modifying a Workspace Theme

Workspace administrators can modify a workspace theme by creating an application using the theme, modifying it, and then manually adding it to the workspace Theme repository

You cannot edit a workspace theme directly. To modify a workspace theme, you must create an application using the theme, modify it, and then manually add it to the Workspace Theme repository.

To modify a workspace theme:

1. Create an application using the theme you want to modify. See Creating Applications.
2. Modify the theme. See Editing Themes.
3. Delete the existing workspace theme. See Deleting a Workspace Theme.
4. Add the modified theme to the Theme repository. See Adding a Workspace Theme to the Theme Repository.

11.2.9.5 Exporting a Workspace Theme

Export a theme in the same way you export any related application files.

Exporting a workspace theme involves the following steps:

1. Create an application using the theme. See Creating Applications.
2. Export the application. See Exporting an Application.
3. Import the Export File into the target Oracle APEX instance. See Importing an Application Export.
4. If you did not install your Export File after importing, you can install it in the Export Repository. See Managing Export Files from the Export Repository.

11.2.10 Managing Instance Themes

Instance administrators manage the Theme repository for an entire Oracle APEX instance. Only an Instance administrators can add or delete a theme from the repository.

> **See Also:**
>
> Managing Public Themes in *Oracle APEX Administration Guide*

11.2.11 About Exporting and Importing Themes

Export a theme in the same way you export any related application files.

Exporting a theme from one development instance to another involves the following steps:

1. Create an application using the theme. See Creating Applications.
2. Export the theme. See Exporting Themes.
3. Import the exported file into the target APEX instance. See Importing Export Files.
4. Install the exported file from the Export Repository. See Managing Export Files from the Export Repository.

11.2.12 Viewing Theme Reports

Access the following reports to better manage themes and templates.

- Viewing All Templates in a Theme
  View all templates that comprise a theme.
- Viewing Theme Template Counts
  The Theme Template Count report lists which template classes currently have associated templates.
- Viewing File References
  The File References report displays a list of all files associated with templates, shared components, or page components in the current application.
- Viewing Class References
  The Class References report displays a list of classes associated with templates, shared components, or page components in the current application.
- Viewing Template Substitution Strings
  Use the Template Substitution Strings report to view all supported substitution strings by component.

11.2.12.1 Viewing All Templates in a Theme

View all templates that comprise a theme.

To view all templates that comprise a theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
   a. Report - Select **Application Templates**.
   b. Theme - Select either **42. Universal Theme** or **All Themes**.

c. Click **Go**.

A list of templates appears with the template type, template name, theme, and template class.

4. To edit a template, select the template name.

11.2.12.2 Viewing Theme Template Counts

The Theme Template Count report lists which template classes currently have associated templates.

To view the Theme Template Count report:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
   a. Report - Select **Theme Template Counts**.
   b. Theme - Select a theme or select **All**.
   c. Click **Go**.

11.2.12.3 Viewing File References

The File References report displays a list of all files associated with templates, shared components, or page components in the current application.

To view the File References report:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
   a. Report - Select **File References**.
   b. Theme - Select a theme or select **All**.
   c. Click **Go**.
4. On the File References page:
   a. Show - Select the type of component to include in the report. If you do not make a selection, no results are returned.

b. Show Files - Select one of the following:
   - **With context** displays the component, the theme identification number, the component name, the image (if applicable), and the page number. Select the page number.
   - **Without context** displays only the file name and the image (if applicable).

c. File Extensions - Select the type of extensions for which to search.

d. Click **Go**.

5. To download a comma-delimited file (.csv) version of this report, click **Download** at the bottom of the page.

11.2.12.4 Viewing Class References

The Class References report displays a list of classes associated with templates, shared components, or page components in the current application.

To view the Class References report:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
   a. Report - Select **Class References**.
   b. Theme - Select a theme or select **All**.
   c. Click **Go**.
4. On the Class References page:
   a. Show - Select the components to check for a class reference. If you do not make a selection, no results are returned.
   b. Show Class Names - Select one of the following:
      - **With context** displays the component, the theme identification number, the component name, the image (if applicable), and the page number.
      - **Without context** displays only the referenced class.
   c. Click **Go**.
5. To download a comma-delimited file (.csv) version of this report, click **Download** at the bottom of the page.

11.2.12.5 Viewing Template Substitution Strings

Use the Template Substitution Strings report to view all supported substitution strings by component.

To view the Substitution String report:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Click **Reports**.
3. On the Theme Reports page:
   a. Report - Select **Template Substitution Strings**.
   b. Theme - Select which themes to include in the report.
   c. Click **Go**.
4. To link to a template definition, select the component name.

> **See Also:**
>
> Using Substitution Strings

11.3 Creating Custom Themes

Create a custom theme by modifying existing templates.

The Oracle APEX engine creates an application user interface based on a named collection of templates called a **theme**. Templates control the look and feel of the components in an application. If you must create a custom template, it is generally easier to start with an existing template and then modify it. Once you have created one or more default templates, you can modify those templates to fit your specific needs.

- About Cascading Style Sheets
  A Cascading Style Sheet (CSS) provides a way to control the style of a web page without changing its structure.
- About Calling the JavaScript File from the Page Template
  Reference a `.js` file in the page template.
- About Using Escaping Syntax in Substitution Strings
  Use escaping syntax in substitution strings.
- Selecting a Default Page Template
  Select a default page template by either selecting a default page template within a specific theme, or by selecting a specific page template on a page-by-page basis.
- Creating a New Template
  To create a custom template, copy and then modify and existing template.
- Using Template Options
  Template options provide a declarative way for developers to apply different styles to components on an Oracle APEX page.

- Viewing Template Reports
  View reports describing template utilization, subscriptions, published templates, and edit history.
- Managing Templates
  Manage templates by navigating to the Templates page in Shared Components.
- Breadcrumb Templates
  Breadcrumb templates control the display of breadcrumb entries.
- Button Templates
  Customize the look and feel of a button using button templates. Button templates are optional.
- Legacy Calendar Templates
  Control the appearance and placement of a legacy calendar using legacy calendar templates.
- Label Templates
  Centrally manage HTML markup of page item labels using label templates.
- List Templates
  Control the appearance of a list using list templates.
- Page Templates
  Define the appearance of a page using page templates.
- Popup LOV Templates
  Use Popup LOV templates control how popup lists display for all items defined as `POPUP`.
- Template Component Type Plug-ins
  Learn about Template Component Type plug-ins.
- Region Templates
  Region templates control the appearance and placement of region attributes.
- Report Templates
  Report column templates provide you with control over the results of a row from a SQL query. This type of template defines a cell, not an entire row.

See Also:

- Using Themes
- Managing Themes and Subscriptions
- Managing Theme Subscriptions

11.3.1 About Cascading Style Sheets

A Cascading Style Sheet (CSS) provides a way to control the style of a web page without changing its structure.

When used properly, a CSS separates visual attributes such as color, margins, and fonts from the structure of the HTML document. APEX includes themes that contain templates that reference their own CSS. The style rules defined in each CSS for a particular theme also determine the way reports and regions display.

When using built-in themes, you can find the theme specific CSS files in the following locations:

```
/i/themes/theme_xx
/i/themes/theme_xx/css
```

Theme specific image can be found in the following locations:

```
/i/themes/theme_xx
/i/themes/theme_xx/images
```

Where *xx* is the theme number. Theme specific CSS files include the APEX version number to preserve backward compatibility for imported applications using older versions of a theme.

> **See Also:**
>
> Using Custom Cascading Style Sheets

11.3.2 About Calling the JavaScript File from the Page Template

Reference a `.js` file in the page template.

Referencing a `.js` file in the page template makes all the JavaScript in that file accessible to the application. This is the most efficient approach since a `.js` file loads on the first page view of your application and is then cached by the browser.

> **See Also:**
>
> JavaScript

11.3.3 About Using Escaping Syntax in Substitution Strings

Use escaping syntax in substitution strings.

Developers can append an exclamation mark (`!`) followed by a predefined filter name to substitution strings to escape special characters in the substitution value. Output escaping is an important security technique to avoid Cross Site Scripting (XSS) attacks in the browser.

> **See Also:**
>
> Controlling Output Escaping in Substitution Strings

11.3.4 Selecting a Default Page Template

Select a default page template by either selecting a default page template within a specific theme, or by selecting a specific page template on a page-by-page basis.

By default, the Oracle APEX engine uses the Page template specified on the Themes page.

- Selecting a Page-level Template Within a Theme
  Select a default page template within a theme in Shared Components.
- Selecting a Page-level Template for a Specific Page
  Select a page-level template for a specific page in Page Designer.

11.3.4.1 Selecting a Page-level Template Within a Theme

Select a default page template within a theme in Shared Components.

To specify a default page template within a theme:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Select a theme name.

   The Create/Edit Theme page appears.
3. Scroll down to Component Defaults.
4. From Page, make a new selection.
5. Click **Apply Changes** at the top of the page.

> **See Also:**
>
> Editing Themes

11.3.4.2 Selecting a Page-level Template for a Specific Page

Select a page-level template for a specific page in Page Designer.

To specify a page-level template for a specific page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Select the application page.

   Page Designer displays the page.
4. To edit the page-level template:
   a. In the Rendering tab, click the page title.
   b. In the Property Editor, locate the **Appearance**. From Page Template, select a new template.
   c. Click **Save**.

11.3.5 Creating a New Template

To create a custom template, copy and then modify and existing template.

Once you have created one or more default templates, you can modify those templates to fit your specific needs.

To create a custom template:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application to which you want to copy the template.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.
5. Click **Create**.
6. Select the type of template you want to create and click **Next**.
7. Select a creation method:
   - **From Scratch**
   - **As a Copy of an Existing Template**
8. Follow the on-screen instructions.

> **Tip:**
>
> Make sure you associate your template with the correct theme.

11.3.6 Using Template Options

Template options provide a declarative way for developers to apply different styles to components on an Oracle APEX page.

- Managing Global Template Options
  Gobal template options are defined at the theme-level.
- Managing Template Options
  Template options enable developers to declaratively apply CSS modifiers to the templates they have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons.

> **See Also:**
>
> Understanding Template Options and About Template Option Groups and Presets

11.3.6.1 Managing Global Template Options

Gobal template options are defined at the theme-level.

> **Tip:**
>
> In order to add or edit template options, you must be the theme owner. See Unsubscribing to a Theme.

- Creating Global Template Options
  Define global template options on the Themes page under Template Options.
- Editing Global Template Options
  Edit existing global template options on the Themes page under Template Options.
- Managing Global Template Option Groups
  Define the purpose of related template options by creating template option groups. Examples of template option groups include Button Size, Button Style, List Examples, List Style, and Form Label Position.

> **See Also:**
>
> About Template Options

11.3.6.1.1 Creating Global Template Options

Define global template options on the Themes page under Template Options.

To create a global template option:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Select a theme.
3. Locate **Template Options** and click **Add Template Option**.

   A new row appears.
4. Edit the attributes:
   a. Template Type - Select the type of template for which the template option is defined.
   b. Group - Select a group name. Group names enable you to group template options and avoid the selection of conflicting option. See Managing Global Template Option Groups.
   c. Name - Enter a name for the template option.
   d. Identifier - Enter an alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
   e. Sequence - Specify the sequence for this option.

f. Classes - Enter one or more CSS classes that define this template option.

g. Help Text - Enter a description of the template option.

5. Click **Save**.

11.3.6.1.2 Editing Global Template Options

Edit existing global template options on the Themes page under Template Options.

To edit a global template option:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Select a theme.
3. Locate **Template Options**.
4. To edit a template option:
   a. Select the row containing the template option.

      The row becomes hightlighted.
   b. Edit the appropriate attributes.
      - Template Type - Select the type of template for which the template option is defined.
      - Group - Select a group name. Group names enable you to group template options and avoid the selection of conflicting option. See Managing Global Template Option Groups.
      - Name - The name for the template option.
      - Identifier - An alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
      - Sequence - Specify the sequence for this option.
      - Classes - Enter one or more CSS classes that define this template option.
      - Help Text - Enter a description of the template option.

      > **WARNING:**
      >
      > If you edit the name of a CSS class, all components that reference that CSS class name must be updated manually. The Option Identifier cannot be modified since it is needed for mapping purposes.

   c. Click **Save**.
5. To delete a template option:
   - Click the **Row Actions** menu and select **Delete Row**.

> **Tip:**
>
> To learn more about the Row Actions menu, see About the Row Actions Menu in an Interactive Grid in *Oracle APEX End User's Guide*

11.3.6.1.3 Managing Global Template Option Groups

Define the purpose of related template options by creating template option groups. Examples of template option groups include Button Size, Button Style, List Examples, List Style, and Form Label Position.

> **Tip:**
>
> In order to add or edit template option groups, you must be the theme owner. See Unsubscribing to a Theme.

- Creating a Global Template Option Group
  Global template option groups are defined at the theme-level and are available for all components of a given type.
- Editing Global Template Option Groups
  Edit global template option groups by navigating to the Themes page.

> **See Also:**
>
> About Template Option Groups and Presets and Understanding Template Options

11.3.6.1.3.1 Creating a Global Template Option Group

Global template option groups are defined at the theme-level and are available for all components of a given type.

When developers create a template option group, they can specify a template option preset at the template-level. For example, suppose you have a button template that has the template options *Large* and *Small*. If you define *Small* as the preset, then any new button referencing the template would automatically have the template option Small applied to it when is it created. Template options that are part of a group are optional unless defined at the template-level.

To create a global template option group:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Select a theme.

3. Locate **Template Options** and click **Template Options Groups**..

   The Template Options Groups window appears.

4. To create a new group:

   a. Click **Add Template Option Group**.

      A new row appears.

   b. Edit the appropriate attributes:

      - Template Type - Specify the type of template for this template option group.
      - Name - Enter a display name for the template option group.
      - Identifier - Defines the internal name of the template option group.
      - Sequence - Specify the sequence for this template option group.
      - Null Text - Default value for the template option group if a user does not select any template option in that group.
      - Help Text - Enter a description of the template option group.
      - Advanced - Determines if the template option group belongs in the Basic or Advanced category of template options. Select **Yes** or **No**.

   c. Click **Save**.

11.3.6.1.3.2 Editing Global Template Option Groups

Edit global template option groups by navigating to the Themes page.

To edit global template option groups:

1. Navigate to the Themes page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Click **Shared Components**.

   d. Under User Interface, select **Themes**.

   The Themes page appears.

2. Select a theme.

3. Locate **Template Options**.

4. Click **Add Template Option Groups**.

   Template Options Groups dialog appears.

5. To edit a template option group:

   a. Select the row containing the template option group.

      The row becomes hightlighted.

   b. Edit the appropriate attributes.

      - Template Type - Specifies the type of template for this template option group.
      - Name - Dsplay name for the template option group.
      - Identifier - Defines the internal name of the template option group.
      - Sequence - Specify the sequence for this template option group.

- Null Text - Default value for the template option group if a user does not select any template option in that group.
- Help Text - Enter a description of the template option group.
- Advanced - Determines if the template option group belongs in the Basic or Advanced category of template options. Select **Yes** or **No**.

c. Click **Save**.

6. To delete a template option group:
   - Click the **Row Actions** menu and select **Delete Row**.

> **Tip:**
>
> To learn more about the Row Actions menu, see About the Row Actions Menu in an Interactive Grid in *Oracle APEX End User's Guide*

11.3.6.2 Managing Template Options

Template options enable developers to declaratively apply CSS modifiers to the templates they have chosen for pages, regions, reports, lists, breadcrumbs, items and buttons.

> **Tip:**
>
> In order to add or edit template options, you must be the theme owner. See Unsubscribing to a Theme.

- Creating Template Options
  Create a template option by declaratively modifying a template.
- Editing Template Options
  Edit an existing template option on the Templates page.
- Creating a Template Option Group
  Define the purpose of related template options by creating template option groups.
- Editing Template Option Groups
  Edit template option groups by navigating to the Templates page.
- Using Quick Edit to Modify Live Template Options
  Using the Live Template Options dialog to alter a component's template options without refreshing the page.

> **See Also:**
>
> Understanding Template Options

11.3.6.2.1 Creating Template Options

Create a template option by declaratively modifying a template.

To create a template option:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. Select a template.
3. Under **Template Options**, click **Add Template Option**.

   A new row appears.
4. Edit the appropriate attributes:
   a. Group - Select a group name. Group names enable you to group template options and avoid the selection of conflicting option. See .
   b. Name - Enter a name for the template option.
   c. Identifier - Enter an alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
   d. Sequence - Specify the sequence for this option.
   e. Classes - Enter one or more CSS classes that define this template option.
   f. Help Text - Enter a description of the template option.
5. Click **Save**.

11.3.6.2.2 Editing Template Options

Edit an existing template option on the Templates page.

To edit a template option:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. Select a template.
3. Select **Template Options**.
4. To edit a template option:
   a. Select the row containing the template option.

      The row becomes hightlighted.
   b. Edit the appropriate attributes.
      - Group - Select a group name. Group names enable you to group template options and avoid the selection of conflicting option.
      - Name - Enter a name for the template option.

- Identifier - Enter an alphanumeric, upper case identifier for the template option. Identifiers map a template option to a corresponding template option when switching themes.
- Sequence - Specify the sequence for this option.
- Classes - Enter one or more CSS classes that define this template option.
- Help Text - Enter a description of the template option.

> **Tip:**
>
> If you edit the name of a CSS class, all components that reference that CSS class name must be updated manually. The Option Identifier cannot be modified since it is needed for mapping purposes.

5. Click **Save**.
6. To delete a template option:
   - Click the **Row Actions** menu and select **Delete Row**.

> **Tip:**
>
> To learn more about the Row Actions menu, see About the Row Actions Menu in an Interactive Grid in *Oracle APEX End User's Guide*

7. To view global template options, expand **Global Template Options**.

11.3.6.2.3 Creating a Template Option Group

Define the purpose of related template options by creating template option groups.

To create a template option group:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. Select a template.
3. Locate **Template Options** and click **Template Options Groups**..

   The Template Options Groups window appears.
4. To create a new group:
   a. Click **Add Template Option Group**.

      A new row appears.
   b. Edit the appropriate attributes:
      - Template Type - Specify the type of template for this template option group.
      - Name - Enter a display name for the template option group.

- Identifier - Defines the internal name of the template option group.
- Sequence - Specify the sequence for this template option group.
- Null Text - Default value for the template option group if a user does not select any template option in that group.
- Help Text - Enter a description of the template option group.
- Advanced - Determines if the template option group belongs in the Basic or Advanced category of template options. Select **Yes** or **No**.

c. Click **Save**.

11.3.6.2.4 Editing Template Option Groups

Edit template option groups by navigating to the Templates page.

To edit template option groups:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. Select a template.
3. Locate **Template Options** and click **Template Options Groups**..

   The Template Options Groups window appears.
4. To edit a template option group:
   a. Select the row containing the template option group.

      The row becomes hightlighted.
   b. Edit the appropriate attributes.
      - Template Type - Specifies the type of template for this template option group.
      - Name - Dsplay name for the template option group.
      - Identifier - Defines the internal name of the template option group.
      - Sequence - Specify the sequence for this template option group.
      - Null Text - Default value for the template option group if a user does not select any template option in that group.
      - Help Text - Enter a description of the template option group.
      - Advanced - Determines if the template option group belongs in the Basic or Advanced category of template options. Select **Yes** or **No**.
   c. Click **Save**.
5. To delete a template option group:
   - Click the **Row Actions** menu and select **Delete Row**.

> **Tip:**
>
> To learn more about the Row Actions menu, see About the Row Actions Menu in an Interactive Grid in *Oracle APEX End User's Guide*

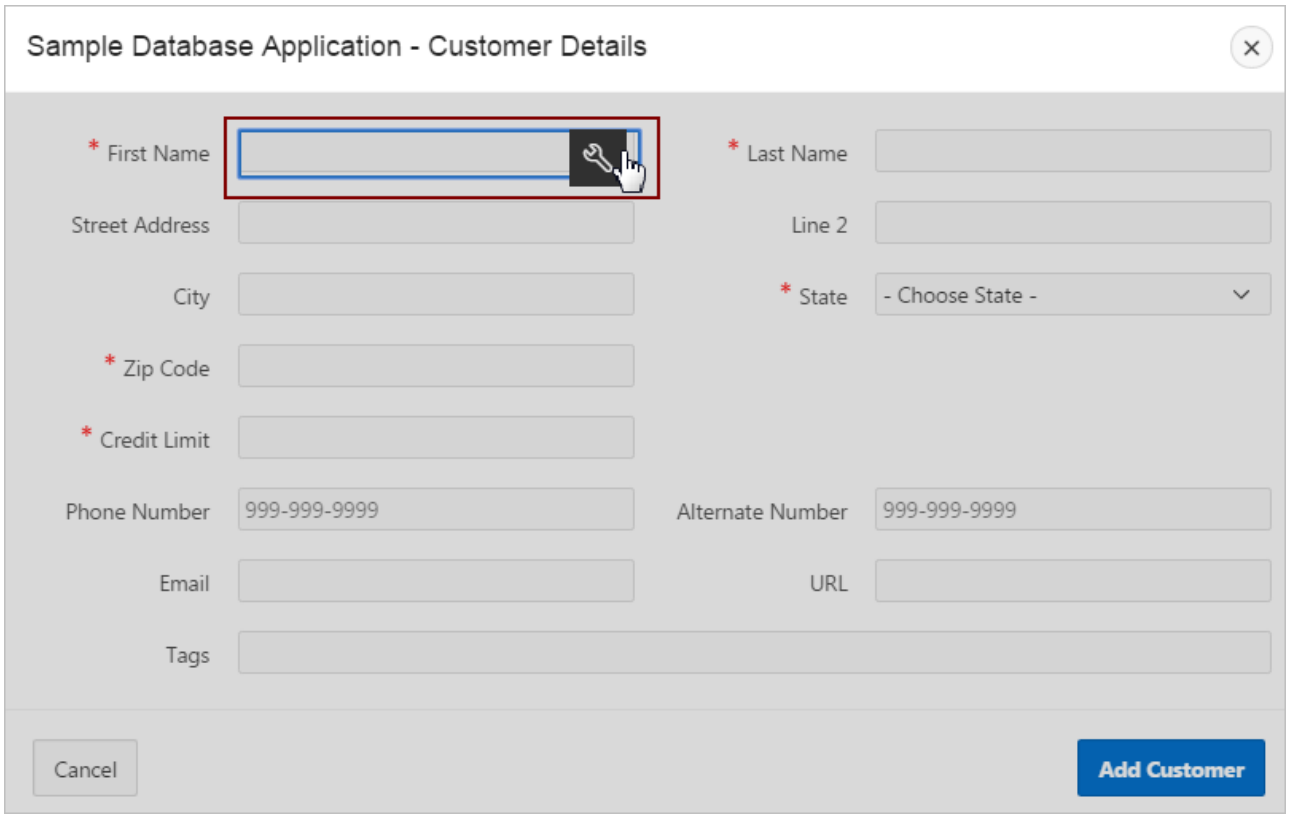
11.3.6.2.5 Using Quick Edit to Modify Live Template Options

Using the Live Template Options dialog to alter a component's template options without refreshing the page.

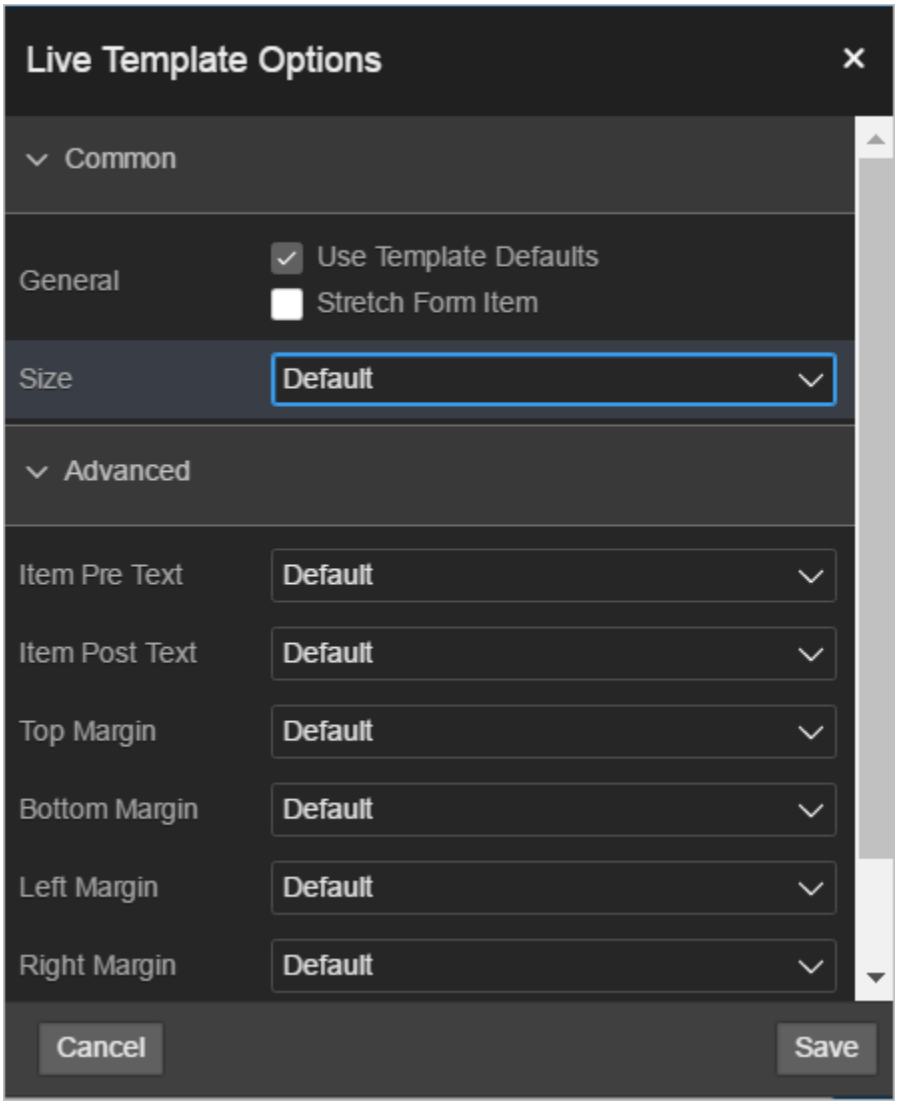
Developers can also use Quick Edit on the Runtime Developer toolbar to access the Live Template Options dialog. Using the Live Template Options dialog, developers can alter component's template options without refreshing the page.

To access the Live Template Options:

1. Run the page.
2. From the Runtime Developer toolbar, click **Quick Edit**.
3. Move the mouse over the component for which you want to modify template options. Click the **Wrench** icon in the upper right corner.



The Live Template Options dialog appears. For region-based components, such as lists and classic reports, the dialog includes an Attributes tab.



4. Select the appropriate tab, edit the appropriate attributes, and click **Save**.

> **See Also:**
>
> Runtime Developer Toolbar

11.3.7 Viewing Template Reports

View reports describing template utilization, subscriptions, published templates, and edit history.

To view template reports for the current application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

The Templates page displays as an interactive report. To customize the view, use the Search bar at the top of the page.

5. To view template reports, click the following tabs:
   - **Subscription** - Displays subscribed templates in your application.
   - **Publish** - Displays templates that have been subscribed to by other templates in your workspace. To publish master template content to subscribing templates, select those you want to publish and click Publish Checked.
   - **Utilization** - Displays template utilization in the current application for all template types. The number in the References column indicates the total number of pages that use the specific template.
   - **History** - Displays recent modifications made to Page and Region Templates in this application.

11.3.8 Managing Templates

Manage templates by navigating to the Templates page in Shared Components.

- Viewing Templates on the Templates Page
  View templates on the Templates page in Shared Components
- Viewing Templates Associated with a Specific Page
  View templates associated with a page by viewing the Page Designer.
- Replacing Templates
  Update the template associated with a component using the Replace Templates Wizard.
- Replacing All Templates within an Application
  Replace all templates within an application with templates from another application.
- Viewing Region Position Utilization by Page Template
  View region position utilization by page template.
- Unsubscribing to Templates
  Unsubscribe to templates.
- Publishing Templates
  Publish master template content to subscribing templates.

> **See Also:**
>
> Viewing All Templates in a Theme

11.3.8.1 Viewing Templates on the Templates Page

View templates on the Templates page in Shared Components

To view existing templates:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.

   d. Under User Interface, select **Templates**.

   The Templates page appears.

2. To view or edit a template definition, click the template name.

   The template definition appears.

3. Edit the appropriate attributes. To learn more about an attribute, see field-level Help.

   If you edit a template, you can make changes in one window and run your application in another by selecting the **Return to Page** checkbox on the right side of the template definition page. Selecting this checkbox keeps the page you are editing current after you click **Apply Changes**.

4. Click **Apply Changes**.

11.3.8.2 Viewing Templates Associated with a Specific Page

View templates associated with a page by viewing the Page Designer.

To view templates associated with a specific page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. View the page in Page Designer:
   a. In the Rendering tab, click the page title.
   b. In the Property Editor, locate **Appearance**. **Page Template** displays the current template.
   c. To save any changes, click **Save**.

> **See Also:**
>
> Viewing a Page in Page Designer

11.3.8.3 Replacing Templates

Update the template associated with a component using the Replace Templates Wizard.

To replace the template associated with a component:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

      To customize the view, use the Search bar at the top of the page.
2. From the Task list, select **Replace Templates**.
3. For Template Type:
   a. Template Type - Identifies the template type to be replaced.

   b. Click **Next**.
4. For Replace Templates:
   a. Change From - Select the template you want to change.
   b. Change To - Select the template you want to change to.
   c. Click **Next**.
5. Click **Finish**.

11.3.8.4 Replacing All Templates within an Application

Replace all templates within an application with templates from another application.

To replace all templates within an application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

   To customize the view, use the Search bar at the top of the page.
5. From the Task list, select **Replace templates in this application with templates from another application**.
6. For Replace Templates:
   - Replace from Application - Select an application.
   - User Interface - Select the type of user interface for which this application has been developed.
   - Click **Next**.
7. On Replace Templates, select a New Template for each component and then select an Action:
   - **Replace** copies the template definition
   - **Replace/Subscribe** copies the templates and adds a subscription.
8. Click **Replace Templates**.

11.3.8.5 Viewing Region Position Utilization by Page Template

View region position utilization by page template.

Regions are organized on a page by position (or Display Point). The possible display points for a region are determined by the page-level template.

To view region position utilization by page template:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Shared Components**.
4. Under User Interface, select **Templates**.

   To customize the view, use the Search bar at the top of the page.
5. From the Task list, select **View page template region position utilization**.

The Region Display Point Utilization report appears.

11.3.8.6 Unsubscribing to Templates

Unsubscribe to templates.

A subscribed template is a template that has its definition maintained in another template referred to as the referenced template. If your application uses subscribed templates, you can unsubscribe to templates on the Unsubscribe Templates page.

To unsubscribe to templates:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. From the Task list, select **Unsubscribe Templates**.

   The Unsubscribe Templates page appears.
3. Select a theme.
4. To unsubscribe to a specific template type within the current theme, select the template type and click **Unsubscribe**.

11.3.8.7 Publishing Templates

Publish master template content to subscribing templates.

Use the Publish Templates page to view templates that are subscribed to by other templates in your workspace.

To publish master template content:

1. Navigate to the Templates page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Templates**.

   The Templates page appears.
2. Click the **Publish** tab.

   The Published Templates page appears.
3. Select those you want to publish and click **Publish Checked**.

11.3.9 Breadcrumb Templates

Breadcrumb templates control the display of breadcrumb entries.

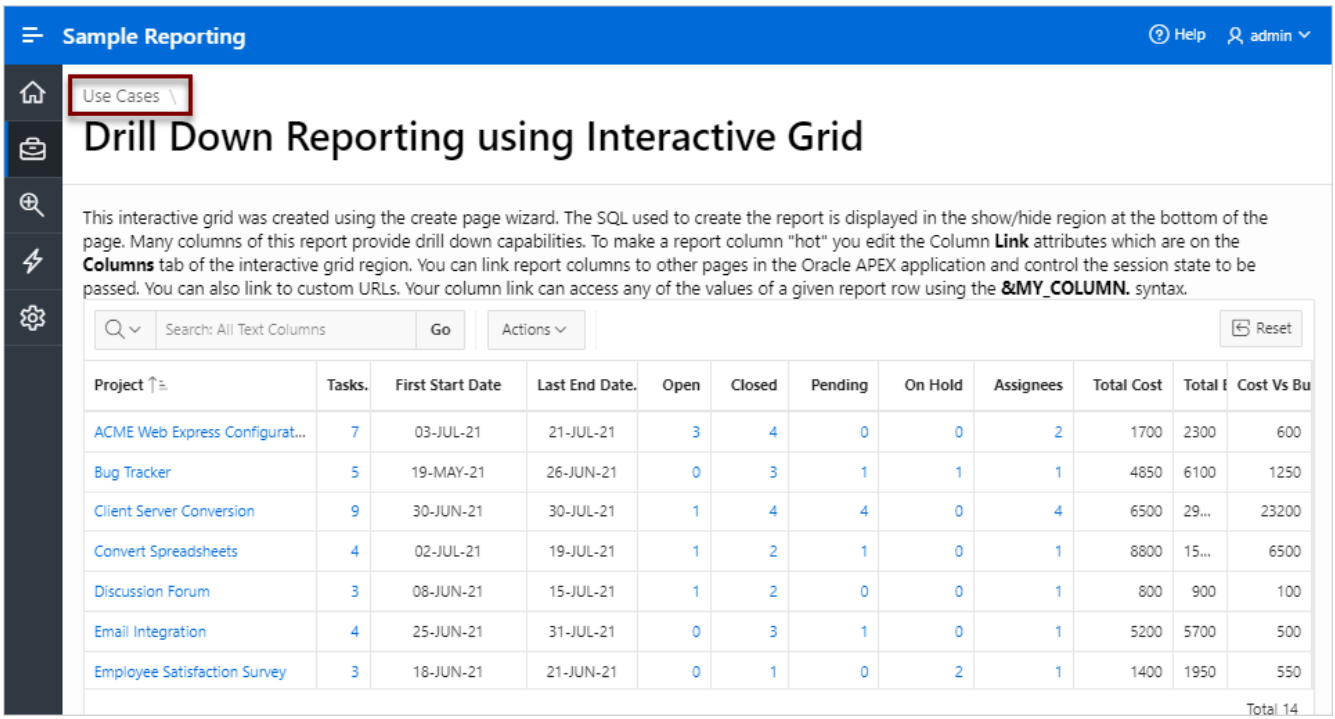
> **See also:**
>
> Using Themes

- About Breadcrumb Style Navigation
  Breadcrumbs provide users with hierarchical navigation and can be displayed as a list of links or as a breadcrumb path.
- Breadcrumb Template Attributes
  Learn about common breadcrumb template attributes.

11.3.9.1 About Breadcrumb Style Navigation

Breadcrumbs provide users with hierarchical navigation and can be displayed as a list of links or as a breadcrumb path.

You select a breadcrumb template when you create a region. Breadcrumbs usually indicate where the current page is relative to other pages in the application. Users can click a specific page to instantly view it. The following is an example of breadcrumb navigation in *Sample Reporting* Sample App.



> **See Also:**
>
> Creating Breadcrumbs

11.3.9.2 Breadcrumb Template Attributes

Learn about common breadcrumb template attributes.

For more information on an attribute, see field-level Help.

- Name
- Definition
- Substitution Strings

11.3.9.2.1 Name

Name identifies the name of the template. Use the **Translatable** checkbox to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member. **Template Class** identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

11.3.9.2.2 Definition

Definition attributes control how a breadcrumb displays Table 11-7 describes Definition attributes.

Table 11-7 Definition attributes

Attribute	Description
Before First	Defines text that displays before the first breadcrumb entry. Use the substitution string `#COMPONENT_CSS_CLASSES#` to apply a CSS class to your breadcrumb template. You edit component CSS classes by editing the region attributes. **See Also:** Editing Region Attributes
Current Page Breadcrumb Entry	Defines the look of a breadcrumb entry that corresponds to the current page.
Non Current Page Breadcrumb Entry	Defines the look of a breadcrumb entry that does not correspond to the current page.
After Last	Defines text that displays after the last breadcrumb entry.
Between Level	Defines text that displays between levels of breadcrumb entries.
Maximum Levels	Specifies the number of levels that appear when displaying breadcrumbs in a breadcrumb style.

11.3.9.2.3 Substitution Strings

Substitution strings are used within templates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the **Substitution Strings** report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

Table 11-8 Breadcrumb Template Substitution Strings

Referenced From	Substitution String	Description
Before First	`#COMPONENT_CSS_CLASSES#`	Component CSS classes. The `#COMPONENT_CSS_CLASSES#` string must be included in the before-first attribute for templates that have template options defined. The `#COMPONENT_CSS_CLASSES#` string is where the corresponding template option classes are injected at runtime.
Current Entry	`#LINK#`	URL Link.
Current Entry	`#LONG_NAME#`	Breadcrumb Long Name.
Current Entry	`#NAME#`	Breadcrumb Label.
Current Entry	`#NAME_ESC_SC#`	Breadcrumb Label (escaping HTML characters).
Non Current Entry	`#LINK#`	URL Link.
Non Current Entry	`#LONG_NAME#`	Breadcrumb Long Name.
Non Current Entry	`#NAME#`	Breadcrumb Label.
Non Current Entry	`#NAME_ESC_SC#`	Breadcrumb Label (escaping HTML characters).

11.3.10 Button Templates

Customize the look and feel of a button using button templates. Button templates are optional.

- Button Template Attributes
  Learn about common button template attributes. You can use multiple images or HTML tags to build a button.

11.3.10.1 Button Template Attributes

Learn about common button template attributes. You can use multiple images or HTML tags to build a button.

For more information on an attribute, see field-level Help.

- Name
- Definition
- Substitution Strings

11.3.10.1.1 Name

Template Name identifies the name of the template. Use the **Translatable** checkbox to indicate if the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

11.3.10.1.2 Definition

Normal Template defines the button template that displays if the button type is **Normal**. For example, `&ITEM_NAME.` values can be substituted at rendering time. **Hot Template** defines the button template that displays if the button type is **Hot**. You have the option of including standard application substitutions. For example, `&ITEM_NAME.` values can be substituted at rendering time.

To learn more about supported substitution strings, see field-level Help.

> **Tip:**
>
> All substitution strings must be in uppercase.

11.3.10.1.3 Substitution Strings

Substitution strings are used within subtemplates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the **Substitution Strings** report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

Referenced From	Substitution String	Description
Hot Template	`#BUTTON_ATTRIBUTES#`	Button attributes.
Hot Template	`#BUTTON_CSS_CLASSES#`	Button CSS classes. You must include the `#BUTTON_CSS_CLASSES#` string for templates that have template options defined. The `#BUTTON_CSS_CLASSES#` string is where the corresponding template option classes are injected at runtime.
Hot Template	`#BUTTON_ID#`	Generated button ID will be either the button's Static ID if defined, or if not will be an internally generated ID in the format `'B' || [Internal Button ID]`.
Hot Template	`#BUTTON_NAME#`	Button name.
Hot Template	`#ICON_CSS_CLASSES#`	Icon CSS classes.
Hot Template	`#JAVASCRIPT#`	Used in an onclick attribute.
Hot Template	`#LABEL#`	Button label.
Hot Template	`#LINK#`	To be used in a `href` attribute.
Hot Template	`#ONCLICK#`	Indicates that the button has an onclick event with JavaScript code, which will be included in a script tag at the bottom of the page.

Referenced From	Substitution String	Description
Template	`#BUTTON_ATTRIBUTES#`	Button attributes.
Template	`#BUTTON_CSS_CLASSES#`	Button CSS classes. You must include the `#BUTTON_CSS_CLASSES#` string for templates that have template options defined. The `#BUTTON_CSS_CLASSES#` string is where the corresponding template option classes are injected at runtime.
Template	`#BUTTON_ID#`	Generated button ID will be either the button's Static ID if defined, or if not will be an internally generated ID in the format `'B' || [Internal Button ID]`.
Template	`#BUTTON_NAME#`	Button name.
Template	`#ICON_CSS_CLASSES#`	Icon CSS classes.
Template	`#JAVASCRIPT#`	Used in an onclick attribute.
Template	`#LABEL#`	Button label.
Template	`#LINK#`	To be used in a `href` attribute.
Template	`#ONCLICK#`	Indicates that the button has an onclick event with JavaScript code, which will be included in a script tag at the bottom of the page.

11.3.11 Legacy Calendar Templates

Control the appearance and placement of a legacy calendar using legacy calendar templates.

Calendar templates frequently use HTML tables to arrange dates. You place calendar attributes using substitution strings such as `#DD#` and `#MONTH#`.

- Supported Legacy Calendar Template Substitution Strings
  Review supported legacy calendar template substitution strings.
- Legacy Calendar Template Attributes
  Learn about legacy calendar template attributes.

> **See Also:**
>
> Managing Legacy Calendars

11.3.11.1 Supported Legacy Calendar Template Substitution Strings

Review supported legacy calendar template substitution strings.

A list of supported substitution strings appears on the right side of the Edit Calendar Template page. Note that template substitution strings must be in uppercase letters and begin and end with a number sign (`#`).

11.3.11.2 Legacy Calendar Template Attributes

Learn about legacy calendar template attributes.

- Name

- Subscription
- Comments

11.3.11.2.1 Name

Name identifies the name of the template. Use the **Translatable** checkbox to indicate if the template contains text strings that require translation.**Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

11.3.11.2.2 Subscription

Use Subscription to apply an existing template to the current application.

Use **Reference Master Label Template From** to select an existing template in this workspace or another template in your workspace's schema. By selecting an existing template, you become a subscriber to that template.

To load a copy of a master template, click **Refresh**.

11.3.11.2.3 Comments

Use this attribute to record comments about this component.

Monthly Calendar, Weekly Calendar, Daily Calendar, Custom Calendar and List Calendar

Use the Monthly Calendar, Weekly Calendar, Daily Calendar, and List Calendar attributes to control the appearance and placement of specific calendars.

11.3.12 Label Templates

Centrally manage HTML markup of page item labels using label templates.

Every item can have an optional label. You can control how these labels display using label templates. For example, you could create a label template called Required Field that references an image (such as an asterisk) to indicate to the user that the field is required.

Label templates also enable you to define a before-and-after text string that gets prepended and appended to the item.

- Label Template Attributes
  Learn about label template attributes.
- About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help
  Learn about using `#CURRENT_ITEM_HELP_TEXT#` to create item help.

11.3.12.1 Label Template Attributes

Learn about label template attributes.

To learn more about a specific attribute, see field-level Help.

- Name
- Definition

- Error Display
- Field Container
- Substitution Strings

11.3.12.1.1 Name

Template Name identifies the name of the template. Use the **Translatable** checkbox to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a theme, all templates in one theme are mapped to corresponding templates in another theme. App Builderaccomplishes this template mapping through the assignment of a template class.

11.3.12.1.2 Definition

Definition attributes include:

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label** - Enter HTML to display before the item label. The `#ITEM_CSS_CLASSES#` substitution string must be included for templates that have template options defined.
- **After Label** - Enter HTML to display after the item label. Since the label automatically displays before the HTML in this region, any open HTML tags in the Before Label region should be closed.
- **Before Item** - Enter HTML to be displayed directly before the form item. This is typically used to wrap an item into a DIV container or to display a help button before or after the form item.
- **After Item** - Enter HTML to be displayed directly after the form item. This is typically used to wrap an item into a DIV container or to display a help button before or after the form item.
- **Item Pre Text** - Enter HTML for the Item Pre Text subtemplate to format the content stored as Pre Text in the item attributes. This subtemplate can be referenced in the **Before Element** template with the `#ITEM_PRE_TEXT#` substitution.
- **Item Post Text** - Enter HTML for the Item Post Text subtemplate to format the content stored as Post Text in the item attributes. This subtemplate can then be referenced in the **After Element** template with the `#ITEM_POST_TEXT#` substitution
- **Help Template** - The help sub template enables developers to define the link or button used to invoke the item level help. This sub template is only rendered for items that have help defined. See About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help.
- **Inline Help Template** - The help sub template enables developers to define inline help text to be displayed immediately with the item. This sub template is only rendered for items that have inline help text defined

11.3.12.1.3 Error Display

Error Display attributes include:

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

- **On Error Before Label** - Enter HTML to precede the item label when an application displays an inline validation error message for the item.
- **On Error After Label** - Enter HTML to be appended to the item label when an application displays an inline validation error message for the item.
- **Error Template** - Enter the HTML rendered to display the inline validation error message. Use the `#ERROR_MESSAGE#` substitution string to reference the message text. The error template can be included in the before and after item attributes using the `#ERROR_TEMPLATE#` substitution string.

11.3.12.1.4 Field Container

Field Container attributes include:

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

- **Before Label and Item** - Enter HTML to be displayed before the label and item. This is typically used to wrap an item into a DIV container or field set. The Before Label and Item attribute is only rendered for pages that do not use a table grid to display form items. This includes any page using a mobile page template.
- **After Label and Item** - Enter HTML to be displayed after the label and item. This is typically used to wrap an item into a DIV container or field set. The after item and label attribute is only rendered for pages that don't use a table grid to display form items. This includes any page using a mobile page template.

11.3.12.1.5 Substitution Strings

Substitution strings are used within templates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the **Substitution Strings** report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

11.3.12.2 About Using #CURRENT_ITEM_HELP_TEXT# to Create Item Help

Learn about using `#CURRENT_ITEM_HELP_TEXT#` to create item help.

Use the `#CURRENT_ITEM_HELP_TEXT#` substitution string to create inline page Help text when a page is rendered. For example, you can use this substitution string to create page Help text which displays as a tooltip when the mouse hovers over the page item label. This placeholder

enables you to include help text on the page when it is rendered without having to perform an expensive Ajax call.

> **Tip:**
>
> Only use this placeholder if the Help text is short. The generated page always includes the help text of all page items.

Example Label Template

Before Label:

```
<label for="#CURRENT_ITEM_NAME#" id="#LABEL_ID#">
<span class="helpText">#CURRENT_ITEM_HELP_TEXT#</span>
```

After Label:

```
</label>
```

11.3.13 List Templates

Control the appearance of a list using list templates.

- About Hierarchical Lists
  Learn about hierarchical lists.
- List Template Attributes
  Learn about list template attributes.

11.3.13.1 About Hierarchical Lists

Learn about hierarchical lists.

A list is a shared collection of links. Oracle APEX supports hierarchical lists. To create a hierarchical list, you must:

- Select a list template that supports hierarchical lists. To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist."

  > **Tip:**
  >
  > To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist."

- Select a Parent List Entry when you create each list entry.

> **See Also:**
>
> Creating Lists

11.3.13.2 List Template Attributes

Learn about list template attributes.

To learn more about a specific attribute, see field-level Help.

- Name
- Before List Entry
- Template Definition
- Template Definitions for First Entry
- Before Sublist Entry
- Sublist Entry
- After Sub List Entry
- After List Entry
- Substitution Strings

11.3.13.2.1 Name

Name identifies the name of the template. Use the **Translatable** checkbox to indicate that the template contains text strings that require translation. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builderaccomplishes this template mapping through the assignment of a template class.

11.3.13.2.2 Before List Entry

Enter HTML that displays before any list elements. You can use this attribute to open an HTML table or HTML table row. The `#COMPONENT_CSS_CLASSES#` substitution string must be included in the List Template Before Rows attribute for templates that have template options defined.

11.3.13.2.3 Template Definition

Defines current and noncurrent list templates. Supported substitution strings include `#LINK#`, `#TEXT#`, `#TEXT_ESC_SC#`, `#ICON_CSS_CLASSES#`, `#APEX_FILES#`, `#IMAGE#`, `#IMAGE_ATTR#`, and `#A01#...#A10#`.

Template Definition attributes include:

- **List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.
- **List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.
- **List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.

- **Between List Elements.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

11.3.13.2.4 Template Definitions for First Entry

Available attributes include:

- **List Template Current (First).** If defined, this is used for the first list item. Enter HTML or text to be substituted for the selected (or current) list template.
- **List Template Current with Sub List Items (First).** If defined, this is used for the first list item. HTML or text to be substituted for the selected (or current) list template used when item has sub list items.
- **List Template Noncurrent (First).** If defined, this is used for the first list item. Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **List Template Noncurrent with Sub List Items (First).** If defined, this is used for the first list item. HTML or text to be substituted for the unselected (or noncurrent) list template used when item has sub list items.

11.3.13.2.5 Before Sublist Entry

Enter HTML that displays before any sublist elements.

11.3.13.2.6 Sublist Entry

Defines current and noncurrent list templates. Supported substitution strings include `#LINK#`, `#TEXT#`, `#ICON_CSS_CLASSES#`, `#APP_FILES#`, `#IMAGE#`, `#IMAGE_ATTR#`, `#ICON_CSS_CLASSES#`, `#A01#`...`#A10#`, `#LIST_ITEM_ID#`, and `#PARENT_LIST_ITEM_ID#`.

Sub List Entry attributes include:

- **Sub List Template Current.** Enter HTML or text to be substituted for the selected (or current) list template.
- **Sub List Template Current with Sub List Items.** Enter HTML or text to be substituted for the selected (or current) list template when an item has sublist items. If not specified, the current list item template is used.
- **Sub List Template Noncurrent.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template.
- **Sub List Template Noncurrent with Sub List Items.** Enter HTML or text to be substituted for the unselected (or noncurrent) list template used when an item has sublist items. If not specified, the current list item template is used.
- **Between Sub List Items.** Enter HTML that displays between list elements. This attribute is ignored if no HTML is specified.

11.3.13.2.7 After Sub List Entry

Enter HTML that displays after displaying sublist elements.

11.3.13.2.8 After List Entry

Enter HTML that displays after displaying all list elements. You can use this attribute to close an HTML table opened in the Before List Elements attribute.

11.3.13.2.9 Substitution Strings

Substitution strings are used within templates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the Substitution Strings report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

11.3.14 Page Templates

Define the appearance of a page using page templates.

- About Page Templates
  Learn about page templates available in the Universal Theme.
- Page Template Attributes
  Learn about page template attributes.
- Required Dialog Page Template Attributes
  Learn about required dialog page template attributes.

11.3.14.1 About Page Templates

Learn about page templates available in the Universal Theme.

Page Templates define the layout and purpose of your pages. The Universal Theme supports two types of navigation menus: side tree-based navigation and top tab-like navigation.

To view samples of all the available page templates:

1. Run the Universal Theme Reference app.
2. From the **Main Navigation** menu, select **Design** and then **Page Templates**.

> **See Also:**
>
> Accessing the Universal Theme Reference App

11.3.14.2 Page Template Attributes

Learn about page template attributes.

A page template consists of a header template, a body template, a footer template, and several subtemplates. If you do not specify a page template as a page-level attribute, then the Oracle APEX engine uses the default page template defined on the Create/Edit Theme page.

Page templates combine static HTML with substitution strings that are replaced at runtime. You use substitution strings to indicate the existence and placement of a component within a page template. You can further specify how a component should display using subtemplates.

This topic describes specific sections of the Edit Page Template page. To learn more about a specific attribute, such as supported substitution strings, see field-level Help.

- Name
- Definition
- JavaScript
- Cascading Style Sheet
- Subtemplate
- Image Based Tab Attributes
- Layout
- Positions
- Dialog
- Error Page Template Control
- Substitution Strings

11.3.14.2.1 Name

Name identifies the name of the template. **Theme** indicates the theme to which the template is a member.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

Use the **Translatable** checkbox to indicate if this template contains text strings that require translation.

Template Type identifies the page template type. If set to **Dialog Page**, the template is only be available for selection by a page with a page mode of **Modal** or **Non-Modal**. If set to **Normal Page**, the template is only be available for selection by a page with a page mode of **Normal**.

> **See Also:**
>
> Creating Dialog Pages and Required Dialog Page Template Attributes

11.3.14.2.2 Definition

Each template consists of a header, a body, a footer, and subtemplates. Use substitution strings to include dynamic content. All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

Definition attributes include:

- **Header** - Enter HTML that defines the `<Head>` section of the HTML document. Oracle recommends you include at least `#TITLE#` and `#HEAD#` in this template section.
- **Body** - Enter HTML that makes up the `<Body>` section of the HTML document. Include substitution strings to define where the body content is placed and to define additional positions for regions.
- **Footer** - Enter HTML to define third section in the page template that displays after the Body.

11.3.14.2.3 JavaScript

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

Available attributes include:

- **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string **#APP_VERSION#** if you want to include the application's version in the file URL.

  JavaScript file URLs you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

  Do not include opening or closing script tags, just write the URL.
- **Function and Global Variable Declaration** - Enter JavaScript code (for example, functions or global variable declarations) for code to be used in this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

  Do not include opening or closing script tags, just include the JavaScript code.

  To reference a shortcut, use `"SHORTCUTNAME"`.
- **Execute when Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle APEX. Code you enter here replaces the `#GENERATED_JAVASCRIPT#` substitution string in the page template.

  You do not need to add any opening or closing script tags, just the JavaScript code.
- **Dialog Initialization Code** - This attribute only displays if the Template Type is **Dialog Page**. Enter JavaScript code to initialize the page as a dialog. The substitution strings will be replaced with their respective values at rendering time.
- **Dialog Closure Code**. This attribute only displays if the Template Type is **Dialog Page**. Enter JavaScript code to close the dialog page.
- **Dialog Cancel Code**. This attribute only applies if the Template Type is **Dialog Page**. Enter JavaScript code to cancel the dialog page.

11.3.14.2.4 Cascading Style Sheet

File URLs

Enter Cascading Style Sheet file URLs to be loaded for this page template. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

Do not include opening or closing script tags, just include the file URL.

Inline

Enter Cascading Style Sheet code to be used for this page template. Consider putting it into an external file to avoid duplication and to leverage browser caching of static files. Code you enter here replaces the `#TEMPLATE_CSS#` substitution string in the template.

Do not include opening or closing style tags, just include the cascading style sheet code.

11.3.14.2.5 Subtemplate

Enter the HTML for subtemplates below. Subtemplates are placed inside the body of the page template using substitution strings.

Subtemplates include:

- **Success Message** - Enter HTML to replace the string `#SUCCESS_MESSAGE#` in the template body, header, or footer.
- **Navigation Bar** - Enter HTML or text to replace the string `#NAVIGATION_BAR#` in the template header, body, or footer. Use the substitution string `#BAR_BODY#` to indicate where in the Navigation Bar the entries should appear.
- **Navigation Bar Entry** - Enter HTML or text to be substituted into the navigation bar `#BAR_BODY#` substitution string for each navigation bar entry.
- **Notification** - Enter HTML or text to be substituted when the `#NOTIFICATION_MESSAGE#` substitution string is referenced in the template header, body, or footer.

11.3.14.2.6 Image Based Tab Attributes

Use this subtemplate for tabs that are entirely based on images. Available attributes include:

- **Current Image Tab -** Enter HTML to be used to indicate that an image-based tab is currently selected. Include the `#TAB_TEXT#` substitution string to show the displayed name of the tab.
- **Non Current Image Tab** - Enter the HTML to be used to indicate that an image tab is not currently selected. Include the `#TAB_TEXT#` substitution string to show the displayed name of the tab.

11.3.14.2.7 Layout

Specified the type of layout used for the page. The Type you select determines what attributes displays. To learn more about an attribute, see field-level Help.

From Type, select a layout:

- **HTML Table** - Uses HTML Tables to position regions, page items, and buttons.
- **Fixed Number of Columns** - Used for grids which divide the screen into a fixed number of grid columns.
- **Variable Number of Columns** - Used for grids which can display up to "n" number of grid columns depending on how the regions, page items and buttons are positioned.

11.3.14.2.8 Positions

Create positions for this page template. Positions enable you to identify where components (that is, regions, items, buttons) display on this page template. The substitution string of a position cannot be one of the reserved page substitution strings, (for example, `NOTIFICATION_MESSAGE`).

Navigate to the Edit Page Template, Positions tab to see a complete list of reserved substitution strings.

11.3.14.2.9 Dialog

Dialog attributes only appear if you select a Template Type of **Dialog Page**. These settings are overwritten by a similar attributes specified at page-level.

Available Dialog attributes include:

- **Width** - Enter the width of the dialog, in pixels or a percentage.
- **Height** - Enter the height of the dialog in pixels, for example, 500.
- **Maximum Width** - Enter the maximum width of the dialog, in pixels.
- **CSS Classes** - Enter additional CSS classes you wish to be applied to the dialog, as a space delimited list.
- **Allow Embed in Frames** - Your page template level browser frame options are listed below, along with the current setting for each. Make a selection to control if the browser is allowed to display your application's pages within a frame. The current setting will be the default for each modal and non-modal page using this page template. To learn more, see field-level Help.

  APEX embeds modal dialogs in iframes. If you set the application security attribute Browser Frame to **Deny**, checking **Allow Embed In Frame** for Modal Dialog will instead use **Allow from same origin** for dialog pages that use this template

> **See Also:**
>
> Required Dialog Page Template Attributes

11.3.14.2.10 Error Page Template Control

Use this attribute when a page template is designated as an error template.

11.3.14.2.11 Substitution Strings

Substitution strings are used within templates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the **Substitution Strings** report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

11.3.14.3 Required Dialog Page Template Attributes

Learn about required dialog page template attributes.

To create a dialog page template, you must define the following Page Template attributes.

Attribute	Description
Name, Template Type	Must be defined as **Dialog Page**.
JavaScript, Dialog Initialization Code	Enter JavaScript code to initialize the page as a dialog. See field-level Help for function call syntax.
JavaScript, Dialog Closure Code	Enter JavaScript code to close the dialog page. See field-level Help for function call syntax.
JavaScript, Dialog Cancel Code	Enter JavaScript code to cancel the dialog page. See field-level Help for function call syntax.
Dialog, Width	Sets the default width for dialogs. The page template-level setting can be overwritten at page-level.
Dialog, Height	Sets the default height for dialogs. The page template-level setting can be overwritten at page-level.
Dialog, Maximum Width	Sets the maximum width for dialogs, in pixels. The page template-level setting can be overwritten at page-level.
Dialog, CSS Classes	Enter additional CSS classes you wish to be applied to dialogs. The page template-level setting can be overwritten at page-level.
Dialog, Allow Embed in Frame	Select the page mode to use this attribute to control if the browser is allowed to display your application's pages within a frame. APEX embeds modal dialogs in iframes. If you set the application security attribute Browser Frame to **Deny**, checking **Allow Embed In Frame** for Modal Dialog will instead use **Allow from same origin** for dialog pages that use this template.

> **See Also:**
>
> Creating Dialog Pages

11.3.15 Popup LOV Templates

Use Popup LOV templates control how popup lists display for all items defined as `POPUP`.

You can only specify one popup LOV template for each theme.

- Popup List of Values Template Attributes
  Learn about popup list of values template attributes.

> **See Also:**
>
> Creating Lists of Values at the Application-Level

11.3.15.1 Popup List of Values Template Attributes

Learn about popup list of values template attributes.

This section describes some sections of the Edit Popup List of Values Template page. For more information on an attribute, see field-level Help.

- Search Field
- Buttons
- Window
- Pagination
- Result Set
- Page Attributes

11.3.15.1.1 Search Field

Use these attributes to specify how a Search field displays. Table 11-9 describes available Search Field attributes.

Table 11-9 Search Field Attributes

Attribute	Description
Before Field Text	Defines text to display before the popup list of values search field displays.
After Field Text	Displays this text after displaying the search field, the search button, and the close button.
Filter Width	Displays the `HTML INPUT TYPE = TEXT` widget using this width.
Filter Max Width	Displays the `HTML INPUT TYPE = TEXT` widget using this maximum width.
Filter Text Attribute	Displays the `HTML INPUT TYPE = TEXT` widget using these attributes. This is included within the HTML input tag.

11.3.15.1.2 Buttons

Use these attributes to define the button name and attributes for the Find, Close, Next, and Previous buttons.

11.3.15.1.3 Window

Popup lists of values are executed using JavaScript. Use these attribute to control the values of `scrollbars=`, `resizable=`, `width=`, and `height=`.

11.3.15.1.4 Pagination

Defines how row count results display.

11.3.15.1.5 Result Set

Use these attributes to define text or HTML to display before and after a result set.

11.3.15.1.6 Page Attributes

Use these attributes to define popup pages.

11.3.16 Template Component Type Plug-ins

Learn about Template Component Type plug-ins.

A Template Component is a type of plug-in which contains an HTML template and includes support for template directives. Template Components which are part of a theme display under **Shared Components**, **Templates**. Standalone Template Components without a theme display under **Shared Components**, **Plug-ins**.

To copy a Template Component Type plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. On the Plug-ins page, narrow the display
4. Click **Create**.

   The Create Plug-in wizard appears.
5. Create Plug-in - Select **As a Copy of an Existing Plug-in** and click **Next**.
6. Copy From:
   a. Copy From Application - Select the application from which to copy the Template Component.
   b. Theme - Identify the theme from which you wish to copy templates.
   c. Click **Next**.
7. Copy:
   - Identify the Plug-in to be copied and select the copy option. Options include:
     - **To Name** - Enter a name for the new Template Component.
     - **Copy** - Select **Yes**.
8. Click **Copy Plug-ins**.

   A new Template Component is created. You can edit it to specify additional custom attributes, action positions, action templates, or upload files.

> **See Also:**
>
> About Template Component Type Plug-ins

11.3.17 Region Templates

Region templates control the appearance and placement of region attributes.

You place region attributes using substitution strings such as `#BODY#`. Template substitution strings must be in upper case and begin and end with a pound sign (#).

- Region Template Attributes
  Learn about region template attributes.

11.3.17.1 Region Template Attributes

Learn about region template attributes.

This topic describes some attributes on the Edit Region Template page. To learn more about a specific attribute, see field-level Help.

- Name
- Definition
- Layout
- Sub Regions
- Image
- JavaScript
- Cascading Style Sheet
- JavaScript
- Substitution Strings

> **See Also:**
>
> Understanding Template Options and Managing Theme Subscriptions

11.3.17.1.1 Name

Theme indicates the theme to which the template is a member. **Name** identifies the name of the template.

Template Class identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class. Use the **Translatable** checkbox to indicate that the template contains text strings that require translation.

11.3.17.1.2 Definition

Region templates provide the appearance for a region on a page. Region templates consist of HTML code to wrap the region content and substitution strings to inject the region content. `#BODY#` is the only required substitution string and identifies where the source of the region should be placed. All other substitution strings are optional. Substitution strings can be either:

- Reserved APEX substitution strings
- Region positions

Region templates provide the appearance for a portion of a page called a region. Use substitution strings to indicate the existence and placement of a component within the region. `#BODY#` is the only required substitution string and identifies where the source of the region should be placed. All other substitution strings are optional. The following are valid substitution strings:

The following reserved APEX substitution strings are used to substitute content from the region defined in Page Designer:

- `#TITLE#`
- `#PAGE_TITLE#`
- `#SUB_REGION_HEADERS#`
- `#FORM_OPEN#`
- `#FORM_CLOSE#`
- `#REGION_ID#`
- `#REGION_STATIC_ID#`
- `#REGION_CSS_CLASSES#`
- `#REGION_ATTRIBUTES#`
- `#REGION_HEADER#`
- `#REGION_FOOTER#`
- `#REGION_IMAGE#`
- `#REGION_IMAGE_URL#`
- `#REGION_IMAGE_ALT_TEXT#`
- `#REGION_IMAGE_ATTRIBUTES#`

The region positions are used to substitute other components inside the region. They are defined in the Positions section of the region template. This is where you will find region positions such as `#BODY#`, `#SUB_REGIONS#` and other custom positions.

> **See Also:**
>
> Using Substitution Strings

11.3.17.1.3 Layout

Page items can be displayed within regions. If the page template layout uses HTML table, each item in a region is part of an HTML table. You can set the attributes of this table using this attribute. Note that this section is not used if the page template layout uses a fixed grid system.

11.3.17.1.4 Sub Regions

Header Template

Sub region header templates together with sub region header entry templates provide the possibility to generate a list of region titles of all the sub regions of the current region.

`#ENTRIES#` is the only required substitution string. It identifies where the sub region header entries should be placed. All other substitution strings are optional.

The following are valid substitution strings: `#ENTRIES#`, `#REGION_ID#`, and `#REGION_STATIC_ID#`.Use `#SUB_REGION_HEADERS#` in the region template to place the output of this template.

Header Entry Template

Sub region header entry templates together with the sub region header templates provide the possibility to generate a list of region titles of all the sub regions of the current region.

The following are valid substitution strings: `#SUB_REGION_ID#`, #`SUB_REGION_TITLE#`, `#REGION_ID#`, and `#REGION_STATIC_ID#`. Use `#ENTRIES#` in the sub region header template to place the output of this template.

Template

Sub Region templates provide the possibility to wrap the sub regions with additional HTML code. `#SUB_REGION#` is the only required substitution string. It identifies where the source of the sub region should be placed. All other substitution strings are optional.

The following are valid substitution strings: `#SUB_REGION#`, `#SUB_REGION_ID#`, `#SUB_REGION_TITLE#`, `#REGION_ID#`, and `#REGION_STATIC_ID#`. Use `#SUB_REGIONS#` in the region template to place the output of this template.

11.3.17.1.5 Image

Template

The Image template provides the possibility of defining a region image with additional HTML code.

`#REGION_IMAGE_URL#` is the only required substitution string and should be used within an img HTML tag. It specifies the img src attribute. All other substitution strings are optional.

The following are valid substitution strings:

- `#REGION_IMAGE_URL#`
- `#REGION_IMAGE_ALT_TEXT#`
- `#REGION_IMAGE_ATTRIBUTES#`
- `#REGION_ID#`

- `#REGION_STATIC_ID#`

Use `#REGION_IMAGE#` in the region template to place the output of this template.

11.3.17.1.6 JavaScript

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

Available attributes include:

- **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

  JavaScript file URLs you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

  Do not include opening or closing script tags, just write the URL.
- **Execute when Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle APEX.

  Do not include opening or closing script tags, just include the JavaScript code.

11.3.17.1.7 Cascading Style Sheet

File URLs

Enter Cascading Style Sheet file URLs to be loaded for this page template. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

You do not need to include opening or closing `link` tags. Just include the file URL.

11.3.17.1.8 JavaScript

> **Tip:**
>
> For a list of supported substitution strings and to view examples, see field-level Help.

Available attributes include:

- **File URLs** - Enter JavaScript file URLs for code to be loaded on this page. Each URL has to be written into a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include `.min` or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view and an empty string if the page is viewed in debug mode.

You also have access to the substitution string `#APP_VERSION#` if you want to include the application's version in the file URL.

JavaScript file URLs you enter here replaces the `#TEMPLATE_JAVASCRIPT#` substitution string in the page template.

Do not include opening or closing script tags, just write the URL.

- **Execute when Page Loads** - Enter JavaScript code to execute when the page loads. The code is executed after the JavaScript code generated by Oracle APEX.

  Do not include opening or closing script tags, just include the JavaScript code.

11.3.17.1.9 Substitution Strings

Substitution strings are used within templates to reference component values. To view the most current list supported substitution strings for a given template, view the template and find the **Substitution Strings** report.

> **Tip:**
>
> All template substitution strings must be in uppercase letters and begin and end with a number sign (#).

11.3.18 Report Templates

Report column templates provide you with control over the results of a row from a SQL query. This type of template defines a cell, not an entire row.

Each report template identifies column names using the syntax `#1#, #2#, #3#`, and so on. You can also name columns using column name substitution syntax such as `#ENAME#` or `#EMPNO#`. You can reference any item from your application within your template. For example, to reference an item called *`ABC`*. In your template, you could include the exact substitution string *`&ABC`*.. The actual value of ABC. would be provided by an end user editing an item in your application named *`ABC`*.

- Generic Column Templates and Named Column Templates
  Oracle APEX includes two types of report templates: Generic Column Templates and Named Column Templates.
- Report Column Template Attributes for Generic Column Templates
  Learn about report column template attributes for generic column templates.
- Report Column Template Attributes for Named Column Templates
  Learn about report column template attributes for named column templates.
- About Using JavaScript in Column Templates
  Learn about about using JavaScript in column templates.

11.3.18.1 Generic Column Templates and Named Column Templates

Oracle APEX includes two types of report templates: Generic Column Templates and Named Column Templates.

- Generic Column Templates
  Learn about generic column templates.

- Named Column Templates
  Learn about named column templates.

11.3.18.1.1 Generic Column Templates

Learn about generic column templates.

A generic column template determines the appearance of a report by defining the look of the column once. This look is then repeated as many times as is necessary, based on the number of columns specified in the report's definition. This type of template is limited to reports that have a standard row and column structure. Additional style can be applied to a report using this type of template through the use of conditions.

The following example demonstrates how to have each column use a specific style:

```
<td class="tabledata" align="#ALIGN#">#COLUMN_VALUE#</td>
```

This example assumes your page template includes a CSS containing the class `tabledata`. This example also demonstrates the use the substitution strings `#ALIGN#` and `#COLUMN_VALUE#`. If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

If your query uses an expression in the select list, it is a good idea to create an alias for the columns to avoid runtime errors. For example, suppose your query was as follows:

```
SELECT ename, (sal + comm) * 12 FROM emp
```

You could rewrite the query to alias the columns as follows:

```
SELECT ename, (sal + comm) * 12 yearly_comp FROM emp
```

11.3.18.1.2 Named Column Templates

Learn about named column templates.

Named column templates allow for more flexibility in report design. However, because they reference columns by name, they can only be used by reports that are based on those columns. For example:

```
<tr><td>#ENAME#</td><td>#SAL#</td></tr>
```

Although named column templates offer flexibility, you may need to create a new template for each query. You can also include a position notation. The following example demonstrates how to use following HTML and substitution strings:

```
<tr><td>#ENAME#</td><td>#SAL#</td></tr>

<tr><td>#1#</td><td>#2#</td></tr>
```

11.3.18.2 Report Column Template Attributes for Generic Column Templates

Learn about report column template attributes for generic column templates.

This topic describes specific sections of the Edit Report Template page for Generic Column Templates. To learn more about a specific attribute, see field-level Help.

- Report Template
- Before Rows
- Column Headings

- Before Each Row
- Column Templates
- After Each Row
- After Rows
- Row Highlighting
- Pagination Subtemplate
- Comments

11.3.18.2.1 Report Template

Template Name identifies the name of the template. **Template Type** indicates the type of template. Named Column templates reference column names in the template. Generic Column Templates reference the `#COLUMN_VALUE#` substitution string in the template.

Theme indicates the theme to which the template is a member. **Template Class** identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class. Use the **Translatable** checkbox to indicate the template contains text strings that require translation.

11.3.18.2.2 Before Rows

In **Before Rows**, enter HTML that displays once at the beginning of a report template.

Opening an HTML table is a common use of this attribute as shown in the following example:

```
<table>
```

You can include pagination above a report by including the substitution string `#TOP_PAGINATION#`. This substitution string generates HTML that starts with an opening `<tr>` tag and ends with a closing `</tr>` tag. For example, to include an open table tag and the `#TOP_PAGINATION#` substitution string, you would enter the following:

```
<table>#TOP_PAGINATION#
```

You can also include the substitution string `#CSV_LINK#` to include support for exporting your report to comma-separated value (CSV) format, a format compatible with most spreadsheet programs.

You can also use the substitution string `#REPORT_ATTRIBUTES#`. This substitution string is set per report region on the report attributes page.

11.3.18.2.3 Column Headings

Use **Before Column Heading** to display text before report heading.

Use **Column Heading Template** to colorize each column header cell. The text of this attribute must include Help to indicate where the cell heading text should be colorized. For example:

```
<th >Item Help Text</th>
```

If you do not want any column headings, enter the following:

```
OMIT
```

If you do use this attribute, APEX engine applies the default column heading template.

Use **After Column Heading** to display text after report headings.

11.3.18.2.4 Before Each Row

In **Before Each Row,** enter text to display before all columns in the report. Use this attribute to open a new HTML row. Before Each Row supports the following substitution strings:

- `#ROWNUM#`

  Use this substitution string to specify the current row.

- `#COLCOUNT#`

  Use this substitution string to specify the number of columns.

- `#HIGHLIGHT_ROW#`

  Use this substitution string to specify the number of highlighted rows.

11.3.18.2.5 Column Templates

Column templates define the look of each column. You can define up to four column templates; each can be conditional. For example, you can have different background colors for even and odd rows, or highlight rows that meet a PL/SQL defined condition.

In each Column Template, you define the look of each column. Column Templates support the substitution strings described in Table 11-10.

Table 11-10 Column Template Substitution Strings

Substitution String	Description
`#ALIGNMENT#`	Determines the column alignment. Specified by the user.
`#COLCOUNT#`	Defines the count of the number of columns.
`#COLNUM#`	Defines the current column number.
`#COLUMN_HEADER#`	Defines the column header.
`#COLUMN_VALUE#`	Replaced with the value of the column.
`#ROWNUM#`	Specifies the current row number.

Consider the following example:

```
<td> #ALIGNMENT#>#COLUMN_VALUE#</td>
```

If you actually ran this report, these substitution strings would be replaced with values generated by the results of a SQL query.

By creating conditions, you can create a report that displays columns differently depending on whether the specified condition is met. To specify a column template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.
- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.
- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

```
#SAL# > 2000
```

You could also use the substitution string `#ROWNUM#`. For example:

```
#ROWNUM# > 2000
```

11.3.18.2.6 After Each Row

In **After Each Row,** enter HTML that displays after all columns in the report display. This attribute is often used to close an HTML table row. For example:

```
</tr>
```

11.3.18.2.7 After Rows

Use **After Rows** to specify text that should display after the last row. A common use of this attribute is to close the HTML table tag. For example:

```
</table>
```

The After Rows attribute supports the following substitution strings:

- `#PAGINATION#`

  Replaced with a pagination attribute.

- `#COLCOUNT#`

  Substituted at runtime with the number of columns defined in the report.

- `#CSV_LINK#`

  Substituted at runtime with CSV download link if CSV export is enabled. Otherwise the string is not shown.

- `#EXTERNAL_LINK#`

  Substituted at runtime with the download link for exports using external processing engine for (for example, PDF export). If this feature is not enabled, the string is not shown.

11.3.18.2.8 Row Highlighting

Background color for checked row controls the background color of a report row when the row selector is checked. **Background color for current row** controls the background color of a report row when the user moves the mouse over the row.

11.3.18.2.9 Pagination Subtemplate

The Pagination Subtemplate section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings `#PAGINATION_NEXT#`, `#PAGINATION_NEXT_SET#`, `#PAGINATION_PREVIOUS#` and `#PAGINATION_PREVIOUS_SET#`. The following table describes these templates.

Pagination Subtemplate Attribute	Description
Pagination Template	Applies to the entire pagination subtemplate. For example: `<span class="instructiontext">#TEXT#</span>` Use the substitution string `#TEXT#` to specify where you want the pagination subtemplate to display. Use the other Pagination Subtemplate attributes to modify individual items.
Next Page Template	Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example: `<a href="#LINK#">next</a>`
Previous Page Template	Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example: `<a href="#LINK#">previous</a>`
Next Set Template	Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example: `<a href="#LINK#">next set</a>`
Previous Set Template	Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example: `<a href="#LINK#">previous set</a>`

11.3.18.2.10 Comments

Use this attribute to record comments about this component.

11.3.18.3 Report Column Template Attributes for Named Column Templates

Learn about report column template attributes for named column templates.

This topic describes specific sections of the Edit Report Template page for Named Column Templates. You can access the sections of the page by either scrolling down the page, or by clicking a navigation button at the top of the page. When you select a button at the top of the page, the selected section appears and all other sections are temporarily hidden. To view all sections of the page, click **Show All**.

- Name
- Subscription
- Row Templates
- Column Headings
- Before first and after last row text
- Pagination
- Comments

11.3.18.3.1 Name

Template Name identifies the name of the template. **Template Type** indicates the type of template. Named Column templates reference column names in the template. Generic Column Templates reference the `#COLUMN_VALUE#` substitution string in the template.

Theme indicates the theme to which the template is a member. Use the **Translatable** checkbox to indicate the template contains text strings that require translation. **Template Class** identifies a specific use for the template. When you switch to a new theme, all templates in one theme are mapped to corresponding templates in another theme. App Builder accomplishes this template mapping through the assignment of a template class.

11.3.18.3.2 Subscription

Use Subscription to apply an existing template to the current application. When you select an existing template, you become a subscriber to that template.

To load a new copy of a master template, click **Refresh**.

11.3.18.3.3 Row Templates

Row templates define the look of each column. You can define up to four row templates, each of which can be conditional.

In each Row Template, you define the look of each row. The following table describes supported substitution strings in row templates.

Substitution String	Description
`#ALIGNMENT#`	Determines the row alignment. Specified by the user.
`#COLCOUNT#`	Defines the count of the number of columns.
`#COLNUM#`	Defines the current column number.
`#COLUMN_HEADER#`	Defines the column header.
`#COLUMN_VALUE#`	Replaced with the value of the column.
`#ROWNUM#`	Specifies the current row number.

By creating conditions, you can create a report that displays rows differently depending on whether the specified condition is met. To specify a row template be used conditionally, select a condition type from the Column Template Condition list. Valid values include:

- **Use Based on PL/SQL Expression.** Conditionally formats columns based on data in that row.
- **Use for Even Numbered Rows.** Conditionally formats even numbered rows.
- **Use for Odd Numbered Rows.** Conditionally formats odd numbered rows.

If you select **Use Based on PL/SQL Expression**, the next step is to enter a PL/SQL expression in the Column Template Expression field. For example, the following expression displays a value in bold if the value is greater than 2000:

```
#SAL# > 2000
```

You could also use the substitution string `#ROWNUM#`. For example:

```
#ROWNUM# > 2000
```

11.3.18.3.4 Column Headings

Use this template to add color to each column header cell. The text of this attribute must include help to indicate where the cell heading text should be colorized. If you do not enter a Column Heading Template, then a default column header template is applied. If you do not want any column headings, then enter `OMIT`. For example:

```
<th #ALIGNMENT#>#COLUMN_HEADER#</th>
```

11.3.18.3.5 Before first and after last row text

In **Before Rows**, enter HTML that displays once at the beginning of a report template. Opening an HTML table is a common use of this attribute, as shown in the following example:

```
<table>
```

You can identify column headers using the syntax `#1#`, `#2#`, `#3#`. For example:

```
<th>#1#</th><th>#2#</th><th>#3#</th>
```

You can include pagination above a report by including the substitution string `#TOP_PAGINATION#`. This substitution string generates HTML that starts with an opening `<tr>` tag and ends with a closing `</tr>` tag. For example, to include an open table tag and `#TOP_PAGINATION#` substitution string, you would enter the following:

```
<table>#TOP_PAGINATION#
```

You can also include the substitution string `#CSV_LINK#` to include support for exporting your report to CSV format, a format compatible with most spreadsheet programs.

Use **After Rows** to specify text that should display after the last row. A common use of this attribute is to close the HTML table tag. For example:

```
</table>
```

The After Rows attribute supports the following substitution strings:

- `#PAGINATION#`

  Replaced with a pagination attribute.

- `#COLCOUNT#`

  Substituted at runtime with the number of columns defined in the report.

11.3.18.3.6 Pagination

The Pagination section contains attributes for editing the Pagination Template, Next Page Template, Previous Page Template, Next Set Template, and Previous Template. Pagination Subtemplates support the substitution strings `#PAGINATION_NEXT#`, `#PAGINATION_NEXT_SET#`, `#PAGINATION_PREVIOUS#` and `#PAGINATION_PREVIOUS_SET#`. The following table describes these templates.

Pagination Subtemplate Attribute	Description
Pagination Template	Applies to the entire pagination subtemplate. For example: `<span class="instructiontext">#TEXT#</span>` Use the substitution string `#TEXT#` to specify where you want the pagination subtemplate to display. Use the other Pagination Subtemplate attributes to modify individual items.
Next Page Template	Enter HTML to modify how the Next Page portion of the pagination subtemplate appears. For example: `<a href="#LINK#">next</a>`
Previous Page Template	Enter HTML to modify how the Previous Page portion of the pagination subtemplate appears. For example: `<a href="#LINK#">previous</a>`
Next Set Template	Enter HTML to modify how the Next Set portion of the pagination subtemplate appears. For example: `<a href="#LINK#">next set</a>`
Previous Set Template	Enter HTML to modify how the Previous Set portion of the pagination subtemplate appears. For example: `<a href="#LINK#">previous set</a>`

11.3.18.3.7 Comments

Use this attribute to record comments about this component.

11.3.18.4 About Using JavaScript in Column Templates

Learn about about using JavaScript in column templates.

You can conditionally display HTML depending upon values in the database using JavaScript. The following example displays an HTML row only if the `GROUP_DESC` query column is not null:

```
<script language="javascript">
IF ( "#GROUP_DESC#" != "" )
document.writeln( "<TR>;
<TD BGCOLOR=#336699>;</TD>
</TR>
</TR>
<TD>#GROUP_DESC#</TD>
              </TR>" );
    </TR>" );
```

See Also:

About Regions

11.4 Using Custom Cascading Style Sheets

Control the style of a web page by uploading and referencing a cascading style sheet.

- About Cascading Style Sheets
  A cascading style sheet (CSS) enables a developer to control the style of a web page without changing its structure.
- Referencing an Uploaded Cascading Style Sheet in the Page Template
  Reference an uploaded cascading style sheet within the Header section of the page template.

11.4.1 About Cascading Style Sheets

A cascading style sheet (CSS) enables a developer to control the style of a web page without changing its structure.

When used properly, a CSS separates visual attributes such as color, margins, and fonts from the structure of the HTML document. Oracle APEX includes themes that contain templates that reference their own CSS. The style rules defined in each CSS for a particular theme also determine the way reports and regions display.

> **Tip:**
>
> If your application uses *Universal Theme - 42*, you can also take advantage of theme styles and update the appearance of your application using Theme Roller. See Using Theme Styles and Using the Theme Roller Dialog.

About Uploading Cascading Style Sheets

You can upload files (including CSS files) for use by a specific application or all applications within a workspace. To learn more, see Managing Static Application Files and Managing Static Application Files.

11.4.2 Referencing an Uploaded Cascading Style Sheet in the Page Template

Reference an uploaded cascading style sheet within the Header section of the page template.

Use the Header section to enter the HTML that makes up the <HEAD> section of the HTML document.

To reference an uploaded cascading style sheet:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.

2. `Select a theme.`
3. On the Tasks list, click **View Templates**.
4. Select the name of the page template you want to edit.
5. Use a `<link>` tag within the Header section to reference the appropriate style sheet.

   To reference an uploaded file that is associated with a specific application, you would use the substitution string `#APP_FILES#`. For example:

```
<html>
<head>
    <title>#TITLE#</title>
    #HEAD#
    <link rel="stylesheet" href="#APP_FILES#sample2.css" type="text/css">
</head>
...
```

   To reference an uploaded file that is associated with a specific workspace, you would use the substitution string `#WORKSPACE_FILES#`. For example:

```
<html>
<head>
    <title>#TITLE#</title>
    #HEAD#
    <link rel="stylesheet" href="#WORKSPACE_FILES#sample3.css" type="text/css">
</head>
...
```

See Also:

- About Cascading Style Sheets
- Creating a New Template
- Managing Templates
- Page Templates
- APP_FILES
- WORKSPACE_FILES

12
Controlling Page Layout

Developers can customize their application's page layout by using a global page, customizing regions, and editing item attributes.

This section also describes how to manage images and static files as well as explains how to render HTML using custom POL/SQL.

- About Page Layout in Oracle APEX
  The Oracle APEX engine renders pages by combining templates with application components defined by the developer and data in the database.
- Optimizing a Page for Printing
  You can optimize a page for printing by creating a specific Printer Friendly template and selecting that template on the Create/Edit Theme page.
- Creating a Global Page to Display Components on Every Page
  A global page (previously referred to as Page 0) functions as a master page.
- Managing Regions
  Regions function as containers for content.
- About Incorporating Content from Other Websites
  To incorporate content from other servers, you can create a region based on a URL to display content.
- About Managing Images
  To reference an image within an application, you must upload it to the static file repository. During the upload process, you specify whether the file is available to all applications or just a specific application.
- Rendering HTML Using Custom PL/SQL
  If you must generate specific HTML content not handled by Oracle APEX forms, reports, and charts, you can use the PL/SQL region type.

See Also:

- Adding Navigation
- Editing User Interface Attributes
- Using Themes and Theme Styles

12.1 About Page Layout in Oracle APEX

The Oracle APEX engine renders pages by combining templates with application components defined by the developer and data in the database.

The overall framework (or structure of a page) is determined by the page template. For example, the page template controls if a page uses tabs and a navigation bar. It can also define if a page includes a bar on the left side that serves as a placeholder for navigation or

secondary content. Finally, a page template can include definitions of region positions, which enable precise control over placement of regions using HTML tables or style sheet definitions. The page template itself is composed of HTML combined with substitution strings, which are substituted with the appropriate components at runtime.

As a developer, you add content to a page by creating a region. A **region** is an area of a page that serves as a container for content. Each region contains a different type of content such as HTML, a report, a form, a chart, a list, a breadcrumb, PL/SQL, a tree, a URL, or a calendar. You position a region either relative to other regions (that is, based on its sequence number and column), or by using a region position defined in the page template. The style of the region is also controlled by the region template. Like the page template, the region template defines the structure of the area that the region takes up on a page. It defines if the region title is displayed and where it is displayed relative to the main content or the body. A region can also define absolute positions for buttons.

See Also:

- Using Component Groups
- Using Themes and Theme Styles

12.2 Optimizing a Page for Printing

You can optimize a page for printing by creating a specific Printer Friendly template and selecting that template on the Create/Edit Theme page.

Generally, a Printer Friendly template optimizes a page for printing. For example, a Printer Friendly template might:

- Not display tabs or navigation bars
- Display items as text instead of as form elements

If the theme you select does not include a printer friendly template, you can create a Printer Friendly template by creating a new page template.

- Selecting a Printer Friendly Template for an Application
  Learn about selecting a printer friendly template for an application.
- Using f?p Syntax to Toggle to Printer Friendly Mode
  Learn about using `f?p` syntax to toggle to Printer Friendly mode.

See Also:

Editing Themes and Accessing New Themes

12.2.1 Selecting a Printer Friendly Template for an Application

Learn about selecting a printer friendly template for an application.

To select a Printer Friendly template:

1. Navigate to the Themes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under User Interface, select **Themes**.

   The Themes page appears.
2. Select a theme.
3. Scroll down to **Component Defaults**.
4. Make a selection from the **Printer Friendly Page** list.
5. Confirm your changes and click **Apply Changes**.

> **See Also:**
>
> Editing Themes

12.2.2 Using f?p Syntax to Toggle to Printer Friendly Mode

Learn about using `f?p` syntax to toggle to Printer Friendly mode.

Once you create a Printer Friendly template and select it, you can use `f?p` syntax to toggle to Printer Friendly mode. Including the ninth `f?p` syntax argument (`PrinterFriendly`) renders the page in printer friendly mode (optimizing printed output). For example, you could include this argument when coding a link or creating a navigation bar icon.

> **See Also:**
>
> About Using f?p Syntax to Link Pages

12.3 Creating a Global Page to Display Components on Every Page

A global page (previously referred to as Page 0) functions as a master page.

- Creating a Global Page
  Create a Global page using the Create Page Wizard.
- Viewing or Editing a Global Page
  View or edit a Global page in Page Designer.

See Also:

- Managing Pages in an Application
- Understanding Conditional Rendering and Processing
- Available Conditions

12.3.1 Creating a Global Page

Create a Global page using the Create Page Wizard.

Developers can create a separate Global page for each user interface. The Oracle APEX engine renders all components you add to a Global page on every page within your application. You can further control whether the APEX engine renders a component or runs a computation, validation, or process by defining conditions.

To create a Global page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Global Page**.

   Tip:

   The **Global Page** option only appears if the application does not have a Global page..
5. Page Number - The page number is an integer value that identifies a page within an application.
6. Click **Create Page**.

12.3.2 Viewing or Editing a Global Page

View or edit a Global page in Page Designer.

To view or edit a Global page:

1. On the Workspace home page, click the **App Builder** icon.

   The App Builder home page appears.
2. Select an application.

   The Application home page appears.
3. Select **Global Page** (or **Page Zero** if this is an older application).

   The page displays in Page Designer.

12.4 Managing Regions

Regions function as containers for content.

- About Regions
  Learn how regions function as a container for content on an application page.
- Creating a Region in Page Designer
  Create a new region manually in Page Designer from either the Rendering tab or the Gallery.
- Editing Regions
  Alter the appearance of a page by editing Region attributes.
- Copying a Region
  Copy a region in Page Designer.
- Deleting Regions
  Delete regions on Page Designer.

12.4.1 About Regions

Learn how regions function as a container for content on an application page.

Each page in an Oracle APEX application contains one or more regions. A region is an area on a page that serves as a container for content. Each page can have any number of regions. You control the appearance of a region through a specific region template. The region template controls the look of the region, the size, determines whether there is a border or a background color, and what type of fonts display. A region template also determines the standard placement for any buttons placed in region positions.

You can use regions to group page controls (such as items or buttons). You can create simple regions that do not generate additional HTML, or create elaborate regions that frame content within HTML tables or images.

Regions display in sequence in the page template body or can be placed explicitly into region positions. The page structure can be defined using HTML tables or more commonly today, using `DIV` tags.

12.4.2 Creating a Region in Page Designer

Create a new region manually in Page Designer from either the Rendering tab or the Gallery.

A region is a container for content. When you create an application or page using a wizard, the wizard creates the appropriate region for you. You can also create regions manually in Page Designer.

- Creating a Region from the Rendering Tab
  Add a region from the Rendering tab by right-clicking a component or container to display a context menu.
- Creating a Region from the Gallery
  Add a region by clicking the Gallery, Region tab and right-clicking a region to display a context menu.
- Supported Region Types
  Learn about supported region types in Page Designer.

12.4.2.1 Creating a Region from the Rendering Tab

Add a region from the Rendering tab by right-clicking a component or container to display a context menu.

The Rendering tab displays in the the left pane of Page Designer and displays all components that are rendered on a page.

To create region from the Rendering tab:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In the left pane, select the **Rendering** tab.
3. Right-click an existing container to display the context menu.
4. From the context menu, select **Create Region**.
5. In the Property Editor, edit the appropriate attributes:
   a. Title - Enter a region title. The region title only displays when it is defined in the region template.
   b. Type - Specify how the region is rendered.

      To learn more about supported region types, see Page Designer Help or Supported Region Types.

      > **Tip:**
      >
      > To select a legacy region, select **Show Legacy...** from the Type list. Legacy region types appear in the list. Although Oracle still supports legacy application components, Oracle does not recommend using them.

   c. Edit the remaining attributes. The attributes that display depend upon the region Type.

      To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
6. Click **Save** or **Save and Run Page**.

12.4.2.2 Creating a Region from the Gallery

Add a region by clicking the Gallery, Region tab and right-clicking a region to display a context menu.

The Gallery displays beneath the Layout tab in the central pane of Page Designer. The Gallery contains three tabs: Regions, Items, and Buttons.

To create a region from the Gallery:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

Page Designer appears.

2. In Page Designer, click the **Layout** tab in the central pane.
3. In the Gallery at the bottom of the central pane:
   a. Click the **Region** tab.
   b. Right-click a region to view the context menu.

   > **Note:**
   >
   > To view legacy components, right-click a region and select **Show Legacy Components**. Legacy components appear. Although Oracle still supports legacy application components, Oracle does not recommend using them.

   c. Select **Add To** and then the location where you want to add the component.

   Page Designer indicates what actions are required next. If there is an error, a **Messages** icon displays in the Page Designer toolbar. Click the **Messages** icon to view a dialog listing all the errors. Select an error to highlight the associated attribute in the Property Editor.
4. Edit the appropriate attributes in the Property Editor.
   a. Title - Enter a region title. The region title only displays when it is defined in the region template.
   b. Type - Specify how the region is rendered.

      To learn more about supported region types, see Page Designer Help or Supported Region Types.

      > **Tip:**
      >
      > To select a legacy region, select **Show Legacy...** from the Type list. Legacy region types appear in the list. Although Oracle still supports legacy application components, Oracle does not recommend using them.

   c. Edit the remaining attributes. The attributes that display depend upon the region Type.

      To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. Click **Save** or **Save and Run Page**.

> **Tip:**
>
> You can also drag and drop components from the Gallery. Click and hold the mouse on the region to be created and drag it the desired location in the Layout tab. When the mouse is over the appropriate location, the Layout tab displays as a darkened yellow tile. Release the mouse to drop the component. You can only drop components into appropriate drop positions, as determined by the component type.

12.4.2.3 Supported Region Types

Learn about supported region types in Page Designer.

The region Type determines how a region is rendered at runtime. When creating a region manually, Page Designer supports the following region types:

Region Type	Description	To Learn More
Breadcrumb	Displays the breadcrumb trail for the current page.	See Creating Breadcrumbs.
Calendar	Displays date based entries on a variety of calendar views. With the inclusion of an end date, the calendar can display duration based events. Calendars can also update records directly if **Drag and Drop** is enabled..	See Creating Calendars.
Cards	Creates a report based on a SQL Query that displays data as an orderly layout of information tiles. Developers choose between three layouts: Grid, Float, or Horizontal (Row). You can customize Header, Media, Body or Actions declaratively.	See Managing Cards.
Chart	Displays data using various built-in chart types.	See Creating Charts.
Classic Report	Creates a simplified report based on a SQL Query. Classic reports are similar to interactive reports, but do not include the numerous end-user customization options available with interactive reports.	See Managing Classic Reports.
Column Toggle Report	Displays a report where end users can select the columns to display at runtime. End users can then change the columns based on the width and orientation of the screen they are using to display the report.	See Creating a Legacy Column Toggle Report.
Dynamic Content	Dynamic Content - Displays the HTML content returned by a function.	n/a
Faceted Search	Displays and filters data using an intuitive experience users recognize from ecommerce sites (a left Search region and report region). Displays search results as cards or a classic report.	See Managing Faceted Search.
Form	Form region which supports different data sources. This form region holds information on the data source (which can be local, remote or REST service) and has page items assigned to it. Each item is assigned to a form region and a display region.	See Developing Forms.
Help Text	Displays a special Help region utilized on the dedicated Help page.	See Creating Help for Your Application.

Region Type	Description	To Learn More
Interactive Grid	Presents data in a searchable, customizable report. Includes similar customization capabilities available in interactive reports plus the ability to rearrange the report interactively using the mouse or keyboard. Defines a single interactive grid and have end users manipulate the data displayed to meet their specific requirements, rather than you needing to develop several, similar regions with different criteria to meet those same requirements.	See Managing Interactive Grids.
Interactive Report	Comprehensive online report based on a SQL Query that provides end users with numerous features for customizing the report layout at run-time. You can control end-user capabilities by enabling or disabling the various report customization options. Define a single interactive report and have end users manipulate the data displayed to meet their specific reporting requirements, rather than you needing to develop several, similar reports with different criteria to meet those same requirements.	See Managing Interactive Reports.
List	Displays values based on either a static or dynamic list. Generally these are used for navigating to pages in the application.	See Creating Lists.
List View	Displays data in an easy to navigate list. You can also incorporate numerous features to enhance the list view.	See Creating a Legacy List View Report.
Map	Report based on a SQL query that displays coordinates data as a map. Supports point, lines, and polygon data.	See Creating Maps.
Reflow Report	Displays a responsive report which transforms from columns to rows to display reports on smaller browser (screen) sizes.	See Creating a Legacy Reflow Report.
Region Display Selector	Displays region names in a horizontal list, enabling end users to select one region to display and hide other regions. Only page regions with **Region Display Selector** set to **Yes** will display in this region.	n/a
Search	Display Search Results, based on Search Configurations, which are defined within Shared Components.	See Adding Search to an Application.

Region Type	Description	To Learn More
Smart Filters	Features a single search field at the top of the page and a search results report (classic report, cards, map, or calendar). While a smart filter behaves similarly to faceted search, it features a more space efficient layout.	See Managing Smart Filters.
Static Content	Displays text content or a region container for page items. To output HTML markup and not have the content escaped, set Output As to HTML.	n/a
Tree	Displays data in a hierarchical structure, based on parent-child relationships between records.	See Managing Trees.
URL	Display remote content sourced from a URL.	n/a
Workflow Diagram	Displays a read-only visual representation of the workflow instance.	See About the Workflow Console, Workflow Details, and Workflow Dashboard.

Depending on your application configuration, you may also have additional region type plug-ins available here. The following plug-ins are installed within this application's shared components:

- Avatar - Displays an image, icon, or initials. Available as a partial template to display a single row.
- Badge - Displays content within a badge. Available for multiple report rows or a single row.
- Comments - Displays user comments and status updates. Supports avatars. Available for a single row or as a report with multiple rows.
- Content Row - Displays content in a formatted row with a title, description, and more. Supports avatars and badges. Available for a single row or as a report with multiple rows.
- Media List - Displays report content in a formatted media list. Supports avatars and badges. Available for a single row or as a report with multiple rows.
- Sample Apps Footer - A region plug-in used to display a custom footer at the bottom of pages with large icons for navigating to other sites such as twitter and linkedin.

  Note: This plug-in should be customized to meet your specific requirements, rather than used as is.
- Source Display - A region plug-in used to display the SQL Source region of an accompanying region.
- Timeline - Displays a series of events. Supports avatars and badges. Available for a single row or as a report with multiple rows.

> **Note:**
>
> Depending on your application configuration, additional region type plug-ins available may display in your environment with the naming convention, *<My Plug-in> [Plug-in]*.

12.4.3 Editing Regions

Alter the appearance of a page by editing Region attributes.

> **Note:**
>
> Edit the region attributes in the Accessibility group to help users of assistive technology to better navigate APEX pages. To learn more, see Regions in *Oracle APEX Licensing Information User Manual*.

- Editing Region Attributes
  Edit Region attributes in Page Designer.
- Updating the Region Name or Title
  Identify a region by editing the Identification, Name and Title attributes.
- Controlling Region Positioning
  Use the Layout, Position attribute to select the template position for this region.
- Specifying a Region Header and Footer
  Edit the Header and Footer attributes to specify additional HTML to be placed above and below a region.
- Enabling Region Caching
  Enable region caching to improve the performance of static regions such as regions containing lists that do not use conditions or regions containing static HTML.
- Enabling Users to Customize a Page
  Use the Customizable attribute to control whether display attributes in a region can be customized by users.
- Specifying a Static Region ID
  Specifying a static region ID is useful when creating custom JavaScript or cascading style sheets.
- Adding a Region Image
  Use Region Image and Image Tag Attributes to add an image to the upper left side of a region.
- Creating a Region Display Selector
  Create a Region Display Selector region to include show and hide controls for each region on a page. to include the current region in the Region Display Selector.

12.4.3.1 Editing Region Attributes

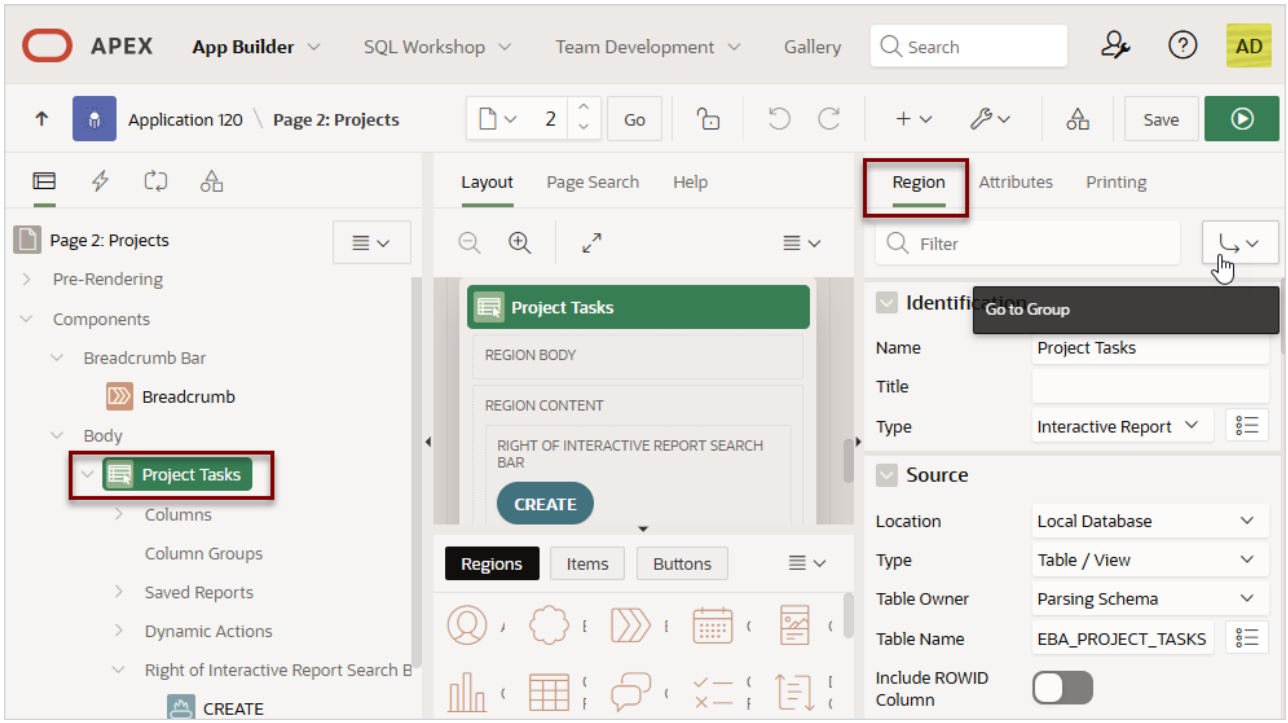
Edit Region attributes in Page Designer.

To edit region attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

The Property Editor displays attributes in the **Region** tab in the right pane.



To find a group or attribute:

- **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
- **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

3. Edit the Region attributes in the Property Editor.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Right Pane of Page Designer (Property Editor)

12.4.3.2 Updating the Region Name or Title

Identify a region by editing the Identification, Name and Title attributes.

Use the **Identification**, **Name** attribute to specify the region name. The **Name** attribute is required and used by Page Designer to identify the region in the Rendering and Layout tabs.

Use the **Identification**, **Title** attribute to specify a region title. The region title only displays at runtime if it is defined in the region template. If the **Title** attribute is empty, the APEX engine uses the region **Name** at runtime.

12.4.3.3 Controlling Region Positioning

Use the Layout, Position attribute to select the template position for this region.

Configure the **Layout**, **Position** attribute to select the position used to display a region. The positions are derived from the appropriate page or region template positions.

12.4.3.4 Specifying a Region Header and Footer

Edit the Header and Footer attributes to specify additional HTML to be placed above and below a region.

In the **Header Text** attribute, enter HTML text to display at the top of the region above the `#BODY#` content.

In the **Footer Text** attribute, enter a region footer that displays directly after the body content, after `#BODY#` in the region template. You may use shortcuts as well as the substitution string `#TIMING#` which displays the time in seconds it took to render the region. You can use this substitution string for debugging purposes.

For classic report regions, the region footer supports the following substitution strings:

- `#ROWS_FETCHED#` shows the number of rows fetched by the Oracle APEX reporting engine (the page size). You can use these substitution strings to display customized messages to the user. For example:

  ```
  Fetched #ROWS_FETCHED# rows in #TIMING# seconds
  ```

- `#TOTAL_ROWS#` displays the total number of rows that satisfy a SQL query used for a report. For example:

  ```
  Z in rows X - Y of Z
  ```

- `#FIRST_ROW_FETCHED#` and `#LAST_ROW_FETCHED#` display the range of rows displayed. For example:

  ```
  Row(s) #FIRST_ROW_FETCHED# through #LAST_ROW_FETCHED# of #ROWS_FETCHED#
  displayed
  ```

12.4.3.5 Enabling Region Caching

Enable region caching to improve the performance of static regions such as regions containing lists that do not use conditions or regions containing static HTML.

When you enable region caching, the APEX engine renders a region from a cached (or stored) repository instead of rendering it dynamically. Keep in mind that the actual session identifiers are not cached. Instead, the APEX engine caches a `&SESSION.` substitution string and the current session rendering the cached region is substituted on display. For example, if a region contains a link and the link includes a session, the exact session is not cached to ensure that the links works for all sessions.

The APEX engine only renders a region from cache if it meets the defined condition. Additionally, regions can be cached specific to a user or cached independent of a user.

To enable region caching:

1. View the page in Page Designer:

a. On the Workspace home page, click the **App Builder** icon.
b. Select an application.
c. Select a page.

Page Designer appears.

2. In the Rendering tab, select the region.

   The Property Editor displays attributes in the **Region** tab in the right pane.

3. Server Cache, Caching - Select one of the following:
   - **Disabled** - Content is not cached and computed for each request.
   - **Enabled** - Content is cached and utilized by all users.
   - **Cached by User** - Content is cached specifically for each user.
   - **Cache by Session** - Content is cached specifically for each session.
4. Click **Save**.

> **See Also:**
>
> Managing Page and Region Caching in *Oracle APEX Administration Guide*

12.4.3.6 Enabling Users to Customize a Page

Use the Customizable attribute to control whether display attributes in a region can be customized by users.

To use this feature, you must include the `#CUSTOMIZE#` substitution string in the Header, Body, or Footer section of the page template.

If at least one region supports end-user customization, a **Customize** link appears wherever you include the `#CUSTOMIZE#` substitution string in the page template. When users click this link, a window appears, enabling them to display or hide regions on the page.

To enable end user customization:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

2. Select the region in the Rendering tab or the Layout tab.

   The Property Editor displays attributes in the **Region** tab in the right pane.

3. Customization, Customizable - Select one of the following:
   - **Not Customizable By End Users**
   - **Customizable and Not Shown By Default**
   - **Customizable and Shown By Default**
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> Creating Custom Themes

12.4.3.7 Specifying a Static Region ID

Specifying a static region ID is useful when creating custom JavaScript or cascading style sheets.

You can use the Static ID attribute on the Edit Region page to uniquely identify a region. You can then reference the region using the `#REGION_STATIC_ID#` substitution string in a region templates, the header, the footer, or the body.

A static region ID is included by assigning it as an HTML element ID to the region container object (that is, the table or DIV). The ID of an HTML element must be unique for the entire page. For example, to keep the page HTML valid you cannot have a button and region with the same ID.

To specify a static region ID:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Select the region in the Rendering tab or the Layout tab.

   The Property Editor displays attributes in the **Region** tab in the right pane.
3. Advanced, Static ID - Enter the static ID for this region. The Static ID is used as the ID for the region, using the substitution string `#REGION_STATIC_ID#`. `#REGION_STATIC_ID#` is useful if you are developing custom JavaScript behavior for the region. If this attribute is not defined, the region generates an internal ID.

   > **Tip:**
   >
   > The region template must include the `#REGION_STATIC_ID#` substitution string, in order for the region to utilize the static ID entered.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

12.4.3.8 Adding a Region Image

Use Region Image and Image Tag Attributes to add an image to the upper left side of a region.

Use the Region Image and Image Tag Attributes to add an image to the upper left side of a region.

To add a region image:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

c. Select a page.

Page Designer appears.

2. Select the region in the Rendering tab or the Layout tab.

   The Property Editor displays attributes in the **Region** tab in the right pane.

3. Under **Advanced**, edit the following attributes:

   - Region Image - Enter a reference to an image that displays in the upper left of the region. There is no control over the table tag used to display this image.
   - Image Tag Attributes - Enter image tag attributes included in the image HTML. This can be used for attributes such as height, width or image ALT text (if the image is non-decorative).

     If no ALT is defined in the image tag attributes, APEX defaults to rendering the image as a decorative image, for example with an empty ALT tag.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

12.4.3.9 Creating a Region Display Selector

Create a Region Display Selector region to include show and hide controls for each region on a page. to include the current region in the Region Display Selector.

When a user clicks on a region name included in the Region Display Selector, only the selected region shows and all other listed regions are hidden. The first entry within the Region Display Selector is Show All, which enables the user to show all of the available regions rather than only the previously selected region.

To create a Region Display Selector:

1. Create the page and regions to be included in the Region Page Selector.
2. For each region to be included in the selector, set the Region Display Selector attribute to **On**.
3. In the Gallery, right-click **Region Display Selector** region, select **Add To**, and select the appropriate location.

   > **Tip:**
   >
   > You can also drag and drop from the Gallery. From the Gallery select the **Region Display Selector** region and drag it to the appropriate location in the Layout tab.

4. Edit the Region attributes in the Property Editor:

   a. In the Rendering tab (left pane), select the region.

      The Property Editor displays attributes in the **Region** tab. Attributes are organized in groups.

   b. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.

   c. Advanced, Region Display Selector - Select **On** if you want this region included in the Region Display Selector. This attribute is only utilized if a region type of Region Display Selector is defined and viewable on the page.

   d. Edit additional attributes as needed.

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Edit Attributes in the Property Editor:
   a. In the Rendering tab (left pane), select the region.
   b. In the Property Editor, select the **Attributes** tab.
   c. Settings, Mode - Select how Region Display Selector regions display on the page. Options include:
      - **View Single Region** - Show regions as tabs. Selecting a tab will make the corresponding region visible and hide the other selections.
      - **Scroll Window** - Always display all the regions on the page. Selecting a tab will scroll your window to the corresponding region.
   d. Settings, Display Region Icons - Region Display Selector tabs will include the icon and title associated with each region.
   e. Settings, Include 'Show All' - Prepend a "Show All" tab which, when selected, displays all regions on the page.
   f. Settings, Remember Last Selection - Select how the Region Display Selectors remember the tab selection on page load. Options include:
      - **By User** - Tab selection will always be remembered across all sessions unless intentionally reset.
      - **By Session** - Tab selection will be remembered for the duration of a single session. Once the session ends, the tab selection will be reset.
      - **No** - Tab selection will be reset to the first tab each time the page reloads.
6. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

12.4.4 Copying a Region

Copy a region in Page Designer.

> **Tip:**
>
> You cannot copy a Tree region since this type of region encompasses more than one region.

To copy a region:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Right-click the region in Layout tab, select **Copy To**, and select a destination from the submenu.

> **Tip:**
>
> To copy a region to another page, select the region in the Rendering tab, right-click, and select **Copy to Other Page**. Follow the on-screen instructions.

12.4.5 Deleting Regions

Delete regions on Page Designer.

To delete regions:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Select one or more regions in the Rendering tab.
3. Right-click and select **Delete**.
4. Click **Save**.

12.5 About Incorporating Content from Other Websites

To incorporate content from other servers, you can create a region based on a URL to display content.

Typically, pages in an application are based on data stored in an Oracle database. For example, suppose you wanted to reference the current Oracle stock price. You could create a region of type URL based on a URL. For example:

```
http://quote.yahoo.com/q?d=b&s=ORCL
```

You could then create an item called STOCK_SYMBOL and base your region on a stock price entered by the user. For example:

```
http://quote.yahoo.com/q?d=b&s=&STOCK_SYMBOL.
```

Sometimes (as is the case with the previous example) the HTML returned to the region is more than is needed. To restrict the HTML displayed, you can use the following region attributes:

- URL (discard until but not including this text)
- URL (discard after and including this text)

> **Tip:**
>
> The previous example may require that you set the Proxy Server application attribute. If you do not set the Proxy Server application attribute, you get an error message. Oracle APEX uses the Oracle `utl_http.request_pieces` function to obtain the HTML generated from the given URL. See Editing the Application Definition.

Developers can also use the Region attribute **Inclusion Mode** to control how APEX retrieves and displays the content. Options include:

- **IFrame** - The browser displays the remote content in an HTML iframe. APEX does not have to fetch this content into the database for display.
- **Inline (escape special characters)** - Causes the database to load the remote content at page rendering time. It then displays the fetched data inline, at the position of the region. Special HTML characters in the content are escaped, to prevent cross site scripting.
- **Inline (no escaping)** - This mode is similar to **Inline (escape special characters)** except the HTML characters are not escaped. Developers should only use this mode if the remote content is always safe.

> **See Also:**
>
> Editing Region Attributes

Working with SSL-Enabled URLs

If you call a SSL-enabled URL (for example, by invoking a Web service), or create a region of type URL that is SSL-enabled, you must create a wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.

> **Tip:**
>
> See Configuring Wallet Information in *Oracle APEX Administration Guide*

12.6 About Managing Images

To reference an image within an application, you must upload it to the static file repository. During the upload process, you specify whether the file is available to all applications or just a specific application.

> **See Also:**
>
> Managing Static Application Files and Managing Static Workspace Files

12.7 Rendering HTML Using Custom PL/SQL

If you must generate specific HTML content not handled by Oracle APEX forms, reports, and charts, you can use the PL/SQL region type.

To generate HTML in this type of region, you need to use the PL/SQL Web Toolkit. You can reference session state using bind variable syntax. Keep in mind that when you generate HTML in this way, you do not get the same consistency and control provided with templates.

To give you more control over HTML dynamically generated within a region, you can use PL/SQL. For example, to print the current date, you could create a region with the following source:

```
htp.p(TO_CHAR(SYSDATE,'Day Month DD, YYYY'));
```

This next example accesses tables:

```
DECLARE
   l_max_sal NUMBER;
BEGIN
   SELECT max(sal) INTO l_max_sal FROM emp;
   htp.p('The maximum salary is: '||TO_CHAR(l_max_sal,'999,999.00'));
END;
```

See Also:

Oracle Database Development Guide

13
Managing Application Controls

Developers can create and manage a variety of application page controls, including page items, dynamic actions, buttons, and trees.

- Managing Page Items
  View existing page items, create new page items, and edit an item to change its appearance or behavior.
- Managing Dynamic Actions
  Dynamic actions enable developers to define complex client-side behavior declaratively without the need for JavaScript.
- Managing Buttons
  You can use buttons to direct users to a specific page or URL, or to post or process information (for example, by creating Create, Cancel, Next, Previous, or Delete buttons). You can also configure buttons to display conditionally or warn users of unsaved changes.
- Managing Trees
  Trees to display hierarchical information in a clear, easy-to-use format. You can create a tree control using a SQL query.
- About Incorporating JavaScript into an Application
  Oracle APEX includes multiple built-in interfaces especially designed for adding JavaScript.

See Also:

- Controlling Page Layout
- Adding Navigation
- Manually Refreshing Oracle APEX Components

13.1 Managing Page Items

View existing page items, create new page items, and edit an item to change its appearance or behavior.

Tip:

See About Item Types to learn more about the item types available in App Builder.

- Understanding Page Items
  An item is part of an HTML form such as a checkbox, date picker, file browse field, popup list of values (LOV), select list, shuttle, flip toggle switch, text field, text area, and so on.

- Viewing Page Items
  Configure page items to control how items display.
- Creating Page Items
  Create page items in Page Designer in either the Rendering tab or by adding an item from the Gallery.
- Editing Page Items
  Edit item attributes to control how an item displays on a page.

13.1.1 Understanding Page Items

An item is part of an HTML form such as a checkbox, date picker, file browse field, popup list of values (LOV), select list, shuttle, flip toggle switch, text field, text area, and so on.

When defining an item, developers must follow defined naming conventions and follow specific rules when referencing item values stored in session state.

- About the Differences Between Page Items and Application Items
  Page items are placed on a page and have associated user interface properties and Application items are not associated with a page.
- About Item Naming Conventions
  When creating an item name, developers must follow very specific item naming conventions.
- Referencing Item Values
  You can reference item values stored in session state in regions, computations, processes, validation, and branches.
- About Referencing Items Using JavaScript
  When you reference an item, the best approach is to reference by ID.
- Working with Multiple Select List Item
  Learn how to handle values returned from a multiple select list item.

> **See Also:**
>
> About Item Types

13.1.1.1 About the Differences Between Page Items and Application Items

Page items are placed on a page and have associated user interface properties and Application items are not associated with a page.

There are two types of items: page items and application items. **Page items** are placed on a page and have associated user interface properties, such as Display Only, Label and Label Template. Examples of page items include a checkbox, date picker, display as text, file browse field, popup list of values, select list, or a text area. In contrast **Application items** are not associated with a page and therefore have no user interface properties. You can use an application item as a global variable.

> **See Also:**
>
> Managing Application Items

13.1.1.2 About Item Naming Conventions

When creating an item name, developers must follow very specific item naming conventions.

When specifying an item name, remember the following rules. Item names must:

- Be unique within an application.
- Not include quotation marks.
- Begin with a letter or a number, and subsequent characters can be letters, numbers, or underscore characters.
- Be case-insensitive.
- Should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax. See Referencing Session State Using Bind Variable Syntax.
- Cannot contain letters outside the base ASCII character set.

As a best practice Oracle recommends including the page number when naming items. By default, wizards prefix page item names with `P<page no>_<item name>` (for example, `P1_NAME`).

13.1.1.3 Referencing Item Values

You can reference item values stored in session state in regions, computations, processes, validation, and branches.

The following table describes the supported syntax for referencing item values.

Type	Syntax	Description
SQL	`:MY_ITEM`	Standard bind variable syntax for items whose names are no longer than 30 bytes. Use this syntax for references within a SQL query and within PL/SQL code.
PL/SQL	`V('MY_ITEM')`	PL/SQL syntax referencing the item value using the `V` function. Use this syntax in PL/SQL code of packages or stored procedures and functions. Avoid this syntax in SQL statements. It may result in performance problems.
PL/SQL	`NV('MY_NUMERIC_ITEM')`	Standard PL/SQL syntax referencing the numeric item value using the `NV` function. Use this syntax in PL/SQL code of packages or stored procedures and functions. Avoid this syntax in SQL statements. It may result in performance problems.
Static Text (exact)	`&MY_ITEM.`	Static text. Exact Substitution. **Note**: Exact substitution syntax should be avoided in SQL or PL/SQL code because it can result in SQL Injection vulnerabilities.

You can set the value of an item in your application using any of the following methods:

- For page items, use the Source Attribute to set the item value.

From the page, select the item name to view the Edit Page Item page. Scroll down to Source and edit the appropriate fields.

You can also set the value of an item in any region based on PL/SQL or a process using the following syntax:

```
BEGIN
 :MY_ITEM :=  'new value';
END;
```

- Pass the value on a URL reference using `f?p` syntax. For example:

```
f?p=100:101:10636547268728380919::NO::MY_ITEM:ABC
```

- Set the value using a computation. Computations are designed to set item values. For example:

```
TO_CHAR(SYSDATE,'Day DD Month, YYYY');
```

- Use the PL/SQL API to set an item value within a PL/SQL context. For example:

```
APEX_UTIL.SET_SESSION_STATE('MY_ITEM',SYSDATE);
```

See Also:

- Clearing Session State
- Understanding Cross-Site Scripting Protection
- Managing Session State Values

13.1.1.4 About Referencing Items Using JavaScript

When you reference an item, the best approach is to reference by ID.

If you view the HTML source of an Oracle APEX page in a web browser, you would notice that all items have an id attribute. This ID corresponds to the name of the item, not the item label. For example, if you create an item with the name `P1_FIRST_NAME` and a label of `First Name`, the ID is `P1_FIRST_NAME`.

You can get and set item attributes and values using the JavaScript functions `$v('P1_FIRST_NAME')` and `$s('P1_FIRST_NAME', 'Joe');`. Consider the following example:

```
function showFirstName(){
  alert('First Name is ' +$v('P1_FIRST_NAME'))
};
function setFirstName(pFirstName){
  $s('P1_FIRST_NAME', pFirstName);
};
```

These functions can be called by other JavaScript functions or with the Execute JavaScript code dynamic action.

> **See Also:**
>
> APEX_JAVASCRIPT in *Oracle APEX API Reference*

13.1.1.5 Working with Multiple Select List Item

Learn how to handle values returned from a multiple select list item.

- About Handling Values Returned from a Multiple Select List Item
  Learn about handling values returned from a multiple select list item.
- Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values
  Learn about using `APEX_UTIL.STRING_TO_TABLE` to convert selected values.

13.1.1.5.1 About Handling Values Returned from a Multiple Select List Item

Learn about handling values returned from a multiple select list item.

A multiple select item renders as a multiple select list form element which can be either a Multiselect List or Shuttle item type. When submitted, selected values are returned in a single colon-delimited string. You can handle values in this format in three ways:

- Using the `INSTR` function
- Using the `APEX_STRING.SPLIT` function
- Creating a shuttle

13.1.1.5.2 Using APEX_UTIL.STRING_TO_TABLE to Convert Selected Values

Learn about using `APEX_UTIL.STRING_TO_TABLE` to convert selected values.

Suppose you had a report on the `EMP` and `DEPT` tables that is limited by the departments selected from a Department multiple select list. First, you create the multiple select item, P1_DEPTNO, using the following query:

```
SELECT dname, deptno
FROM dept
```

Second, you return only those employees within the selected departments as follows:

```
SELECT ename, job, sal, comm, dname
FROM emp e, dept d
WHERE d.deptno = e.deptno
AND instr(':'||:P1_DEPTNO||':',':'||e.deptno||':') > 0
```

Next, assume you want to programmatically step through the values selected in the multiple select item, P1_DEPTNO. To accomplish this task, convert the colon-delimited string into a PL/SQL array using the `apex_string.split` function. The following example demonstrates how to insert the selected departments into an audit table containing the date of the query.

```
DECLARE
  l_selected apex_t_varchar2;
BEGIN
  --
  -- Convert the colon separated string of values into
  -- a PL/SQL array
```

```
  l_selected := apex_string.split(
                    p_str => :P1_DEPTNO,
                    p_sep => ':' );

  --
  -- Loop over array to insert department numbers and sysdate
  --

  FOR i IN 1..l_selected.count
  LOOP
    INSERT INTO report_audit_table (report_date, selected_department)
        VALUES (sysdate, l_selected(i));
  END LOOP;
END;
```

> **See Also:**
>
> STRING_TO_TABLE Function in *Oracle APEX API Reference*

13.1.2 Viewing Page Items

Configure page items to control how items display.

To view page attributes in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays Page Item attributes. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. Edit the appropriate attributes in the Property Editor.

   Edited attributes display a green marker to the left of the attribute name until the page is saved.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Tip:**
>
> See About Item Types to learn more about the item types available in App Builder.

13.1.3 Creating Page Items

Create page items in Page Designer in either the Rendering tab or by adding an item from the Gallery.

- Creating a Page Item from the Rendering Tab
  Create a page item in Page Designer from the Rendering tab.
- Adding a Page Item from the Gallery
  Create a page item in Page Designer by adding it from the Gallery.
- Creating a Static List of Values
  Create a static list of values locally with the definition of the item.
- Creating a Cascading List of Values
  Create a cascading list of values locally with the definition of the item.
- Creating a Shuttle Item on the Form Page
  Create a shuttle item in Page Designer.

13.1.3.1 Creating a Page Item from the Rendering Tab

Create a page item in Page Designer from the Rendering tab.

> **Tip:**
>
> To learn more about available item types, see About Item Types.

To create a page item from the Rendering tab:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. If necessary, create a region to contain the item.
3. In the Rendering tab, right-click the region to contain the item and select **Create Page Item**.

   > **Note:**
   >
   > You can also right-click an existing item and select **Create Page Item Below**.

   The Property Editor displays Page Item attributes. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. Edit the attributes in the Property Editor:

To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane. Edited attributes display a green marker to the left of the attribute name until the page is saved.

a. Identification, Name - Enter the name of this item. Item names must be valid Oracle identifiers. See About Item Naming Conventions to learn about item naming best practices.

b. Identification, Type - Specify how the item is rendered.

c. Edit the appropriate attributes in the Property Editor.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> About Regions

13.1.3.2 Adding a Page Item from the Gallery

Create a page item in Page Designer by adding it from the Gallery.

> **Tip:**
>
> To learn more about available item types, see About Item Types.

To create a page item by adding if from the Gallery:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. If necessary, create a region to contain the item.

3. In the central pane, click **Layout** and then the **Items** tab in the Gallery.

   Passing the cursor over an item displays a tooltip that describes it.

4. Select an item and drag it to the appropriate location in the Layout tab.

   > **Tip:**
   >
   > You also right-click the item, click **Add To** and select the appropriate location.

   The Property Editor displays Page Item attributes.

5. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.

   a. Click the **Show Messages** icon on the Page Designer toolbar.

   The Messages dialog displays errors that must be addressed.

b. Select an error to highlight the associated attribute in the Property Editor.

6. Edit the attributes in the Property Editor:

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane. Edited attributes display a green marker to the left of the attribute name until the page is saved.

   a. Identification, Name - Enter the name of this item. Item names must be valid Oracle identifiers. See About Item Naming Conventions to learn about item naming best practices.

   b. Identification, Type - Specify how the item is rendered.

   c. Edit the appropriate attributes in the Property Editor.

7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> About Regions

13.1.3.3 Creating a Static List of Values

Create a static list of values locally with the definition of the item.

Note that local a list of values is not reusable. As a best practice, Oracle recommends creating a list of values as a shared component whenever possible.

To create a static list of values locally:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. Create a new item from the Rendering tab or by adding it from the Gallery.

   The Property Editor displays the Page Item attributes. Attributes are organized in groups. To search for a group or an attribute, enter a keyword in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. Edit the appropriate attributes in the Property Editor.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Under Identification:

   a. Identification, Name - Enter the name of this item. Item names must be valid Oracle identifiers. Oracle recommends that item names be no longer then 30 bytes so they can be used as bind variables in SQL Queries.

   b. Identification, Type - Select **Select List**.

5. Under List of Values:

   a. List of Values, Type - Select **Static Values**.

b. List of Values, Static Values - Click the **Display1, Display2** button.

   In the Static Values dialog:

   - **Display Value** - Enter the Display Value for each entry.
   - **Return Value** - Enter the Return Value for each entry. If you do not include a Return Value, the return value equals the Display Value
   - **Move Up** and **Move Down** - Click the **Move Up** and **Move Down** arrow buttons to change the order.
   - **Sort** - To sort the list at alphabetically at runtime, expand the Sort region and enable **Sort at Runtime**.

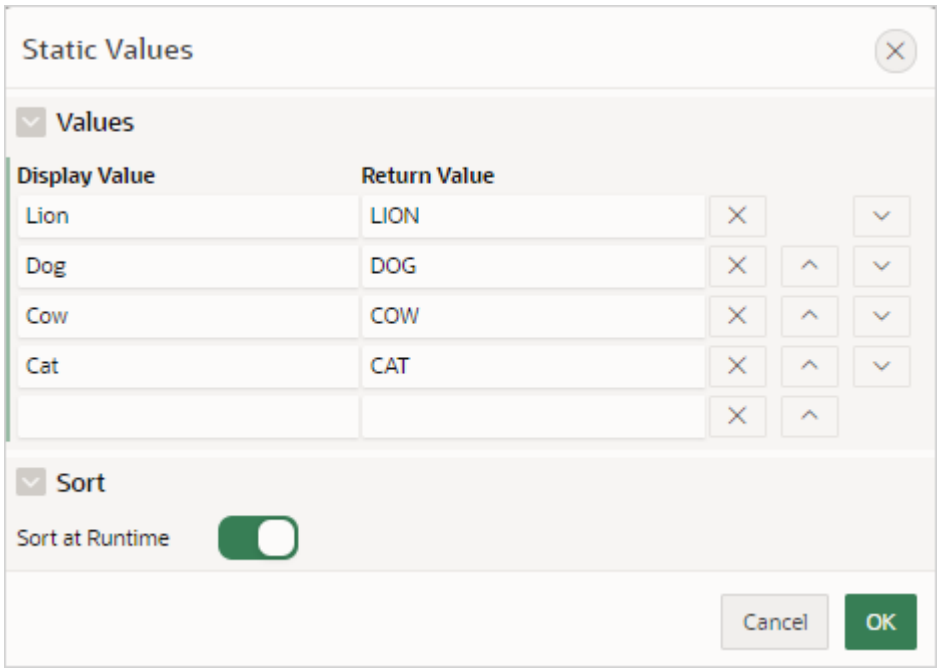
   > **Tip:**
   >
   > See the sections that follow to view examples.

c. Click **OK**.

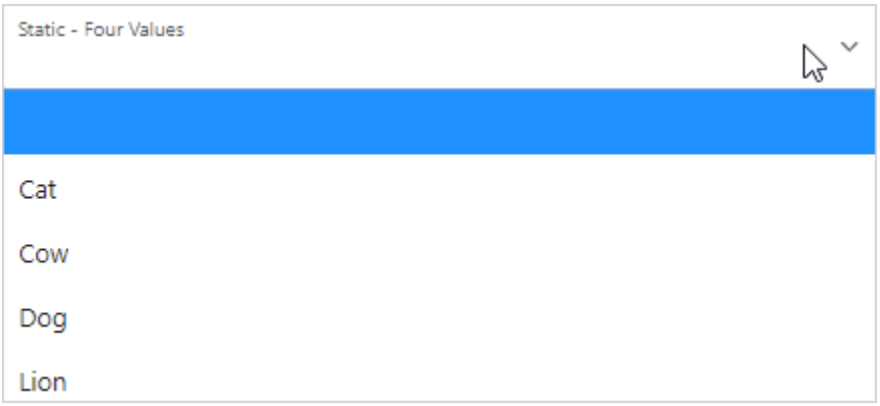
6. Edit the remaining List of Values attributes as appropriate. To learn more about an attribute, select the attribute and click the **Help** tab in the central pane.
7. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

Example 13-1 Four Values Displayed in Alphabetical Order

This example shows the Static Values dialog with four values defined: `Lion`, `Dog`, `Cow`, and `Cat`. The return value of each entry is capitalized. **Sort at Runtime** is set to **On** so that the list displays alphabetical order.

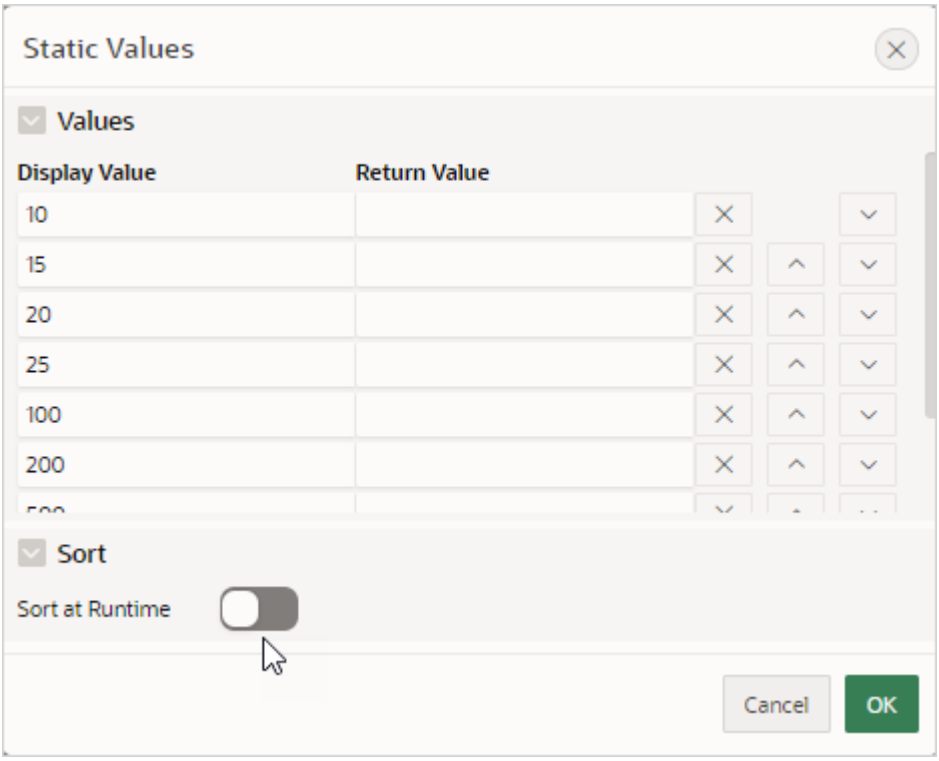


In a running application this select list would look similar to the following illustration.

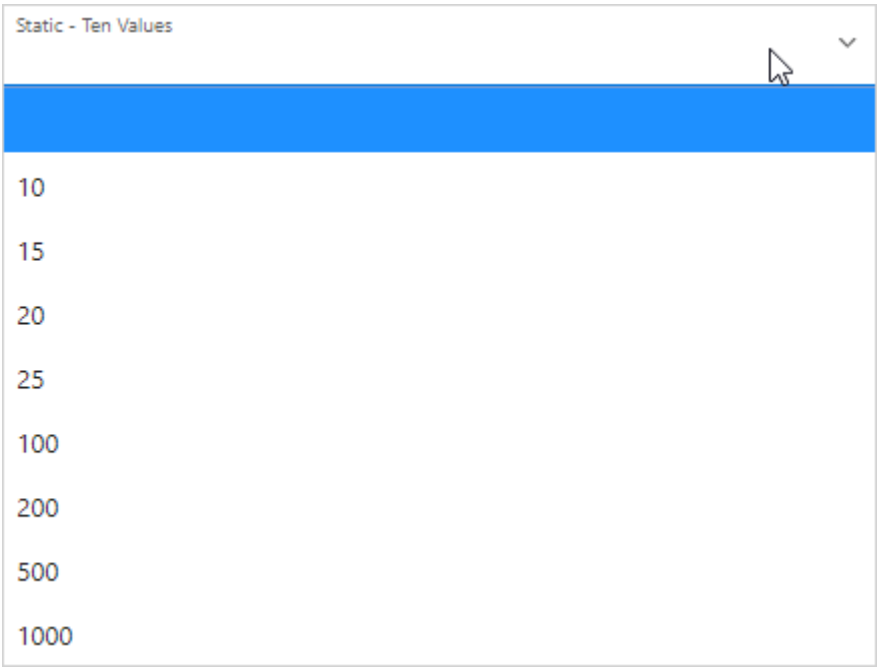


Example 13-2 Values Displayed in the Order Listed

This example shows the Static Values dialog with ten values defined: `10`, `15`, `20`, `25`, `50`, `100`, `200`, `500`, `1000`, and `10000` which display in the order listed. The return value of each entry equals the display value.

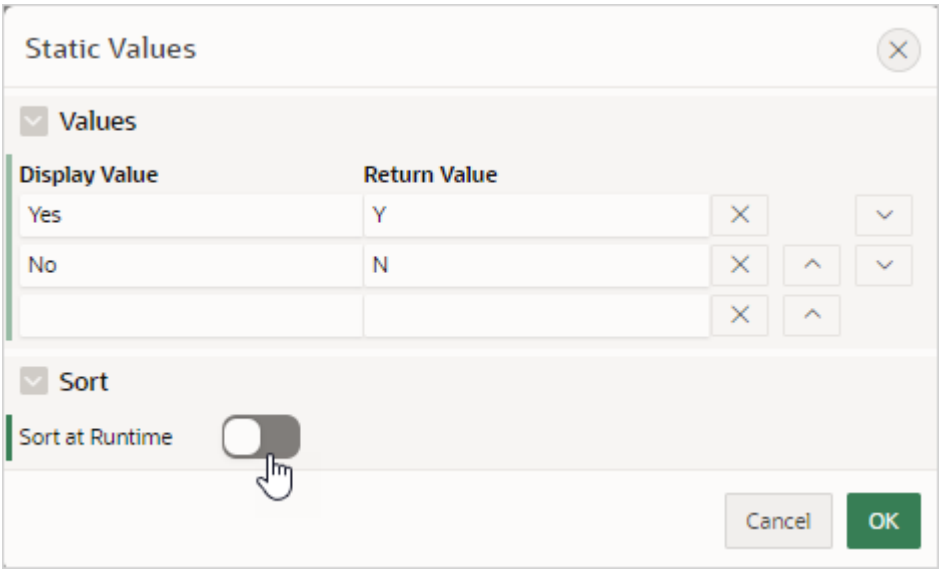


In a running application this select list would look similar to the following illustration.



Example 13-3 A List of Values with Having Both a Return and Display Value

This example shows the Static Values dialog with two values defined. The first value has a **Display Value** of `Yes` and a **Return Value** of `Y`. The second value has a **Display Value** of `No` and a **Return Value** of `N` **Sort at Runtime** is set to **Off** to make sure Yes always displays first.



In a running application this select list would look similar to the following illustration.

See Also:

- Creating Lists of Values at the Application-Level
- About Item Naming Conventions

13.1.3.4 Creating a Cascading List of Values

Create a cascading list of values locally with the definition of the item.

By creating a cascading list of values, one item on a page determines the list of values for another item. App Builder includes cascading list of values support for the following item types: Select List, Shuttle, Checkbox, Radio Group, Popup LOV, and List Manager.

You create a cascading list of values (LOV) by creating two items. To populate the second item, you specify a Cascading LOV Parent Item when running the Create Item Wizard or by editing the item's attributes.

The following example demonstrates how to create two related select lists where the value of the first list populates the second.

To create a cascading list of values:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create the first item, `P1_DEPTNO`:
   a. Create a new item from the Rendering tab or by adding it from the Gallery.

      The Property Editor displays the Page Item attributes. Attributes are organized in groups. To search for a group or an attribute, enter a keyword in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
   b. In the Property Editor, edit the Identification and Label attributes:
      - Identification, Name - Enter `P1_DEPTNO`
      - Identification, Type - Select **Select List**.
      - Label, Label - Enter `Department`
   c. Under List of Values:
      - List of Values, Type - Select **SQL Query**.
      - List of Values, SQL Query - Enter:

        ```
        SELECT dname as d,
               deptno as r
          FROM dept
         ORDER BY dname
        ```
      - List of Values, Display Null Value - Select **On**.

- List of Values, Null Display Value - Enter:

  ```
  - Select Department -
  ```

- Accept the remaining defaults.
- Click **Save**.

3. Create the second item, `P1_EMPNO`:
   a. Create a select list item.
   b. In the Property Editor, edit the Identification and Label attributes:
      - Identification, Name - Enter `P1_EMPNO`
      - Identification, Type - Select **Select List**.
      - Label, Label - Enter `Employee`
   c. Under List of Values:
      - List of Values, Type - Select **SQL Query**.
      - List of Values, SQL Query, enter:

        ```
        SELECT ename as d,
               empno as r
          FROM emp
         WHERE deptno = :P1_DEPTNO
        ```

      - List of Values, Display Null Value - Select **On**.
      - List of Values, Null Display Value - Enter:

        ```
        - Select Employee -
        ```

      - Accept the remaining defaults.
      - Click **Save**.
4. To save and run the page, click **Save and Run Page**.

   Two select lists appear. Making a selection in the Department select list, determines which individuals display in the Employees select list.

13.1.3.5 Creating a Shuttle Item on the Form Page

Create a shuttle item in Page Designer.

This section demonstrates how to create shuttle item type. In this example, you first create a report and form on the `DEPT` table. Then, you add a shuttle item to the form that lists employees alphabetically to make it easier to assign employees to a department.

To create a shuttle item on a form:

1. Create a report and form on the `DEPT` table.
2. View the form page in Page Designer:

   Page Designer appears.
3. Create a new item from the Rendering tab or by adding it from the Gallery.

   The Property Editor displays the Page Item attributes. Attributes are organized in groups. To search for a group or an attribute, enter a keyword in the Filter Properties field. The

Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. Select the new page item in the Rendering tab and edit attributes in the Property Editor.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. Under Identification:

   a. Identification, Name - Enter the name of this item (for example, `PX_EMP_LIST`).

      Where `PX` in the item name (for example `P2_EMP_LIST`) indicates the page on which the item resides.

   b. Identification, Type - Select **Shuttle**.

6. Under List of Values:

   a. List of Values: Type - Select **SQL Query**.

   b. List of Values, SQL Query, enter:

```
SELECT ename, empno FROM emp ORDER BY 1
```

7. Under Source:

   a. Source, Type - Select **SQL Query (return multiple values)**.

   b. Source, Item Source Value - Enter the SQL Query that returns one or more rows to be used as the source for this item value. If the result contains multiple rows then the value from each row is formed into a single colon delimited value. For example:

```
SELECT empno FROM emp WHERE deptno = :P2_DEPTNO ORDER BY ename
```

8. Under Multiple Values - Use these attribute to control how attribute lists are stored.

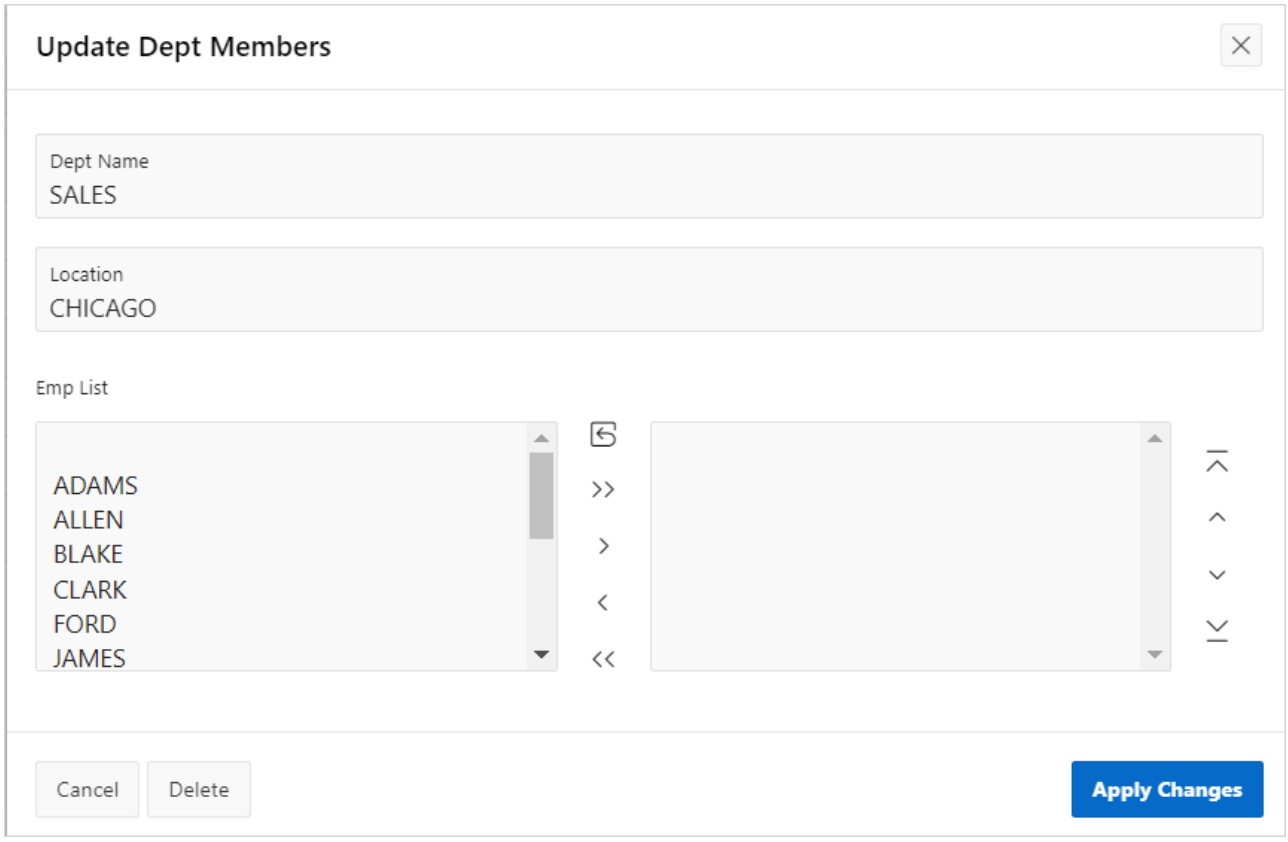
   a. Multiple Values, Type - Determines whether the Source Column contains multiple values, and how these are encoded.

   b. Multiple Values, Separator - Identifies the character which separates values from each other.

> **Note:**
>
> To learn more and view examples, see Page Designer Help or Configuring Multiple Value Attributes.

9. To save your changes click **Save**.

10. Run the application and view the revised form with the new Shuttle item.



13.1.4 Editing Page Items

Edit item attributes to control how an item displays on a page.

For example, the Type attribute determines if the item displays as text field, text area, checkbox, date picker, switch, hidden item and so on. Depending upon the item Type, item attributes can determine where a label displays, how large an item is, and if the item displays next to or below the previous item. Item attributes also control item behavior such as the item's default value, whether the item displays conditionally, or if the item is read-only.

- Editing Page Item Attributes in the Property Editor
  Learn about editing page item attributes in the Property Editor.
- Positioning Page Items
  Position page items at a page or region level.
- Defining Default Values for Page Items
  Define default values for an item using the Default attribute.
- Configuring Page Item Security
  Learn about editing page item Security attributes in the Property Editor.
- Changing an Existing Item to Use Quick Picks
  Change an item to use a quick picks list by edting the Quick Picks attributes in Page Designer.
- Configuring Multiple Value Attributes
  Control how lists are stored or define custom separators using Multiple Value attributes.
- Displaying Conditional Page Items
  Display an item conditionally by editing Server-side Condition attributes.

- Displaying Read Only Page Items
  Display a read-only item by editing the Read Only attribute in Page Designer.
- Applying a Format Mask to an Item
  Apply a format mask to an item by editing the Format Mask attribute.
- Configuring Item Attributes to Warn Users of Unsaved Changes
  Configure the Warn on Unsaved Changes attribute to specify if the page item should be included in the unsaved changes check.
- Controlling How Page Items Write Session State
  Control how a page item maintains session state by configuring the Session State, Storage attribute.
- Viewing Item Utilities
  Access the Utilities page to view Grid Edit pages and reports that enable you to edit items on multiple pages within a selected application.

13.1.4.1 Editing Page Item Attributes in the Property Editor

Learn about editing page item attributes in the Property Editor.

To edit item attributes in the Property Editor:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the Page Item attributes. Attributes are organized in groups. To search for a group or an attribute, enter a keyword in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. Edit the appropriate attributes in the Property Editor.

   Edited attributes display a green marker to the left of the attribute name until the page is saved.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.2 Positioning Page Items

Position page items at a page or region level.

To position a page item at the page-level:

1. Select the page item in Page Designer.
2. In the Property Editor leave the **Layout**, **Region** attribute empty.
3. To place a page item at a specific location on the page, configure the **Layout**, **Position** attribute.

To position a page item at a region-level:

1. Select the page item in Page Designer.
2. In the Property Editor, edit the **Layout**, **Region** attribute and select the region where the page item belongs.
3. Configure the **Layout**, **Position** within the region to place a Page Item at a specific location inside the region.

13.1.4.3 Defining Default Values for Page Items

Define default values for an item using the Default attribute.

The default value is used when the item's value is not derived from session state and when the source value is `NULL`.

To define a default value for an item:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the Page Item attributes. Attributes are organized in groups. To search for a group or an attribute, enter a keyword in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. In the Property Editor, find the **Default** group.
4. Default, Type - Select a default value type for this item. Options include:
   - **Static** - Set to the value entered in **Static Value**.
   - **Item** - Set to the value returned from the session state for the selected Item.
   - **SQL Query** - Set to the value returned from the entered SQL Query.
   - **SQL Query returning Colon Delimited List** - Set to the colon delimited list of values returned from the entered SQL Query . Colon delimited lists are required when the column is displayed as a checkbox, radio group, or shuttle.
   - **PL/SQL Expression** - Set to the value returned from the entered PL/SQL Expression.
   - **PL/SQL Function Body** - Set to the value returned from the PL/SQL Function Body entered.
   - **Sequence** - Set to the value returned from the Sequence entered.
5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.4 Configuring Page Item Security

Learn about editing page item Security attributes in the Property Editor.

To configure page item security:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

c. Select a page.

Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the item attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field, or click **Go to Group** and select the group.

3. Under Security:

   - Security, Authorization Scheme - Select an authorization scheme to control this component. The authorization must evaluate to TRUE in order for this component to be rendered or executed.

     See Also: Providing Security Through Authorization

   - Security, Session State Protection level - Select a Session State Protection level. Options include:

     – **Unrestricted** - The item can be set by passing the item in a URL or in a form. No checksum is required in the URL.

     – **Checksum Required - Application Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace and application.

     – **Checksum Required - User Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace, application and user.

     – **Checksum Required - Session Level** - The item can be set by passing the item in a URL that includes a checksum specific to the session.

     – **Restricted - May not be set from browser** - The item cannot be altered using the URL or POSTDATA. Select this option to restrict what can set the item value to internal processes, computations, and so on. This attribute only applies to items that are not used as data entry items and is always observed, even if Session State Protection is disabled. Use this attribute for page or application items that have the following Display As types

       * Display Only (Save State=No)

       * Text Field (Disabled, does not save state)

     See Also: Preventing URL Tampering

   - Security, Store value encrypted in session state - Specify whether to encrypt this item when stored in session state. If the contents of an item contains sensitive data, then you should encrypt the value when it is stored in the Oracle APEX session state management tables. Otherwise, anyone with rights to read the Oracle APEX meta data tables could potentially write a query to extract this sensitive data.

     See Also: Understanding Cross-Site Scripting Protection

   - Security, Restricted Characters - Select how to restrict the characters that can be saved in session state. APEX displays an error message if a user tries to save data that does not conform to the selected character restriction. Restricting the characters hardens application security and is a possible counter measure to cross-site scripting (XSS) attacks.

     Available options include:

     – **All characters can be saved** - No restriction applies.

     – **Allowlist for a-Z, 0-9 and space** - Only allow characters a-z, A-Z, 0-9, and space.

- **Blocklist HTML command characters (<>")** - Reserved HTML characters ***cannot*** be saved.
- **Blocklist &<>"/;,*|=% and --:** - The following characters ***cannot*** be saved:

  &, <, >, ", /, ;, ",", *, |, =, % and "--" (PL/SQL comment)
- **Blocklist &<>"/;,*|=% or -- and new line** - The following characters ***cannot*** be saved:

  &, <, >, ", /, ;, ",", *, |, =, %, "--", and new line characters

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.5 Changing an Existing Item to Use Quick Picks

Change an item to use a quick picks list by edting the Quick Picks attributes in Page Designer.

Quick picks enable users to select predefined values with just one click, rather than typing in text or selecting from a list of available options. If the item type supports quick pick selection, use the Quick Picks attributes to define up to ten selections that display under a give item. Clicking on a quick pick sets the value of item. The following example shows quick picks for 5%, 10%, 15%, and 20%.

To create a quick pick:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the item attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field, or click **Go to Group** and select the group.
3. Under Quick Picks:
   a. Quick Picks, Show Quick Picks - Select **On** .
   b. Quick Picks, Link Attributes - Enter HTML attributes that are associated with the display of each Quick Pick link
   c. Quick Picks, Label - Enter the label for the each Quick Pick. This label is the text the end user of the application sees.
   d. Quick Picks, Value - Enter the value associated with each Quick Pick. This value replaces the existing item value when the user clicks the quick pick.
   e. Repeat the previous two steps until all Quick Pick labels and values have been defined.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.6 Configuring Multiple Value Attributes

Control how lists are stored or define custom separators using Multiple Value attributes.

You can configure **Multiple Values** attributes for the folowing item types: Checkbox Group, Combobox, List Manager, Popup LOV, Select List, Select Many, and Shuttle. Configuration of these attributes is required for Checkbox Group, List Manager, Select Many, and Shuttle.

these
The **Multiple Values** group contain two attributes. **Type** determines whether the Source Column contains multiple values and how these are encoded. **Separator** enables you to define the character which separates values from one other.

To edit Multiple Values attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the item attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field, or click **Go to Group** and select the group.
3. Under Multiple Values:
   a. Multiple Values, Type - Determines whether the Source Column contains multiple values and how these are encoded.

      Options include:

      - **No** - Column does not contain multiple values.
      - **Delimited List** - Column contains values as a delimited list. Example:

        `APEX,SQL,PL/SQL`

        Use the `APEX_STRING.SPLIT` function for processing, for example:

        ```
        select column_value from apex_string.split ( :ITEM_NAME, ',' );
        ```

      - **JSON Array** - The Source column contains a JSON array of scalar values. If the column contains invalid JSON syntax, APEX will raise error messages, for example:

        ```
        ["APEX", "SQL", "PL/SQL"]
        ```

        Use the `JSON_TABLE SQL` function for processing, for example:

        ```
        select
            element
        from
            json_table ( :ITEM_NAME, '$[*]'
                columns
                    element varchar2 path '$'
            );
        ```

   b. Multiple Values, Separator - Identifies the character which separates values from one other. To view examples, see Page Designer Help.

> **Note:**
>
> Double-quote ("), backslash (/) characters or characters in the `\xxxx` format are not supported as a Value Separator.

4. To save your changes click **Save**.

13.1.4.7 Displaying Conditional Page Items

Display an item conditionally by editing Server-side Condition attributes.

To display a conditional item:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays Page Item attributes. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. Server-side Condition, Type - Select a condition type that must be met in order for this component to be rendered or processed.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.8 Displaying Read Only Page Items

Display a read-only item by editing the Read Only attribute in Page Designer.

To display a read-only item:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays attributes for the item. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field, or click **Go to Group** and select the group.
3. Under Read Only:
   a. Read Only, Type - Select a condition type that must be met in order for this component to render as read-only.

Not selecting a condition causes the item to render normally. A read-only item is rendered with the same settings as a Display Only item.

To always render the item as an enterable field, set the condition type to **Never**. This setting overwrites any read-only condition setting on the region or page level.

b. The UI may change depending on the Type you select. Edit the appropriate attributes.

To learn more, select the attribute in the Property Editor and click the **Help** tab in the central pane

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.9 Applying a Format Mask to an Item

Apply a format mask to an item by editing the Format Mask attribute.

Developers commonly select a format mask to determine how numbers and dates display.

To apply a format mask to an item:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays Page Item attributes. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. Appearance, Format Mask - Select a format mask.

   > **Tip:**
   >
   > Only apply number format masks to items that contain numbers and only apply date format masks to items that contain dates. Otherwise, is raised since the component contains a value that can not be converted using the specified format mask.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.1.4.10 Configuring Item Attributes to Warn Users of Unsaved Changes

Configure the Warn on Unsaved Changes attribute to specify if the page item should be included in the unsaved changes check.

Developers can use the **Warn on Unsaved Changes** item attribute to specify if the page item should be included in the unsaved changes check. This check warns the user when they try to navigate away from a page and when the page contains unsaved changes. Items not based on a database column (for example, a search item) may not need to be checked. For those items, set **Warn on Unsaved Changes** to **Do Not Check**.

To configure the Warn on Unsaved Changes attribute:

1. View the page in Page Designer:

a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

c. Select a page.

Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays Page Item attributes. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

3. Advanced, Warn on Unsaved Changes - Select one of the following:

   - **Page Default** - Include item in the unsaved changes check if **Warn on Unsaved Changes** is enabled at the page-level.
   - **Ignore** - Changes to the item are ignored by the unsaved changes check. Select this option for items not based on a database column (for example, a search item).

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **Tip:**
>
> The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.

> **See Also:**
>
> - Configuring Pages to Warn Users of Unsaved Changes
> - warnOnUnsavedChanges in *Oracle APEX JavaScript API Reference*

13.1.4.11 Controlling How Page Items Write Session State

Control how a page item maintains session state by configuring the Session State, Storage attribute.

To configure page item writes session state:

1. View the page containing the item to be configured in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the item to edit.

   The Property Editor displays the item attributes. Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field, or click **Go to Group** and select the group.

3. In the Property Editor, find the **Session State** group.

4. Session State, Storage - Select how session state is maintained. Options include:
   - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
   - **Per Session (Disk)**- Maintain for each session by storing the value in the database, to access it across requests.
   - **Per User (Disk)** - Maintain for a user by storing the value in the database, to access it across sessions.

> **Note:**
>
> When creating database items that work with a Form region (for example as part of a wizard), **Per Request (Memory Only)** is the default. Per **User (Disk)** is not available for these items.

5. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

> **See Also:**
>
> About Setting Session State

13.1.4.12 Viewing Item Utilities

Access the Utilities page to view Grid Edit pages and reports that enable you to edit items on multiple pages within a selected application.

To access the Utilities page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Utilities**.
4. From Page Specific Utilities region, click **Item Utilities**.
5. Select one of the following reports:
   - **Edit Item Help Text** - Edit item help text to ensure consistency across your application.
   - **All Page Items** - Edit all page items in your application.
   - **Password Items** - Edit password items in your application. Oracle recommends password items not save session state. To learn more, see the **Password Items** region on the right side of the page.
   - **Conditional Items** - Edit conditional items in your application.

13.2 Managing Dynamic Actions

Dynamic actions enable developers to define complex client-side behavior declaratively without the need for JavaScript.

- About Dynamic Actions
  Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.
- About Dynamic Action Events
  You can define dynamic actions to fire based on events that happen on the page. Oracle APEX includes four different categories of events: Browser events, Framework events, Component events, and Custom events. This section describes all supported events, including the internal JavaScript event name in brackets.
- Creating a Dynamic Action
  Creating a dynamic action involves specifying when the action happens (with optional conditions), what action or actions are performed, and what elements are affected by the action. To learn more about any Property Editor attribute, select the attribute and click the **Help** tab in the central pane.
- Editing Dynamic Actions
  Once you create a dynamic action, you can modify attributes defined during the creation process, specify attributes not available during the process (such as specifying an Authorization Scheme) and add additional true actions.
- Defining Dynamic Action Event Scope
  After creating the dynamic action, the scope of the action can be modified to trigger only once, for the lifetime of the current page, or until triggering elements are updated by a Partial Page Refresh (PPR).
- Deleting a Dynamic Action
  Delete a dynamic action by selecting it in Page Designer and selecting Delete from the context menu.
- About Calling JavaScript Using a Dynamic Action
  You can execute JavaScript code by creating a dynamic action.
- Debugging Dynamic Actions
  Learn how to debug dynamic actions in APEX.

> **See Also:**
>
> Accessing Page Specific Utilities

13.2.1 About Dynamic Actions

Dynamic actions provide a way to define complex client-side behavior declaratively without the need for JavaScript. Using the Dynamic Action Create wizard, you specify an action that is performed when a defined set of conditions occur. You can also specify which elements are affected by the action, and when and how they are affected.

Without any complex coding or knowledge of JavaScript, you can create declaratively define how componentsrespond to various user interactions such as button clicks, items changing, and so. An APEX dynamic actioin consists two parts: **Triggers** and **Actions**. The Trigger defines the condition upon which the Actions take place. Actions enable you to dynamically do anything you would normally achieve using a combination of JavaScript and CSS such as showing/hiding page components, setting values, executing PL/SQL code and so on.

When working with dynamic actions, you should be mindful of the fact that the more dynamic actions you add to a page, the greater your overall page size. This is because the dynamic action framework emits additional code to the client for each dynamic action defined, which then must be downloaded and executed by the framework in the client.

The process of implementing a dynamic action involves the following steps:

1. Create (or use an existing) page component such as an item, button or region. This component is referenced within the dynamic action, in defining when it fires.
2. Create a dynamic action on the application page that invokes the action.
3. Run your application to test the dynamic action.

> **Tip:**
>
> See Debugging Dynamic Actions for information on how to debug problems.

> **Tip:**
>
> To view dynamic action examples, install the sample app, *Sample Dynamic Actions*. To learn more, see Installing Apps from the Gallery.

13.2.2 About Dynamic Action Events

You can define dynamic actions to fire based on events that happen on the page. Oracle APEX includes four different categories of events: Browser events, Framework events, Component events, and Custom events. This section describes all supported events, including the internal JavaScript event name in brackets.

Browser Events

> **Note:**
>
> Refer to the W3C UI Events specification for details about browser events or consult a reputable JavaScript reference. The events that begin with `apex` are events that wrap native touch events.

- Change (`change`) - Fires when a control loses the input focus and its value has been modified since gaining focus.
- Click (`click`) - Fires when the pointing device button is clicked over the triggering element.
- Double Click (`dblclick`) - Fires when the pointing device button is double clicked over the triggering element.
- Double Tap (`apexdoubletap`) - Fires when the pointer is doing a double tap/click.
- Get Focus (`focusin`) - Fires when the triggering element receives focus by either a pointing device or by tabbing into the element.
- Input (`input`) - This event is triggered every time the value of the element changes. It differs from the change event, which is only triggered when the value is committed, for

example, by pressing the enter key or selecting a value from a list of options. Use the input event when you want to respond immediately to user input, whether it's typed using the keyboard or pasted from the clipboard.

- Key Down (`keydown`) - Fires when a key on the keyboard is pressed. Use this event when you want to capture special keystrokes such as arrow keys, after a key has been pressed.
- Key Press (`keypress`) - Fires when a key on the keyboard is pressed resulting in text being entered. Use this event when you want to capture actual text entry.
- Key Release (`keyup`) - Fires when a key on the keyboard is released. Use this event when you want to capture special keystrokes such as arrow keys, after a key has been released.
- Lose Focus (`focusout`) - Fires when the triggering element loses focus either by the pointing device or by tabbing out of the element.
- Mouse Button Press (`mousedown`) - Fires when the pointing device button is pressed over the triggering element.
- Mouse Button Release (`mouseup`) - Fires when the pointing device button is released over the triggering element.
- Mouse Enter (`mouseenter`) - Fires once when the pointing device is moved into the triggering element.
- Mouse Leave (`mouseleave`) - Fires once when the pointing device is moved away from the triggering element.
- Mouse Move (`mousemove`) - Fires when the pointing device is moved while it is over the triggering element.
- Pan (`apexpan`) - Fires when the pointer is down, then moved in a horizontal direction.
- Page Load - Fires when the page DOM content has loaded using the jQuery ready handler.
- Page Unload (`unload`) - Fires when a page is unloaded.
- Press (`apexpress`) - Fires when the pointer is down for greater than 250ms.
- Resize (`resize`) - Fires when the browser window is resized.
- Resource Load (`load`) - When the triggering element is the window element (using a JavaScript Expression value of `window` in the When attributes), the event fires when the browser finishes loading all content within a document, including window, frames, objects and images. For other elements, this event can only be used for elements associated with a URL: images, scripts, frames, iframes.
- Scroll (`scroll`) - Fires when a scrollable triggering element is scrolled. This could be the browser window (using a JavaScript Expression value of `window` in the When attributes), scrollable frames or elements with the `overflow` CSS property set to `scroll` (or `auto` when the element's explicit height is less than the height of its contents).
- Select (`select`) - Fires when a user selects some text in a text field.
- Swipe (`apexswipe`) - Fires when the pointer is moving fast in a horizontal direction.
- Tap (`apextap`) - Fires when the pointer is doing a small tap click.

Framework Events

- After Refresh (`apexafterrefresh`) - Fires after the triggering element has been refreshed. The event is only valid for triggering elements that perform Partial Page Refresh and fire this event. The native components that support this are Interactive Reports, Classic Reports, Charts, List View and all item types with cascading LOV support. Plug-ins might

support this event as well. This event can be sent by the `apex.server.plugin` and `apex.server.process` APIs when the `refreshObject` option is provided. See apex.server in *Oracle APEX API Reference*

- Before Page Submit (`apexbeforepagesubmit`) - Fires before a page being submitted.
- Before Refresh (`apexbeforerefresh`) - Fires before the triggering element has been refreshed. The event is only valid for triggering elements that perform Partial Page Refresh and fire this event. The native components that support this are Interactive Reports, Classic Reports, Charts, List View and all item types with cascading LOV support. Plug-ins might support this event as well. This event can be sent by the `apex.server.plugin` and `apex.server.process` APIs when the refreshObject option is provided. See apex.server in *Oracle APEX JavaScript API Reference*.
- Dialog Closed (`apexafterclosedialog`) - Fires when an APEX dialog is closed. This event only fires when the dialog is closed using the 'Close Dialog' page process, or the 'Close Dialog' dynamic action.

Component Events

These events are available when there is a component (either an item, region, or dynamic action) available to your application that triggers a custom event. These events appear in the following format Event name [Component Name], for example the Change Order event triggered by the Shuttle native item type appears as Change Order [Shuttle]. Component events are either triggered from native components shipped with APEX, or from plug-in components you have installed into your application.

- Events triggered by native components:

  These will be in the format Event name [Component Name]. For help related to events raised by our native components, see `interactiveGrid` and `treeView`. See interactiveGrid and treeView in *Oracle APEX JavaScript API Reference*.
- Events triggered by plug-in components:

  These will be available when added to your current application and will be in the format Event name [Component Name]. For help related to events raised by plug-ins, refer to Help text on the plug-in configuration page, by navigating to Shared Components, Plug-ins, *plug-in name*, Help Text, where the plug-in author may have included documentation.

Custom Events

- Custom Events:

  By selecting Custom an additional field displays enabling you to define of a custom event. This is useful when the native or plug-in provided events are insufficient.

13.2.3 Creating a Dynamic Action

Creating a dynamic action involves specifying when the action happens (with optional conditions), what action or actions are performed, and what elements are affected by the action. To learn more about any Property Editor attribute, select the attribute and click the **Help** tab in the central pane.

To create a dynamic action:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

c. Select a page.

Page Designer appears.

2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, right-click **Events** and select **Create Dynamic Action**.

> **Tip:**
>
> You can also select an event type and then right-click to select **Create Dynamic Action**.

The Property Editor displays Dynamic Action attributes.

Page Designer indicates what actions are required next. If there is an error, a **Messages** icon displays next to the **Save** button. Click the **Messages** icon to view a dialog listing all the errors. Select an error to highlight the associated attribute in the Property Editor.

4. In the Property Editor, edit the following Dynamic Action attributes:

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

a. **Identification**, **Name** - Enter the name of the dynamic action.

b. **Execution** , **Sequence** - Specify the sequence for this component. The sequence determines the order of execution.

c. **Execution** , **Event Scope** - Select the scope of the event. Event Scope determines when the event is evaluated a second or successive time. Options include:

- **Static** - Binds the event handler to the triggering element(s) for the lifetime of the current page, but is no longer bound if a triggering element is updated using Partial Page Refresh (PPR).
- **Dynamic** - Binds the event handler to the triggering element(s) for the lifetime of the current page, irrespective of any triggering elements being recreated using Partial Page Refresh (PPR).
- **Once** - Binds the event handler to the triggering element(s) only once. The dynamic action is not triggered again until after the page has been fully refeshed.

d. **Execution** , **Static Container (jQuery Selector)** - Only displays if **Event Scope** is **Dynamic**. Enter the jQuery Selector of a static container for use by this dynamic action.

You can improve the performance of dynamic actions that have an event scope of Dynamic. This must be an element on the page that itself does not get recreated, but contains the triggering elements that are recreated using Partial Page Refresh (PPR). This element reference needs to be defined as a jQuery selector.

e. **Execution** , **Type** - Choose the type of execution method for the dynamic action. This defines if the actions are fired immediately, debounced, or throttled. Options include:

- **Immediate** - Actions are executed immediately.

- **Debounce** - Actions are only executed once after a specified time at the end of a single or series of events.

  Example 1: Update a text field if the user has stopped typing for 2 seconds and use the values last entered.

  Example 2: Refresh a region once after a click and ignore all other clicks in the next 5 seconds.

- **Throttle** - Actions are executed at timed intervals when the action is fired multiple times.

  Example 1: During scrolling, update the current screen position at least every 2 seconds. The action will continue to fire if the user keeps scrolling.

  Example 2: As a user types, update a text field with the values entered at least every 2 seconds. The action will continue to fire as the user keeps typing.

f. **When**, **Event** - Specify the event that causes the dynamic action to fire.

g. **When**, **Selection** - Select the type of page element or construct to be used to trigger the event.

> **Note:**
>
> Only available if the event selected supports definition of a page element. Selecting any of the following events hides this attribute: Page Load, Page Unload, Resize, Before Page Submit, Orientation Change. All other event types show this field.

h. **Client-side Condition**, **Type** - Optionally select the type of condition you want to control the true and false action processing of the dynamic action. If no client-side condition is defined, only true actions will fire. If a client-side condition is defined, the true action will fire when the condition is met, and the false action will fire when it is not.

> **Note:**
>
> Other properties will display conditionally based on the type, to declaratively allow you to define the condition. For example for the condition type `Item = Value`, you will see an `Item` and `Value` property, which will be checked in evaluating the condition.

Next, define the action that to be performed if the event evaluates to True or False.

5. To edit an existing action:

   a. Expand the Dynamic Action tree and select an existing action (that is, select either **True** or **False**).

   b. In Property Editor, edit the following **Action** attributes:

      - **Identification**, **Name** - Enter the name of the action to help identify what the action does in Page Designer.
      - **Identification**, **Action** - Specify which action you want to perform.

- **Affected Elements**, **Selection Type**- Select the page components to be affected when this action is executed. Additional options display depending upon the type of component selected.
- **Execution Options**, **Fire When Event Result Is** - Specify whether this action fires when the triggering element condition is met by selecting True (True Action), or when it is not met by selecting False (False Action). If no condition is specified, only True actions fire.
- **Fire on Initialization** - Specify if the action fires on initialization.

  Initialization has a slightly different meaning depending on how the dynamic action is defined. For dynamic actions defined to fire on interactive grid columns, this specifies if the action fires when the interactive grid row is activated for editing. For all other dynamic actions, this specifies if the action fires when the page loads.

6. To add a new action:
   a. Expand the Dynamic Action tree.
   b. Right-click the dynamic action and select either **Create TRUE Action** or **Create FALSE Action**.
   c. Edit the action in the Property Editor as described in the previous step.
7. Click **Save**.

> **See Also:**
>
> About Dynamic Action Events

13.2.4 Editing Dynamic Actions

Once you create a dynamic action, you can modify attributes defined during the creation process, specify attributes not available during the process (such as specifying an Authorization Scheme) and add additional true actions.

To edit a dynamic action:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane. Dynamic actions are organized by events.
3. Expand the appropriate event and select the dynamic action.

   Attributes for the dynamic action display in the Property Editor.
4. In Property Editor, edit the appropriate Dynamic Action attributes.

> **Tip:**
>
> To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

5. To edit an existing action:
   a. Expand the Dynamic Action tree to view the True or False nodes.
   b. Under True or False, select the action.
   c. Edit the action in the Property Editor.
6. To add a new action:
   a. Expand the Dynamic Action tree.
   b. Right-click the dynamic action and select either **Create TRUE Action** or **Create FALSE Action**.
   c. Edit the action in the Property Editor.
7. Click **Save**.

13.2.5 Defining Dynamic Action Event Scope

After creating the dynamic action, the scope of the action can be modified to trigger only once, for the lifetime of the current page, or until triggering elements are updated by a Partial Page Refresh (PPR).

To specify event scope:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.
3. Under Dynamic Actions, select the dynamic action.

   The Property Editor displays attributes for the dynamic action. Attributes are organized in groups.

   To find a group or attribute:

   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
4. **Execution**, **Event Scope** - Select the scope of the event, which determines when the event is evaluated a second or successive time. Options include:
   - **Static** - Binds the event handler to the triggering element(s) for the lifetime of the current page, but is no longer bound if a triggering element is updated using Partial Page Refresh (PPR).

- **Dynamic** - Binds the event handler to the triggering element(s) for the lifetime of the current page, irrespective of any triggering elements being recreated using Partial Page Refresh (PPR).
- **Once** - Binds the event handler to the triggering element(s) only once. The dynamic action is not triggered again until after the page has been fully refeshed.

5. **Execution** , **Static Container (jQuery Selector)** - Only displays if **Sequence** is **Dynamic**. Enter the jQuery Selector, of a static container, for use by this dynamic action.

   You can improve the performance of dynamic actions that have an event scope of Dynamic. This must be an element on the page that itself does not get recreated, but contains the triggering elements that are recreated using Partial Page Refresh (PPR). This element reference needs to be defined as a jQuery selector.

6. **Execution** , **Type** - Choose the type of execution method for the dynamic action. This defines if the actions are fired immediately, debounced, or throttled. Options include:

   - **Immediate** - Actions are executed immediately.
   - **Debounce** - Actions are only executed once after a specified time at the end of a single or series of events.

     Example 1: Update a text field if the user has stopped typing for 2 seconds and use the values last entered.

     Example 2: Refresh a region once after a click and ignore all other clicks in the next 5 seconds.
   - **Throttle** - Actions are executed at timed intervals when the action is fired multiple times.

     Example 1: During scrolling, update the current screen position at least every 2 seconds. The action will continue to fire if the user keeps scrolling.

     Example 2: As a user types, update a text field with the values entered at least every 2 seconds. The action will continue to fire as the user keeps typing.

   > **Note:**
   >
   > If a Client-side Condition is defined, it will be evaluated when the dynamic action and itsActions are executed, not when the event fires.

7. Click **Save**.

13.2.6 Deleting a Dynamic Action

Delete a dynamic action by selecting it in Page Designer and selecting Delete from the context menu.

To delete a dynamic action:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Dynamic Actions** tab in the left pane.

3. Expand the appropriate event and locate the dynamic action to be deleted.
4. Right-click the dynamic action and select **Delete**.
5. Click **Save**.

13.2.7 About Calling JavaScript Using a Dynamic Action

You can execute JavaScript code by creating a dynamic action.

You can also execute JavaScript code by creating a dynamic action with the **ActionExecute** and **Set Value**. You can also use JavaScript code for the condition of a dynamic action by setting the Client-side Condition type to **JavaScript Expression**.

> **See Also:**
>
> Managing Dynamic Actions

13.2.8 Debugging Dynamic Actions

Learn how to debug dynamic actions in APEX.

- About Debugging Dynamic Actions
  Learn about debugging dynamic actions.
- Debugging Dynamic Actions
  Debug an dynamic action by viewing the page in runtime and clicking Debug on the Runtime Developer toolbar.

13.2.8.1 About Debugging Dynamic Actions

Learn about debugging dynamic actions.

Debugging dynamic actions in APEX is slightly different than other debugging, because much of the processing done with the dynamic action framework is done on the client, not on the server. To debug dynamic actions, APEX outputs debug information to the browser's JavaScript console if your browser supports it (for example Firefox with Firebug installed shows the debug information in its Console pane). The debug information tells you when an action of a dynamic action fires, along with some additional information about the dynamic action, in the following format:

```
Dynamic Action Fired: [Dynamic Action name] ([Action name]) {JavaScript
object containing all Dynamic Action information}
```

This format enables you to identify the dynamic action name, the action name which indicates which action is triggered, and the JavaScript object which contains a lot of information about the dynamic action, including the when element, the affected elements, the event object, and any data that may be associated with the dynamic action.

13.2.8.2 Debugging Dynamic Actions

Debug an dynamic action by viewing the page in runtime and clicking Debug on the Runtime Developer toolbar.

To debug a dynamic action:

1. Ensure the application containing the dynamic action has Debugging enabled. See Utilizing Debug Mode.
2. Run the page containing the dynamic action.
3. Open the browser's JavaScript console.
4. From the Runtime Developer toolbar, find **Debug**.
5. On Runtime Developer toolbar, click **Debug**, **Enable Debug**. Use **Enable Debug** to specify the level of detail included when you click **View Debug** .

   Options include:

   - **No Debug** - Enables and disables Debug mode
   - **Info (default)** - The debug log contains the default or normal amount of information. This corresponds with a log level of 4 and a Debug URL argument of `YES`.
   - **App Trace** - The debug log contains messages up to level 6. This corresponds with a Debug URL argument of `LEVEL6`.
   - **Full Trace** - The log will contain the maximum amount of information. This corresponds with a Debug URL argument of `LEVEL9`. **Full Trace** can slow request processing but provides the most amount of detail.

   Since debug information is only output when running in Debug mode, leaving Debug mode switched on enables you to further test if dynamic actions are firing when you expect them to. For example if you have defined a dynamic action that fires when a certain item's value changes, change that item's value and the console shows the debug output if the dynamic action fires.
6. On Runtime Developer toolbar, click **Debug**, **View Debug**.

   Clicking **View Debug** refreshes the page. If you have any dynamic actions that are set to fire on page load, you will see the debug output in the browser console.

13.3 Managing Buttons

You can use buttons to direct users to a specific page or URL, or to post or process information (for example, by creating Create, Cancel, Next, Previous, or Delete buttons). You can also configure buttons to display conditionally or warn users of unsaved changes.

- About Actions Buttons Can Perform
  Use buttons to submit a page or redirect to a different page or different URL.
- About Branching with Buttons
  Learn about branching with buttons.
- Creating a Button
  Create buttons in Page Designer.
- About the Relationship Between Button Names and REQUEST
  The name you give a Submit button (that is, a button with an Action of Submit Page) determines the value of the built-in attribute `REQUEST` when the page is submitted.
- Editing a Button
  Edit buttons in Page Designer.
- Displaying a Button Conditionally
  You can choose to have a button display conditionally by editing the Server-side Condition attribute in Page Designer.

- Configuring Button Attributes to Warn Users of Unsaved Changes
  Use the Warn on Unsaved Changes attribute to warn users of unsaved changes when they attempt to navigate away from a page.
- Avoiding Multiple Page Submits by Enabling Show Processing
  Prevent users from double-clicking a button by enabling the Show Processing attribute.
- Displaying a Button Confirmation Dialog
  Display a Confirmation dialog after the button is clicked by enabling the Requires Confirmation attribute.

> **See Also:**
>
> Calling a Page from a Button URL and Accessing Page Specific Utilities

13.3.1 About Actions Buttons Can Perform

Use buttons to submit a page or redirect to a different page or different URL.

The **Behavior**, **Action** attribute determines the action to be performed when a button is clicked. Available options include:

- **Submit Page** - Submit the current page with a `REQUEST` value equal to the Button Name.
- **Redirect to Page in this Application** - Redirect to a page in the current application with optional additional attributes for resetting pagination, setting the request value, clearing cache, and setting item values on the target page.
- **Redirect to Page in a different Application** - Redirect to a page in a different application with optional additional attributes for resetting pagination, setting the request value, clearing cache, and setting item values, on the target page.

> **Note:**
>
> Redirecting to a page in different application relies on deep linking being enabled in the target application. To enable deep linking, go to the application's Security attributes and enable Deep Linking.

- **Redirect to URL** - Redirect to the target URL you specify.
- **Defined by Dynamic Action** - Does not submit the page or redirect to another page. Instead the button's behavior is defined by a Dynamic Action.

13.3.2 About Branching with Buttons

Learn about branching with buttons.

Each page can include any number of branches. A branch links to another page in your application or to a URL. The Oracle APEX engine considers branching at different times during page processing. You can choose to branch before processing, before computation, before validation, and after processing. Like any other control in App Builder, branching can be conditional. For example, you can branch when a user clicks a button. When you create a branch, you associate it with a specific button. The branch is only be considered if a user clicks the button.

> **See Also:**
>
> Controlling Navigation Using Branches

13.3.3 Creating a Button

Create buttons in Page Designer.

To create a button in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. If necessary, create a region to contain the item.
3. In the Gallery, click the **Buttons** tab.

   Passing the cursor over a button displays a tooltip that describes it.
4. Right-click the button, select **Add To**, and select the appropriate location.

   > **Tip:**
   >
   > You also select the button with the mouse and drag it to the appropriate location in the Layout tab.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.

   To find a group or attribute:

   - **Search for a group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
5. Edit the appropriate attributes in the Property Editor.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
6. Click **Save**.

> **See Also:**
>
> About Regions

13.3.4 About the Relationship Between Button Names and REQUEST

The name you give a Submit button (that is, a button with an Action of Submit Page) determines the value of the built-in attribute `REQUEST` when the page is submitted.

You can reference the value of `REQUEST` from within PL/SQL using the bind variable `:REQUEST`. By using this bind variable, you can conditionally process, validate, or branch based on which button the user clicks. You can also create processes that execute when the user clicks a button. And you can use a more complex condition as demonstrated in the following examples:

```
If :REQUEST in ('EDIT','DELETE') then ...
If :REQUEST != 'DELETE' then ...
```

These examples assume the existence of buttons named EDIT and DELETE. You can also use this syntax in PL/SQL Expression conditions. Be aware, however, that the button name capitalization (case) is preserved. In other words, if you name a button LOGIN, then a request looking for the name *Login* fails. For example:

```
<input type="BUTTON" value="Finish" onclick="apex.submit('Finish');">
```

In this example *Finish* is the name of the `REQUEST` and this example is case-sensitive.

13.3.5 Editing a Button

Edit buttons in Page Designer.

To edit a region button in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.

   To find a group or attribute:

   - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
   - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.
3. Edit the button attributes.

> **Note:**
>
> To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Click **Save**.

13.3.6 Displaying a Button Conditionally

You can choose to have a button display conditionally by editing the Server-side Condition attribute in Page Designer.

To have a button display conditionally:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.
3. Configure the Server-side Condition:
   a. Server-side Condition, Type - Select a condition type that must be met in order for this component to be rendered or processed. What displays next, depends upon selected Type.

      What displays next, depends upon the Server-side Condition, Type you select.
   b. Configure the remaining attributes.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. Click **Save**.

> **See Also:**
>
> Referencing Session State Using Bind Variable Syntax

13.3.7 Configuring Button Attributes to Warn Users of Unsaved Changes

Use the Warn on Unsaved Changes attribute to warn users of unsaved changes when they attempt to navigate away from a page.

To configure the Warn on Unsaved Changes attribute:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the button to edit.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.
3. Behavior, Warn on Unsaved Changes - Select one of the following:
   - **Page Default** - Check for unsaved changes when the button is clicked if **Warn on Unsaved Changes** is enabled at page level.
   - **Do Not Check** - The unsaved changes check will not be performed when the button is clicked. Use this setting for Cancel, Delete, and Apply Changes buttons.
4. Click **Save**.

> **Tip:**
>
> The **Warn on Unsaved Changes** attribute is implemented using the `apex.page.warnOnUnsavedChanges` API.

> **See Also:**
>
> Configuring Pages to Warn Users of Unsaved Changes and warnOnUnsavedChanges in *Oracle APEX JavaScript API Reference*

13.3.8 Avoiding Multiple Page Submits by Enabling Show Processing

Prevent users from double-clicking a button by enabling the Show Processing attribute.

The **Show Processing** attribute is useful if you have a slow process and provides feedback to the user that the button has already been clicked. Enabling **Show Processing** grays out the page and displays a processing style animation when the page is submitted.

To enable the Show Processing attribute, the button **Action** must be **Submit Page**.

To enable Show Processing:

1. View the page in Page Designer:

a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

c. Select a page.

Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the button to edit.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.

3. Behavior - Configure the attributes:

   a. Behavior, Action - Select **Submit Page**.

   b. Behavior, Show Processing - Enable this attribute to gray out the page and display a processing style animation during page submission. This attribute is useful to prevent end user action during long running operations.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

13.3.9 Displaying a Button Confirmation Dialog

Display a Confirmation dialog after the button is clicked by enabling the Requires Confirmation attribute.

Enabling the **Requires Confirmation** attribute displays a confirmation dialog after a button is clicked. The button action, whether it be it to submit the page, to redirect to another page, or to invoke a dynamic action, will be delayed until the user has confirmed the action. If the user cancels, the action will not be performed. You can configure the dialog message and apply style options such as Information, Warning, Danger, or Success. The dialog's Confirm button label will match the actual button label.

To display a button confirmation dialog:

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. In either the Rendering tab or the Layout tab, select the button to edit.

   The Property Editor displays button attributes in the right pane. Attributes are organized in groups.

3. Behavior, Requires Confirmation - Enable this attribute.

   The Confirmation group appears.

4. Configure the attributes under Confirmation:

   a. Confirmation, Message - Enter the message to be displayed in the confirmation dialog. This attribute supports substitutions (application, page items, or system variables).

   b. Confirmation, Style - Select a style for the dialog. Options include:

      - **Default**
      - **Information**
      - **Warning**

- **Danger**
- **Success**

5. Click **Save**.

13.4 Managing Trees

Trees to display hierarchical information in a clear, easy-to-use format. You can create a tree control using a SQL query.

- About Trees
  Tree controls in Oracle APEX use APEX Tree.
- Creating a Tree Using the Create Page Wizard
  Create a tree on a new page by running the Create Page Wizard.
- Creating a Tree in Page Designer
  Create a tree control by providing a SQL query that specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view.
- Editing Tree Attributes
  Control tree behavior by editing region attributes and report Attributes.

13.4.1 About Trees

Tree controls in Oracle APEX use APEX Tree.

App Builder includes a built-in wizard for generating a tree hierarchical navigation mechanism. Trees are implemented using a single hierarchical query that identifies the row to be used as the start of your query and the relationship between parent rows and child rows of the hierarchy. Trees use the APEX Tree implementation. This is a JavaScript-based, cross browser tree component that features optional keyboard navigation, and optional state saving.

When you implement a tree control, the SQL query specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view. The tree query must utilizes a `START WITH .. CONNECT BY` clause to generate the hierarchical query.

> **Tip:**
>
> The creation of APEX Tree regions is only supported on pages using a Desktop user interface.

About Desupported jsTrees

Prior to release 5.2, APEX supported the rendering of jsTree tree regions. APEX no longer supports jsTree tree regions. APEX now supports the generation of an APEX Tree tree region. APEX Tree is a JavaScript based, cross browser tree component. jsTree regions are automatically upgraded to APEX Tree regions when upgrading from a previous release or when importing to release 18.1.

> **Tip:**
>
> To view tree examples, install the sample app, *Sample Trees*. To learn more, see Installing Apps from the Gallery.

13.4.2 Creating a Tree Using the Create Page Wizard

Create a tree on a new page by running the Create Page Wizard.

A tree is based on a query and returns data that can be represented in a hierarchy. When you create a tree using the Create Page Wizard, the wizard generates the hierarchical query for you based on the options you select.

To create a tree on a new page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Under Component, select **Tree**.
5. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

      > **Tip:**
      >
      > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
6. Data Source - Select a Data Source for the page:
   a. Table/View Owner - Select the owner of the table on which you are building the page.
   b. Table/View Name - Select the table or view on which the page will be based. When entering manually, note that the table name is case sensitive.
7. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
8. Click **Next**.
9. Tree Attributes:

A tree is based on a query and returns data that can be represented in a hierarchy. A start `with .. connect by` clause will be used to generate the hierarchical query for your tree. Use this page to identify the column you want to use as the ID, the Parent ID, and text that should appear on the nodes. The **Start With** column will be used to specify the root of the hierarchical query, and its value can be based on an existing item, static value or SQL query returning a single value.

a. ID - Select the column to use as the ID. For example, select the **EMPNO** column for the `EMP` table.

b. Parent ID - Select the column to use as the parent ID. For example, select the **MGR** column for the `EMP` table.

c. Node Text - Select the text to appear on the tree nodes. For example, select the **ENAME** column for the `EMP` table.

d. Start With - Select the column to be used to specify the root of the hierarchical tree query. For example, select the **MGR** column for the `EMP` table.

e. Start Tree - Choose how to start your query. Options include:

- **Based on Existing Item** - Select an existing application or page item.
- **Based on a SQL Query** - Enter a SQL query that returns a single row or single column.
- **Based on a Static Value** - Enter a static value.
- **Value is NULL**.

f. Click **Next**.

g. Include Buttons - Select the buttons to include.

10. Click **Create Page**.

11. To view page, click **Save and Run Page**.

    The rendered page appears.

13.4.3 Creating a Tree in Page Designer

Create a tree control by providing a SQL query that specifies a hierarchical relationship by identifying an ID and parent ID column in a table or view.

The tree query you provide must utilizes a `START WITH .. CONNECT BY` clause to generate the hierarchical query.

To create a tree in Page Designer

1. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

2. Click the **Layout** tab in the central pane.

3. In the Gallery, click the **Regions** tab. Right-click **Tree** and select **Add To**, and select the appropriate location.

> **Tip:**
>
> You can also select the Tree region in the Gallery and drag it to the appropriate location in the Layout tab.

4. Page Designer indicates what to do next. If there are errors, the **Show Messages** icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
5. Edit the **Region** attributes:
   a. In the Property Editor, select the **Region** tab (if not already selected).

      To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
   b. **Identification**, **Title** - Enter a region title. The region title only displays when it is defined in the region template.
   c. Under **Source** -
      - **Location** - Select the location of the data.
      - **Type** - Select how the data is queried.
      - **Table Name** - Enter the case-sensitive table or view name. You can type in the name or pick from the list.

      Depending upon your selections, additional attributes may appear.
   d. Under **Layout** -
      - **Sequence** - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
      - **Parent Region** - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region.
      - **Position** - Select the template position used to display this region. The selections available are derived from the appropriate template positions defined within the current theme, for the page's user interface.
   e. **Appearance**, **Template** - Specify the template. Select a region template to define the appearance and layout of this region.
6. Edit the tree **Attributes**:
   a. In the Rendering tab, select the region and click **Attributes** tab.

      Tree Attributes display in the Property Editor.

      To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
   b. Under **Settings**:
      - **Node Label Column** - Select the data source column containing the value for the node label. The value cannot contain markup.

- **Node Value Column** - Select the data source column containing the value for the node value. This value is not displayed but added to the tree adapter node id property that can be accessed using JavaScript.
- **Hierarchy** - Indicate whether APEX should compute the tree hierarchy from the data or whether the data already contains all hierarchy information. The user interface changes depending upon your selection.
  - **Computed With SQL**:
    - **Node ID Column** – Select the data source column containing the ID for a node. This is required when the hierarchy is being computed by APEX. Typically, a primary key column is chosen here.
    - **Parent Key Column** – Select the data source column containing the value for the parent key of a node. Required when APEX computes the hierarchy.
  - **Not Computed**:
    - **Node Status Column** – Select the data source column containing the value for the node status. Required when the **Hierarchy** attribute is set to **Not Computed**. The column must contain **0** for leaf nodes and **1** for nodes with children.
    - **Hierarchy Level Column** – Select the data source column containing the value for the hierarchy level of the node. Required when the **Hierarchy** attribute is set to **Not Computed**.
- **Tooltip** - Select whether tooltips are displayed, and the source for the tooltip.
- **Link** - Specify the link to be executed when a node is clicked.
- **Link Column** - Select the data source column containing the value for the link to be executed when a node is clicked.
- **Icon CSS Class Column** - Select the data source column containing the value for the icon CSS class to be applied to the node.

7. Click **Save**.

13.4.4 Editing Tree Attributes

Control tree behavior by editing region attributes and report Attributes.

By default, a tree does not have focus and node links are activated with a single click action.

To access the Tree attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Select the tree region in the Rendering tab or the Layout tab.

   The Property Editor displays attributes in the right pane. Attributes are organized in groups.

   To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
3. Edit Region attributes:

   a. In the Property Editor (right pane), select the **Region** tab.
   b. Edit attributes as needed.
4. Edit tree Attributes
   a. In the Property Editor (right pane), select the **Attributes** tab.
   b. Edit attributes as needed.
5. Click **Save**.

13.5 About Incorporating JavaScript into an Application

Oracle APEX includes multiple built-in interfaces especially designed for adding JavaScript.

Adding JavaScript to a web application is a great way to add features that mimic those found in client/server applications without sacrificing all the benefits of web deployment.

Remember that JavaScript is not appropriate for data intensive validations. For example, to verify that a name is contained within a large database table, you must pull down every record to the client, creating a huge HTML document. In general, complex operations are much better suited for server-side APEX validations instead of JavaScript. To learn more, see the JavaScript discussion for the appropriate control or component.

See Also:

- Supported Shortcut Types for information about referencing a shortcut inside of a JavaScript literal string
- Understanding Validations

14
Adding Navigation

Add navigation to an application by creating breadcrumbs, branches, tabs, or classic navigation bars.

> **Tip:**
>
> You can also add navigation by creating a list (or a shared collection of links). Newer themes support the creation of lists that render as a navigation menu or navigation bar. See Managing Navigation Menus and Managing Navigation Bar Lists.

- About Using Lists as Navigation
  Add navigation to your application by creating a list (or a shared collection of links). Newer themes, such as *Universal Theme - 42*, support the creation of navigation menus and navigation bars.
- Controlling Navigation Using Branches
  Add navigation to application pages by creating branches. A branch is an instruction to link to a specific page, procedure, or URL after a given page is submitted.
- Creating Breadcrumbs
  Breadcrumbs provide hierarchical navigation and display as a list of links.
- Managing Tabs
  If your application uses an older theme, add navigation by creating tabs.
- Managing a Classic Navigation Bar
  If your application uses an older theme, add navigation by creating classic navigation bar entries.

> **See Also:**
>
> - Managing Trees
> - Adding Search to an Application

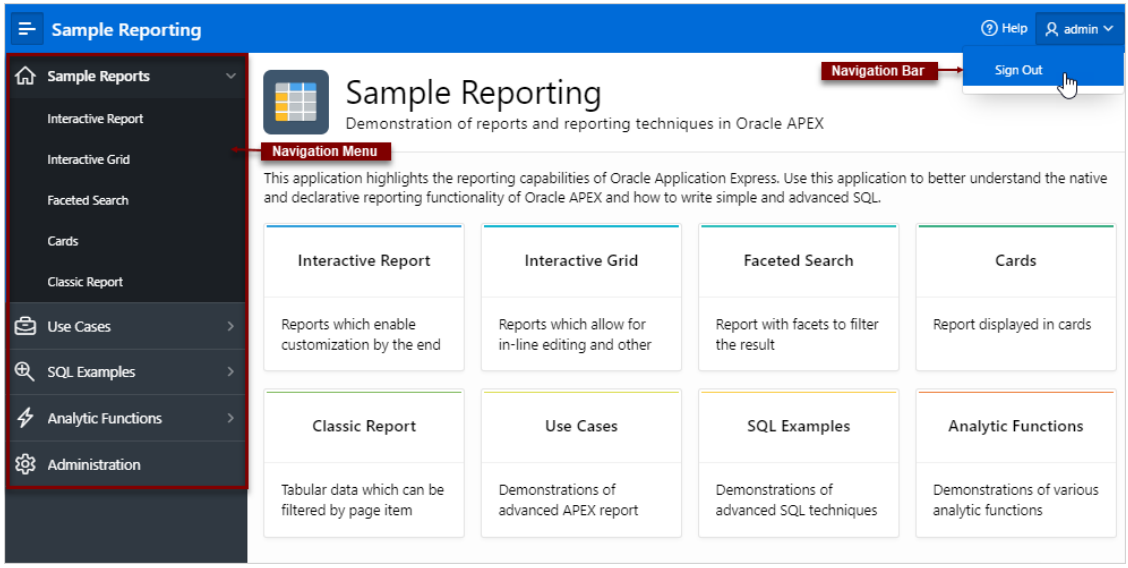
14.1 About Using Lists as Navigation

Add navigation to your application by creating a list (or a shared collection of links). Newer themes, such as *Universal Theme - 42*, support the creation of navigation menus and navigation bars.

You add a list to a page by creating a region and specifying the region type as List. You control the appearance of a list through list templates. For example, you can create a static list that functions as navigation using the template **Vertical Unordered List with Bullets**. **Navigation menus** are lists that render at the top of the page (similar to tabs) or display as a side bar. A

navigation bar displays with a list template in the `#NAVIGATION_BAR#` position on your page template.

The *Sample Reporting* sample app, shown below, includes both a navigation menu and a navigation bar.



Tip:

To install the sample app, *Sample Reporting*, see Installing Apps from the Gallery.

See Also:

- Creating Lists
- Managing Navigation Menus
- Managing Navigation Bar Lists

14.2 Controlling Navigation Using Branches

Add navigation to application pages by creating branches. A branch is an instruction to link to a specific page, procedure, or URL after a given page is submitted.

- Creating a Branch
  Create a branch in Page Designer.
- About Branching Conditionally
  Make a branch conditional by editing Server-side Condition attributes in Page Designer.

> **See Also:**
>
> Accessing Page Specific Utilities

14.2.1 Creating a Branch

Create a branch in Page Designer.

To create a branch:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create a branch in the left pane:
   - **Rendering** tab - Right-click and select **Create Branch**
   - **Processing** tab - Right-click and select **Create Branch**.

   The Property Editor displays the Branch attributes.
3. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
4. In the Property Editor, edit the attributes:

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

   a. **Identification**, **Name** - Specify the name of the branch for easy identification by developers.
   b. **Execution Options**:
      i. **Sequence** - Specify the sequence for this component. The sequence determines the order of execution.
      ii. **Point** - Select the point in the processing at which this branch action is considered. Options include:
         - **After Submit** (previously named **Before Computation**)
         - **Validating** (previously named **Before Validation**)
         - **Processing** (previously named **Before Processing**)
         - **After Processing** (previously named **After Processing**)

- **Before Header** (previously named **Before Header**)

c. **Behavior**, **Type** - Select a page branch type. To learn more, click the **Help** tab in the central pane.

   The attributes that display in the Property Editor depend on the branch type you select.

   If Type is **Page or URL (Redirect)**, click **Target** and edit the attributes in the Link Builder Target dialog.

d. **Server-Side Condition**:

   i. **When Button Pressed** - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.

   ii. **Condition Type** - Select a condition type from the list that must be met in order for this component to be rendered or processed. To learn more, click the **Help** tab in the central pane.

5. Click **Save**.

> **See Also:**
>
> - About the When Button Pressed Attribute
> - Creating Pre-Rendering Branches, Computations, and Processes
> - Creating Page Submission Branches, Validations, Computations, and Processes

14.2.2 About Branching Conditionally

Make a branch conditional by editing Server-side Condition attributes in Page Designer.

To make a branch conditional, view the branch attributes in the Property Editor. Under **Server-side Condition**, edit the appropriate attributes.

14.3 Creating Breadcrumbs

Breadcrumbs provide hierarchical navigation and display as a list of links.

- About Breadcrumbs
  Use breadcrumbs to create hierarchical list of links that indicates where the user is within the application from a hierarchical perspective.
- Creating a Breadcrumb While Creating a Page
  Create a breadcrumb when creating a new page.
- Creating a Breadcrumb Manually
  Create a breadcrumb while creating a new page, as a Shared Component using the Create Breadcrumb Wizard, or by adding a the breadcrumb region in Page Designer. Add breadcrumb entries from Shared Components.
- Editing Breadcrumbs
  Edit breadcrumbs and breadcrunb entries in Shared Components and edit breadcrumb attributes in Page Designer.

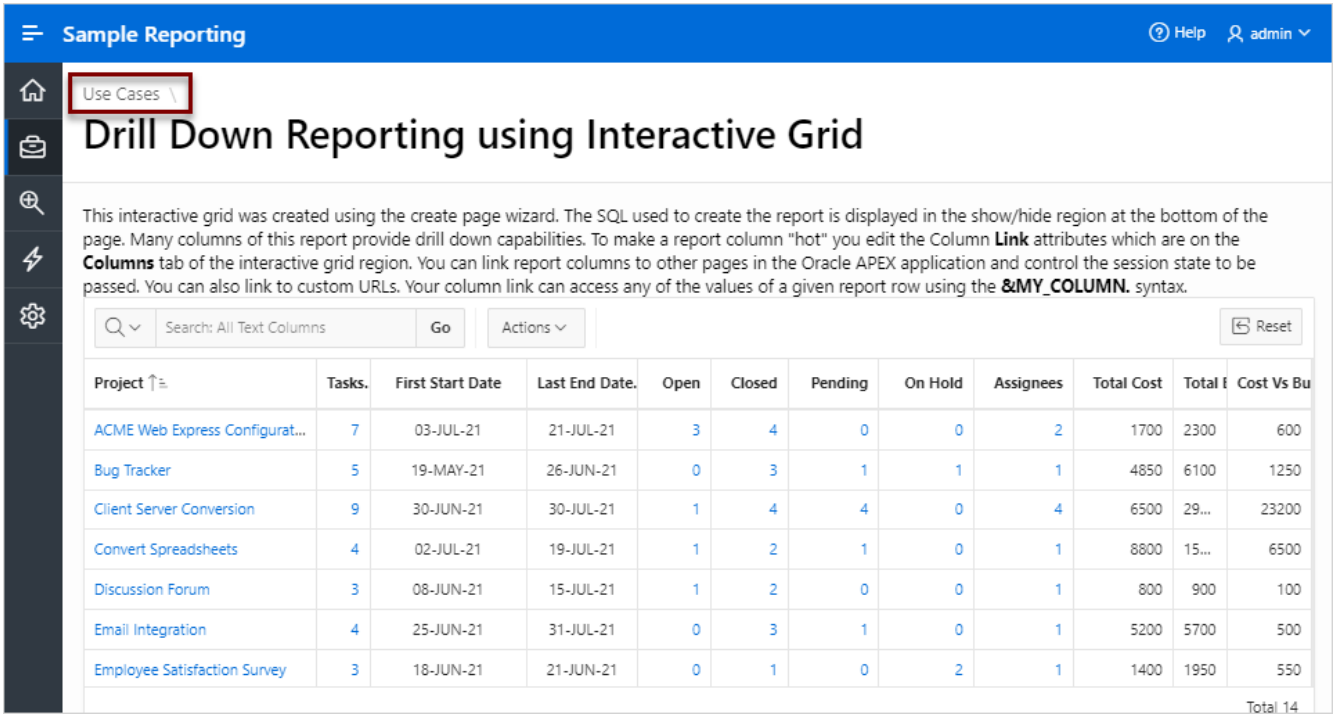
- Accessing Breadcrumb Reports
  Access the Grid Edit, Utilization, and History reports by clicking the appropriate tab at the top of the Breadcrumbs page.

> **See Also:**
>
> Breadcrumb Templates

14.3.1 About Breadcrumbs

Use breadcrumbs to create hierarchical list of links that indicates where the user is within the application from a hierarchical perspective.

A breadcrumb provides navigational context to end users and offer an easy navigation path back to the app home page. Users can click a specific breadcrumb link to instantly view the target page. Within the Universal Theme, breadcrumbs have a special location for placement and should generally be used with the Title Bar region template. A breadcrumb can also display within a blank region. The following example shows breadcrumb in the Sample App, *Sample Reporting*.



14.3.2 Creating a Breadcrumb While Creating a Page

Create a breadcrumb when creating a new page.

Add a breadcrumb while creating a new page using the Create Page Wizard. If the **Breadcrumb** flag is enabled, the **Page Name** is used as **Breadcrumb Entry Name**.

To add a breadcrumb while running the Create Page Wizard:

1. Run the Create Page Wizard to add a new page.

   During the wizard, enable the **Navigation**, **Breadcrumb** flag. The actual page on which this option displays depends upon the type of page you are creating.

2. Navigation - Expand the region and enable or disable navigation. If either **Use Breadcrumb** or **Use Navigation** is enabled, additional attributes appear.
   - Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   - Breadcrumb Parent Entry - Select parent breadcrumb entry for the current page. To have the breadcrumb display at the top level, select **- No parent entry -**.
   - Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as breadcrumb entry name.
   - Parent Navigation Menu Entry - Identify the parent for this list entry.
   - Icon - Identify a icon class for the icon associated with this list entry. To learn more about an attribute, see field-level Help.

3. Follow the on-screen instructions.

> **See Also:**
>
> Adding a New Page to an Application

14.3.3 Creating a Breadcrumb Manually

Create a breadcrumb while creating a new page, as a Shared Component using the Create Breadcrumb Wizard, or by adding a the breadcrumb region in Page Designer. Add breadcrumb entries from Shared Components.

- About Creating Breadcrumbs Manually
  Create a breadcrumb manually by creating the breadcrumb, adding breadcrumb entries, and then adding the breadcrumb to a page by creating a region.
- Creating Breadcrumbs from Page Designer
  Create a breadcrumb and breadcrumb entries from Page Designer.
- Creating Breadcrumbs from Shared Components
  Create a breadcrumb and breadcrumb entries in Shared Components.
- Adding a Breadcrumb Region with a Wizard
  Add a breadcrumb region from the Gallery.
- Adding a Breadcrumb Region from the Gallery
  Add a breadcrumb region from the Gallery.
- About Creating Dynamic Breadcrumbs

14.3.3.1 About Creating Breadcrumbs Manually

Create a breadcrumb manually by creating the breadcrumb, adding breadcrumb entries, and then adding the breadcrumb to a page by creating a region.

The following summarizes the steps needed to create breadcrumbs manually:

1. Create the breadcrumb.

a. You can create a breadcrumb in two ways:
   - From Shared Components. See Creating Breadcrumbs from Shared Components.
   - From Page Designer. Creating Breadcrumbs from Page Designer.

b. In the Create Breadcrumb dialog, enter a name for the new breadcrumb and click **Create**.

2. Add entries to the breadcrumb.
3. Add the breadcrumb to a page by creating a breadcrumb region.

14.3.3.2 Creating Breadcrumbs from Page Designer

Create a breadcrumb and breadcrumb entries from Page Designer.

Once you create a named breadcrumb, you can then add breadcrumb entries. The breadcrumb name provides a way to name a collection of breadcrumb entries.

To manually create a breadcrumb from Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create the breadcrumb:
   a. In the left pane of Page Designer, click the **Page Shared Components** tab.
   b. Right-click **Breadcrumb** and select **Create**.

      The Create Breadcrumb dialog appears.
   c. Breadcrumb Name - Enter a name for this new breadcrumb. Once you create a named breadcrumb, you can then add breadcrumb entries.
   d. Click **Create**.

   The Edit Breadcrumb page appears displaying the breadcrumb in the left pane.
3. Next, create a breadcrumb entry. Click **New Entry**.

   Breadcrumb Entry appears.
4. Under Breadcrumb Entry:
   a. Page - Select the page where this breadcrumb entry displays.
   b. Parent Entry - Identify the parent of this entry. Any given breadcrumb should have only one root node, identified as an item with no parent ID.
   c. Sequence - Indicate the order in which breadcrumb entries appear. The sequence determines the order of evaluation.
   d. Short Name - Identify the short name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
   e. Synchronize page name and title - Enable this option to update the corresponding Page Name and Title with breadcrumb short name.
   f. Long Name - Identify the long name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.

5. Target - Specify the target location. For the **Target is a** attribute, select either:
   - **Page in this Application**
   - **URL**

   The UI changes depending upon what you select.
6. If the Target is **Page in this Application**:
   a. Target is a - Select **Page in this Application**.
   b. Page - Specify the target page number.

      To reset pagination for that page, select **reset pagination for this page**.
   c. Request - Enter text for which you would like to set the built-in application item called `REQUEST`. This is also the item that is set with the name of a button that was clicked.
   d. Clear Cache - Enter the page numbers (separated by commas) for those pages for which you would like to clear the user's session state.
   e. To set session state (that is, give a listed item a value):
      - Set these items - Enter the list of application item names (separated by commas) for which you would like to set the session state (give the listed item a value).
      - With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`).

      You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).
7. If the Target is **URL**:
   a. Target is a - Select **URL**.
   b. URL Target - Identify URL to go to when the breadcrumb entry is clicked..
8. For Conditions:
   a. Condition Type - Select a condition type from the list. This condition must be met in order for this component to be rendered or processed.
   b. Enter expressions in the fields provided.
9. For Authorization - Optionally select an authorization scheme. This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.
10. For Configuration - Optionally select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.
11. Click **Create Entry**.

    The Edit Breadcrumb page appears. A new entry appears under the breadcrumb in the left pane.
12. Create another breadcrumb entry:
    a. Click **New Entry**.

       The Breadcrumb Entry appears under the breadcrumb in the left pane.
    b. Starting at step 4, repeat previous steps for each breadcrumb entry you want to create.

Next, add the breadcrumb to a page by creating a region.

14.3.3.3 Creating Breadcrumbs from Shared Components

Create a breadcrumb and breadcrumb entries in Shared Components.

Once you create a named breadcrumb, you can then add breadcrumb entries. The breadcrumb name provides a way to name a collection of breadcrumb entries.

To manually create a breadcrumb from Shared Components:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application.
   c. Click **Shared Components**.
   d. Under Navigation and Search, click **Breadcrumbs**.

   The Breadcrumbs page appears.
2. Create the breadcrumb:
   a. Click **Create Breadcrumb**.
   b. In the Create Breadcrumb dialog:

      The Create Breadcrumb dialog appears.
   c. Breadcrumb Name - Enter a name for this new breadcrumb. Once you create a named breadcrumb, you can then add breadcrumb entries.
   d. Click **Create**.

   The Edit Breadcrumb page appears displaying the breadcrumb in the left pane.
3. Next, create a breadcrumb entry. Click **New Entry**.

   Breadcrumb Entry appears.
4. Under Breadcrumb Entry:
   a. Page - Select the page where this breadcrumb entry displays.
   b. Parent Entry - Identify the parent of this entry. Any given breadcrumb should have only one root node, identified as an item with no parent ID.
   c. Sequence - Indicate the order in which breadcrumb entries appear. The sequence determines the order of evaluation.
   d. Short Name - Identify the short name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
   e. Synchronize page name and title - Enable this option to update the corresponding Page Name and Title with breadcrumb short name.
   f. Long Name - Identify the long name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.
5. Target - Specify the target location. For the **Target is a** attribute, select either:
   - **Page in this Application**
   - **URL**

   The UI changes depending upon what you select.
6. If the Target is **Page in this Application**:

a. Target is a - Select **Page in this Application**.

b. Page - Specify the target page number.

   To reset pagination for that page, select **reset pagination for this page**.

c. Request - Enter text for which you would like to set the built-in application item called `REQUEST`. This is also the item that is set with the name of a button that was clicked.

d. Clear Cache - Enter the page numbers (separated by commas) for those pages for which you would like to clear the user's session state.

e. To set session state (that is, give a listed item a value):

   - Set these items - Enter the list of application item names (separated by commas) for which you would like to set the session state (give the listed item a value).
   - With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`).

   You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).

7. If the Target is **URL**:

   a. Target is a - Select **URL**.

   b. URL Target - Identify URL to go to when the breadcrumb entry is clicked..

8. For Conditions:

   a. Condition Type - Select a condition type from the list. This condition must be met in order for this component to be rendered or processed.

   b. Enter expressions in the fields provided.

9. For Authorization - Optionally select an authorization scheme. This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.

10. For Configuration - Optionally select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.

11. Click **Create Entry**.

    The Edit Breadcrumb page appears. A new entry appears under the breadcrumb in the left pane.

12. Create another breadcrumb entry:

    a. Click **New Entry**.

       The Breadcrumb Entry appears under the breadcrumb in the left pane.

    b. Starting at step 4, repeat previous steps for each breadcrumb entry you want to create.

Next, add the breadcrumb to a page by creating a region.

14.3.3.4 Adding a Breadcrumb Region with a Wizard

Add a breadcrumb region from the Gallery.

> **Tip:**
>
> Before creating a breadcrumb region, create the breadcrumb. The wizard will propmt you to create breadcrumb entry if one has not yet been defined.

To create a breadcrumb region with a wizard:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. On Page Designer Toolbar at the top of the page, click the **Create** menu and select **Breadcrumb Region**.

   The Create Breadcrumb Wizard appears.
3. For Region Attributes:
   a. Region Title - Enter a title for the region such as `Breadcrumb`.
   b. Region Template - Select a region template.

      For the Universal Theme, the application breadcrumb has a special location for placement. Oracle recommends using **Title Bar** region template.
   c. Display Point - Identify the display point.
   d. Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
   e. Click **Next**.
4. Identify the Breadcrumb:
   a. Breadcrumb - Select the breadcrumb to be associated with this region.
   b. Breadcrumb Template - Identify breadcrumb template.
   c. Click **Next**.
5. If no Breadcrumb Entry has been created, the wizard prompts you to create one. For Breadcrumb Entry:
   a. Breadcrumb Entry Label - Specify the label for this breadcrumb entry.
   b. Parent Entry - Select this breadcrumb entry's hierarchical parent..
   c. Click **Next**.
6. Confirm your selections and click **Finish**.

14.3.3.5 Adding a Breadcrumb Region from the Gallery

Add a breadcrumb region from the Gallery.

> **Tip:**
>
> Before creating a breadcrumb region, Oracle recommends creating the breadcrumb and adding breadcrumb entries.

To create a breadcrumb region from the Gallery:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

   > **Note:**
   >
   > Note the Gallery at the bottom of the page. Pass the cursor over a control or component to view a tooltip that describes it.

2. From the Gallery at the bottom of the page:
   a. Click the **Regions** tab.
   b. Right-click **Breadcrumb**.
   c. Select **Add To** and then select the appropriate location such as **Breadcrumb Bar**.

   > **Tip:**
   >
   > You can also select the Breadcrumb region in the Gallery and drag it to the appropriate location in the Layout tab.

   Page Designer indicates what actions are required next. If there is an error, a **Messages** icon displays next to the **Save** button. Click the **Messages** icon to view a dialog listing all the errors. Select an error to highlight the associated attribute in the Property Editor.

3. In the Property Editor, edit the following Region attributes:

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.

   a. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.

b. Source, Breadcrumb - Select the breadcrumb to be added to the page.

c. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.

d. Layout, Position - Select the template position used to display this region. The selections available are derived from the appropriate template positions defined within the current theme, for the page's user interface.

e. Appearance, Template - Select a region template to define the appearance and layout of this region.

   For the Universal Theme, the application breadcrumb has a special location for placement. Oracle recommends using **Title Bar** region template.

f. Appearance, Render Components - Select if the region components display above or below the main region content.

4. Click **Save**.

Repeat these procedures for each page where you would like to add breadcrumbs.

14.3.3.6 About Creating Dynamic Breadcrumbs

To give users more exact context, you can include session state in breadcrumbs, making your breadcrumbs dynamic. For example, suppose a page in your application displays a list of orders for a particular company and you want to include the following breadcrumb:

```
Home > Orders > Orders for ACME Inc
```

In this example, `ACME Inc` not only indicates the page a user is on but also the navigation path. The Oracle APEX engine stores the value of ACME Inc. in session state.

To create this type of dynamic menu, you must include a reference to a session state item in the breadcrumb's short name or long name. For example:

```
&COMPANY_NAME.
```

14.3.4 Editing Breadcrumbs

Edit breadcrumbs and breadcrunb entries in Shared Components and edit breadcrumb attributes in Page Designer.

- Editing Breadcrumb and Breadcrumb Entries
  Edit breadcrumbs and breadcrumb entries on the Breadcrumbs page in Shared Components.
- Adding Breadcrumb Entries
  Add breadcrumb entries on the Breadcrumb Entry page.
- Reparenting Breadcrumb Entries
  Select a new parent for selected breadcrumb entries on the Reparent Entries page.
- Deleting Unused Breadcrumb Entries
  Delete unused breadcrumb entries.
- Editing Breadcrumb Attributes in Page Designer
  Edit Region attributes and breadcrumb Attributes in Page Designer.

14.3.4.1 Editing Breadcrumb and Breadcrumb Entries

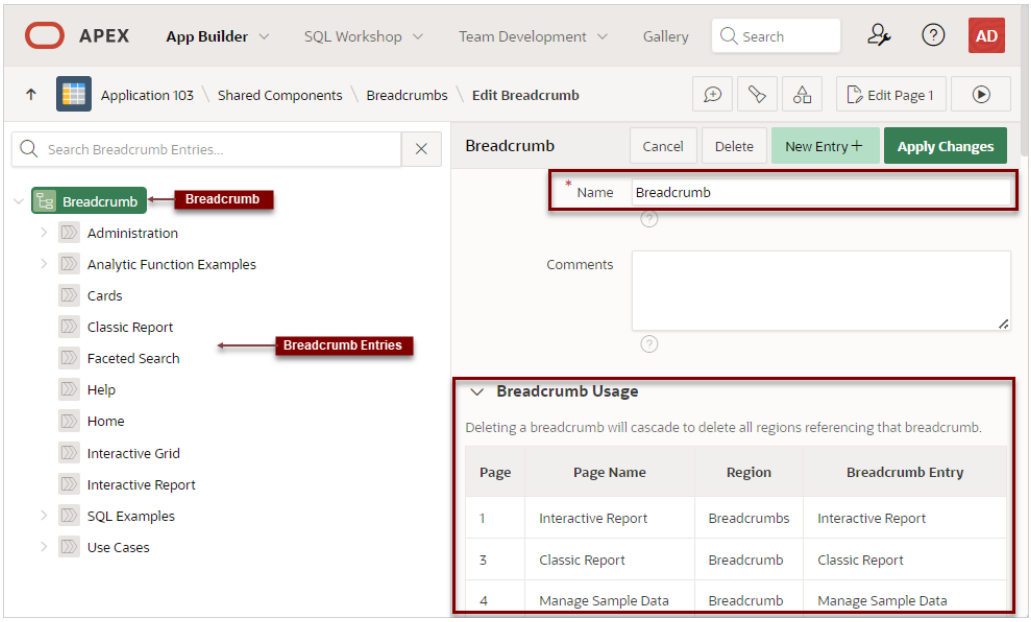
Edit breadcrumbs and breadcrumb entries on the Breadcrumbs page in Shared Components.

To edit a breadcrumb:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, select **Breadcrumbs**.

      The Breadcrumbs page appears.
2. To edit the breadcrumb name:
   a. On the Breadcrumbs page, select the breadcrumb.

      The Edit Breadcrumb page appears. A breadcrumb tree displays in the left pane.



   b. Select the root node.

      The Breadcrumb attributes appear in the right pane.
   c. Edit the Name attribute and click **Apply Changes**.
3. To delete a breadcrumb:
   a. On the Breadcrumbs page, select the breadcrumb.

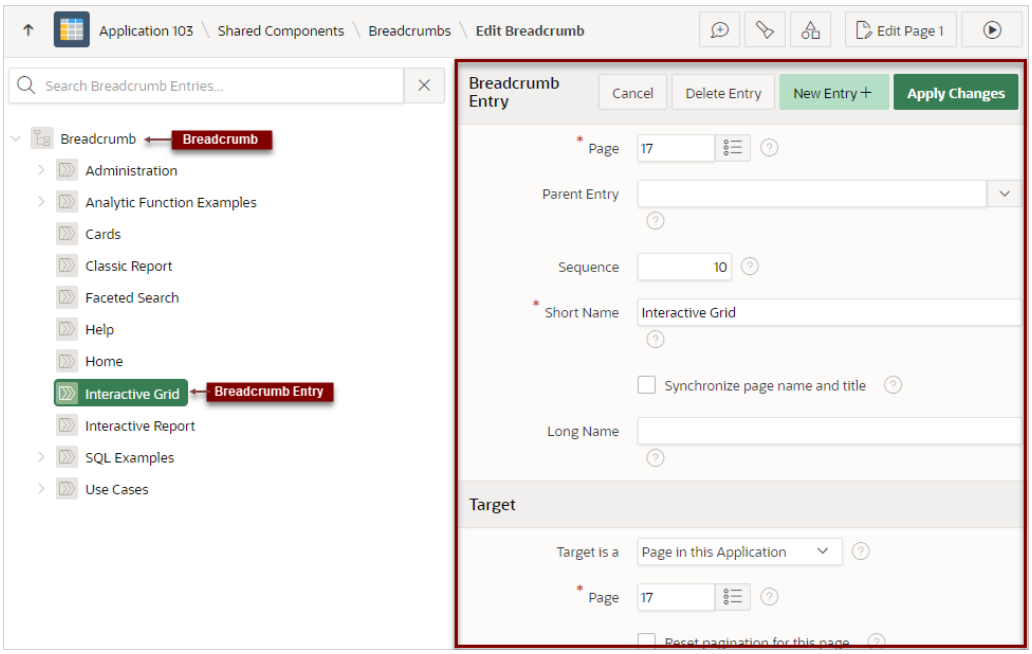
      The Edit Breadcrumb page appears. A breadcrumb tree displays in the left pane.
   b. Select the parent node.

      The Breadcrumb attributes appear in the right pane.
   c. Expand and review the **Breadcrumb Usage** region. Deleting a breadcrumb delete all regions referencing that breadcrumb.
   d. Click **Delete**.

4. To edit breadcrumb entries:
   a. On the Breadcrumbs page, select the breadcrumb.

      The Edit Breadcrumb page appears. A breadcrumb tree displays in the left pane.
   b. In the breadcrumb tree, select a breadcrumb entry.

      The Breadcrumb Entry attributes appear in the right pane.



   c. Edit the appropriate Breadcrumb Entry attributes.

      To learn more about a specific attribute, see field-level Help.
   d. Click **Apply Changes**.

14.3.4.2 Adding Breadcrumb Entries

Add breadcrumb entries on the Breadcrumb Entry page.

To add entries to a breadcrumb:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Navigation and Search, click **Breadcrumbs**.

      The Breadcrumbs page appears.
2. In the left pane, select the breadcrumb to which to add entries.
3. Click **New Entry**.

   The Edit Breadcrumb page appears.
4. Under Breadcrumb Entry:
   a. Page - Select the page where this breadcrumb entry displays.

b. Parent Entry - Identify the parent of this entry. Any given breadcrumb should have only one root node, identified as an item with no parent ID.

c. Sequence - Indicate the order in which breadcrumb entries appear. The sequence determines the order of evaluation.

d. Short Name - Identify the short name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.

e. Long Name - Identify the long name of this breadcrumb entry. Both the short and long names can be referenced from the breadcrumb template.

5. Under Target, specify the target location. For the **Target is a** attribute, select either:

   - **Page in this Application**
   - **URL**

   The UI changes depending upon what you select.

6. If the target is a page in the application:

   a. Target is a - Select **Page in this Application**.

   b. Page - Specify the target page number.

      To reset pagination for that page, select **reset pagination for this page**.

   c. Request - Enter text for which you would like to set the built-in application item called `REQUEST`. This is also the item that is set with the name of a button that was clicked.

   d. Clear Cache - Enter the page numbers (separated by commas) for those pages for which you would like to clear the user's session state.

   e. To set session state (that is, give a listed item a value):

      - Set these items - Enter the list of application item names (separated by commas) for which you would like to set the session state (give the listed item a value).
      - With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`).

      You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).

7. If the target is a URL:

   a. Target is a - Select **URL**.

   b. URL Target - Identify URL to go to when the breadcrumb entry is clicked..

8. For Conditions:

   a. Condition Type - Select a condition type from the list. This condition must be met in order for this component to be rendered or processed.

   b. Enter expressions in the fields provided.

9. For Authorization - Optionally select an authorization scheme. This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.

10. For Configuration - Select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.

11. Click **Create Entry**.

Repeat these procedures for each breadcrumb entry you create.

14.3.4.3 Reparenting Breadcrumb Entries

Select a new parent for selected breadcrumb entries on the Reparent Entries page.

To reparent breadcrumb entries:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Breadcrumbs**.

      The Breadcrumbs page appears.
2. On the Breadcrumbs page, select the breadcrumb.

   The Edit Breadcrumb page appears. A breadcrumb tree displays in the left pane.
3. Click the **Reparent Entries** tab.

   The Reparent Entries page appears.
4. You can reparent entries in two ways:
   - Drag and Drop - In the breadcrumb tree, drag and drop entries to a new location in the tree.
   - Reparent Context Menu - Right-click the entry and select **Reparent**. In the Select a Parent Node dialog, select a new location..

14.3.4.4 Deleting Unused Breadcrumb Entries

Delete unused breadcrumb entries.

> **Tip:**
>
> The **Delete Unused** button described in this section is conditional and only appears if the application has unused (or orphaned) breadcrumb entries.

To delete unused breadcrumb entries:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, select **Breadcrumbs**.

      The Breadcrumbs page appears.
2. Click the **Utilization** tab.

   The Utilization report appears. The Orphaned Breadcrumb Entries region displays any unused breadcrumbs.
3. To delete unused breadcrumb entries, click **Delete Unused**.

   Delete Unused only appears if there are orphaned breadcrumb entries.

14.3.4.5 Editing Breadcrumb Attributes in Page Designer

Edit Region attributes and breadcrumb Attributes in Page Designer.

Region attributes define the area of the page which functions as a container for the breadcrumb. Edit **breadcrumb Attributes** to change the Breadcrumb Template, Template Options, and CSS Classes.

To edit region attributes:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. To edit Region attributes:
   a. In the Rendering tab, select the region containing the breadcrumb.
   b. The Property Editor displays the region attributes in the **Region** tab.
   c. Edit the appropriate attributes.

      To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
3. To edit breadcrumb Attributes:
   a. In the Rendering tab, select the region containing the breadcrumb.

      The Property Editor displays the region attributes in the **Attributes** tab.
   b. Edit the appropriate attributes.

      To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
4. Click **Save**.

14.3.5 Accessing Breadcrumb Reports

Access the Grid Edit, Utilization, and History reports by clicking the appropriate tab at the top of the Breadcrumbs page.

> **Note:**
>
> Breadcrumb reports only appear after you create a breadcrumb.

To view Breadcrumb reports:

1. Navigate to the Breadcrumbs page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

d. Under Navigation and Search, select **Breadcrumbs**.

   The Breadcrumbs page appears.

2. Click the appropriate tab:
   - **Grid Edit** - Edit all breadcrumb entry names at once. Click the page number to link to a page.
   - **Utilization** - View the Breadcrumb Utilization report. This report lists breadcrumbs by page. Click the page number to go to a specific page.
   - **History** - View the Breadcrumb History report. This report lists recent changes to breadcrumbs.

14.4 Managing Tabs

If your application uses an older theme, add navigation by creating tabs.

> **Tip:**
>
> Tabs only appear in applications using an older theme. Applications using the *Universal Theme - 42* theme provide navigation support through the creation of navigation menus.

- About Tabs
  For applications that use an older theme, tabs are an effective way to navigate users between pages.
- About the Tabs Page
  The Tabs page describes the tabs defined in your application.
- Checking for Tab Template Support
  Prior to creating parent and standard tabs, check that your default template has positions defined for both standard and parent tabs using the appropriate substitution strings.
- Creating a Tab
  Create tabs on the Tabs page.
- Editing Tabs
  Edit tabs on the Tabs page.
- Accessing Tab Reports
  Access the following reports by clicking the tabs at the top of the Tabs page: Conditional Display, Utilization, and History.

> **See Also:**
>
> - Creating Lists
> - Managing Navigation Menus
> - Creating Applications

14.4.1 About Tabs

For applications that use an older theme, tabs are an effective way to navigate users between pages.

The ability to include tabs in your application depends upon your application theme. Older themes, such as such as *Theme 26 - Productivity Applications*, include tab-based navigation. Newer themes, such a *Universal Theme - 42*, include list-based navigation menus.

An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you define both Parent tabs and Standard tabs.

> **Tip:**
>
> As an alternative to tabs, you can use lists to display tab controls. List templates provide greater control over HTML generation.

App Builder includes two different types of tabs:

- **Standard tabs**

  An application having only one level of tabs uses a standard tab set. A standard tab can have a one-to-one relationship with a page and is associated with a specific page and page number. You can use standard tabs to link users to a specific page.

- **Parent tabs**

  Parent tabs can control the display of standard tab sets and can be current for many pages. Clicking a parent tab displays the corresponding standard tab, with the default page as the current page. Parent tabs give users another level of navigation and a context (or sense of place) within the application. You can use parent tabs to link users to a specific URL associated with a specific page.

You can group tabs into collections called a tab set. Each tab must be part of a tab set.

14.4.2 About the Tabs Page

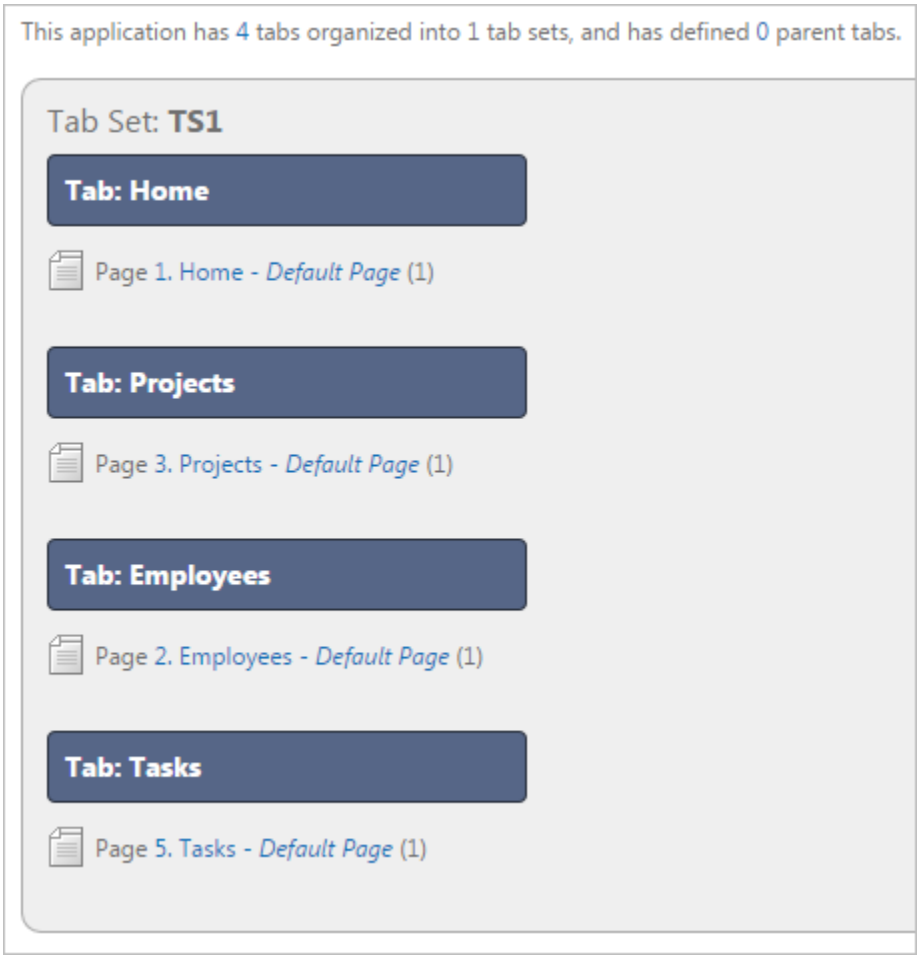
The Tabs page describes the tabs defined in your application.

To access the Tabs page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Navigation, click **Tabs**.

   The Tabs page appears.



The Tab Display section provides additional information about how the current tabs or tab set are configured to display.

The Page Templates region at the bottom of the page displays a report of page templates associated with the current application. This region is hidden by default. To view it, click the greater than (>) icon to the left of Page Templates.

> **See Also:**
>
> Checking for Tab Template Support

14.4.3 Checking for Tab Template Support

Prior to creating parent and standard tabs, check that your default template has positions defined for both standard and parent tabs using the appropriate substitution strings.

You also must make sure you do not override this template at the page-level.

To view page templates:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.

   b. Select an application.
   c. On the Application home page, click **Shared Components**.
2. Under Navigation, click **Tabs**.

   The Tabs page appears.
3. Locate **Page Templates** at the bottom of the page. If needed, expand the Page Templates region.

Page Templates

Page Template Name	Supports Standard Tabs	Supports Parent Tabs	Template Class
Login	No	No	Login
No Tabs - Left Sidebar (fixed-width / DIV based)	No	No	No Tabs with Sidebar
No Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right)	No	No	No Tabs with Sidebar
No Tabs - Right Sidebar (fixed-width / DIV based)	No	No	No Tabs
No Tabs - Right Sidebar (optional / table-based)	No	No	No Tabs
One Level Tabs - Left Sidebar (fixed-width / DIV based)	Yes	No	One Level Tabs with Sidebar
One Level Tabs - Left and Right Sidebar (fixed-width / DIV left and optional table-based right)	Yes	No	One Level Tabs with Sidebar

See Also:

- Page Templates
- Using Themes for information about setting a default page template at the application level

14.4.4 Creating a Tab

Create tabs on the Tabs page.

An application can have pages with no tabs, one level of tabs, and two levels of tabs. Standard tabs enable you to display only one level of tabs. To display two levels of tabs, you must define both Parent tabs and Standard tabs.

To create a tab:

1. Access the Tabs page:
   a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

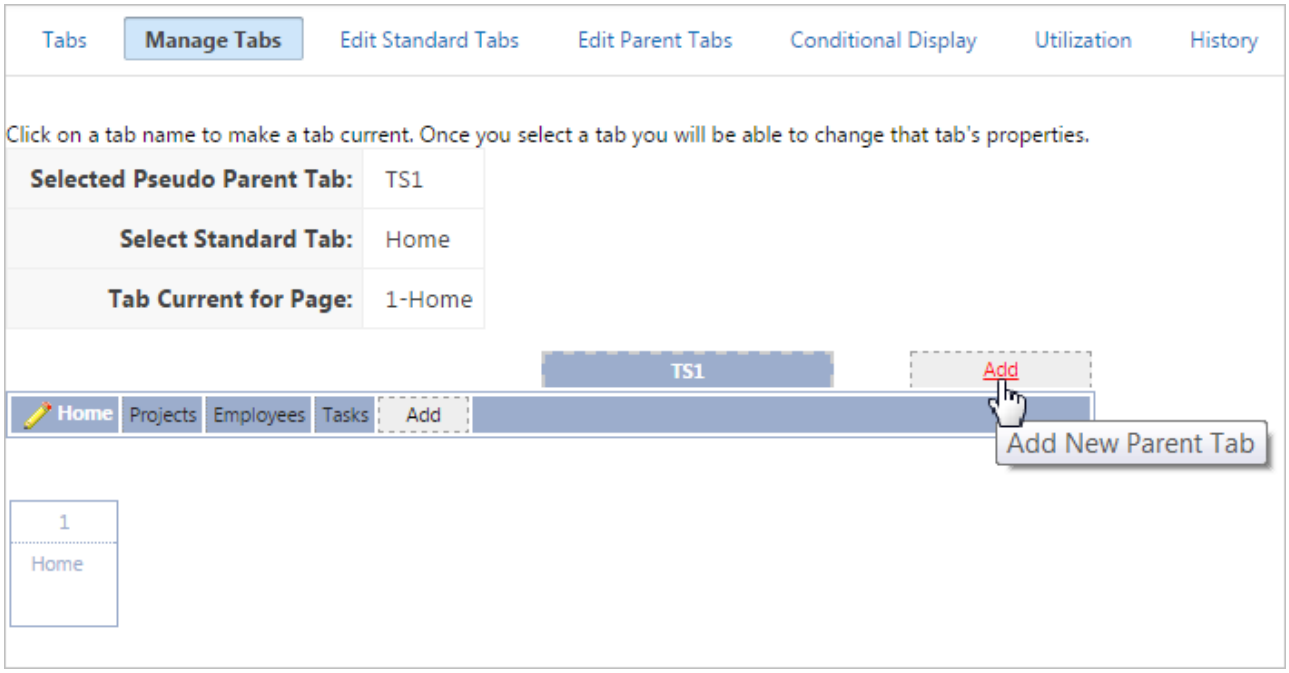
c. On the Application home page, click **Shared Components**.

d. Under Navigation, click **Tabs**.

2. Click **Manage Tabs**.

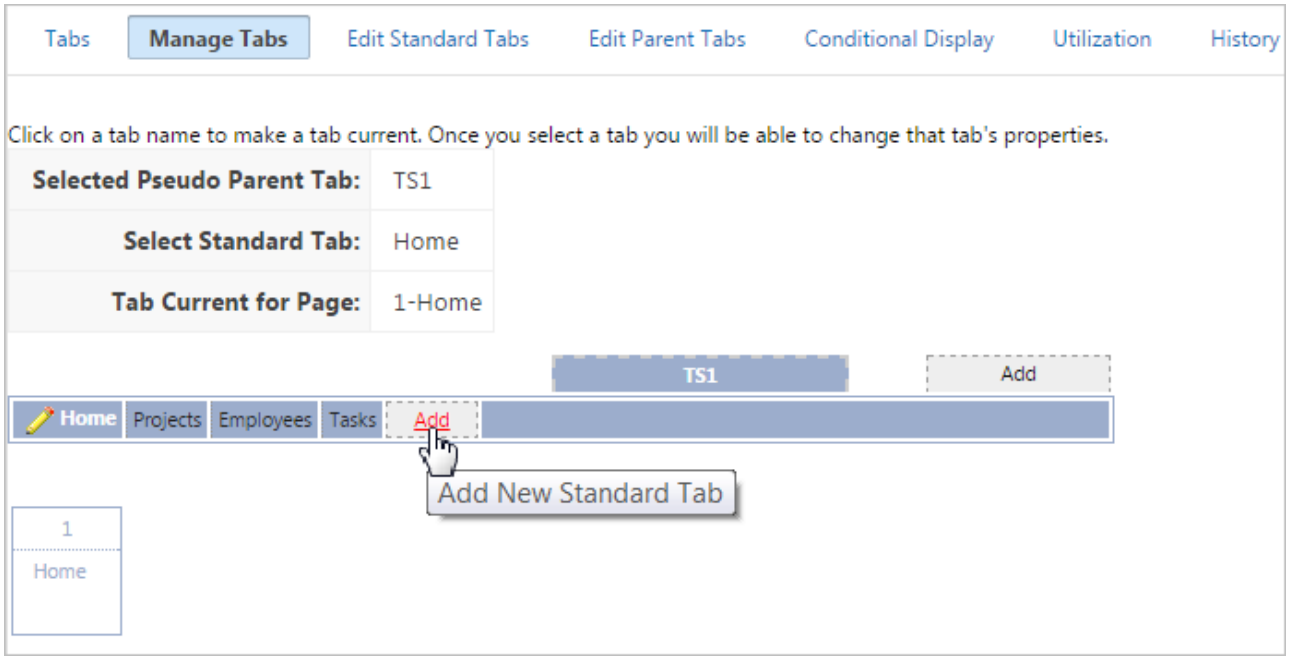
3. To add a Parent tab, click the **Add** button in the upper row.

   A graphical representation of the tabs defined in your application displays.



   Parent tabs can control the display of standard tab sets. Clicking a parent tab displays the corresponding standard tab, with the default page as the current page.

4. To create a Standard tab, click the **Add** button in the lower row.



The Create Parent Tab or Create Standard Tab Wizard appears.

5. Follow the on-screen instructions.

> **See Also:**
>
> About Tabs

14.4.5 Editing Tabs

Edit tabs on the Tabs page.

- Editing Multiple Tabs Simultaneously
  Learn about editing multiple tabs simultaneously.
- Managing Standard Tabs
  Learn about accessing the Standard Tab Task list.

14.4.5.1 Editing Multiple Tabs Simultaneously

Learn about editing multiple tabs simultaneously.

To edit multiple tabs simultaneously:

1. Navigate to the Tabs page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Navigation, click **Tabs**.

2. Click one of the following tabs at the top of the page:
   - **Edit Standard Tabs**
   - **Edit Parent Tabs**

   A report appears.
3. To edit a specific tab, click the **Edit** icon.
4. Edit the appropriate attributes and click **Apply Changes**.

14.4.5.2 Managing Standard Tabs

Learn about accessing the Standard Tab Task list.

The Standard Tab Task list displays on the right side of the Tabs page. You can access the links on this list to move a standard tab to different parent tab, rename a standard tab set, resequence the display order, associate pages with a tab set, create a new standard tab, or create a new standard tab set.

To access the Standard Tab Task list:

1. Navigate to the Tabs page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Navigation, click **Tabs**.
2. Click **Manage Tabs**.
3. Make a selection from the Standard Tab Task list on the right side of the page:
   - **Move Highlighted Standard Tab t a Different Parent Tab**
   - **Rename Standard Tab Set**
   - **Resequence Display Order**
   - **Associate Page(s) with Selected Standard Tab**
   - **Create New Standard Tab**
   - **Create New Standard Tab Set**

14.4.6 Accessing Tab Reports

Access the following reports by clicking the tabs at the top of the Tabs page: Conditional Display, Utilization, and History.

This section describes the Conditional Display, Utilization, and History reports.

To view the Conditional Display report:

1. Navigate to the Tabs page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Tabs**.

The Tabs page appears.

2. Click the appropriate tab:
   - **Conditional Display** - The Conditional Display report displays Standard Tabs and Parent tabs that are configured to display conditionally. To filter the display, select tab type and click **Go**.
   - **Utilization** - The Utilization report lists the standard tabs used in the current application.
   - **History** - History report displays a history of changes to tab attributes for the current application.

14.5 Managing a Classic Navigation Bar

If your application uses an older theme, add navigation by creating classic navigation bar entries.

> **Tip:**
>
> Applications using the *Universal Theme - 42* theme provide navigation support through the creation of navigation menus.

- About Classic Navigation Bar Entries
  For applications using older themes, navigation bar entries offer an easy way to move users between application pages.
- Creating a Classic Navigation Bar Entry
  Before adding a navigation bar, you must create entries for the navigation bar.
- Copying a Classic Navigation Bar Entry
  Copy a classic navigation bar entry by selecting **As a Copy of an Existing Navigation Bar** in the Create Navigation Bar Entry Wizard.
- Editing a Classic Navigation Bar Entry
  Edit an existing navigation bar entry on the Navigation Bar Entries page accessible from Shared Components..
- Editing Multiple Classic Navigation Bar Entries Simultaneously
  Edit multiple navigation bar entries simultaneously.
- Accessing Classic Navigation Bar Entry Reports
  Access the Subscription and History reports by clicking the tabs at the top of the Navigation Bar Entries page.

> **See Also:**
>
> - Creating Custom Themes
> - Managing Navigation Menus

14.5.1 About Classic Navigation Bar Entries

For applications using older themes, navigation bar entries offer an easy way to move users between application pages.

The associated page template determines the location of a navigation bar. A navigation bar entry can be an image, text, or an image with text beneath it. You must supply the images and text to use in the navigation bar entries.

Navigation bars are different from other shared components in that you do not need to reference them on a page-by-page basis. If your page template includes the `#NAVIGATION_BAR#` substitution string, the Oracle APEX engine automatically includes any defined navigation bars when it renders the page.

14.5.2 Creating a Classic Navigation Bar Entry

Before adding a navigation bar, you must create entries for the navigation bar.

You can create a navigation bar entry from scratch or by copying an existing entry.

- Creating a Classic Navigation Bar Entry from Scratch
  Learn about creating a classic navigation bar entry from scratch.
- Creating a Classic Navigation Bar Entry for Feedback
  Learn about creating a classic navigation bar entry for feedback,

14.5.2.1 Creating a Classic Navigation Bar Entry from Scratch

Learn about creating a classic navigation bar entry from scratch.

To create a navigation bar entry from scratch:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

   The Create Navigation Bar Entry Wizard appears.
3. For Method, select **From Scratch**.
4. For Type, select **Navigation to URL**.
5. For Attributes, specify the following:
   a. Sequence - Specify the order of evaluation for this component.

> **Tip:**
>
> To review the existing entries, click the **Existing Navigation Bar Entries** link at the bottom of the page.

b. Entry Label - Enter display text for this navigation bar entry.

c. Icon Image Name - Enter an image name. For naming conventions, see field-level Help.

d. Image ALT - Enter ALT text for navigation icons that are images. If you do not specify an image name, then this text displays.

e. Image Height - Define the height of the image in pixels.

f. Width - Defines the width of the image.

g. Click **Next**.

6. For Target, specify the following

   a. If the target location is a URL, specify:

      - Target is a - Select **URL**.
      - URL Target - Enter a URL. For example:

        `http://www.yahoo.com`

   b. If the target location is a page, specify:

      - Target is a - Select **Page in this Application**.
      - Page - Specify the target page number.

        To reset pagination, select **reset pagination for this page**.

        Select **Printer Friendly** to display the target page using the application's Printer Friendly template. Printer friendly templates optimize a page for printing.
      - Request - Enter text that defines the built-in application item called `REQUEST`.
      - Clear Cache - Enter the page numbers where you would like to clear the user's session state. Separate multiple entries with commas.

   c. To set session state (that is, give a listed item a value):

      - Set these items - Enter the list of application item names for which you would like to set session state. Separate multiple items with commas. Setting the session state gives the listed item a value.
      - With these values - Enter a comma-delimited list of values for the items your specified.

        You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).

   d. Click **Next**.

7. To set a condition for displaying the navigation bar entry, select a condition type.

8. Click **Create**.

> **Tip:**
>
> In order for a navigation bar to appear, your page template must include the `#NAVIGATION_BAR#` substitution string. To learn more, see Page Templates.

See Also:

- Creating a Classic Navigation Bar Entry for Feedback
- Viewing Field-Level and Page Designer Help
- Editing Themes
- Optimizing a Page for Printing

14.5.2.2 Creating a Classic Navigation Bar Entry for Feedback

Learn about creating a classic navigation bar entry for feedback,

Feedback is the process of gathering real-time comments, enhancement requests, and bugs from your application users. To add a feedback page to an application, you add a Feedback page.

To create a navigation bar entry for feedback:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

   The Create Navigation Bar Entry Wizard appears. Each wizard page displays a series of blocks on the left, representing the sequence of steps. The highlighted block indicates the step you are performing.
3. For Method, select **From Scratch**.
4. For Type, select **Feedback**.
5. For Feedback:
   a. Feedback Page - Select your feedback page.
   b. Entry Label - Enter the label of the new navigation bar entry.
   c. Display Sequence - The sequence for the new navigation bar entry. The sequence determines the order of evaluation.
6. Click **Create**.

Tip:

In order for a navigation bar to appear, your page template must include the `#NAVIGATION_BAR#` substitution string. SeePage Templates.

> **See Also:**
>
> Managing Feedback

14.5.3 Copying a Classic Navigation Bar Entry

Copy a classic navigation bar entry by selecting **As a Copy of an Existing Navigation Bar** in the Create Navigation Bar Entry Wizard.

To copy a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click **Create**.

   The Create Navigation Bar Entry Wizard appears.
3. For Method, select **As a Copy of an Existing Navigation Bar** and click **Next**.
4. For Copy From Application, select an application and click **Next**.
5. For New Names, enter the name on the new entry, and select whether to copy and subscribe.
6. Click **Copy Navigation Bar Entries**.

14.5.4 Editing a Classic Navigation Bar Entry

Edit an existing navigation bar entry on the Navigation Bar Entries page accessible from Shared Components..

To edit a navigation bar entry:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Select a navigation bar entry.

   The Edit page appears.
3. Edit the appropriate attributes.
4. To learn more about a specific item on a page, see field-level Help.
5. Click **Apply Changes**.

> **See Also:**
>
> Viewing Field-Level and Page Designer Help

14.5.5 Editing Multiple Classic Navigation Bar Entries Simultaneously

Edit multiple navigation bar entries simultaneously.

To edit multiple navigation bar entries simultaneously:

1. Navigate to the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder**.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Select a list.
3. Click **Grid Edit** at the top of the page.
4. Edit the appropriate attributes and click **Apply Changes**.

14.5.6 Accessing Classic Navigation Bar Entry Reports

Access the Subscription and History reports by clicking the tabs at the top of the Navigation Bar Entries page.

> **Note:**
>
> The Subscription and History buttons only appear after you create a navigation bar.

To view Classic Navigation Bar Entry reports:

1. Navigate to the Navigation Bar Entries page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation, click **Classic Navigation Bar Entries**.
2. Click the appropriate tab:
   - **Subscription** - Click **Subscription** to access the Subscribed NavBars report. This report displays subscribed navigation bar entries in your application.
   - **History** - Click **History** to view the Navigation Bar History report. This report lists recent changes to navigation bars.

15

Managing Computations, Validations, and Processes

Enhance your application by including computations, validations, and processes.

Computations enable you to assign values to items.**Validations** are edit checks on specific items, pages, or columns. A **page process** performs an action at a specified point during the rendering or submission of the page. **Application processes** are blocks of PL/SQL logic that are set to run at specific points using processes from multiple pages of an application.

- Understanding Page Computations
  Use computations to assign a value to an identified item when a page is submitted or displayed.
- Understanding Validations
  Use validations to create edit checks on specific items, pages, or columns.
- Understanding Page Processes
  Create **page processes** to perform an action at a specified point during the rendering or submission of the page.
- Understanding Application Processes
  Create an application process to run a block of PL/SQL logic at a specific point from multiple pages of an application.
- About Branches
  A branch is an instruction to go to a specific page, procedure, or URL. For example, you can branch from page 1 to page 2 after page 1 is submitted. When you create a branch, you specify a Branch Point and Branch Type.
- Understanding Application Computations
  Create an application computation to set the value of a single page or application-level item. Application computations run at the same point across multiple pages in an application.
- Using the Attribute Dictionary
  Use the Manage Attribute Dictionary page to update values on the current page with those in the Attribute Dictionary, or to update the Attribute Dictionary with the values on the current page.

15.1 Understanding Page Computations

Use computations to assign a value to an identified item when a page is submitted or displayed.

- About Page Computations
  A page computation assigns a value to an identified item when a page is displayed or submitted (rendered and processed).
- Creating a Page Computation
  Create a Page Computation in Page Designer.

- Editing Page Computation Attributes
  Edit a page computation in Page Designer.
- Editing the Computation Point and Source
  Control when a computation executes by editing the Sequence and Point attributes.
- Example: Computation Points and Computation Syntax
  View computation points and computation syntax example.

> **See Also:**
>
> Understanding Application Computations

15.1.1 About Page Computations

A page computation assigns a value to an identified item when a page is displayed or submitted (rendered and processed).

You create a page computation by running the Create Page Computation Wizard. For each computation, specify the item for which you are creating the computation and a computation type.

You can also use application-level computations to assign values to items. Most page-level computations populate page items. In contrast, most application-level computations populate application items.

15.1.2 Creating a Page Computation

Create a Page Computation in Page Designer.

To create a page computation:

1. Navigate to the appropriate page in Page Designer.
2. In the Rendering tab, expand the **Pre-Rendering** or **Post-Rendering** node and locate where the computation should be.
3. Right-click the location and select **Create Computation**.

   Page Designer appears.

   In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. Edit the following attributes:
   a. Identification, Item Name - Enter the page or application item populated by this computation.
   b. Execution Options, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

c. Execution Options, Point - Select at what point in page rendering or processing this computation is executed. The most typical computation point is **After Submit**.

d. Computation, Type - Select a computation type. To view a list of options, see Help.

5. Click **Save**.

> **See Also:**
>
> - Viewing a Page in Page Designer
> - Creating Pre-Rendering Branches, Computations, and Processes
> - Creating Page Submission Branches, Validations, Computations, and Processes

15.1.3 Editing Page Computation Attributes

Edit a page computation in Page Designer.

To edit a page computation:

1. View the page in Page Designer.
2. In the Rendering tab, expand the Pre-Rendering or Post-Rendering node and select the computation.

   Page Designer appears.

   In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. In the Property Editor, edit the appropriate attributes.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
4. Click **Save**.

> **See Also:**
>
> Viewing a Page in Page Designer.

15.1.4 Editing the Computation Point and Source

Control when a computation executes by editing the Sequence and Point attributes.

You control when a computation executes by editing the **Sequence** and **Point** attributes. The **Sequence** attribute determines the order of evaluation. The **Point** attribute determines the

point in page rendering or processing at which the computation executes. The most typical computation point is After Submit.

To edit a computation point and source:

1. View the page in Page Designer.
2. In the Rendering tab, expand the Pre-Rendering or Post-Rendering node and select the computation.

   Page Designer appears.

   In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
3. In the Property Editor, edit the following attributes.
   a. Execution Options, Sequence - Specify the sequence for this component. The sequence determines the order of execution. If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.
   b. Execution Options, Point - Select at what point in page rendering or processing this computation is executed.

      For example, selecting **On New Instance** executes the computation when a new session (or instance) is generated. The most typical computation point is **After Submit.**
   c. Computation - Define expression or query to compute an item's value.
   d. Error, Error Message - Enter an error message that displays if this computation fails.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
4. Click **Save**.

> **See Also:**
>
> Viewing a Page in Page Designer

15.1.5 Example: Computation Points and Computation Syntax

View computation points and computation syntax example.

A good example of using computations can be illustrated by a page containing form fields for entering phone numbers. In this example, the phone number is stored in one database column; however, the data entry form breaks the phone number into three components: area code, prefix, and line number. In this example, the page items are called `P10_AREA_CODE`, `P10_PREFIX`, and `P10_LINE_NUMBER`.

Next, suppose you need to combine the values stored in these items into a single string. You could accomplish this by using an After Submit computation and store the combined values in an item called `P10_PHONE_NUMBER`.

To create a computation to store the combined values of `P10_AREA_CODE`, `P10_PREFIX`, and `P10_LINE_NUMBER` in new items:

1. Navigate to the appropriate page.
2. Create an item named `P10_PHONE_NUMBER` to store the combined values of `P10_AREA_CODE`, `P10_PREFIX`, and `P10_LINE_NUMBER`.
3. Create the computation:
   a. In the Rendering tab, expand the **Pre-Rendering**.
   b. Select a location, and then right-click and select **Create Computation**.

      Page Designer appears.

      In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

      > **Tip:**
      >
      > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

4. In the Property Editor, edit the following attributes:
   a. Identification, Item Name - Select `P10_PHONE_NUMBER`.
   b. Execution Options, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.
   c. Execution Options, Point - Select at what point in page rendering or processing this computation is executed. The most typical computation point is **After Submit**.
5. For Computation Type, you can create the following computation types:
   a. Static Value:
      - Computation, Type - Select **Static Value** .
      - Computation, Static Value - Enter the following computation:

        ```
        (&P10_AREA_CODE.) &P10_PREFIX.-&P10_LINE_NUMBER.
        ```

   b. PL/SQL Function Body:
      - Computation, Type - Select **PL/SQL Function Body** .
      - Computation, PL/SQL Function Body - Enter the computation:

        ```
        DECLARE
        l_return_value  VARCHAR2(300) DEFAULT NULL;
        BEGIN
            l_return_value :=
        '('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER;
        RETURN l_return_value;
        END;
        ```

c. SQL Query (return colon separated value):
   - Computation, Type - Select **SQL Query (return colon separated value)** .
   - Computation Type, SQL Query - Enter the computation:

```
SELECT '('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER
FROM DUAL
```

d. PLSQL Expression:
   - Computation, Type - Select **PLSQL Expression** .
   - Computation Type, PLSQL Expression - Enter the computation:

```
'('||:P10_AREA_CODE||')'||:P10_PREFIX||'-'||:P10_LINE_NUMBER
```

6. Click **Save**.

> **See Also:**
>
> About the Differences Between Page Items and Application Items

15.2 Understanding Validations

Use validations to create edit checks on specific items, pages, or columns.

- About Validations
  Learn about validations.
- What Happens When a Validation Fails?
  Learn what happens when a validation fails.
- Creating a Validation
  Create validations in Page Designer.
- Editing a Validation
  Edit validations in Page Designer.
- About Determining When Validations Execute
  Determine when a validation executes by editing the Always Execute attribute.
- About Defining How Validation Error Messages Display
  Control how validation messages display by editing the Error Message and Error Message attributes.
- Processing Validations Conditionally
  Control when and if a validation executes by configuring the Server-side Condition attributes.
- About Altering the Go to Error Link
  Alter the Go to Error text to change the default text or to translate it into a language that is not translated by default.

15.2.1 About Validations

Learn about validations.

A validation is an edit check. Validations specific to a single item are **page item validations**. Validations that apply to an entire page are **page validations**. Validations for tabular forms specific to a single column are **column level validations**. Validations for tabular forms that do not apply to a single column are **tabular form row validations**.

You can define a validation declaratively by selecting a validation method. You enter the actual validation edit check in the Validation Messages field. Be aware that if a validation fails, subsequent page processes or computations do not occur. Also remember that the validation you enter must be consistent with the validation type you selected.

15.2.2 What Happens When a Validation Fails?

Learn what happens when a validation fails.

If a page is submitted and some of the validations fail, Oracle APEX redisplays the existing page with all inline validation errors. While displaying inline errors, APEX does not execute computations, application processes, or page processes (for example, Automated Row Fetch) which are defined to execute during Page Rendering (in other words, all "On Load %" display points). The one exception is if the computations and processes use the condition type Inline Validation Errors Displayed.

The reason for skipping these computations and processes is that any of they could potentially alter the data entered by the user. For example, an Automated Row Fetch process would fetch and overwrite the modified data with the values from the database when the user actually wants to get the entered data in order to fix the validation error.

If a computation is skipped, the following displays in the debug output:

```
Do not perform computation because inline validation errors found.
```

If an application or page process is skipped, the following displays in the debug output:

```
Skip because inline validation errors found.
```

15.2.3 Creating a Validation

Create validations in Page Designer.

> **Note:**
>
> Text entered for validations may not exceed 3,950 characters.

To create a validation in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. Right-click **Validations** and select **Create Validation**.

   Validation attributes display in the Property Editor. Attributes are organized in groups.

> **Note:**
>
> To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

4. **Identification**, **Name** - Enter the name of the validation. Make this name should be descriptive so that developers can easily identify conditions being checked.
5. **Execution Options**, **Sequence** - Enter the sequence for this validation. The sequence determines the order of execution.
6. Edit the following **Validation** attributes:
   a. **Validation**, **Editable Region** - Select the associated region.
   b. **Validation**, **Type** - Select the type of equality to be tested for this validation. Depending on your selection, one or more additional attributes are required to fully define this validation.

      If the validation passes the equality test, or evaluates to TRUE, then the validation error message does not display. Validation error messages display when the validation fails the equality test, or evaluates to FALSE, or a non-empty text string is returned.
   c. **Validation**, **Always Execute** - Specify whether this validation always executes.

      If set to **On**, this validation is always evaluated, irrespective of the Execute Validations setting defined against the button that submitted the page. If set to **Off**, this validation is only evaluated if the triggering button has an Execute Validations setting of **On**.
7. Edit the following **Error** attributes:
   a. **Error**, **Error Message** - Enter text for the error message.
   b. **Error**, **Display Location** - Select where the error message displays for this validation.

      Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the Associated Item label and/or in a Notification area, defined as part of the page template.
   c. **Error**, **Associated Item** - Select the item where this validation error message displays.

      If you select an item and the Error message display location includes **Inline with Field**, the error message displays underneath the label of the specified item. If the Error message display location does not include **Inline with Field**, this attribute has no impact on the display of the error message.
8. Click **Save**.

> **See Also:**
>
> - Creating Pre-Rendering Branches, Computations, and Processes
> - Creating Page Submission Branches, Validations, Computations, and Processes

15.2.4 Editing a Validation

Edit validations in Page Designer.

To edit a validation in Page Designer:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. Expand **Validating** and then **Validations** and select the validation.

   Validation attributes display in the Property Editor. Attributes are organized in groups.

   > **Note:**
   >
   > To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. In the Property Editor, edit the appropriate attributes.

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the center pane.
5. Click **Save**.

15.2.5 About Determining When Validations Execute

Determine when a validation executes by editing the Always Execute attribute.

The **Always Execute** attribute determines if validations execute when a page is submitted. Use this attribute for buttons or certain page items that submit a page. The **Always Execute** attribute has two options:

- **On** - The validation always executes independent of validation settings for buttons or items on a page.

  For example, select **On** if your validation is a security check which determines if the current user is allowed to modify or delete the current record. This ensures the validation executes for any operation and prevents unprivileged users from modifying data.
- **Off** - The validation only executes if the button or item used to submit the page has the **Execute Validations** attributes set to **On**.

  For example, it is not useful to verify that an item be numeric and display an error message if the user is simply deleting a record. In that scenario, the validations should only execute when the user clicks the Create or Apply Changes buttons.

To accomplish this behavior:

- For the validation, set **Always Execute** to **On**
- For the Create and Apply Changes buttons, set **Execute Validations** to **On**.
- For the Delete button, set **Execute Validations** to **Off** to avoid unnecessary validations.

> **See Also:**
>
> Editing a Validation

15.2.6 About Defining How Validation Error Messages Display

Control how validation messages display by editing the Error Message and Error Message attributes.

You can choose to have validation error messages display inline (that is, on the page where the validation is performed) or on a separate error page.

You define how a validation error message displays by configuring the following validation attributes:

1. **Error**, **Error Message** - Enter the error message text in this field. You can use `#LABEL#` as placeholder for the label of the associated page item or `#COLUMN_HEADER#` for the heading of the associated editable region column. The placeholders are not supported if error display location is set to On Error Page.
2. **Error**, **c** - Select where the error message displays for this validation. Validation error messages display on a separate error page, or inline with the existing page. Inline error messages display underneath the Associated Item label and/or in a Notification area, defined as part of the page template.

   Options include:

   - **Inline with Field and in Notification** - Error displays inline within the field label and within a notification area (defined as part of the page template).
   - **Inline with Field** - Error displays inline within the field label.
   - **Inline in Notification** - Error displays inline within a notification area (defined as part of the page template).
   - **On Error Page** - Displays the error on a separate error page. To create a hard error that stops all processing (including validations), you must display the error on an error page.
3. **Error**, **Associated Item** (or Column) - Select the item (or column) where this validation error message displays.

   If you select an item and the Error message display location includes **Inline with Field**, the error message displays underneath the label of the specified item. If the Error message display location does not include **Inline with Field**, this attribute has no impact on the display of the error message.

> **See Also:**
>
> Editing a Validation

15.2.7 Processing Validations Conditionally

Control when and if a validation executes by configuring the Server-side Condition attributes.

You can control when and if a validation executes by configuring the following **Server-side Condition** attributes:

1. **Server-side Condition**, **When Button Pressed** - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.

   If no button is pressed, other conditions are evaluated before performing the validation.

2. **Server-side Condition**, **Type** - Select a condition type from the list that must be met in order for this component to be rendered or processed.

   To view a list of options, select the **Type** attribute in the Property Editor and click the **Help** tab in the central pane.

> **See Also:**
>
> - Editing a Validation
> - Understanding Conditional Rendering and Processing

15.2.8 About Altering the Go to Error Link

Alter the Go to Error text to change the default text or to translate it into a language that is not translated by default.

If you define an associated item, Oracle APEX renders a **Go to Error** link in the Notification display location next to the corresponding error message. When clicked, the user's focus is set to the associated item.

You can alter the **Go to Error** text to change the default text or to translate it into a language that is not translated by default with APEX. In either case, define a text message called `APEX.GO_TO_ERROR` with the text and language of your choice. Additionally, you can disable this functionality and remove the **Go to Error** link by setting the text of the `APEX.GO_TO_ERROR` text message to just a single space.

> **See Also:**
>
> Translating Messages Used Internally by APEX

15.3 Understanding Page Processes

Create **page processes** to perform an action at a specified point during the rendering or submission of the page.

- About Page Processes
  A page process performs an action at a specified point during the rendering or submission of the page.
- Creating Page Processes
  Create a page process in Page Designer.
- Editing Page Processes
  Learn about editing page process attributes in Page Designer.
- Understanding Background Page Processing
  Background page processing enables developers to run a sequence of page processes that can execute in the foreground or in the background.

15.3.1 About Page Processes

A page process performs an action at a specified point during the rendering or submission of the page.

For example, you can create a page process to execute logic or to make a call to the APEX engine. A page process is a unit of logic that runs when a specific event occurs, such as loading or submitting a page.

From a functional perspective, there is no difference between page-level and application-level processes. The difference between these two process types is where the process is defined, that is at the page-level or at the application level.

You create a page process in Page Designer. In either the Rendering or Processing tab, locate the node where the process will occur, right-click, and select **Create Process**. The process Type determines what attributes you need to configure.

Supported Process Types

> **Note:**
>
> Depending on your application configuration, you may also have additional process type plug-ins. These display as <*My Plug-in*> [*Plug-in*].

Supported page process Types include:

- **Clear Session State** - Process to clear session state stored within Oracle APEX.
- **Close Dialog** - Process to close the current modal or non-modal dialog page.
- **Data Loading** - Process to load data into the target table or collection based on the given data load definition.
- **Download** - Downloads one or multiple files. This process type works in Before Header position at Page Rendering or in Page Processing when submitting a page. The processing stops after this process is executed.
- **Execute Code** - Process to execute PL/SQL code.

- **Execution Chain** - Execute a sequence of page processes, either in the foreground or in the background..
- **Form - Automatic Row Processing (DML)** - Process to insert, update, or delete a form region row(s).
- **Form - Initialization** - Process to initialize form region items. Initialization can either be fetching data from the region source, using the primary key value(s) or simple initialization of the form region items. Human Task - Create - Process to create a human task, using an existing task definition from the application.
- **Human Task - Create** - Process to create a human task, using an existing task definition from the application.
- **Human Task - Manage** - Process to manage a human task. The different types of task management are Approve, Reject, Claim, Release, Delegate, Forward, Cancel, Comment and Invite Participant.
- **Interactive Grid - Automatic Row Processing (DML)** - Process to insert, update, or delete Interactive Grid rows.
- **Invoke API** - Process to invoke a procedure or function stored in the local database, or an operation defined in a REST Data Source.
- **Print Report** - Process to print a report using a report query.
- **Reset Pagination** - Process to reset pagination of reports on the current page.
- **Send E-Mail** - Process to send an email, and optionally one or more attachments, from the application.
- **Send Push Notification** - Sends a push notification to a user. All devices that the user subscribed to will receive the push notification.
- **Server Side Geocoding** - Only supported on Oracle Autonomous Database.

  This process type provides Geocoding (turning a postal address to a coordinate) functionality. Geocoding is performed by calling `SDO_GCDR.ELOC_GEOCODE`, which does a REST request to the Oracle Elocation Geocoding Service (elocation.oracle.com). For further information on this function, see SDO_GCDR.ELOC_GEOCODE in *Oracle Spatial Developer's Guide*.

  Geocoding input data comes from page items, which are mapped to address parts like Street, House Number, Postal Code or City. The process populates a coordinate and match vector page item with the results of the first matched address. In addition, all matches can be loaded into a collection for further processing. If **Sanitize Address** is enabled, address part page items will be overwritten with the sanitized values of the first matched address.
- **User Preferences** - Process to set User Preferences for the end user.
- **Web Service** - Process to consume the specified Web service.
- **Work Flow** - Process to perform the following operations on a Workflow:
  - Start a new Workflow Instance based on the selected Workflow Definition.
  - Terminate, Suspend, Resume or Retry an existing Workflow Instance.

  See About Using Workflows in Page Processes.

See Also:

- Understanding Application Processes
- Accessing Page Specific Utilities

15.3.2 Creating Page Processes

Create a page process in Page Designer.

To create a page process:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

      Page Designer appears.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Page Designer indicates what to do next. If there is an error, the Show Messages icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.

      To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.
4. Identification, Name - Enter the name of the process for easy identification by developers.
5. Identification, Type - Specify which process type to execute. The Type you select determines what attributes display next.

   Consider the following examples.

   - If Identification, Type is **Execute Code**:
     – Identification, Type - Select **Execute Code**.
     – Source, Location - Select where the process executes. Select either **Local Database** or **REST Enabled SQL**.
     – Source, Language - Select the language used to execute the entered code.
     – Source, PL/SQL - Enter the PL/SQL code to be executed.
   - If Identification, Type is **Invoke API**:
     – Identification, Type - Select **Invoke API**.
     – Settings, Type - Select what process executes. Options include: **PL/SQL Package**, **PL/SQL Procedure or Function**, or **REST Source**.
       What displays next, depends upon the Settings, Type you select. Consider the following examples:

       If Settings, Type is **PL/SQL Package**:

* Settings, Type - Select **PL/SQL Package**.
* Settings, Package - Enter the case-sensitive PL/SQL package name. You can type in the name or pick from the list.
* Settings, Procedure or Function - Enter the case-sensitive procedure or function name defined in the selected PL/SQL package. You can type in the name or pick from the list.

If Settings, Type is **PL/SQL Procedure or Function**:

* Settings, Type - Select **PL/SQL Procedure or Function**.
* Settings, Procedure or Function - Enter the case-sensitive procedure or function name defined in the selected PL/SQL package. You can type in the name or pick from the list.

If Settings, Type is **REST Source**:

* Settings, Type - Select **REST Source**.
* Settings, REST Source - Select the REST Source which contains the Operation to be invoked.
* Settings, Operation - Select the Operation to invoke. The select list shows the name, if configured for the REST Operation. Otherwise, its HTTP method and URL pattern will be shown.

6. Under **Execution** :
   a. Execution, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

      If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.

   b. Execution, Point - Select the point at which this process is executed.
   c. Execution, Run Process - Select the frequency that this process executes.
7. Under **Success Message**:
   - Success Message, Success Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed. If you are branching to another page using a URL redirect, you may need to check the **preserve success message** attribute.

     To learn more, select the attribute in the Property Editor and click the Help tab in the central pane.
8. Under **Error**:
   a. Error, Error Message - Enter the error message for this process.

      This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.

   b. Error, Display Location - Select where this process error message displays, when the process fails to complete successfully.

      Option include:

      - **Inline in Notification** - The process error message displays in the Notification area (defined as part of the page template).
      - **On Error Page** - Error displays on a separate error page.

9. Click **Save**.

> **See Also:**
>
> - Creating Pre-Rendering Branches, Computations, and Processes
> - Creating Page Submission Branches, Validations, Computations, and Processes

15.3.3 Editing Page Processes

Learn about editing page process attributes in Page Designer.

- Editing Page Process Attributes
  Edit a page process in Page Designer.
- Controlling When a Page Process Executes
  Control when a process executes by editing the Sequence and Point attributes.
- Making a Page Process Conditional
  Create a conditional page process by editing Server-side Condition attributes.
- Creating Page Process Success and Error Messages
  Define both success and error messages that display when a process runs.

15.3.3.1 Editing Page Process Attributes

Edit a page process in Page Designer.

To edit an existing page process:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

      Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. In either the Rendering or Processing tab, expand the nodes to locate and select the process.

   Process attributes display in the Property Editor. Attributes are organized in groups.

   To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.
4. Edit the appropriate attributes:

   > **Tip:**
   >
   > To learn more about an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
5. Click **Save**.

15.3.3.2 Controlling When a Page Process Executes

Control when a process executes by editing the Sequence and Point attributes.

To control when a process executes, edit the Sequence and Point attributes:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. In either the Rendering or Processing tab, expand the nodes to locate and select the process.

   Process attributes display in the Property Editor. Attributes are organized in groups.
4. Under **Execution** :
   a. Execution, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

      If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.
   b. Execution, Point - Select the point at which this process is executed.
   c. Execution, Run Process - Select the frequency that this process executes.
5. Click **Save**.

> **See Also:**
>
> Editing Page Process Attributes

15.3.3.3 Making a Page Process Conditional

Create a conditional page process by editing Server-side Condition attributes.

To create a conditional page process:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. In either the Rendering or Processing tab, expand the nodes to locate and select the process.

   Process attributes display in the Property Editor. Attributes are organized in groups.

4. Under **Server-side Condition**:
   a. Server-side Condition, When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.
   b. Server-side Condition, Execution Type - Select a condition type that must be met in order for this component to be rendered or processed:

      To learn more, select the attribute in the Property Editor and click the Help tab in the central pane.
5. Click **Save**.

15.3.3.4 Creating Page Process Success and Error Messages

Define both success and error messages that display when a process runs.

You can define both success and error messages that display when a process runs. How an error message displays depends upon the processing point. For processes with a processing point of `On Submit - Before Computation and Validations` or `On Submit - After Computations and Validations`, you can specify whether the error message displays inline on the current page (the default) or on a separate page. For all other page processes, error messages display on a separate error page.

To create a page process error or success message:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Click the **Processing** tab in the left pane.
3. In either the Rendering or Processing tab, expand the nodes to locate and select the process.

   Process attributes display in the Property Editor. Attributes are organized in groups.
4. Under **Success Message**:
   - Success Message, Success Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed. If you are branching to another page using a URL redirect, you may need to check the **preserve success message** attribute.

     To learn more, select the attribute and click the Help tab in the central pane.
5. Under **Error**:
   a. Error, Error Message - Enter the error message for this process. This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.
   b. Error, Display Location - Select where this process error message displays, when the process fails to complete successfully.

      Option include:

- **Inline in Notification** - The process error message displays in the Notification area (defined as part of the page template).
- **On Error Page** - Error displays on a separate error page.

6. Click **Save**.

> **See Also:**
>
> Page Templates

15.3.4 Understanding Background Page Processing

Background page processing enables developers to run a sequence of page processes that can execute in the foreground or in the background.

- About Background Page Processes
  Learn about background page processes.
- Creating a Background Page Page Process
  Create a background page process using the Execution Chain process type, which contains child processes executing one after the other. Execution chains can run in the background or in the foreground.
- Monitoring Currently Running Background Processes
  Monitor background processes by clicking Session on the Runtime Developer toolbar.

15.3.4.1 About Background Page Processes

Learn about background page processes.

The Execution Chain process type enables page processing to execute in the background so that users don't have to wait until processing finishes. Typical use cases are long running operations such as data loading or expensive data processing. An Execution Chain process type may contain one or multiple child processes, which do the actual processing, and are executed one after the other. An execution chain can run either in the background or in the foreground.

Creating a background page process involves the following general steps:

- In Page Designer, create a Execution Chain page process.
- To the Execution Chain process, add the child processes which do the actual processing (for example, Execute PL/SQL code, Data Loading, and so on).
- Edit the Execution Chain attributes and enable the **Execute in Background** switch.

> **Tip:**
>
> You can monitor background procecces programatically, using the `APEX_BACKGROUND_PROCESS` API. See APEX_BACKGROUND_PROCESS in *Oracle APEX API Reference*

About Reimporting an Application with Background Processes

Reimporting an application with background processes has a slightly different workflow compared to a standard Application Export. During the import, the wizard prompts you on how to handle existing background processes. See Reimporting an Application Containing Background Processes.

Configuring Limits for Concurrently Running Scheduler Jobs

Background page processing is actually executed by the database scheduler (`DBMS_SCHEDULER`). This means that the application's parsing schema must have the `CREATE JOB` privilege granted in order to run background page processing jobs.

You can configure limits for concurrently running background page processing jobs as follows. If a new background execution would exceed the configured limits, it will wait in a queued state until one of the running executions finishes and a slot becomes available.

- **Instance level** (as a default). Configure the **Background Jobs** attributes in Instance Settings. The instance level setting acts as a default for all workspaces and can be overridden at the workspace level.

  See Configuring Background Jobs in *Oracle APEX Administration Guide*

- **Workspace level**. Edit the **Maximum Background Page Process Jobs** attribute under Workspace Isolation. The workspace limit acts as a *maximum* for all applications within that workspace.

  See Configuring Workspace Isolation Attributes in *Oracle APEX Administration Guide*

- **Application level**. Edit the Application Definition, **Maximum Background Page Process Jobs** attribute. This limit is specific to this application. Use this setting to distribute the workspace limit across applications in that workspace. If you configure a higher value than the workspace limit, this limit will be ignored and the workspace limit will be used instead.

  Application level. Edit the Application Definition, **Maximum Background Page Process Jobs** attribute. See Properties.

15.3.4.2 Creating a Background Page Page Process

Create a background page process using the Execution Chain process type, which contains child processes executing one after the other. Execution chains can run in the background or in the foreground.

To create background page process using the Execution Chain process type:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

      Page Designer appears.

Create Execution Chain page process:

2. In the Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.

In the Property Editor, configure the Execution Chain process.

3. Under **Identification**:

a. Identification, Name - Enter the name of the process for easy identification by developers.

b. Identification, Type - Select **Execution Chain**.

c. Identification, Execution Chain - This attribute enables support for nested execution chains. Use this attribute to define another execution chain as the parent for this chain. For this example, select **None**.

4. Under **Settings**:

   a. Settings, Execute in Background - Enable this option to execute this chain in the background.

   b. Settings, Serialize Executions - Specify whether to serialize executions of this execution chain. When enabled, all executions (from all APEX sessions) are serialized to avoid concurrent executions in the background. Use this setting when processes of this chain are likely to work on the same resources, for instance, performing DML on the same row in a table..

   c. Settings, Return Execution ID into Item - Choose a page item to return the execution ID into. Based on the execution ID, the current status of the background execution can be fetched.

   d. Settings, Temporary File Handling - Determine how to deal with temporary files, uploaded through a File Browse page item. Options include:

      - **Ignore** - Neither move nor copy temporary files to the working session for background execution. Choose this option if the background processes will not work with uploaded files.
      - **Move** - Move temporary files to the working session for background execution. Choose this option if access to temporary files is only needed by one background execution chain.
      - **Copy** - Copy temporary files to the working session for background execution. Choose this option if you have multiple background execution chains, and all of them require access to the temporary files.

   e. Settings, Executions Limit - If an execution limit is specified, Oracle APEX will raise an error message, if an application user requests more than the specified limit of background executions. For example, specifying the value `3`, users see an error message when attempting to submit the fourth background execution. This attribute is useful to prevent users from submitting large amounts of background executions to the server.

5. Under **Execution** :

   a. Execution, Sequence - Specify the sequence for this computation. The sequence determines the order of execution.

      If two components have the same sequence value, then they may be executed in a different order when the application is exported and imported into another environment.

   b. Execution, Point - Select the point at which this process is executed.

   c. Execution, Run Process - Select the frequency that this process executes.

6. Under **Success Message**:

   - Success Message, Success Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed. If you are

branching to another page using a URL redirect, you may need to check the **preserve success message** attribute.

To learn more, select the attribute in the Property Editor and click the Help tab in the central pane.

7. Under **Error**:
   a. Error, Error Message - Enter the error message for this process.

      This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays.

   b. Error, Display Location - Select where this process error message displays, when the process fails to complete successfully.
8. Click **Save**.

Create one or more child process:

9. Create a child process:
   a. In the Processing tab, select the Execution Chain process, right-click and select **Create Child Process**.

      The new child process displays under **Processes**.

   b. In the Property Editor, configure the process attributes.

      To learn more about an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.

   c. Click **Save**.
10. Repeat the previous step to create additional child processes.

15.3.4.3 Monitoring Currently Running Background Processes

Monitor background processes by clicking Session on the Runtime Developer toolbar.

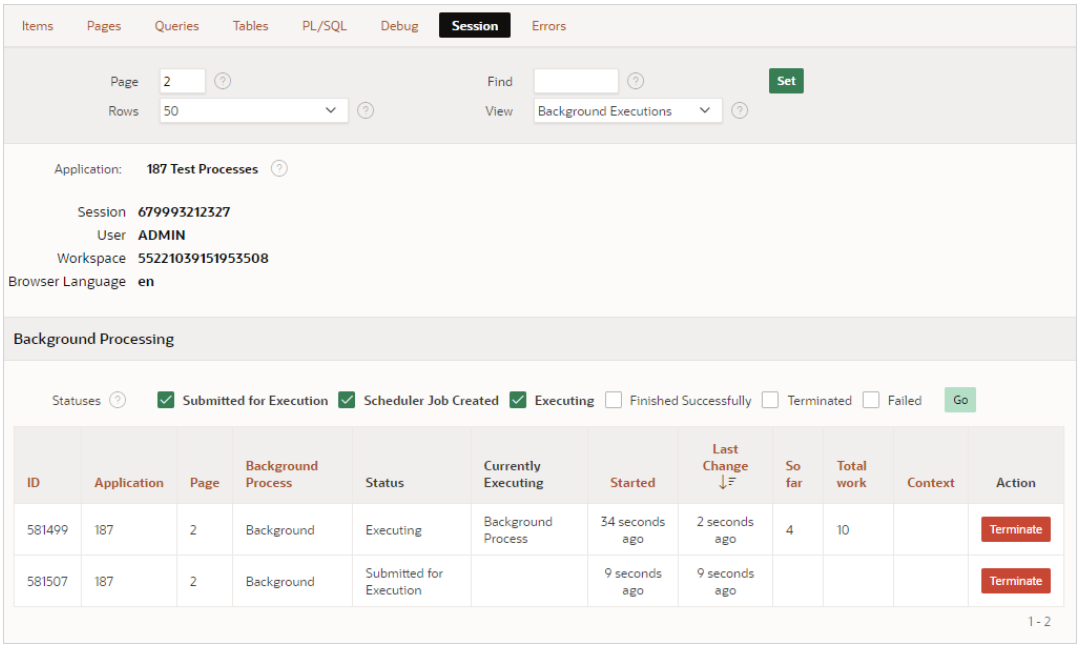
> **Tip:**
>
> You can also view a Background Processing report for Active Sessions in Workspace Administration. From the Administration menu, select **Monitor Activity** and then **Active Sessions**. Select an active session and then **Background Processing**.

To monitor background processes:

1. Run the application as a developer.
2. Locate the Runtime Developer toolbar. By default, the Runtime Developer toolbar displays at the bottom of the page.
3. On the Runtime Developer toolbar, click **Session** and then **View Session State**.

   The Session page appears and displays the following information displays at the center of the page.
4. From View, select **Background Executions** and click **Set**.

The Background Processing region appears displaying currently running background processes as shown in the following example.



5. Statuses - Configure which background executions to include and click **Go**. Available Statuses include:
   - Submitted for Execution
   - Scheduler Job Created
   - Executing
   - Finished Successfully
   - Terminate
   - Failed
6. To terminate a currently running background process, click **Terminate**.

15.4 Understanding Application Processes

Create an application process to run a block of PL/SQL logic at a specific point from multiple pages of an application.

By default, application processes execute at the same point for every page in the application. However, you can apply conditions for specific pages to control when the process executes.

> **Tip:**
>
> To learn more about execution behavior in the case of a validation error, see What Happens When a Validation Fails?

- On Demand Application Processes
  On Demand processes are useful when you have PL/SQL logic that you would like to run from different execution points across multiple pages.

- About Application Processes that Execute On New Instance
  Processes with a Process Point of **On New Instance** enable you to retrieve information once within a user's session.
- About Running an On Demand Process from a Page Request
  Run an On Demand process from a page request.
- Example: Application Process
  View an application process example.
- Creating an Application Process
  Create an application process from Shared Components.
- Editing Application Process Attributes
  Edit application process attributes from Shared Components.
- Copying or Subscribing to Application Processes
  Copy application processes from the current application or from another application. When copying an application process from another app, you can also subscribe to it.
- Subscribing to an Application Process
  Subscribe to an application process on the Application Processes, Edit page.
- Creating Application Process Error Messages
  Learn how to define application process error message.
- Viewing the Application Processes History Report
  To view the Application Process History report.

> **See Also:**
>
> Understanding Page Processes

15.4.1 On Demand Application Processes

On Demand processes are useful when you have PL/SQL logic that you would like to run from different execution points across multiple pages.

An **On Demand** process is special type of application process which has a Process Point of On Demand, is of type PL/SQL, and executes when called from a page-level On Demand process or from an Ajax call from the browser.

On Demand processes should typically be created on a page and not at the application-level. On Demand processes created at the application-level are created with an initial authorization scheme of **Must Not Be Public User** to prohibit the processes from being invoked from users in unauthenticated sessions.

> **See Also:**
>
> Creating Page Processes

15.4.2 About Application Processes that Execute On New Instance

Processes with a Process Point of **On New Instance** enable you to retrieve information once within a user's session.

Typically an application process runs at the same point across multiple pages in an application. Processes having a Process Point of **On New Instance** are the exception. These types of processes are useful when you only need to retrieve information once within a user's session. For example, if the application items do not depend on the logged in user, you can initialize them using the Process Point **On New Instance**. To look up information that depends on the user, you can also use **After Authentication**.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, **On New Instance** and **After Login**, the application processes fire whenever Oracle APEX first processes a request for the application (that is, On New Instance), or processes a request of an authenticated user for the application (that is, **After Authentication**).

15.4.3 About Running an On Demand Process from a Page Request

Run an On Demand process from a page request.

You can have a page request run an On Demand process by using the following syntax:

```
f?p=application_id:page_id:session:APPLICATION_PROCESS=process_id
```

Where:

- *application_id* is the application ID or alphanumeric alias
- *page_id* is the page number or alphanumeric alias
- *session* is the session ID
- *APPLICATION_PROCESS=process_id* is the keyword *APPLICATION_PROCESS=* followed by either the process ID or an alphanumeric name of an application-level process having a Process Point of On Demand

When you use this syntax, the APEX engine recognizes the request and processes it using the following rules:

- The page number in the URL can be the current page number or alias. A page number or alias is required in the request only as a syntactic placeholder because no specific page is accessed for this type of request.
- The process authorization scheme, the application's authorization scheme, and the process conditions are supported.
- Session state (that is, item names and values) may be set in the URL, but clear cache options are ignored.
- Any failures of authentication, authorization, or process conditions do not result in visible error messages or other indicators of such failures and most often result in a blank page being displayed. Note that if you are logged in to App Builder as a developer, an error messages displays.
- Specifying the process by name locates the first process with the specified (case-preserved) name.

See Also:

Clearing Session State

15.4.4 Example: Application Process

View an application process example.

A shopping cart application is a good example of when you might use an application process. For example, to display the contents of a user's shopping cart with each page view, you create a region on page zero of your application that displays the values of the application-level items `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`.

Instead of writing a process for each page to set the values of `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`, you could write an application process of type **On Load: Before Header** to compute these values. Then, the APEX engine would execute the process on each page as it renders the application. As a result, each page, would display the most current values for `TOTAL_CART_ITEMS` and `TOTAL_PURCHASE_PRICE`.

> **See Also:**
>
> Creating a Global Page to Display Components on Every Page

15.4.5 Creating an Application Process

Create an application process from Shared Components.

To create an application process:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Processes**.

   The Application Processes page appears.
3. Click **Create**.
4. For Identification:
   a. Name - Enter a name for the application process.
   b. Sequence - Specify the sequence number for this process. The sequence number determines the order in which the process is evaluated relative to other processes that execute at the same point.
   c. Point - Identify the point during the processing of each page in the application that this process would run. Note the option, **On Demand,** only runs when requested by a page process.
   d. Click **Next**.
5. For Source:
   a. Language - Select a language that is used to evaluate the entered programming code.

   b. Code - Enter the code of your application process.
   c. Do not validate code (parse code at runtime only) - Select if applicable.
   d. Click **Next**.
6. For Conditionality:
   a. Condition Type - Select a condition type that must be met in order for this process to execute.Note that the UI changes slightly depending upon the selected Condition Type.
   b. Expression 1 - Use this attribute to conditionally control whether or not a component is rendered or processed. Enter values in this attribute based on the selected condition type. The component will be rendered or processed if the specified condition is met.
   c. Click **Create Process**.

> **See Also:**
>
> Creating Application Process Error Messages

15.4.6 Editing Application Process Attributes

Edit application process attributes from Shared Components.

To edit an existing page process:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Processes**.

   The Application Processes page appears.
3. Click the process name.
4. Edit the appropriate attributes. For more information, see field-level Help.
5. Click **Apply Changes**.

15.4.7 Copying or Subscribing to Application Processes

Copy application processes from the current application or from another application. When copying an application process from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

> **Note:**
>
> When copying an application process from another app, shared component dependencies are automatically resolved. To learn more, see About Automatic Dependency Resolution.

To copy or subscribe to an application process:

1. Navigate to the Application Processes page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Application Processes**.

      The Application Processes page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. To copy an application process within the current application:
   a. Search for the application process and click **Copy** in the Copy column.

      The Copy Application Process Wizard appears.
   b. New Application Process Name - Enter the name of the application process.
   c. Click **Copy Application Process**.
3. To copy an application process from another application:
   a. From the Tasks region, click **Copy from another app**.

      The Copy Application Process appears.
   b. Application - Select the application from which you want to copy.
   c. Copy Application Process - Select the application process to copy.
   d. Subscribe - Set this flag to **On** to subscribe to the application process.
   e. Click **Copy Application Process**.

15.4.8 Subscribing to an Application Process

Subscribe to an application process on the Application Processes, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a search configuration by copying it and then subscribing. See Copying or Subscribing to Application Processes.

To subscribe to an application process:

1. Navigate to the Application Processes page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Application Processes**.

      The Application Processes page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an application process.

   The Application Processes, Edit page appears.
3. To subscribe to an application process:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master application process to subscribe to and click **Apply Changes**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscribed application process:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Application Process - If the application process is already subscribed, click **Refresh Application Process**.
5. To unsubscribe from a subscribed application process:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

15.4.9 Creating Application Process Error Messages

Learn how to define application process error message.

If a process raises an error, you can define an error messages that displays to the user. How the error message displays depends upon the processing point. For processes with a processing point of `On Submit - Before Computation and Validations` or `On Submit - After Computations and Validations`, you can specify whether the error message displays inline on the current page (the default) or on a separate page. For all other application processes, error messages display on a separate error page.

To create an application process error message:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Processes**.

   The Application Processes page appears.
3. Click the process name.

4. Edit the following attributes:
   a. **Source**, **Process Error Message** - Enter the message that displays when an unhandled exception is raised.
   b. **Error message display location** - Applies to processes with a processing point of `On Submit - After Page Submission - Before Computation and Validations` or `On Submit - After Page Submission - After Computations and Validations`, you can specify where the error message displays.

      Select one of the following:

      - **Inline in Notification** - Error displays inline on the page in the "notification" area (defined as part of the page template).
      - **On Error Page** - Error displays on a separate error page.

        If the application contains a manually created tabular form (using `APEX_ITEM` calls in the SQL statement), select **On Error Page**. Otherwise, changes made by the user are discarded when the page is re-rendered. When errors are displayed on a separate error page, the user can use the browser's Back button to preserve the entered values.
5. Click **Apply Changes**.

> **See Also:**
>
> Editing Application Process Attributes

15.4.10 Viewing the Application Processes History Report

To view the Application Process History report.

The Application Process History report displays recent modifications made to application processes.

To view

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Processes**.

   The Application Processes page appears.
3. Select the History tab at the top of the page.
4. You can customize the appearance the page using the Search bar at the top of the page. Available controls include:
   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) to search for a process by name and click **Go**.

- **Go button** - Executes a search or applies a filter.
- **Actions menu** - Use the Actions menu to customize the report view.

> **See Also:**
>
> About the Actions Menu

15.5 About Branches

A branch is an instruction to go to a specific page, procedure, or URL. For example, you can branch from page 1 to page 2 after page 1 is submitted. When you create a branch, you specify a Branch Point and Branch Type.

> **See Also:**
>
> Controlling Navigation Using Branches

15.6 Understanding Application Computations

Create an application computation to set the value of a single page or application-level item. Application computations run at the same point across multiple pages in an application.

> **Tip:**
>
> To learn more about execution behavior in the case of a validation error, see What Happens When a Validation Fails?

- About Application Computations
  Application Computations are units of logic that set the value of a single page or application-level item and are run at the same point across multiple pages in an application.
- About Application Computations that Execute On New Instance
  Use the On New Instance Computation Point to retrieve information once within a user's session.
- Creating an Application Computation
  Create an application computation from Shared Compoents.
- Copying or Subscribing to Application Computations
  Copy application computations from the current application or from another application. When copying an application computation from another app, you can also subscribe to it.
- Subscribing to an Application Computation
  Subscribe to an application computation on the Application Computations, Create / Edit page.
- Accessing the Application Computation History Report

15.6.1 About Application Computations

Application Computations are units of logic that set the value of a single page or application-level item and are run at the same point across multiple pages in an application.

Similar to page-level computations, application computations can be based on static values, item values, PL/SQL, or SQL. A common use of an application item is to store the value of the last page viewed in the application. By storing the value in an item, you can add a back button and then redirect the user to the page number captured by the computation. This type of computation works well, for example, when you need to enable users to back out of an error page.

Example: Storing the Last Visited Page

The following is an example of a computation that stores the last page visited. In this example, the computation:

- Stores the last application page visited to an item named `LAST_PAGE`
- Checks that the value of a `CURRENT_PAGE_ITEM` is of type PL/SQL Function Body with a Computation body of:

```
BEGIN
   :LAST_PAGE := nvl(:CURRENT_PAGE,:APP_PAGE_ID);
   :CURRENT_PAGE := :APP_PAGE_ID;
   RETURN :LAST_PAGE;
END;
```

v

15.6.2 About Application Computations that Execute On New Instance

Use the On New Instance Computation Point to retrieve information once within a user's session.

Typically an application computation runs at the same point across multiple pages in an application. The exception is computations having a **Computation Point** of **On New Instance**. These types of computations are useful when you only need to retrieve information once within a user's session. For example, if the application items do not depend on the logged in user, you can initialize them using the Computation Point **On New Instance**. To look up information that depends on the user, you can also use **After Authentication**.

If you configure applications to share the same session by setting the same cookie name in the authentication scheme, **On New Instance** and **After Login**, the application processes fire whenever Oracle APEX first processes a request for the application (that is, **On New Instance**), or processes a request of an authenticated user for the application (that is, **After Authentication**).

15.6.3 Creating an Application Computation

Create an application computation from Shared Compoents.

To create an application computation:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.

   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Application Logic, select **Application Computations**.
3. Click **Create**.
4. Edit the following attributes:
   a. **Item**, **Sequence** - Specify the sequence for this component. The sequence determines the order of evaluation.
   b. **Item**, **Computation Item** - Select the item this computation affects.
   c. **Frequency**, **Computation Point** - Select a process point at which this computation should be performed. Selecting **After Submit** causes the computation to be performed only after the page is displayed and then submitted.
   d. **Computation**, **Computation Type** - Select the manner in which this computation is performed.
   e. **Computation**, **Computation** - Enter the computation logic that corresponds to the computation type. See field-level Help for examples.
   f. **Computation**, **Computation Error Message** - Enter the error message that displays if the computation fails.
   g. **Authorization**, **Authorization Scheme** (optional) - Select an authorization scheme which must evaluate to True in order for this computation to execute.
   h. **Conditions**, **Condition Type** (optional) - Select a condition type that must be met in order for this computation to execute.
   i. **Configuration**, **Build Option** (optional) - Select a build option for this component.
   j. **Comments**, **Comments** (optional) - Enter any comments.
5. Click **Create Computation**.

> **See Also:**
>
> Using Build Options to Control Configuration

15.6.4 Copying or Subscribing to Application Computations

Copy application computations from the current application or from another application. When copying an application computation from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

> **Note:**
>
> When copying an application computation from another app, shared component dependencies are automatically resolved. To learn more, see About Automatic Dependency Resolution.

To copy or subscribe to an application computation:

1. Navigate to the application computations page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Application Computations**.

      The Application Computations page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. To copy an application computation within the current application:
   a. Search for the application computation and click **Copy** in the Copy column.

      The Copy Application Computation Wizard appears.
   b. New Computation Item - Select an application-level or a page-level item.
   c. Click **Copy Application Computation**.
3. To copy an application computation from another application:
   a. From the Tasks region, click **Copy from another app**.

      The Copy Application Computation Wizard appears.
   b. Application - Select the application from which you want to copy.
   c. Copy Application Computation - Select the application computation to copy.
   d. Subscribe - Set this flag to **On** to subscribe to the application computation.
   e. Click **Copy Application Computation**.

15.6.5 Subscribing to an Application Computation

Subscribe to an application computation on the Application Computations, Create / Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a search configuration by copying it and then subscribing. See Copying or Subscribing to Application Computations.

To subscribe to an application computation:

1. Navigate to the Application Computations page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Application Logic, click **Application Computations**.

      The Application Computations page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an application computation.

   The Application Items, Create / Edit page appears.
3. To subscribe to an application computation:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master application computation to subscribe to and click **Apply Changes**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscribed application computation:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Application Computation - If the application computation is already subscribed, click **Refresh Application Computation**.
5. To unsubscribe from a subscribed application computation:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

15.6.6 Accessing the Application Computation History Report

Once you create an application computation, you can view the Application Computation History report. The Application Computation History report displays a history of recently changed application computations by date.

To access the Application Computation History report:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Computations**.
3. Select the **History** at the top of the page.
4. You can customize the appearance the page using the Search bar at the top of the page. Available controls include:
   - **Select columns to search** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.

- **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
- **Go button** - Executes a search or applies a filter.
- **Actions menu** - Use the Actions menu to customize the report view.

> **See Also:**
>
> About the Actions Menu

15.7 Using the Attribute Dictionary

Use the Manage Attribute Dictionary page to update values on the current page with those in the Attribute Dictionary, or to update the Attribute Dictionary with the values on the current page.

- About the Attribute Dictionary
  The Attribute Dictionary contains a set of attributes about a column that are used in creating forms and reports.
- Accessing the Manage Attribute Dictionary
  Access the Manage Attribute Dictionary page:

15.7.1 About the Attribute Dictionary

The Attribute Dictionary contains a set of attributes about a column that are used in creating forms and reports.

These definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

15.7.2 Accessing the Manage Attribute Dictionary

Access the Manage Attribute Dictionary page:

To access the Manage Attribute Dictionary page:

1. On the Workspace home page, click the **App Builder** icon.

   The App Builder home page appears.
2. Select an application.
3. Click **Utilities**.
4. Click **Utilities** and click **Attribute Dictionary**.

   The page is divided into two sections: **Page Items** and **Report Columns**.

   Use Page Item and Report Column definitions to update the Attribute Dictionary. You can also use the Attribute Dictionary to update Page Items and Report Columns.

   Hidden objects (that is, those in hidden regions and button items) are not counted in the number of candidate Items and Report Columns as these are not used in the updates.
5. To update the current page or update the Attribute dictionary, select the appropriate page link.

The Attribute Dictionary appears.

Only Items and Report Columns that have different values for at least one attribute are shown. You choose which objects to insert or update and which attributes to include. The number of Items and Report Columns that can be used might be less than the total because hidden objects, objects in hidden regions, and button items are not included.

16
Managing Shared Components

Shared components can display or be applied on any page within an application. Developers can use the tools and wizards on the Shared Components page either at the application-level or on specific pages.

- Accessing the Shared Components Page
  Access the Shared Components page by selecting an application and then clicking **Shared Components**.
- Shared Components Page
  Use the tools and wizards on the Shared Components page either at the application-level or on specific pages.
- Using Shared Component Subscriptions
  Use subscriptions to reuse shared components across several applications in a workspace.
- Managing Application Settings
  Create Application Settings to define application-level configuration options.
- Managing Application Items
  Application items do not display, but are used to maintain session state.
- Managing Component Settings
  Use Component Settings to set application level values for built-in APEX components and installed plug-ins.
- Creating Lists of Values at the Application-Level
  Create a list of values at the application-level that can be referenced by any page item or report field.
- Creating Lists
  Add a shared collection of links (or a list) to an application. To add a list to a page, create a region and specify the region type as List. You control the appearance of a list through list templates.
- Adding Search to an Application
  Add declarative searches to an application by creating a search configuration and then a search page that displays the search results.
- Using Shortcuts
  Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions.
- Using Component Groups
  Use component groups to easily copy or subscribe to a collection of shared components.
- Managing Email Templates
  Enable users to send emails from your application by creating email templates.
- Managing Static Application Files
  Manage static application files in the Oracle database on the Shared Components, Static Application Files page or in remote storage solution such as OCI Object Storage.
- Managing Static Workspace Files
  Navigate to Shared Components to upload, view, download, and delete static workspace files (including images, CSS files, and other files which must be managed independently).

- Printing a Report Region by Defining a Report Query
  Print a report region by defining a report query.
- Formatting a Report Region or Report Query Using Report Layouts
  Create a report layout to format a report region or report layout.

> **See Also:**
>
> - App Builder Concepts
> - Creating Applications
> - Controlling Page Layout
> - Adding Navigation

16.1 Accessing the Shared Components Page

Access the Shared Components page by selecting an application and then clicking **Shared Components**.

To access the Shared Components page:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. On the Application home page, click **Shared Components** in the center of the page.

> **Tip:**
>
> Once you select an application, you can also access the Shared Components page by clicking **Shared Components** ( ) at the top of the page. The Shared Components button displays at the top of most App Builder pages including the Application home page, Page Designer, Supporting Objects, and Utilities.

   The Shared Components page appears and is divided into regions. Badges with counts display next to some component list items.

4. To create or edit a shared component, select the appropriate link.

   To learn more about the links on the Shared Components page, see Shared Components Page.

16.2 Shared Components Page

Use the tools and wizards on the Shared Components page either at the application-level or on specific pages.

The Shared Components page is divided into regions. Badges with counts display next to some components.

- Application Logic
- Security
- Other Components
- Navigation and Search
- User Interface
- Files and Reports
- Data Sources
- Workflows and Automations
- Globalization
- Generative AI
- Tasks Region
- Workspace Objects Region

16.2.1 Application Logic

The following table describes the links under **Application Logic** on the Shared Components page.

Link	Description
Application Definition	Links to the Edit Application Definition page. Use this page to edit attributes for an application. Attributes are grouped into four categories: Definition, Security, Globalization, and User Interface. **See Also:** Managing Application Attributes
Application Items	Application items do not display, but are used to maintain session state. Application Items can be set using computations, processes, or by passing values on a URL. **See Also:** Managing Application Items
Application Processes	Use application processes to run PL/SQL logic: • At specific points for each page in an application • As defined by the conditions under which the process is set to execute • Upon the creation of a new session Note that **On Demand** processes execute only when called from a page-level On Demand process or when called using Ajax from the browser. **See Also:** Understanding Application Processes
Application Computations	Use application-level computations to assign values to application and page items for each page displayed or upon the creation of a new application session. You can also create an application-level computation and execute it conditionally on multiple pages. **See Also:** Understanding Application Computations
Application Settings	Application Settings enable developers to define application level configuration options. **See Also:** Managing Application Settings

Link	Description
Build Options	Use build options to conditionally display or process specific functionality within an application. You can use build options to control which features of an application are turned on for each application deployment. **See Also:** Using Build Options to Control Configuration

16.2.2 Security

The following table describes the links under **Security** on the Shared Components page.

Link	Description
Security Attributes	Use the Edit Security Attributes page to configure general security attributes for all pages within an application. **See Also:** Configuring Security Attributes
Authentication Schemes	Authentication is the process of establishing each user's identity before they can access your application. Authentication may require a user to enter a user name and password or may involve verification of a user's identity or use of a secure key. **See Also:** Establishing User Identity Through Authentication
Authorization Schemes	Authorization restricts user access to specific controls or components based on predefined user privileges. **See Also:** Providing Security Through Authorization
Application Access Control	Manage application access control roles and user role assignments. **See Also:** Managing Roles and User Assignments
Session State Protection	Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy. **See Also:** Preventing URL Tampering

16.2.3 Other Components

The following table describes the links under **Other Components** on the Shared Components page.

Link	Description
List of Values	A list of values (LOV) is a static or dynamic set of values used to display a popup list of values, select list, checkbox, or radio group. **See Also:** Creating Lists of Values at the Application-Level
Plug-ins	App Builder includes built-in item types, region types, dynamic actions, and processes. Use plug-ins to add new declarative types in to your application. **See Also:** Implementing Plug-ins
Component Settings	Use Component Settings to set application-level values for built-in APEX components and installed plug-ins. **See Also:** Managing Component Settings

Link	Description
Shortcuts	Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions. You can create a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML. Once you define a shortcut, it is stored in a central repository so you can reference it from various locations within your application. **See Also:** Using Shortcuts
Component Groups	A component group is a collection of shared components that you can easily copy or subscribe to across different applications. Component groups enable bulk refresh of the shared components in the group. **See Also:** Using Component Groups
Data Load Definitions	Use Data Load Tables to define tables for use in the Data Loading create page wizard. A Data Load Table is an existing table in your schema that has been selected for use in the data loading process to upload data. **See Also:** Creating Applications with Data Loading Capability and Managing Applications with Legacy Data Loading Capability

16.2.4 Navigation and Search

The following table describes the links under **Navigation** on the Shared Components page.

Link	Description
Lists	A list is a shared collection of links. You control the appearance of a list through list templates. Each list element has a display condition that enables you to control when it displays. **See Also:** Creating Lists
Navigation Menu	A navigation menu is a list links that enables users to navigate the pages in an application. Navigation menus are only supported in applications using the *Universal Theme - 42*. **See Also:** Managing Navigation Menus
Tabs	**Note**: Tabs only appear if the associated application uses an older theme. In new themes, Tabs have been replaced with Navigation Menu. Tabs are an effective way to navigate users between pages in an application. You can create two types of tabs: standard tabs or parent tabs. A standard tab set is associated with a specific page and page number. A parent tab set functions as a container to hold a group of standard tabs. **See Also:** Managing Tabs
Breadcrumbs	Breadcrumbs provide users with hierarchical navigation. A breadcrumb is a hierarchical list of links that display using templates. You can display a breadcrumb as a list of links or as a breadcrumb path. **See Also:** Creating Breadcrumbs
Navigation Bar List	Navigation bar lists offer users a simple navigation path for moving between pages in an application. The location of a navigation bar depends upon the associated page template. A list entry can be an image, an image with text beneath it, or text. **See Also:** Managing a Classic Navigation Bar

Link	Description
Search Configurations	Added declarative searches to an application by creating search configuration that contain information about a searchable data source. **See Also:** Adding Search to an Application

16.2.5 User Interface

The following table describes the links under **User interface** on the Shared Components page.

Link	Description
User Interface Attributes	Use User Interface page to specify user interface options for an application. **See Also:** Editing User Interface Attributes
Progressive Web App	Progressive Web Apps (PWAs) leverage modern web APIs giving users an experience that looks and feels like a native app.. **See Also:** Creating a Progressive Web App (PWA)
Themes	A theme is a named collection of templates that defines the application user interface. **See Also:** Using Themes
Templates	Templates control the look and feel of specific constructs within your application, such as pages, regions, items, and menus. **See Also:** Creating Custom Themes
Email Templates	Create templates to define the HTML format and Plain Text formats for the emails you wish to send from an application. **See Also:** Managing Email Templates
Map Backgrounds	Create custom maps backgrounds for map regions, Display Map items, and Geocoded Address items. **See Also:** Creating Custom Map Backgrounds

16.2.6 Files and Reports

The following table describes the links under **Files and Reports** on the Shared Components page.

Link	Description
Static Application Files	Use this link to upload, edit, and delete static files associated with the current application, including style sheets (CSS), images and JavaScript files. **See Also:** Managing Static Application Files
Static Workspace Files	Use this link to upload, edit, and delete static files associated with the current application, including style sheets (CSS), images and JavaScript files. **See Also:** Managing Static Workspace Files
Report Queries	Use the Report Queries link to view a report of stored queries within the current application. **See Also:** Printing a Report Region by Defining a Report Query

Link	Description
Report Layouts	Use Report Layouts with a report or shared query to render data in a printer-friendly format, such as Adobe Portable Document Format (PDF), Microsoft Word Rich Text Format (RTF), or Microsoft Excel (XLS) format. **See Also:** Formatting a Report Region or Report Query Using Report Layouts

16.2.7 Data Sources

The following table describes the links under **Data Sources** on the Shared Components page.

Link	Description
REST Data Sources	REST Data Sources enable developers to access data sourced from an external REST API. **See Also:** Managing REST Data Sources
Duality Views	Duality Views enable developers to access data sourced from a Duality View in the local or a remote database. **See Also:** Managing Duality Views
JSON Sources	JSON Sources enable developers to access data sourced from a JSON Collection Table or table with JSON columns, in the local or a remote database. **See Also:** Managing JSON Sources
Legacy Web Service References (SOAP)	The APEX engine can use Web Service References to access a Web service across the network. The Web service performs an action and then sends back a response. **See Also:** Managing Legacy Web Services

16.2.8 Workflows and Automations

The following describes the links under Workflows and Automations on the Shared Components page.

Link	Description
Task Definitions	Task Definitions define the participants, parameters, and actions for a task that requires human action. **See Also:** Managing Task Definitions
Automations	Create automations to initiate a sequential set of actions based on a schedule. Use automations to monitor data and then perform the appropriate action. **See Also:** Managing Automations
Workflows	Workflows represent the automation of a business process according to a set of procedural rules. **See Also:** About Workflows

16.2.9 Globalization

The following table describes the links under **Globalization** on the Shared Components page.

Link	Description
Globalization Attributes	Develop applications that can run concurrently in different languages. Click this link to specify globalization options such as the Application Primary Language and Application Language Derived From attributes. **See Also:** Configuring Globalization Attributes and Understanding Application Translation and Globalization Support
Text Messages	Text messages are named text strings that can be called from the PL/SQL code you write. This PL/SQL can be anonymous blocks within page processes and page regions, or in packages and procedures. **See Also:** Translating Messages
Application Translations	Develop applications in Oracle APEX that can run concurrently in different languages. A single Oracle database and Oracle APEX instance can support an application in multiple languages. Translating an application involves multiple steps. **See Also:** Understanding Application Translation and Globalization Support and Understanding the Translation Process

16.2.10 Generative AI

The following table describes the links under **Globalization** on the Shared Components page.

Link	Description
AI Attributes	Develop applications that can run concurrently in different languages. Click this link to specify globalization options such as the Application Primary Language and Application Language Derived From attributes. **See Also:** Configuring Generative AI Attributes
AI Configurations	**Generative AI Configurations** contain information such as System Prompt, Welcome Message, and Retrieval-Augmented Generation (RAG) Sources to enable richer interactions with Generative AI services. These configurations can define one or more RAG Sources that can be used by the AI service to retrieve relevant information and improve response quality. **See Also:** Managing AI Configurations and RAG Sources
AI Services	Create Generative AI Services on the Generative AI Services page in Workspace Utilities. Generative AI Services are stored at the workspace-level, they are visible and available to all applications in the workspace. **See Also:** Managing Generative AI Services

16.2.11 Tasks Region

The following table describes the Tasks region on the right side of the Shared Components page.

Link	Description
Export Application Components	Links to the Component Export page. Use this page to identify the components of an application to be exported either for backup purposes or to load the components on another instance. **See Also:** Exporting Application Components
Manage Supporting Objects	Links to the Supporting Objects page. Use this page to create a custom application. **See Also:** How to Create a Custom Application
REST Synchronizations	Enable Data Synchronization to keep the contents of a local table automatically in sync with the data from a REST service. **See Also:** Managing Data Synchronization
Edit Application Comments	Links to the Application Comments page. Use this page to enter comments specific to the currently selected application. **See Also:** Adding Application Comments
Developer Comments	Links to Developer Comments page. **See Also:** Adding Developer Comments

16.2.12 Workspace Objects Region

The Workspace Objects region displays on the right side of the Shared Components page. Workspace Objects are stored at the workspace-level and are therefore visible in all applications.

Link	Description
Credentials	Use Credentials to connect to REST Enabled SQL or other external REST services. Oracle APEX stores these securely stores and encrypts these credentials for use by Oracle APEX components. Credentials cannot be retrieved back in clear text. **See Also:** Managing Web Credentials
REST Enabled SQL Databases	Use REST Enabled SQL to execute SQL or PL/SQL defined in APEX components on a remote Oracle database. **See Also:** Managing REST Enabled SQL References
Remote Servers	Links to the Remote Servers page. Remote Servers can be shared among multiple REST Data Sources. Remote Servers are stored at the Workspace-level and therefore visible in all applications. When an application is being exported, the used Remote Servers are being added to the export file. **See Also:** Managing Remote Servers

16.3 Using Shared Component Subscriptions

Use subscriptions to reuse shared components across several applications in a workspace.

- About Shared Component Subscriptions
  Learn about shared component subscriptions.
- About Automatic Dependency Resolution
  Learn about the automatic copy of dependent shared components.
- Subscribing to a Shared Component
  Learn about subscribing to a shared component.

- About Subscribed Component Subscription Status
  Learn about viewing the Subscription Status of subscribed shared components.
- Refreshing a Subscribed Shared Component
  Access changes in a master shared component by refreshing the subscribed shared component.
- Publishing a Master Shared Component
  Publish a master shared component to push changes to all subscribing components.
- Unsubscribing to a Shared Component
  Unsubscribe to a shared component on the shared component edit page.

16.3.1 About Shared Component Subscriptions

Learn about shared component subscriptions.

Shared components can display or be applied on any page within an application and are accessible from the Shared Components page in App Builder. Shared component subscriptions enable developers to reuse shared components across several applications in a workspace.

This following shared components support subscriptions:

- Application Access Control (Roles)
- Application Computations
- Application Items
- Application Processes
- Application Settings
- Authentication Schemes
- Authorization Schemes
- Build Options
- Component Groups
- Data Load Definitions
- Email Templates
- Lists
- Lists of Values
- Map Backgrounds
- Plug-ins
- REST Data Source
- Report Layouts
- Search Configuration
- Shortcuts
- Text Messages

Subscribing to a shared component causes the following to happen:

- All the data from the master component and any dependent shared components are copied over to the subscribing shared component.

- The subscribing shared component becomes read-only.
- Only the master shared component can be edited.
- To make modifications, make changes to the master shared components. You can then push the changes to subscribed components by publishing them. Alternatively, you can also refresh individual subscribed shared components using the provided Refresh option.

16.3.2 About Automatic Dependency Resolution

Learn about the automatic copy of dependent shared components.

Shared components that refer other shared components are called **dependent shared components**.

The following table describes shared components that have dependents.

Shared Component Type	Dependent Shared Components
Application Computation	Build Option
Application Item	Build Option
Application Process	Build Option Authorization Scheme Application Item (if the computation is based on an Application Item)
Application Settings	Build Option
Lists	Build Option Authorization Scheme Lists (if any list entry points to a sub-list)
Search Configuration	Build Option Authorization Scheme Lists (if search configuration is based on a list)

About the Logic Resolving Dependencies

Shared component dependencies are automatically resolved when:

- Copying a shared component from one application to another application.
- Subscribing to a shared component.
- Refreshing a subscribed shared component.
- Publishing a master shared component.

The logic used to resolve dependencies is as follows:

1. In the destination or subscribing application, APEX engine checks if the dependent shared component exists with the same name. If it exists, then APEX re-uses the dependent shared component.
2. If the dependent shared component does not exist with the same name:
   - APEX copies the dependent shared component to the destination or subscribing application.
   - While copying, if the Subscribe flag is set to **On** then the subscription data is also copied.

- While subscribing, refreshing, publishing, the dependent shared component subscription data is also copied.

16.3.3 Subscribing to a Shared Component

Learn about subscribing to a shared component.

You can subscribe to a shared component during the copy process or for selective shared components on the shared component edit page.

> **Tip:**
>
> To learn more about subscribing to a shared component during the copy process, see the copy discussion for the respective shared component.

To subscribe to a shared component:

1. Navigate to the shared component edit page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

      > **Tip:**
      >
      > Shared component pages can display either in icon or report view. If the page displays in icon view, Oracle recommends changing to report view so you can view the subscription status. Click **View Table** on the toolbar.

2. To subscribe to a shared component by copying:
   a. From the Tasks list, select **Copy from another app**.
   b. Copy From Application - Select an application from which to copy the shared component and click **Next**.
   c. Copy:
      - To Name - Optionally edit the name of the Data Load Definition.
      - Action - Select an Action:
        - **Yes** - Copy the shared component.
        - **No** - Do not copy the shared component.
        - **Copy and Subscribe** - Subscribing to a shared component copies it to your application and sets the master reference.
   d. Click **Copy *shared component***.
3. For some shared components, you can subscribe on the shared component edit page:

   > **Note:**
   >
   > This procedure does not apply to REST Data Sources or Data Load Definitions.

a. Find and access the edit page of the shared component.

b. Find the **Subscription** region.

c. Subscription, Subscribe From - Select a master shared component to subscribe to and click **Apply Changes**.

When you select a master shared component and click **Apply Changes**, the subscribed shared component are be automatically refreshed from the selected master

> **See Also:**
>
> Refreshing a Subscribed Shared Component

16.3.4 About Subscribed Component Subscription Status

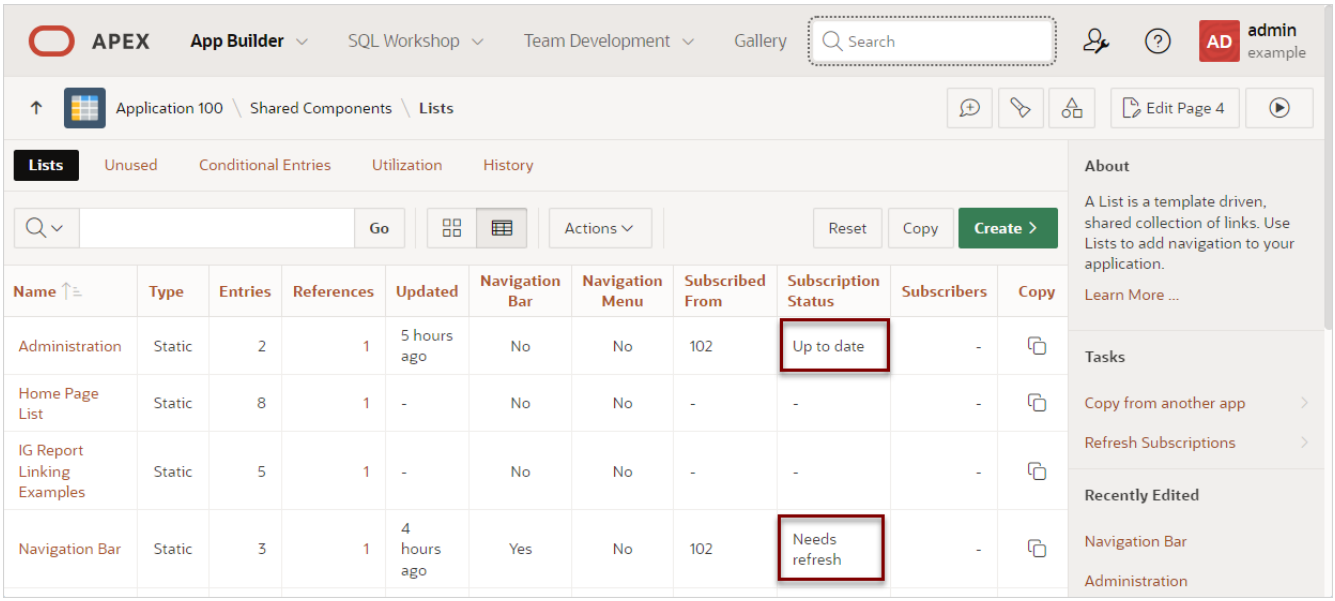
Learn about viewing the Subscription Status of subscribed shared components.

Each subscribed shared component has a Subscription Status of either:

- `Up to date` - Indicates the subscribed shared component is the same as master shared component.
- `Needs refresh` - Indicates the subscribed component is not same as master and needs to be refreshed.

> **Tip:**
>
> Shared component pages can display either in icon or report view. To view a Subscription Status for all shared components, access table view. On the toolbar, click **View Table**.

In report view, each *shared component* page includes the following columns: **Subscribed From**, **Subscription Status**, and **Subscribers**.



In the previous example, the **Administration** list is subscribed from application 102 and has a Subscription Status of `Up to date`. The **Navigation Bar** is also subscribed from application 102 but has a Subscription Status of `Needs Refresh`.

> **Tip:**
>
> The subscription status also displays on the *shared component* edit page in the Subscription region.

> **See Also:**
>
> Refreshing a Subscribed Shared Component

16.3.5 Refreshing a Subscribed Shared Component

Access changes in a master shared component by refreshing the subscribed shared component.

If a master shared component has changes, the subscribed shared component has a Subscription Status of `Needs refresh` and a **Refresh** button appears on the *shared component* edit page. The **Refresh** button does not appear if the subscription status is `Up to date`.

You can refresh a subscribed shared component on the shared component edit page, or perform a bulk refresh of all subscriptions by clicking **Refresh Subscriptions** on the Shared Components page.

> **Note:**
>
> The `static_id` is not updated during refresh for email templates, REST data sources, data load definitions, or search configurations.

To refresh a subscribed shared component:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application containing the subscribed shared component.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. To perform a bulk refresh for all shared components subscriptions:
   a. On the Shared Components page, find the **Tasks** region on the side bar.
   b. Under **Tasks**, click **Refresh Subscriptions**.
   c. On the Refresh Subscriptions page, click **Refresh All** to perform a bulk refresh of all shared components
3. To refresh a specific shared component subscription:

a. Find and access the edit page of the shared component.

   By default, each shared component page displays as a report. If there are new changes in the master shared component, the Subscription Status changes to `Needs refresh`.

   To access the edit page, click the shared component name.

b. On the shared component edit page, find the **Subscription** region and click the **Refresh *shared component*** button.

   The following confirmation message appears:

   `shared component` `refreshed from the master.`

16.3.6 Publishing a Master Shared Component

Publish a master shared component to push changes to all subscribing components.

> **Note:**
>
> The `static_id` is not updated during publish for email templates, REST data sources, data load definitions, or search configurations.

To publish a master shared component:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application containing the master shared component.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Find and access the edit page of the master shared component.

   By default, each shared component page displays as a report. To access the edit page, click the shared component name.
3. Edit the master shared component.
4. To push the changes to subscribed shared components, under Subscription, click **Publish**.
5. In the Confirm Publish dialog, click **Publish** again.

   The following confirmation message appears:
   `All subscribing` *`shared components`* `are refreshed.`

16.3.7 Unsubscribing to a Shared Component

Unsubscribe to a shared component on the shared component edit page.

To unsubscribe to a shared component:

1. Navigate to the shared component edit page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

d. Find and access the edit page of the shared component.

By default, each shared component page displays as a report. To access the edit page, click the shared component name.

2. On the shared component edit page, find the **Subscription** region and click **Unsubscribe**.

16.4 Managing Application Settings

Create Application Settings to define application-level configuration options.

> **Tip:**
>
> You also use the `APEX_APP_SETTING` API to access application settings. See APEX_APP_SETTING in *Oracle APEX API Reference*.

- Creating an Application Setting
  Create an Application Settings to enable developers to define application-level configuration options.
- Editing or Deleting Application Setting
  Edit or delete an existing setting on the Application Settings page.
- Copying or Subscribing to Application Settings
  Copy an application setting from the current application or from another application. When copying an application setting from another application, you can also subscribe to it.
- Subscribing to a Application Settings
  Subscribe to application settings on the Application Settings page.

16.4.1 Creating an Application Setting

Create an Application Settings to enable developers to define application-level configuration options.

To create an Application Setting:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Settings**.
3. Click **Create Setting**.
4. On Application Setting:
   a. Name - Enter a descriptive name for this application setting.
   b. Value - Enter the default value for this application setting.
   c. Value Required - Select whether this Application Setting requires a value. Select **On** or **Off**.

d. Valid Values - Enter comma separated valid values. If defined, APEX only accepts valid values when the settings change.

e. On Upgrade Keep Value - Select whether the application setting value in the deployed application should be kept, or overwritten with the specified value when the application is upgraded. Selecting **On** to keep the deployed value is useful if the application supports application setting configuration (using the `APEX_APP_SETTING.SET_VALUE` API) to change the application setting value in the deployed application, and where you want to respect those values upon upgrade.

f. Comments - Enter comments that describe this setting.

5. Click **Create Application Setting**.

   The new Application Setting displays on the Application Settings page.

16.4.2 Editing or Deleting Application Setting

Edit or delete an existing setting on the Application Settings page.

To edit or delete an Application Setting:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Settings**.

   The Application Settings page appears.
3. To edit an existing setting:
   a. Select the setting.

      The Application Setting dialog appears.
   b. Edit the appropriate attributes.

      To learn more about an attribute, see field-level Help.
   c. Click **Apply Changes**.
4. To delete an existing setting:
   a. Select the setting.

      The Application Setting dialog appears.
   b. Click **Delete**.

16.4.3 Copying or Subscribing to Application Settings

Copy an application setting from the current application or from another application. When copying an application setting from another application, you can also subscribe to it.

To copy an application setting:

1. Navigate to the Shared Components, Application Settings page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.

c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

d. Under Application Logic, select **Application Settings**.

   The Application Settings page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. To copy an application setting within the current application:

   a. Search for the application setting and click **Copy** in the Copy column.

      The Copy Application Setting Wizard appears.

   b. New Application Setting Name - Enter the name of the application setting. Application setting name must be unique within the application.

   c. Click **Copy Application Setting**.

3. To copy an application setting from another application:

   a. From the Tasks region on the Application Settings page, click **Copy from another app**.

      Copy Application Setting Wizard appears.

   b. Application - Select the application from which you want to copy.

   c. Copy Application Setting - Select the application setting to copy.

   d. Subscribe - Enable this flag to subscribe to the application setting.

   e. Click **Copy Application Setting**.

> **Note:**
>
> While copying an application setting from another application, build option dependency is automatically resolved during the copy process. To learn more see About Automatic Dependency Resolution.

> **See Also:**
>
> Using Shared Component Subscriptions

16.4.4 Subscribing to a Application Settings

Subscribe to application settings on the Application Settings page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an application setting by copying it and enabling the Subscribe option. See Copying or Subscribing to Application Settings.

To subscribe to application settings in the current application:

1. Navigate to the Application Settings page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Application Settings**.

      The Application Settings page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an application setting.

   The Application Settings page appears.
3. To subscribe to an application setting:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master application setting to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscription:
   a. Find the **Subscription** region.
   b. Click **Apply Changes**.
   c. Subscription, Refresh Application Settings - If the application setting is already subscribed, click **Refresh Application Settings**.
5. To unsubscribe from an application setting:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.5 Managing Application Items

Application items do not display, but are used to maintain session state.

Application items can be set using computations, processes, or by passing values on a URL. Use **On New Instance** computations to set the value of items once for a session. Use application items to maintain session state that is not displayed and is not specific to any one page.

- Creating an Application Item
  Create a application item on the shared Shared Components page.
- Editing Application Item Attributes
  Once you create an application item, you can edit it on the Create/Edit Application Item page.
- Copying or Subscribing to Application Items
  Copy application items from the current application or from another applications. When copying an application item from another app, you can also subscribe to it.

- Subscribing to an Application Item
  Subscribe to an application item on the Application Item, Create / Edit page.
- Accessing Application Item Reports
  View the application item Utilization and History reports by clicking the appropriate tab on the Application Items page.
- About Creating an Application Item that Functions as a Global Variable
  You can create a global variable by creating an application item with the same name across all applications and setting the scope to Global.

> **See Also:**
>
> - About the Differences Between Page Items and Application Items
> - Referencing Item Values

16.5.1 Creating an Application Item

Create a application item on the shared Shared Components page.

To create an application item:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Items**.

   The Application Items page appears.
3. To create an application item, click **Create**.

   The Create/Edit page appears.
4. For Name:
   a. Name - Enter a name for the application item.
   b. Scope - Options include:
      - **Global** - Select this option if the Oracle APEX session is shared by more than one application and the item's value should be the same for all applications.

        Applications can share the same session if their authentications have the same Session Cookie Attributes. The Scope attribute of application items has to be the same across these applications.
      - **Application** - Select this option if the application does share session state with another application.
5. Security, Session State Protection - Set session state protection for this item. Options include:
   - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.

> **Note:**
>
> If you must set this item's value in session state using Ajax, then an Unrestricted protection level must be used for the item (for example in Dynamic Actions, Set Value, Page Items to Submit or Cascading LOVs, Page Items to Submit).

- **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is only applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled. This attribute may be used for application items or for page items with any of these Display As types:
  - Display Only (Save State=No)
  - Display Only (Save State=No)
  - Stop and Start Grid Layout (Displays label only)
- **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the schema is provided. A user-level checksum or a session-level checksum will also suffice (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
- **Checksum Required: User Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the workspace, application, and user is provided. A session-level checksum will also suffice (see next bullet). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.
- **Checksum Required: Session Level** - The item's session state may be set by passing the item name/value in a URL if a checksum specific to the current session is provided. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

6. Security, Escape Special Characters - Controls whether the application item's value is automatically escaped in substitutions (for exmple, when referencing the item using `&ITEM.` syntax.

   To prevent Cross-Site Scripting (XSS) attacks, set Escape Special Characters to **On**. If you need to render HTML tags stored in the application item, you can set this flag to **Off**.
7. Configuration, Build Option - Select a build option for this component. Build options are predefined settings that determine whether components within an application are enabled.
8. Comments - Enter any notes regarding this item.
9. Click **Create Application Item**.

> **See Also:**
>
> About Creating an Application Item that Functions as a Global Variable

16.5.2 Editing Application Item Attributes

Once you create an application item, you can edit it on the Create/Edit Application Item page.

To edit application item attributes:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Items**.

   The Application Items page appears.
3. Select an application item.

   The Create/Edit page appears.
4. Edit the attributes.

   To learn more about an attribute, see field-level Help
5. Click **Apply Changes**.

> **See Also:**
>
> - Viewing Field-Level and Page Designer Help
> - Understanding Cross-Site Scripting Protection

16.5.3 Copying or Subscribing to Application Items

Copy application items from the current application or from another applications. When copying an application item from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

> **Note:**
>
> When copying an application item from another app, shared component dependencies are automatically resolved. To learn more, see About Automatic Dependency Resolution.

To copy or subscribe to an application item:

1. Navigate to the Application Items page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Application Items**.

      The Application Items page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. To copy an Application Item within the current application:
   a. Search for the application item and click **Copy** icon in the Copy column.

      The Copy Application Item Wizard appears.

      > **Tip:**
      >
      > To copy an Application Item from another app, click **Copy from another app** and see Step 3.

   b. New Application Item - Enter the name of the application item. This name must be unique within the application.
   c. Click **Copy Application Item**.
3. To copy an application item from another application:
   a. From the Tasks region, click **Copy from another app**.

      The Copy Application Item Wizard appears.
   b. Application - Select the application from which you want to copy.
   c. Copy Application Item - Select the application item to copy.
   d. Subscribe - Set this flag to **On** to subscribe to the application item.
   e. Click **Copy Application Item**.

> **See Also:**
>
> Using Shared Component Subscriptions

16.5.4 Subscribing to an Application Item

Subscribe to an application item on the Application Item, Create / Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an application item by copying it and then subscribing. See Copying or Subscribing to Application Items.

To subscribe to an application item:

1. Navigate to the Application Items page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Application Logic, click **Application Items**.

      The Application Items page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an application item.

   The Application Items, Create / Edit page appears.
3. To subscribe to an application item:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master application item to subscribe to and click **Apply Changes**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscribed application item:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Application Item - If the application item is already subscribed, click **Refresh Application Item**.
5. To unsubscribe from a subscribed application item:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.5.5 Accessing Application Item Reports

View the application item Utilization and History reports by clicking the appropriate tab on the Application Items page.

To view the application item reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Application Logic, select **Application Items**.

   The Application Items page appears.

3. Click the appropriate tab:
   - **Utilization** - Displays where each application item is used.
   - **History** - Displays recent modifications made to application items in this application.

16.5.6 About Creating an Application Item that Functions as a Global Variable

You can create a global variable by creating an application item with the same name across all applications and setting the scope to Global.

Large applications are often separated into several physical applications that reside in the same workspace. These applications often share session state by configuring the same cookie name in each application.

Another requirement of this type of architecture is the need to share a few global variables. While the number global variables should be kept small, global variables function as an effective means to facilitate communicate between the applications. For example:

- Properties of the current user (such as email address, employee number, and so on).
- Current company in a multitenant application.

To create an application item that functions as a global variable:

1. Create an application item with the same name in each application to share session state.
2. For Scope, select **Global**.
3. For Security, Session State Protection, select **Restricted**.

> **See Also:**
>
> Creating an Application Item

16.6 Managing Component Settings

Use Component Settings to set application level values for built-in APEX components and installed plug-ins.

> **Tip:**
>
> The Component Settings in your environment will vary based on the currently selected application.

- Configuring Component Settings
  Configure Component Settings to set application-level values for built-in APEX components and installed plug-ins.
- Configuring Checkbox
  Configure the default value stored in a page item for an enabled or disabled checkbox in Component Settings, Checkbox.

- Configuring Color Picker Display
  Configure the default display behavior for a Color Picker in Component Settings, Color Picker.
- Configuring Date Picker
  Configure default Date Picker behavior in Shared Components, Component Settings.
- Configuring Geocoded Address
  Configure Geocoded Address behavior in Component Settings, Geocoded Address.
- Configuring Interactive Report Action Menu Structure
  Configure if interactive reports use the new Action Menu or keep the legacy structure from earlier releases.
- Configuring Vector Tile Layers for Maps
  Configure Map component settings to include Vector Tile Layers as the background map.
- Configuring Oracle Cloud Applications (BOSS) REST Service
  Configure Component Settings for an Oracle Cloud Applications (BOSS) REST Service.
- Configuring an Oracle Cloud Applications (SaaS) REST Service
  Specify a sandbox name for use in App Builder and at runtime when using a Oracle Cloud Applications (SaaS) REST Service.
- Dynamically Adding Sliders to Regions
  Dynamically add a slider to regions if the displayed region name exceeds the width of the region by configuring the Component Settings, Region Display Selector.
- Configuring Select Many
  Specify how to display values when the item is not focused in Component Settings, Select Many.
- Configuring Server Side Geocoding
  Configure Component Settings for Server Side Geocoding.
- Configuring Show AI Assistant
  Configure Show AI Assistant component settings to set application-level values for the Show AI Assistant plug-in.
- Configuring Star Rating
  Configure default Star Rating behavior in Component Setting, Star Rating.
- Configuring Switch Defaults
  Configure default Switch item type behavior Component Settings, Switch.
- Viewing the Component Settings History Report
  View a report of recent modifications made to Component Settings in the current application.

16.6.1 Configuring Component Settings

Configure Component Settings to set application-level values for built-in APEX components and installed plug-ins.

> **Tip:**
>
> To learn more about a specific component type, see the topics that follow.

To configure Component Settings:

1. Navigate to the Shared Components page:

a. On the Workspace home page, click **App Builder**.

b. Select an application.

c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
>
>
>) on the Page Designer toolbar.

2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click the component name.
4. On Component Settings, configure the appropriate settings.

   To learn more about a specific setting, see item Help.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.2 Configuring Checkbox

Configure the default value stored in a page item for an enabled or disabled checkbox in Component Settings, Checkbox.

To configure the default value stored in a checkbox page item:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Checkbox**.
4. Settings:

   a. Checked Value - Enter the value stored in the page item if the end user checks the checkbox. The value can be up to 255 characters long.

   b. Unchecked Value - Enter the value stored in the page item if the user deselects the checkbox. This attribute can also be left blank if you want to store NULL when checked. The value can be up to 255 characters long.

5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.3 Configuring Color Picker Display

Configure the default display behavior for a Color Picker in Component Settings, Color Picker.

To configure the default display behavior for a Color Picker:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Color Picker**.
4. Settings:
   a. Display As - Select how the color picker displays. Options include:
      - **Popup** - A color can be entered in the text field or chosen from the popup color picker.
      - **Inline** - A color picker is shown inline.
      - **Color-only Popup** - A button opens a popup color picker.
      - **Native Color Picker** - A native browser color input. The UI is dependent on the browser.
   b. Mode - Select the amount of information shown in the color picker. Options include:
      - **Simple** - Shows just the color spectrum controls and color presets if any are defined.
      - **Full** - Shows additional fields to view or enter colors as Hex, HSL, or RGB values with optional Alpha value depending on Return Value As. Also shows the Initial and Current colors.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.4 Configuring Date Picker

Configure default Date Picker behavior in Shared Components, Component Settings.

To configure default Date Picker behavior:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Date Picker**.
4. Settings:
   a. Appearance & Behavior - Control the appearance and behavior of the calendar used to select dates. Only applies to Date Pickers when **Display As** is either **Popup** or **Inline**.

      Options include:

      - **Show Week** - Determines if the week of the year is shown.
      - **Month Picker**- Controls if the month is rendered as a button which allows selection instead of text.
      - **Year Picker**- Determines whether the year is rendered as a button which allow selection instead of text.
      - **Paginate by Multiple Months** - Determines whether the pagination size when displaying several months is only one month or the number of months.
      - **Show Today Button** - Controls whether the today button displays to select the current date.
      - **Show Clear Button** - Controls whether the clear button displays to clear the value of the date picker. Note: This is only used when **Display As** is **Inline**.
   b. Days Outside Month - Dictates the behavior of days outside the current viewing month. Only applies to Date Pickers when **Display As** is either **Popup** or **Inline**.

      Options include:

      - **Hidden** - Days outside the current viewing month will be hidden.
      - **Selectable** - Days outside the current viewing month will be visible and selectable.
      - **Visible** - Days outside the current viewing month will be visible.
   c. Time Increment - Time increment in minutes used by the time selection control. Use this attribute to select the time of increments that can still be manually entered and saved. Only applies to Date Pickers when **Display As** is either **Popup** or **Inline**.
   d. Show On - Controls when the calendar popup displays. Only applies to Date Pickers when **Display As** is **Popup**.

      Options include:

      - **Item Focus** - When the item receives focus or when the calendar icon is clicked.
      - **Icon Click** - Only when the calendar icon is clicked.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

16.6.5 Configuring Geocoded Address

Configure Geocoded Address behavior in Component Settings, Geocoded Address.

A Geocoded Address item provides geocoding functionality that turns a postal address to a coordinate. To learn more, see Managing Page Items and Geocoded Address.

To configure default Geocoded Address behavior:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Geocoded Address**.
4. Settings:
   a. Geocoder Match Mode - Determines how closely the attributes of an input address must match the data being used for geocoding. Options include:
      - **Exact** - All provided address parts must match. However, if the house number, street name, or street type do not all match, the first match in the following is returned: postal code, city or town, and state. For example, if the street name is incorrect but a valid postal code is specified, a location in the postal code is returned.
      - **Relax Street Type** - The provided street type can be different from the data used for geocoding. For example, `Main Street` or `Main Blvd` match `Main St`, if there is no other `Main Blvd` or `Main Street` in the relevant area.
      - **Relax House Number** - The house number and street type can be different from the data used for geocoding. For example, `123 Main St` matches `123 Main Lane` and `124 Main St`, as long as there are no ambiguities or other matches.
      - **Relax Street Name** - The base name of the street, the house number, and the street type can be different from the data used for geocoding. For example, `Pleasant Vale` matches `Pleasant Valley` as long as there are no ambiguities or other matches in the data.
      - **Relax Postal Code** - The postal code (if provided), street name, street type and house number can be different from the data used for geocoding.
      - **Relax All** - The address can be outside the city specified as long as it is within the same county. Also includes the characteristics of **Relax Postal Code**.
   b. Display As - Select how the geocoding results display. Options include:
      - **List View** - The results display in a list view.
      - **Grid View** - The results display in a grid view.
   c. Map Preview - Determines whether to support previewing the geocoded address on a map. Options include:
      - **Results Popup** - Show the map in the Geocoding results popup, so that users can review the position before selecting a result. This is only supported when **Display As**, **Grid View** is enabled.

- **Item** - Show a map visualizing the address location at the item position.

d. Background - Select an out-of-the-box map background or choose a custom map background defined in Shared Components. Options include:

- **Default** - Use the default map background. (either **OpenStreetMap Bright** or **OpenStreetMap Dark** for dark themes).
- **OpenStreetMap Bright** - General purpose basemap based on OpenStreetMap data. This basemap style provides detailed location context for your data.
- **OpenStreetMap Dark** - Non-obtrusive dark basemap based on OpenStreetMap data. This basemap style accentuates visualizations of your data.
- **OpenStreetMap Positron** - Non-obtrusive light basemap based on OpenStreetMap data. This basemap style accentuates visualizations of your data.
- **Oracle World Map** - General purpose basemap based on HERE Map data. This base map style provides location context for your data and accentuates visualization of your data.
- **Shared Component** - Use custom map backgrounds defined in Shared Components.

> **Note:**
>
> To learn more about contributors and licensing for out-of-the-box map backgrounds, or to view them on Gitbub, see item Help.

e. Custom Background - Only displays if **Background** is set to **Shared Component**. Select a background map to use.

f. Height - Specify the height (in pixels) of the displayed map. Map's default height is calculated based on it's width with an 3:2 aspect ratio.

5. Subscription:

- Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
  If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

16.6.6 Configuring Interactive Report Action Menu Structure

Configure if interactive reports use the new Action Menu or keep the legacy structure from earlier releases.

The structure of the interactive report Actions menu has changed to be consistent with interactive grids.

To configure how the Actions menu displays in interactive reports:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Interactive Report**.
4. Settings, Actions Menu Structure - Specify how the Actions menu displays in interactive reports. Options include:

   Options include:

   - **Interactive Grid** - Select **Interactive Grid** to have the interactive report Actions menu match interactive grid.
   - **Legacy** - Select **Legacy** to have the interactive report Actions menu use the legacy structure.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.7 Configuring Vector Tile Layers for Maps

Configure Map component settings to include Vector Tile Layers as the background map.

Some of the background maps are also available as Vector Tile Layers which include the layers: `OpenStreetMap Position`, `OpenStreetMap Bright`, and `OpenStreetMap DarkMatter`. Vector maps provide superior resolution over raster maps for all map zoom levels. Started with Oracle APEX release 22.2, all new applicationsVector Tile Layer usage is enabled by default. For aapplications created before release 22.2, you must enable Vector Tile Layers in Shared Components, Component Settings.

To enable Vector Tile Layers for maps:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Maps**.
4. Settings, Use Vector Tile Layers - Select **Yes** to use Vector Tile Layers as the background map. If maps.oracle.com provides a Vector Tile Layer for the chosen background map and this attribute is enabled, the map region will use it.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

16.6.8 Configuring Oracle Cloud Applications (BOSS) REST Service

Configure Component Settings for an Oracle Cloud Applications (BOSS) REST Service.

To configure Component Settings for an Oracle Cloud Applications (BOSS) REST Service:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Oracle Cloud Applications (BOSS) REST Service**.
4. Settings, Advanced Settings - Provide the case-sensitive names of Advanced Settings to be used with the guidance of Oracle Support. Add one setting name per line.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

> **See Also:**
>
> Working with Oracle Cloud Applications BOSS REST Services

16.6.9 Configuring an Oracle Cloud Applications (SaaS) REST Service

Specify a sandbox name for use in App Builder and at runtime when using a Oracle Cloud Applications (SaaS) REST Service.

To configure a sandbox for use in the App Builder and at runtime when using this Oracle SaaS REST service.:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Oracle Cloud Applications (SaaS) REST Service**.

4. Settings, Sandbox Name - Provides the name of a metadata sandbox (if any) to use in the App Builder and at runtime when using this Oracle SaaS REST service. .

   Oracle SaaS REST endpoints automatically reflect application customizations made to the system. If you need to develop an APEX application against in progress customizations being made in a named metadata sandbox, enter a case sensitive name of the sandbox that APEX should use. If the value is null, then APEX works with the version of the REST service that only reflects published customizations

5. Subscription:

   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

> **See Also:**
>
> - Working with REST Data Sources for Oracle Cloud SaaS Applications
> - Working Against a Sandbox

16.6.10 Dynamically Adding Sliders to Regions

Dynamically add a slider to regions if the displayed region name exceeds the width of the region by configuring the Component Settings, Region Display Selector.

To dynamically add a slider to a region if the displayed region name exceeds the width of the region:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Other Components, click **Component Settings**.

3. On the Component Settings page, click **Region Display Selector**.

4. Settings, Include Slider - Select **Yes** to dynamically add a slider to the region if the displayed region name exceeds the width of the region.

5. Subscription:

   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

16.6.11 Configuring Select Many

Specify how to display values when the item is not focused in Component Settings, Select Many.

To configure Component Settings for Select Many:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Select Many**.
4. Settings, Display Value As - Specify how to display values when the item is not focused.

   Available options include:
   - **Chips**
   - **Comma-separated list**
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.12 Configuring Server Side Geocoding

Configure Component Settings for Server Side Geocoding.

To configure Component Settings for Server Side Geocoding:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Server Side Geocoding**.
4. Settings:
   - Match Mode - Determines how closely the attributes of an input address must match the data being used for geocoding
     - **Exact** - All provided address parts must match. However, if the house number, street name, or street type do not all match, the first match in the following is returned: postal code, city or town, and state. For example, if the street name is

incorrect but a valid postal code is specified, a location in the postal code is returned.

- **Relax Street Type** - The provided street type can be different from the data used for geocoding. For example, `Main Street` or `Main Blvd` match `Main St`, if there is no other `Main Blvd` or `Main Street` in the relevant area.
- **Relax House Number** - The house number and street type can be different from the data used for geocoding. For example, `123 Main St` matches `123 Main Lane` and `124 Main St`, as long as there are no ambiguities or other matches.
- **Relax Street Name** - The base name of the street, the house number, and the street type can be different from the data used for geocoding. For example, `Pleasant Vale` matches `Pleasant Valley` as long as there are no ambiguities or other matches in the data.
- **Relax Postal Code** - The postal code (if provided), street name, street type and house number can be different from the data used for geocoding.
- **Relax All** - The address can be outside the city specified as long as it is within the same county. Also includes the characteristics of **Relax Postal Code**.

5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.13 Configuring Show AI Assistant

Configure Show AI Assistant component settings to set application-level values for the Show AI Assistant plug-in.

Configure the Show AI Assistant Settings to change the **AI Avatar** and **User Avatar** that display in Show AI Assistant.

To configure Show AI Assistant:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**.
3. On the Component Settings page, click **Show AI Assistant**.
4. Settings, AI Avatar Type - Select the type of avatar for the AI assistant.

   Options include:
   - Default
   - Icon Classes - If selected, enter the Icon Classes.
   - Image URL- If selected, enter the Image URL.

- Initials - If selected, enter the Image URL.

Depending upon the option you select, additional settings appear.

5. User Avatar Type - Select the type of avatar for the current application user. Options include:

   Depending upon the option you select, additional settings appears.

   - Default
   - Icon Classes - If selected, enter the Icon Classes.
   - Image URL- If selected, enter the Image URL.
   - Initials - If selected, enter the Image URL.

6. Subscription:

   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

7. Click **Apply Changes**.

16.6.14 Configuring Star Rating

Configure default Star Rating behavior in Component Setting, Star Rating.

To configure default Star Rating behavior:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Other Components, click **Component Settings**.

3. On the Component Settings page, click **Star Rating**.

4. Settings:

   a. Default Icon - Choose the default icon to use (such as `fa-star`, `fa,-thumbs-up`, or `fa-database`).

   b. Active Color - Choose the default color for active stars.

   c. Inactive Color - Choose the default color for inactive stars.

   d. Tooltip - Specify the template for the tooltip when end users hover the mouse over the item. This rating value replaces the `#VALUE#` substitution (for example, the movie rating is `#VALUE#`).

5. Subscription:

   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.

6. Click **Apply Changes**.

16.6.15 Configuring Switch Defaults

Configure default Switch item type behavior Component Settings, Switch.

Display options include On/Off switch toggle, pill button, or select list.

To configure default Switch item type behavior:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Settings**
3. On the Component Settings page, click **Switch**.
4. Settings:
   a. Display Style - Options include:
      - **Switch**
      - **Pill Button**
      - **Select List**
   b. On Value - Enter the value stored in the page item if the end user picks the **On** option. The value can be up to 255 characters long.
   c. On Label - Enter the text displayed for the **On** option. If no text is entered, the system default label is used.
   d. Off Value - Enter the value stored in the page item if the user picks the **Off** option. The value can be up to 255 characters long. This value is used if the page item is populated with a value which does not match either the On Value or **Off Value**.
   e. Off Label - Enter the text displayed for the **Off** option. If no text is entered, the system default label is used.
5. Subscription:
   - Subscribe From - Select a master component to subscribe from. When you select a master component and click on the **Apply Changes**, the component settings is automatically refreshed from the selected master.
     If the component setting is already subscribed, then click **Refresh Component Settings** to refresh the component settings from the master component.
6. Click **Apply Changes**.

16.6.16 Viewing the Component Settings History Report

View a report of recent modifications made to Component Settings in the current application.

To view the Component Settings History report:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.

b. Select an application.

c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

2. Under Other Components, click **Component Settings**.

3. Click the **History** tab.

The History page displays as an interactive report. To customize the report, use the Search bar at the top of the page.

16.7 Creating Lists of Values at the Application-Level

Create a list of values at the application-level that can be referenced by any page item or report field.

> **Tip:**
>
> To learn about defining a column as a list of values, go to Developing Reports and find the "Defining a Column as a List of Values" discussion under the appropriate report type.

- About Lists of Values
  A list of values is a static or dynamic definition used to display a specific type of page item.
- Creating a Static List of Values at the Application-Level
  Run the Create List of Values Wizard to define a named static list of values at the application-level.
- Creating a Dynamic List of Values at the Application-Level
  Run the Create List of Values Wizard to define a named dynamic list of values at the application-level.
- Editing an Existing List of Values
  Edit an existing list of values from Shared Components.
- Copying or Subscribing to a List of Values
  Copy a list of values from the current application or from another applications. When copying a list of values from another app, you can also subscribe to it.
- Subscribing to a List of Values
  Subscribe to a list of values on the Lists of Values, Edit page.
- About Referencing Session State Within a List of Values
  Reference session state in a list of values by using bind variables.
- Accessing Lists of Values Reports
  Access lists of values reports in Shared Components.

> **See Also:**
>
> - Creating Page Items
> - Managing Shared Components

16.7.1 About Lists of Values

A list of values is a static or dynamic definition used to display a specific type of page item.

A list of values can be static or dynamic. A **static** list of values is based on values the developer enters. A **dynamic** list of values is based on a local data source, REST Enabled SQL reference, or REST Data Source. A list of values can be referenced by page items such popup lists of values, a select list, a checkbox, a radio group, or multiple select lists.

Advantages of creating a list of values as a shared component include:

- A list of values created as a shared component can be added to any page within an application.
- All list of values definitions are stored in one location, making them easy to locate and update.
- You can define lists of values on different data sources and make use of new functionality only included in Shared LOVs, for example multi-column support, icon and grouping columns.

Creating and using a named list of values at the application level, involves two steps:

1. Navigate to Shared Components and run the Create List of Values Wizard and add it to the List of Values repository.
2. In Page Designer, configure the page item attributes to display the item using the named list of values.

16.7.2 Creating a Static List of Values at the Application-Level

Run the Create List of Values Wizard to define a named static list of values at the application-level.

Oracle APEX stores named (or shared) lists of values in the List of Values repository in Shared Components. A static list of values contains display values, return values, and the sort sequence that you specify. Once you add a list of values to the repository, you edit the item in Page Designer and select the display Type.

The following procedure demonstrates how to create a static list of values in Shared Components for the form item, Status, and then display it as a Radio Group. The following example is built on the `EBA_DEMO_IR_PROJECTS` table which available in the sample dataset, Project Data.

To create a named static list of values:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
>
>
>) on the Page Designer toolbar.

2. Under Other Components, select **Lists of Values**.
3. Run the Create List of Values Wizard. On the Lists of Values page, click **Create**.
4. In the Create List of Values Wizard:
   a. Source:
      - Create List of Values - Specify how to create a list of values. For this example, select **From Scratch**.
      - Click **Next**.
   b. Name and Type:
      - Name - Enter a name for the list of values. For example:

        ```
        Status
        ```

      - Type - Select **Static**.
      - Click **Next**.
   c. Static Values: Enter static display and return values. For this example, enter the following
      - Sequence 1:
        - Display Value - Pending
        - Return Value - Pending
      - Sequence 2:
        - Display Value - Open
        - Return Value - Open
      - Sequence 3:
        - Display Value - On-Hold
        - Return Value - On-Hold
      - Sequence 4:
        - Display Value - Closed
        - Return Value - Closed

      Values will display in the order entered. You can display additional attributes including build option controls and item level conditional display by editing the List of Values later.

> **Note:**
>
> Return Value does not display, but is the value that is returned to the Oracle APEX engine. If you do not specify a Return Value then it is equal to the Display Value.

d. Click **Create List of Values**.

   The new list of values appears on the Lists of Values page.

5. In Page Designer, update the form item to reference the list of values:

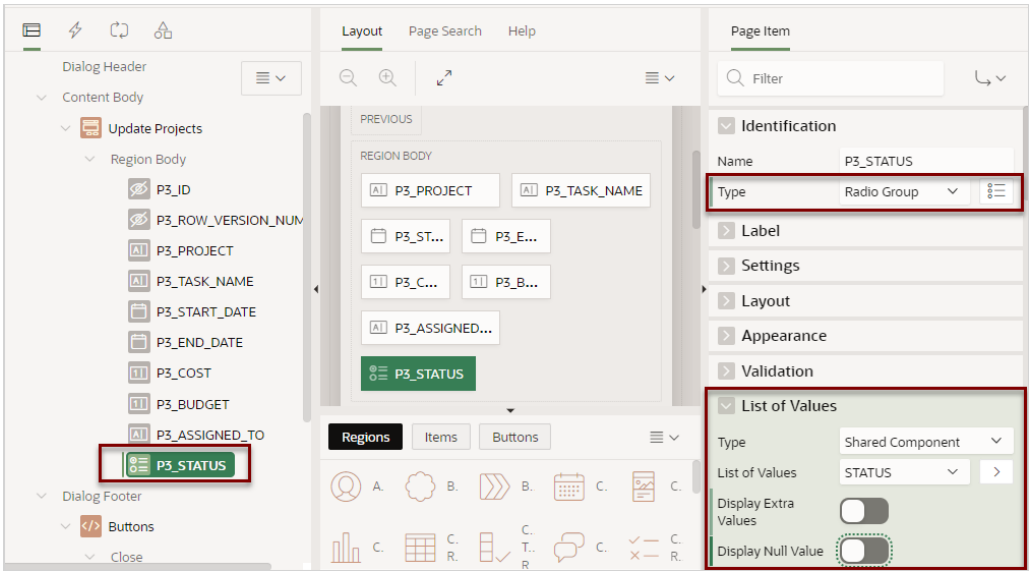
   a. Return to Page Designer. Click **Edit Page X** on the Page Designer toolbar where *X* is the page number.

      Page Designer appears, displaying the form page, Update Projects.

   b. In the Rendering tab, select the form page item. For example, select **P3_STATUS**.

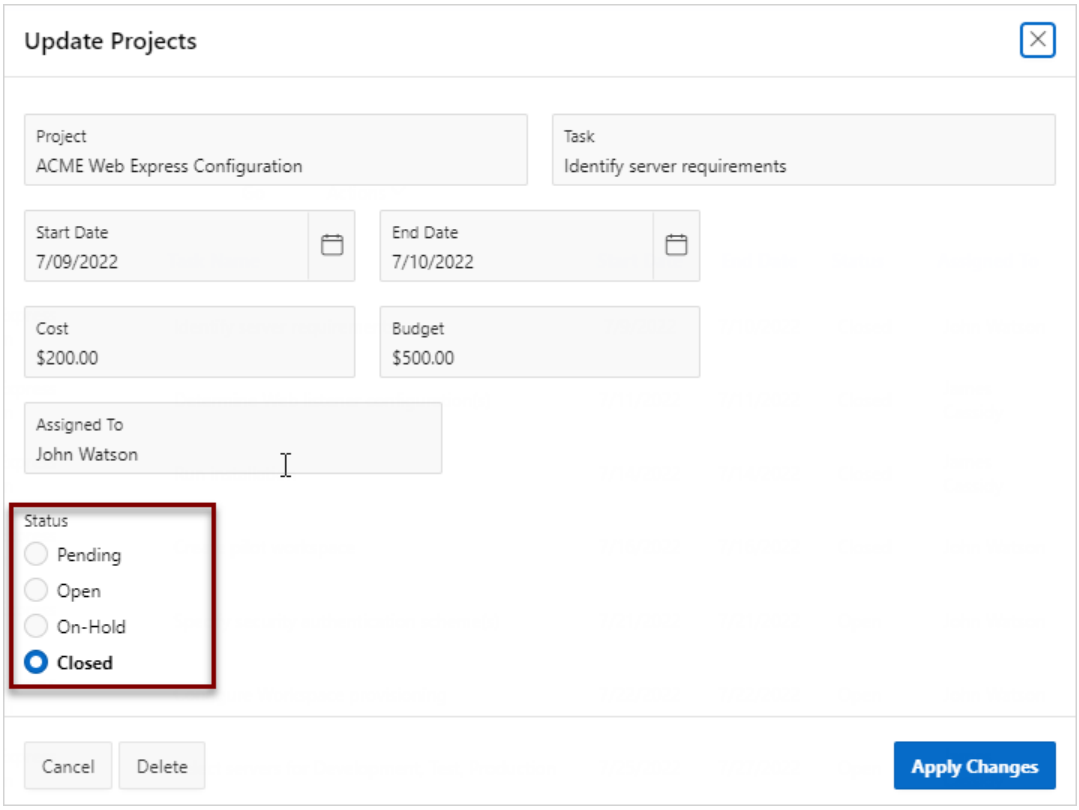
   c. In the Property Editor, edit the following attributes:

      - Identification, Type - Select **Radio Group**.
      - List of Values, Type - Select **Shared Component**.
      - List of Values, List of Values - Select **STATUS**.
      - List of Values, Display Extra Values - Disable this option.
      - List of Values, Display Extra Values - Disable this option.



   d. Click **Save** and then run the page.

As shown in the folllowing example, the form item, Status, now displays as Radio Group with four values: **Pending**, **Open**, **On-Hold** and **Closed**.



16.7.3 Creating a Dynamic List of Values at the Application-Level

Run the Create List of Values Wizard to define a named dynamic list of values at the application-level.

Oracle APEX stores named (or shared) list of values in the List of Values repository. A dynamic list of values retrieves data from a local data source, a remote connection defined using REST Enabled SQL, or from a RESTful web service defined using REST Data Sources.

The following procedure creates dynamic list of values for the form item, **Task Name** and then displays it as a Select List. This following example is built on the `EBA_DEMO_IR_PROJECTS` table which available in the sample dataset, Project Data.

To create a named dynamic list of values:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components** (
>
>
>
>) on the Page Designer toolbar.

2. Under Other Components, select **Lists of Values**.
3. Run the Create List of Values Wizard. On the Lists of Values page, click **Create**.
4. In the Create List of Values Wizard:
   a. Source:
      - Create List of Values - Specify how to create the list of values. For this example, select **From Scratch**.
      - Click **Next**.
   b. Name and Type:
      - Name - Enter a name for the list of values. For example:

        ```
        Task
        ```

      - Type - Select **Dynamic**.
      - Click **Next**.
   c. List of Values Source:
      - Data Source - Select **Local Database**.
      - Source Type - Lists using a Local Database can be based on a Table, SQL Query or PL/SQL Function Returning SQL.

        For this example, select **Table**.
      - Table / View Owner - Select the owner of the table on which you are building the list of values.
      - Table / View Name - Select the table or view on which the list of values will be based.

        Select **EBA_DEMO_IR_PROJECTS**.
      - Click **Next**.
   d. Column Mappings:
      - Return Column - Defines the column used as the return value for your list of values. Accept the default, **ID**.
      - Display Column - Define the column that used as the display value for your list of values. Accept the default, **TASK_NAME**.

        You can define additional display columns after creating a list of values for item types that support displaying multiple columns (such as the Popup LOV).
   e. Click **Create**.

      The new list of values appears on the Lists of Values page.

5. In Page Designer, return to the form page and update the form item to reference the list of values:

   a. Return to Page Designer. Click **Edit Page X** on the Page Designer toolbar where *X* is the page number.

      Page Designer appears.

   b. View the form page (for example, Update Projects) in Page Designer.

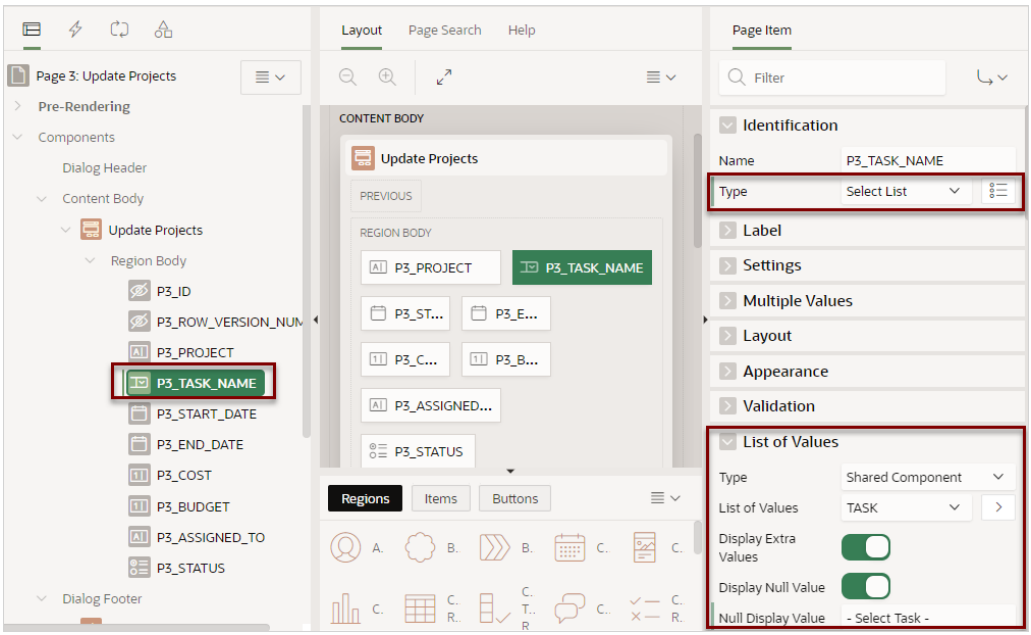
   c. In the Rendering tab, select the form page item. For example, select **P3_TASK_NAME**.

   d. In the Property Editor, edit the following attributes:

      - Identification, Type - Select **Select List**.
      - List of Values, Type - Select **Shared Component**.
      - List of Values, List of Values - Select **TASK**.
      - List of Values, Display Extra Values - An item may have a session state value which does not occur in the given list of values definition. Select whether this list of values displays this extra session state value. Accept the default.
      - List of Values, Display Null Value - When enabled, a `NULL` value displays at the top of the list. Accept the default.
      - List of Values, Null Display Value - Enter the text to be displayed within the list NULL option at the top of this list. Enter:

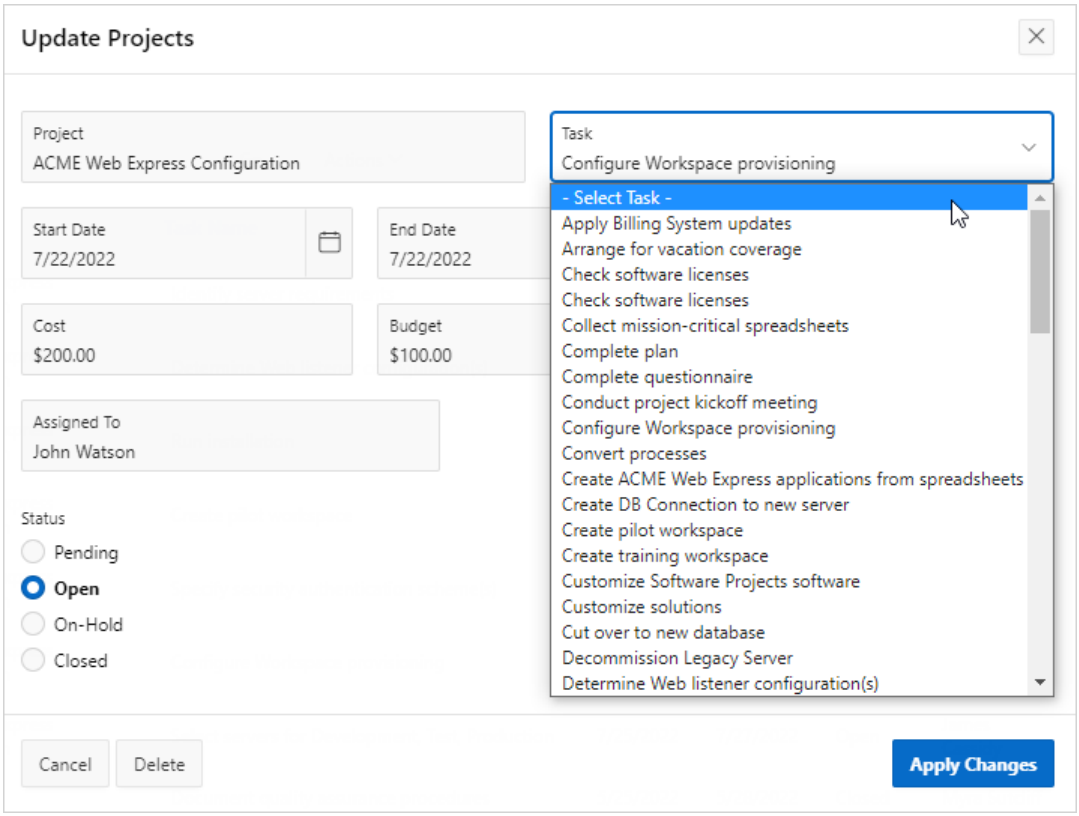
        ```
        - Select Task -
        ```

        This value also displays when rendering the item if the value is null. If you do not enter any text, it is displayed as a blank line in the list and a blank item when rendered.



   e. Click **Save** and then run the page.

The form item, **Task Name**, now displays as select list.



16.7.4 Editing an Existing List of Values

Edit an existing list of values from Shared Components.

- Editing a List of Values in Shared Components
  Edit a list of values by navigating to Shared Components, Other Components and selecting List of Values.
- Editing List of Values in Bulk
  Edit the display values of all static list of values by clicking the Grid Edit tab on the List of Values page.
- Subscribing to a List of Values
  Subscribe to a list of values on the Lists of Values, Edit page.

16.7.4.1 Editing a List of Values in Shared Components

Edit a list of values by navigating to Shared Components, Other Components and selecting List of Values.

To edit a list of values from the Shared Components page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

> **Tip:**
>
> You can also access Shared Components from Page Designer by clicking **Shared Components**
> (
>
>) on the Page Designer toolbar.

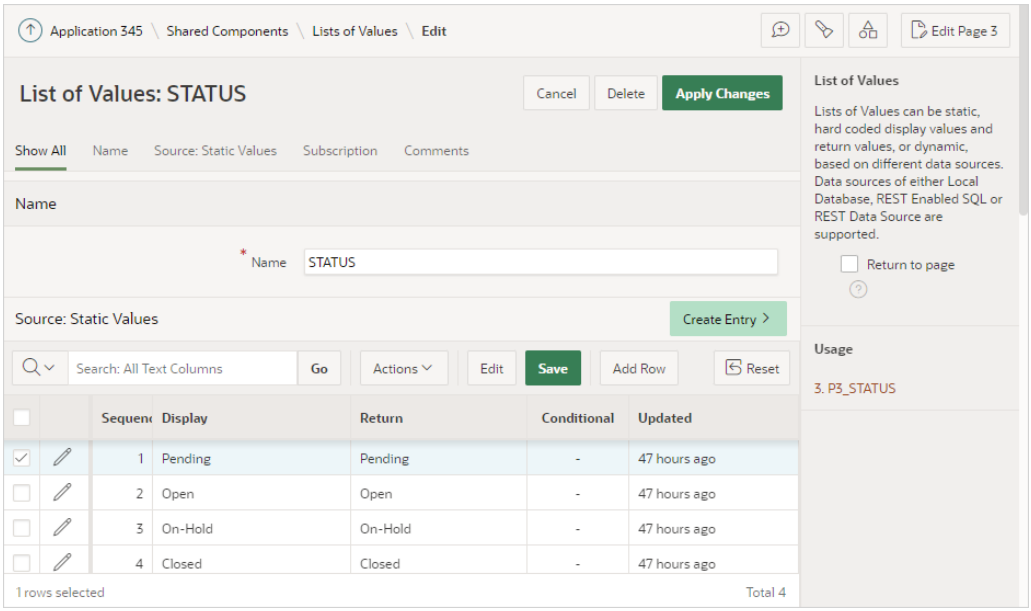
2. Under Other Components, select **Lists of Values**.

   The Lists of Values page appears.

3. Select a list of values.

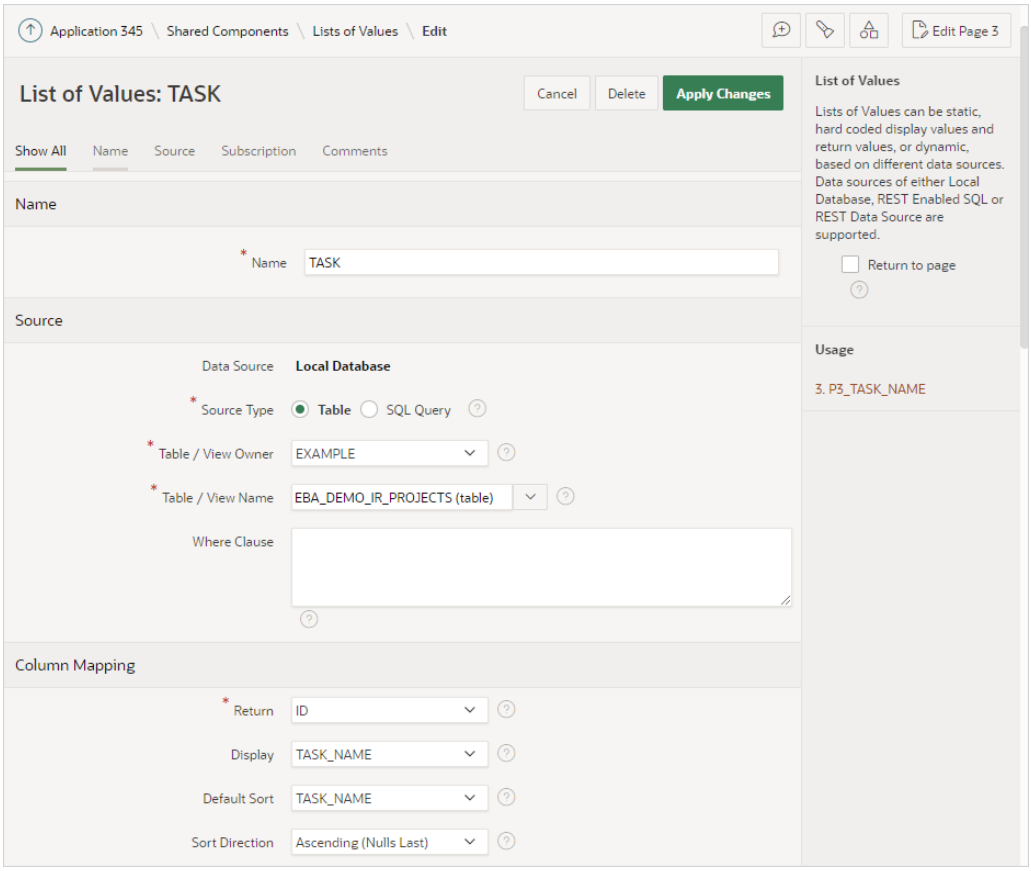
   The Edit page appears. The appearance of the Edit page changes depending upon the type of list as shown in the following examples:

   - **Static List of Values** - For Static lists, the Edit page resembles the following example. In this example, STATUS contains four Display and Return values: Pending, Open, On-Hold, and Closed.



   Click the **Edit** icon edit a value or the sequence. Click **Add Row** to add more Display/Return value combinations.

   - **Dynamic List of Values** - For Dynamic lists, the Edit field resembles the following example. In this example, the list is based on the `EBA_DEMO_PROJECTS` table in a local database.



Edit the attributes to reconfigure the Source or Column Mappings.

Use **Additional Display Columns** to define additional display columns for item types that support multiple display columns, such as Popup LOV. If adding additional display columns, ensure that the return column is included in the column list. If you do not want it displayed to users, set the return column to **Visible No** and **Searchable No** .

> **Tip:**
>
> To learn more about an attribute, see field-level Help.

4. Click **Apply Changes** to save your changes.

16.7.4.2 Editing List of Values in Bulk

Edit the display values of all static list of values by clicking the Grid Edit tab on the List of Values page.

To perform a bulk edit of static LOVs:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **List of Values**.

3. Click the **Grid Edit** tab.
4. Edit the Display Text and click **Apply Changes**.

16.7.4.3 Subscribing to a List of Values

Subscribe to a list of values on the Lists of Values, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a list of values by copying it and then subscribing. See Copying or Subscribing to a List of Values.

To subscribe to a list of values:

1. Navigate to the Lists of Values page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Other Components, click **Lists of Values**.

      The Lists of Values page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select a list of values.

   The Lists of Values, Edit page appears.
3. To subscribe to a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master list of values to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Refresh List - If the list of values is already subscribed, click **Refresh List**.
   c. Click **Apply Changes**.
5. To unsubscribe from a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.7.5 Copying or Subscribing to a List of Values

Copy a list of values from the current application or from another applications. When copying a list of values from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a list of values:

1. Navigate to the Lists of Values page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Other Components, click **Lists of Values**.

      The Lists of Values page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. From the Tasks list, click **Copy from another app**.
3. Copy From:
   a. Copy From Application - Select an application from which to copy.
   b. Click **Next**.
4. New Lists of Values:
   a. To Name - Optionally edit the list of values name.
   b. Copy - Select an action:
      - **Yes** - Copy the list of values.
      - **No** - Do not copy the list of value.
      - **Copy and Subscribe** - Copy and subscribe to this list of values. Subscribing to a list of values copies it to application and sets the plug-in master reference.
5. Click **Copy List of Values**.

16.7.6 Subscribing to a List of Values

Subscribe to a list of values on the Lists of Values, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a list of values by copying it and then subscribing. See Copying or Subscribing to a List of Values.

To subscribe to a list of values:

1. Navigate to the Lists of Values page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Other Components, click **Lists of Values**.

      The Lists of Values page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select a list of values.

   The Lists of Values, Edit page appears.
3. To subscribe to a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master list of values to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Refresh List - If the list of values is already subscribed, click **Refresh List**.
   c. Click **Apply Changes**.
5. To unsubscribe from a list of values:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.7.7 About Referencing Session State Within a List of Values

Reference session state in a list of values by using bind variables.

While referencing session state makes a list of values a bit less reusable, Oracle still recommends this approach as a best practice. In the following example, the list of values only works if the item called *my_deptno* contains a valid department number.

```
SELECT ename, empno FROM emp WHERE deptno = :P1_DEPTNO
```

16.7.8 Accessing Lists of Values Reports

Access lists of values reports in Shared Components.

To access lists of values reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **List of Values**.
3. Select one of the following tabs at the top of the page:
   - **Lists of Values** - Displays all lists of values that are defined locally with the definition of an item. Lists of values that are usable by multiple pages can be converted into shared lists of values. Converting locally defined lists of values into a shared list of values promotes reuse.
   - **Edit Null Text** - Edit null text for all lists of values in your application.
   - **Locally Defined** - View all lists of values that are defined locally with the definition of the item. Lists of values that are usable by multiple pages can be converted into shared lists of values. Converting locally defined lists of values into a shared list of values promotes reuse..
   - **Grid Edit** - Find and edit static list of value entries over all static lists of values. Use this page to standardize display text.
   - **Conditional Entries** - Displays all lists of values that display conditionally.
   - **Subscription** - Displays all subscribed lists of values in your application.
   - **Utilization** - Displays where lists of values are used in the current application.
   - **History** - Displays displays recent modifications made to lists of values in in the current application.
4. Follow the on-screen instructions.

16.8 Creating Lists

Add a shared collection of links (or a list) to an application. To add a list to a page, create a region and specify the region type as List. You control the appearance of a list through list templates.

- About Static and Dynamic Lists
  Oracle APEX supports two types of lists: static lists and dynamic lists.
- Creating Static Lists
  A static list is based on predefined display and return values.
- Creating Dynamic Lists
  A Dynamic List is based on a SQL query or a PL/SQL function executed at runtime.
- Editing List Attributes
  Edit a list in Shared Components.

- Adding a List to a Page
  Once you create a list and list entries, the next step is to add it to a page by creating a list region.
- Deleting a List
  Delete a list on the List Details page.
- Copying or Subscribing to a List
  Copy a list from the current application or from another application. When copying a list from another app, you can also subscribe to it.
- Subscribing to a List
  Subscribe to a list on the List Details page.
- Accessing List Reports
  View reports on the Lists by clicking the following tabs: Unused, Conditional Entries, Utilization, and History. View List Details to access a Grid Edit of all list entries.
- Managing Navigation Menus
  Applications using newer themes, such as *Universal Theme - 42,* provide navigation with navigation menus.
- Managing Navigation Bar Lists
  Applications using newer themes, such as *Universal Theme - 42,* include navigation bar lists.

See Also:

- Creating a New Template
- List Templates

16.8.1 About Static and Dynamic Lists

Oracle APEX supports two types of lists: static lists and dynamic lists.

A **static list** is based on predefined display and return values. When you create a static list you define a list entry label and a target (either a page or URL). You can add list entries when you create the list (creating from scratch), by copying existing entries, or by adding the list entries. You can control when list entries display by defining display conditions.

A **dynamic list** is based on a SQL query or a PL/SQL function executed at runtime. A dynamic list enables you to dynamically create styled list items that support mobile frameworks.

The list definition displays a specific type of page item, such as progress bars, sidebar, bullet navigation list, or navigation menu. You can control how a list displays through templates.

See Also:

- Creating Static Lists
- Creating Dynamic Lists

16.8.2 Creating Static Lists

A static list is based on predefined display and return values.

- About Creating a Static List
  Learn about the steps involved in creating a static list.
- About Adding Entries and Sublists to a Static List
  Once you create a list, you must add entries to it.
- Creating a Static List from Scratch
  Create a static list from scratch by running the Create List Wizard.
- Adding Entries or Sublists Manually
  Add list entries or sublists from the List Details page.
- Copying Static List Entries Between Lists
  Copy static list entries from one list to another.
- Reparenting Static List Entries
  Reparent static list entries to manage the list entry hierarchy.
- Managing Orphaned Static List Entries
  Manage orphaned list entries by removing the parent entry.
- Resequencing Static List Entries
  Resequence list entries in increments of 10 on the Clean Up page.

16.8.2.1 About Creating a Static List

Learn about the steps involved in creating a static list.

A **static list** is based on predefined display and return values. When you create a static list you define a list entry label and a target (either a page or URL). You can add list entries when you create the list, by copying existing entries, or by adding the list entries. You can control when list entries display by defining display conditions.

The process of creating a static list involves the following steps:

- **Step 1**: Create the static list by either:
  - Creating a list from scratch by running the Create List Wizard. When prompted enter the list name and select the list type, **Static**.
  - Copying an existing static list by running the Copy List Wizard.

  Each list element has a display condition, which enables you to control when it displays. You can define a list element to be *current* for a specific page or for multiple pages.
- **Step 2:** Add list entries. You can add list entries:
  - When running the Create List Wizard.
  - By copying entries from within an existing list. If you copy an entire list, you also copy all of its list entries.
  - By adding list entries manually.
- **Step 3:** Add the list to a page by creating a List region.

How you perform these steps is a personal preference. You can perform all of these steps when you run the Create List Wizard. Alternatively, you can perform these steps one at a time.

See Also:

- Adding a List to a Page
- Editing a Dynamic List
- Accessing List Reports

16.8.2.2 About Adding Entries and Sublists to a Static List

Once you create a list, you must add entries to it.

You can add list entries when running the Create List Wizard, copying a list entry from within a list, or copying existing entries from one list to another.

You can also create hierarchical lists that contain sublists. To create a hierarchical list, you must:

- Select a list template that supports hierarchical lists.

  Tip:

  To determine if a list template supports hierarchical lists, go to Shared Components, Templates. On the Templates page, find and click the list template name. The **List Template Current with Sublist Items** attribute on the Edit List Template page determines whether the list template supports hierarchical lists. If this attribute is not null, the list template supports hierarchical lists. If this attribute is null, then the list template does not support hierarchical lists.

- Select a Parent List Item when you create each list entry.

See Also:

- Managing Templates
- Creating Custom Themes
- List Templates

16.8.2.3 Creating a Static List from Scratch

Create a static list from scratch by running the Create List Wizard.

To create a static list by running the Create List Wizard:

1. Access the Create List Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Lists**.

   e. Click **Create**.

   The Create List Wizard appears.

2. For Name and Type:
   a. Name - Enter a numeric or alphanumeric name for the list.
   b. Type - Select **Static**.
   c. Click **Next**.
3. For Query or Static Values:
   a. List Entry Label - Enter a numeric or alphanumeric name for the list.
   b. Target Page ID or custom URL - Select a target page or enter a custom URL.
   c. Click **Next**.
4. For Confirm:
   a. Create List Regions - Select whether to create a list region. Options include:
      - **Do not create list region(s)**
      - **Create list region on current page**
      - **Create list region for each target page**
   b. Region Position - If creating a new region, select the region position.
   c. Region Template - If creating a new region, select a region template.
   d. List Template - Select the list template to use to control the appearance of your list.
   e. Click **Create List**.

16.8.2.4 Adding Entries or Sublists Manually

Add list entries or sublists from the List Details page.

To add an entry or sublist manually:

1. Navigate to the Lists page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Lists**.

      The Lists page appears.
2. Select a list.

   The List Details page appears.
3. Click **Create Entry**.
4. Under Entry:
   a. Parent List Entry - If applicable, select the parent for this list entry. Use this attribute if you are creating a hierarchical list that contains a sublist.

      Note this attribute does not appear if you are creating the parent list entry.
   b. Sequence - Indicate the order in which list the list entry appears. The sequence determines the order of evaluation.

c. Image/Class - Identify an image file name or the the icon class for the icon associated with this list entry.

   Most Universal Theme apps use the Font APEX icon library. To pick an icon, use the built-in icon picker and select an icon from the popup window. See examples below:

   - `fa-cloud` - Displays the cloud icon.
   - `fa-exclamation-circle u-danger-text` - Displays a red exclamation icon.

   To learn more, see field-level Help.

d. Attributes - Identify the image attributes (such as `width="12" height="12"`) for the list element image.

   Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string enables the title image attribute to be automatically set based on the value of the list label text. For example:

   ```
   title="#LIST_LABEL#"
   ```

e. Alt Attribute - Identify the image alt attribute for the list element image.

   Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string enables the Alt attribute to be automatically set based on the value of the list label text. For example:

   ```
   ...alt="#LIST_LABEL#"...
   ```

f. List Entry Label - Enter the label text for this link.

5. Specify a target type. Options include: **URL** or **Page in this Application**.

   If the target location is a **URL**, specify the following:

   a. Target Type - Select **URL**.

   b. URL Target - Enter a URL. For example:

      http://www.example.com

   If the target location is a page:

   a. Target Type - Select **Page in this Application**.

   b. Page - Specify the target page number.

      Other options include:

      - **reset pagination for this page** - Select this option to have the page to return to the first set of data meeting a user's new query.
      - **Printer Friendly** - This option displays the target page using the application's Printer Friendly template. Printer friendly templates optimize a page for printing.

   c. Request - Enter text to set the built-in application item `REQUEST`. This is also the item that is set with the name of a button that was clicked.

   d. Clear Cache - Enter the page numbers, separated by commas, for which you would like the user's session state to be cleared. To learn more, see field-level Help.

   e. To set session state (that is, give a listed item a value):

      - Set these items - Enter a comma-delimited list of item names for which you would like to set session state.
      - With these values - Enter a comma-delimited list of values for the items specified in the previous step.

You can specify static values or substitution syntax (for example, `&APP_ITEM_NAME.`). Note that item values passed to `f?p=` in the URL cannot contain a colon. Additionally, item values cannot contain commas unless you enclose the entire value in backslashes (for example, `\1234,56\`).

6. Under Current List Entry:
   a. List Entry Current for Pages Type - Specify when this list entry should be current based on the page type.

      List items can be current or non-current. Current list items use the current template; non-current list items use the non-current list item template. The actual condition and templates are defined in subsequent attributes.

   b. List Entry Current for Condition - Displays conditionally. Based on the selection above, define a condition to evaluate. When this condition is true, then the list item becomes current. To learn more, see field-level Help.
7. Under Conditions, specify the appropriate information to make the list entry conditional.
8. Under Authorization, you can specify an authorization scheme.

   This authorization scheme must evaluate to TRUE in order for this component to be rendered or otherwise processed.
9. Under Configuration, select a build option for this component.

   Build options are predefined settings that determine whether components within an application are enabled.
10. Under Click Counting, specify if you want the list entries to be included in the click count.

    If this is a link to an external page, you can count clicks.
11. In User Defined Attributes, specify additional attributes. To learn more, see field-level Help.
12. When you are finished defining list attributes, click **Create List Entry**.

> **See Also:**
>
> - Editing Themes
> - Optimizing a Page for Printing
> - COUNT_CLICK Procedure in *Oracle APEX API Reference*

16.8.2.5 Copying Static List Entries Between Lists

Copy static list entries from one list to another.

To copy list entries between lists:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

The Lists page appears.

2. Select the list.
3. From the Tasks region on the side bar, click **Copy List Entries from one List to Another**.
4. On Copy List Entries:
   a. Copy List Entries Select the list to copy.
   b. Click **Next**.
5. Click **Copy List Entries**.

16.8.2.6 Reparenting Static List Entries

Reparent static list entries to manage the list entry hierarchy.

To reparent list entries:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears.
2. Select the list.
3. From the Tasks region on the side bar, click **Reparent List Entries within this List**.
4. Use the navigation bar to customize the display and click **Go**.
   - List - Select the list you would like to edit.
   - Start With - Restrict your view to a subset of the hierarchy.
5. To reparent entries:
   a. Select entries to reparent.
   b. Reparent To - Select a new parent.
   c. Select the entries to move and click **Reparent Checked Entries**.

16.8.2.7 Managing Orphaned Static List Entries

Manage orphaned list entries by removing the parent entry.

An orphaned list entry is a list entry which has a parent, but the parent is no longer a member of the current list. When a list entry becomes orphaned, remove the parent entry.

To manage orphaned list entries:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.

   e. Under Navigation and Search, click **Lists**.

      The Lists page appears.

2. Select the list.
3. From the Tasks region on the side bar, click **Manage Orphaned List Entries**.
4. On Manage Orphaned List Entries:
   a. Select the entries to remove.
   b. Click **Clear Parent Entries**.

16.8.2.8 Resequencing Static List Entries

Resequence list entries in increments of 10 on the Clean Up page.

To resequence list entries:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears.
2. Select the list.
3. From the Tasks region on the side bar, click **Resequence List Entries**.
4. Click **Resequence**.

16.8.3 Creating Dynamic Lists

A Dynamic List is based on a SQL query or a PL/SQL function executed at runtime.

- About Creating a Dynamic List
  Learn about the steps involved in creating a dynamic list.
- Understanding Dynamic List Syntax
  Learn about dynamic list syntax.
- Creating a Dynamic List from Scratch
  Create a dynamic list from scratch by running the Create List Wizard.

16.8.3.1 About Creating a Dynamic List

Learn about the steps involved in creating a dynamic list.

A **dynamic list** enables you to create a List component based on items from a SQL query or a PL/SQL function returning a SQL query. In a dynamic list, the number of list entries depends on the list query output. When you delete a list, then corresponding list regions are deleted as well. These dynamic lists can then be rendered on a page using any List Template from your theme.

The process of creating a dynamic list involves the following steps:

- **Step 1**: Create the dynamic list by either:

  - Creating a list from scratch by running the Create List Wizard. When prompted enter the list name and select the list type, **Dynamic**.
  - Copying an existing dynamic list by running the Copy List Wizard.
- **Step 2:** Define the List Query:
  - Select a Query Source Type. Options include:
    - **SQL Query**
    - **Function Returning a SQL Query**
  - Enter a SQL query.
- **Step 3:** Add the list to a page by creating a List region.

> **See Also:**
>
> - Adding a List to a Page
> - Editing a Dynamic List
> - Accessing List Reports

16.8.3.2 Understanding Dynamic List Syntax

Learn about dynamic list syntax.

You create a dynamic list using a SQL query or a function returning a SQL query. Oracle APEX interprets your query's results by using the `SELECT` list column position as indicated in the table below. Since column aliases are ignored, choose any names that best document the information your query returns. Your SQL statement must include a minimum of two columns in its `SELECT` list, but can return up to 17 columns depending on your needs. All optional columns accept a `NULL` value as a placeholder to allow you to specify a non-null value for a later column position. For example, if you select **Badge list** for your list region's Appearance, List Template attribute, then the value of column 8 corresponding to User Attribute 1 determines the text that will appear inside the badge.

> **Tip:**
>
> The documentation for the List region explains which user attribute values are used by different Appearance template options.

Column	Purpose	Datatype	Required?	Notes
1	Level	`NUMBER`	Yes	Depth level for hierarchical lists, otherwise use NULL. Note that the root parent level must have a value of 1 or higher. When using 0, that level is skipped for creating the hierarchy.
2	Label	`VARCHAR2`	Yes	Text to appear at list entry
3	Target URL	`VARCHAR2`	No	Target URL to branch to when list entry is selected
4	Is Current?	`VARCHAR2`	No	Controls the behavior of the list entry. Valid values include 'YES', 'NO', and NULL

Column	Purpose	Datatype	Required?	Notes
5	Icon Name	VARCHAR2	No	The name of icon to be displayed on the list entry
6	Image attributes	VARCHAR2	No	Attributes of the image, such as the width or height
7	Image ALT Text	VARCHAR2	No	Value for Image ALT tag, required for accessibility purposes in templates where the user must click the image.
8	User attribute 1	VARCHAR2	No	N/A
9	User attribute 2	VARCHAR2	No	N/A
10	User attribute 3	VARCHAR2	No	N/A
11	User attribute 4	VARCHAR2	No	N/A
12	User attribute 5	VARCHAR2	No	N/A
13	User attribute 6	VARCHAR2	No	N/A
14	User attribute 7	VARCHAR2	No	N/A
15	User attribute 8	VARCHAR2	No	N/A
16	User attribute 9	VARCHAR2	No	N/A
17	User attribute 10	VARCHAR2	No	N/A

For example, a simple query for a dynamic list of employee names might look like this:

```
select null  c1_level,
       ename c2_name_for_label
from emp
order by ename
```

To include a person icon next to each employee's name in the list, select NULL for the intervening columns and return the name of an icon like fa-user in the fifth column position like this:

```
select null      c1_level,
       ename     c2_name_for_label,
       null      c3_target_url,
       null      c4_is_current,
       'fa-user' c5_icon_name
from emp
order by ename
```

To have the list entries link to page 10, passing the value of `EMPNO` for the `P10_EMPNO` page item, adjust the query to provide a value for the target URL in the third column like this:

```
select null      c1_level,
       ename     c2_name_for_label,
       apex_page.get_url(p_page   => 10,
                         p_items  => 'P10_EMPNO',
                         p_values => EMPNO) c3_target_url,
       null      c4_is_current,
       'fa-user' c5_icon_name
from emp
order by ename
```

The following query returns a value for a user attribute. It expands on the previous query to return each employee's salary (`sal`) in the column 8 slot as User Attribute 1. The List region's **Appearance**, **List Template** option, **Badge List**, uses this User Attribute 1 value as the text to display on the badge.

```
select null      c1_level,
       ename     c2_name_for_label,
       apex_page.get_url(p_page   => 10,
                         p_items  => 'P10_EMPNO',
                         p_values => EMPNO) c3_target_url,
       null      c4_is_current,
       'fa-user' c5_icon_name,
       null      c6_icon_attrs,
       null      c7_icon_alt_text,
       sal       c8_user_attr1_badge_text
from emp
order by ename
```

To create a hierarchical list, return a non-null value for the first column that indicates the depth level of the hierarchy. This example returns a hierarchical list of employees starting with the employee(s) having no manager at the top of the tree.

```
select level                                      c1_level,
       ename                                      c2_name_for_label
       apex_page.get_url(p_page   => 10,
                         p_items  => 'P10_EMPNO',
                         p_values => EMPNO)       c3_target_url,
       null                                       c4_is_current,
       'fa-user'                                  c5_icon_name
from emp
connect by prior empno = mgr
start with mgr is null
```

If you need to return a SQL query defined by a function, the same rules apply. You may find it useful to use PL/SQL's multi-line string literal syntax `q'[ ... ]'` as shown in the following example to more easily return query text that spans multiple lines. Note that any syntax that returns a correctly-formatted SQL statement is allowed.

```
return q'[
   select null      c1_level,
          ename     c2_name_for_label,
```

```
            apex_page.get_url(p_page   => 10,
                              p_items  => 'P10_EMPNO',
                              p_values => EMPNO) c3_target_url,
            null      c4_is_current,
            'fa-user' c5_icon_name
   from emp
   order by ename
]';
```

16.8.3.3 Creating a Dynamic List from Scratch

Create a dynamic list from scratch by running the Create List Wizard.

To create a dynamic list:

1. Access the Create/Edit Lists Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Lists**.
   e. Click **Create**.

   The Create List Wizard appears.
2. For Name and Type:
   a. Name - Enter a numeric or alphanumeric name for the list.
   b. Type - Select **Dynamic**.
   c. Click **Next**.
3. For Query or Static Values:
   a. Query Source Type - Select **SQL Query** or **Function Returning SQL Query**.
   b. Query - Enter a SQL query or function returning a SQL query. To view SQL Query examples, expand the Examples region.

      Build Query - Click **Build Query** to use a wizard to build a query on the table or view you select. Follow the on-screen instructions.
   c. Click **Next**.
4. For Confirm:
   a. Create List Regions - Select whether to create a list region. Options include:
      - **Do not create list region(s)**
      - **Create list region on current page**
   b. Region Position - If creating a new region, select the region position.
   c. Region Template - If creating a new region, select a region template.
   d. Click **Create**.

> **See Also:**
>
> Understanding Dynamic List Syntax

16.8.4 Editing List Attributes

Edit a list in Shared Components.

Each list may contain any number of list entries. Edit the list to set the list name and list template.

- Editing a Static List
  Edit a static list by selecting it on the Lists page in Shared Components.
- Editing a Dynamic List
  Edit a dynamic list by selecting it on the Lists page in Shared Components.

16.8.4.1 Editing a Static List

Edit a static list by selecting it on the Lists page in Shared Components.

Edit static list attributes on the List Details page. To edit a list entry, select the list entry, Name.

To edit a static list:

1. Navigate to the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears. By default, the Lists page displays as a table. Note that the Type column identifies the list type as **Dynamic** or **Static**.
2. Select a Static list.

   The List Details page appears.
3. Name, Name - Defines the name for the list. Both numeric or alphanumeric list names are supported.
4. List Entries - List entries display in a table. To edit a list entry, select the Name of the list entry.

   On the Create/Edit page, edit the List Entry attributes.

   a. Entry:
      - Entry, Parent List Entry - Identify the parent for this list entry. Use this attribute if you are creating a hierarchical list. To create a hierarchical list, you must select a list template that supports hierarchical lists.

        To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist." For more information on hierarchical list item display, see list template item help.

- Entry, Sequence - Specify the sequence for this component. The sequence determines the order of evaluation.
- Entry, Image/Class - Identify an image file name or the icon class for the icon associated with this list entry. To learn more, see field-level Help.
- Entry, Attributes - Identify the image attributes (such as width="12" height="12") for the list element image. Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string allows for the title image attribute to be automatically set based on the value of the list label text. For example:

  ```
  ...title="#LIST_LABEL#"...
  ```

- Entry, Alt Attribute - Identify the image alt attribute for the list element image. Use the `#LIST_LABEL#` substitution string to reference the list label text. This substitution string allows for the image alt attribute to be automatically set based on the value of the list label text. For example:

  ```
  ...alt="#LIST_LABEL#"...
  ```

- Entry, List Entry Label - Identify the text of the link. The display of this link text is controlled by list templates.

b. Target, Target Type - Select a target type. Options include:

- **Page in this Application**
- **URL**

The **Target Type** you select determines what attributes are editable. To learn more about an attribute, see field-level Help.

c. Subentries, List - Use this attribute when creating a hierarchical list. Identify the list populating subentries for this list entry.

To create a hierarchical list, you must select a list template that supports hierarchical lists. To determine which list templates support hierarchical lists, look for templates having the naming convention "with Sublist." For more information on hierarchical list item display, see list template item help.

d. Current List Entry, List Entry Current for Pages Type - Specify when this list entry should be current based on the page type. List items can be current or non-current. Current list items use the current template, non current list items use the non current list item template. The actual condition and templates are defined in subsequent attributes.

e. Conditions, Condition Type - Select a condition type from the list that must be met in order for this component to be rendered processed.

f. Authorization, Authorization Scheme - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.

g. Configuration, Build Option - Select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.

h. Click Counting:

- Click Counting, Count Clicks - If this is a link to an external page (for example www.google.com) you can count clicks.
- Click Counting, Click Count Category - A category to classify the click.

See COUNT_CLICK Procedure in *Oracle APEX API Reference*.

i. User Defined Attributes - Specify the theme and list template and define additional attributes. Click **Apply Changes** to save your edits. To learn more about an attribute, see field-level Help.

5. Subscription:
   - Subscription, Reference Master List From - Select a master list to subscribe from. When you select a master list and click **Apply Changes**, the list details are automatically refreshed from the master list.
   - Subscription, Refresh List - If the list is already subscribed, the **Refresh List** button appears. Click **Refresh List** to refresh list details from the master list.
6. Configuration, Build Option - From Build Option select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.
7. Comments, Comments - Enter any comments or notes for developers.
8. Click **Apply Changes**.

16.8.4.2 Editing a Dynamic List

Edit a dynamic list by selecting it on the Lists page in Shared Components.

To edit a dynamic list:

1. Navigate to the Lists page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears. By default, the Lists page displays as a table and the Type column identifies the list type as **Dynamic** or **Static**.
2. Select a dynamic list.

   The List Details page appears.
3. Edit the list attributes.
4. Name, Name - Defines the name for the list. Both numeric or alphanumeric list names are supported.
5. List Query:
   a. List Query, Query Source Type - Select the query source type. Options include **SQL Query** or **Function Returning a SQL Query**.
   b. List Query, Query - Enter a list SQL Query or Function Returning a SQL Query to generate a dynamic list. To learn more and view examples, see field-level Help.
6. Subscription:
   - Subscription, Reference Master List From - Select a master list to subscribe from. When you select a master list and click **Apply Changes**, the list details are automatically refreshed from the master list.
   - Subscription, Refresh List - If the list is already subscribed, the **Refresh List** button appears. Click **Refresh List** to refresh list details from the master list.

7. Configuration, Build Option - From Build Option select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.
8. Comments, Comments - Enter any comments or notes for developers.
9. Click **Apply Changes**.

16.8.5 Adding a List to a Page

Once you create a list and list entries, the next step is to add it to a page by creating a list region.

To add a list to a page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create a new list region:
   a. In the Gallery at the bottom of the central pane, click **Regions** and locate **List**.
   b. Right-click **List** and from the context menu, select **Add To** and then the location.

   > **Tip:**
   >
   > In addition using the context menus, you can also use your mouse to drag and drop components to the appropriate location.
3. In the Property Editor, edit the appropriate attributes:

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. Identification:
   a. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.
   b. Identification, Type - Select **List**.
5. Source, List - Select the source for this list.
6. Layout:
   a. Layout, Sequence - Enter the display sequence for this item. The sequence and other layout settings determine where this item is displayed in relation to other items within the region.
   b. Layout, Parent Region - Select the parent region to which this region belongs. If a parent region is selected then this region is rendered completely inside the parent region..

   c. Layout, Position - Identify a display point for this region.

7. Appearance, Template - Choose a template to control the look of the region.
8. Server-side Condition, Type - Optionally select a condition type from the list that must be met in order for this component to be rendered or processed. Additional attributes appear based on your selection.
9. Security, Authorization Scheme - Optionally select an authorization scheme to control this component. The authorization must evaluate to TRUE in order for this component to be rendered or executed.
10. Click **Save**.

Repeat these procedures for each page where you would like to add a list.

> **See Also:**
>
> - Understanding Conditional Rendering and Processing
> - Providing Security Through Authorization
> - Creating a New Template
> - List Templates for information about altering list display

16.8.6 Deleting a List

Delete a list on the List Details page.

Deleting a list also deletes any corresponding list regions.

To delete a list:

1. Navigate to the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.
2. On the Lists page, select a list to view.

   The List Details page appears.
3. On the List Details page, click **Delete**.

16.8.7 Copying or Subscribing to a List

Copy a list from the current application or from another application. When copying a list from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a list:

1. Navigate to the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. From the Tasks list, click **Copy from another app**.

   > **Tip:**
   >
   > You can also access the Copy List Wizard on the Lists page by clicking **Copy** in the Copy column. Alternatively, **Copy List** also appears on Tasks list on the List Details page.

   The Copy List Wizard appears.
3. Copy From - Choose where to copy the list from. Options include:.
   - **This Application**
   - **Other Application**
4. To copy a list from the current application:
   a. Copy From - Select **This Application**.
   b. Copy List - Select the list to copy.
   c. New List Name - Enter the name of the list you would like to create.
   d. Click **Copy**.
5. To copy a list from another application:
   a. Copy From - Select **Other Application**.
   b. Application - Select the application from which you want to copy.
   c. Copy List - Select the list to copy.

d. Subscribe - Enable **Subscribe** to subscribe to the list.

e. Click **Copy**.

16.8.8 Subscribing to a List

Subscribe to a list on the List Details page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a list by copying it and enabling the Subscribe option. See Copying or Subscribing to a List.

To subscribe to a list in the current application:

1. Navigate to the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The List Details page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select a list.

   The List Details page appears.
3. To subscribe to a list:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master list to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a list:
   a. Find the **Subscription** region.
   b. Subscription, Refresh List - If the list is already subscribed, click **Refresh List**.
   c. Click **Apply Changes**.
5. To unsubscribe from a list:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.8.9 Accessing List Reports

View reports on the Lists by clicking the following tabs: Unused, Conditional Entries, Utilization, and History. View List Details to access a Grid Edit of all list entries.

Note that these reports only display after you create a list.

To view list reports:

1. To view reports available on the Lists page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Lists**.

      The Lists page appears.
   f. On the Lists page, click the appropriate tab:
      - **Unused** - Click the **Unused** tab to identify lists that are not used in the current application.
      - **Conditional Entries** - Click the **Conditional Entries** tab to view conditional lists.
      - **Subscription** - Click the **Subscription** tab to view subscribed Lists in your application. To refresh Lists, select one or more lists and click **Refresh Checked**.
      - **Utilization** - Click the **Utilization** tab to view all lists included in the current application. From the Utilization report, you can:
        - Edit list entries by selecting the list name.
        - View the pages on which the list appears by clicking the number in the Pages column.
      - **History** - Click the **History** tab to view recent modifications made to lists and list entries in this application.
2. To view the Grid Edit List Entries report for a specific list:
   a. On the Lists page, select a list.

      The List Details page appears.
   b. Under Tasks, click **Grid Edit**.

      The Grid Edit List Entries report appears.

16.8.10 Managing Navigation Menus

Applications using newer themes, such as *Universal Theme - 42,* provide navigation with navigation menus.

- About Navigation Menus
  A **navigation menu** is list with hierarchical list entries.
- About Editing Navigation Menu Lists
  Since a Navigation menu is a list, you create and edit in the same way as any other list.

- Changing Where and How Navigation Menus Display
  Change how and where a navigation menu displays by editing Navigation Menu attributes on the User Interface page.
- About Switching from Tabs to Navigation Menus
  You can switch from a theme using tabs to a theme using navigation menus if the existing application used one level of tabs was used.

16.8.10.1 About Navigation Menus

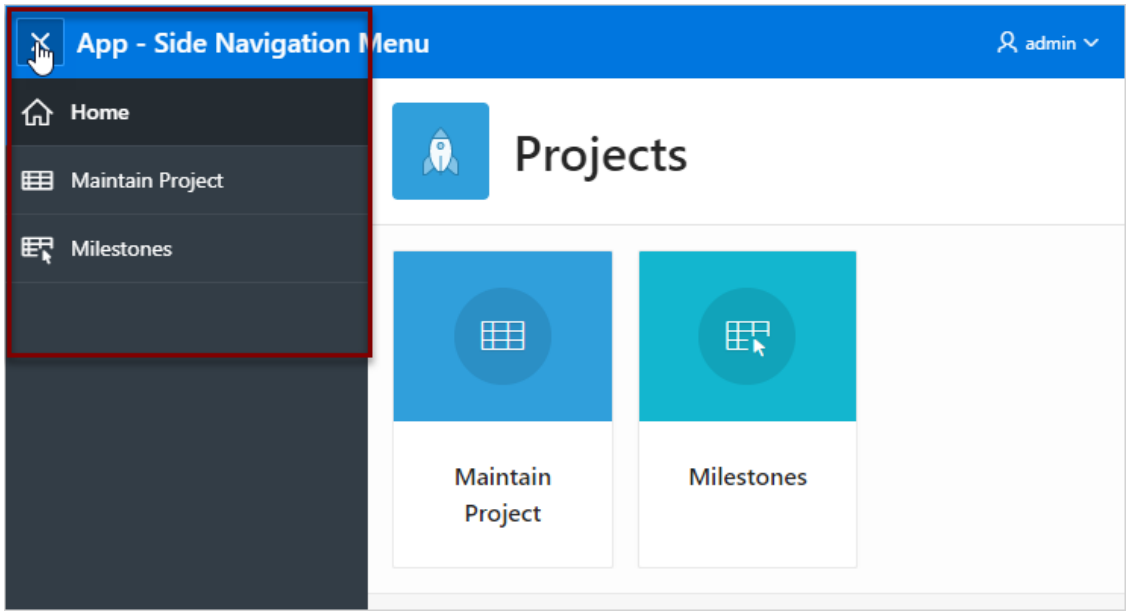
A **navigation menu** is list with hierarchical list entries.

When you create an application, the Create Application Wizard automatically creates a navigation menu for you and populates it with list entries linking to the application pages you have created. Navigation menus are controlled at the theme and user interface level. You can change how and where a navigation menu displays by editing Navigation Menu attributes on the User Interface page. Types of navigation menus include **Side Menu**, **Top Menu**, or **Mega Menu**.

Side Navigation Menu

Users can expand or collapse the **Side Navigation Menu** by clicking on the menu icon from the header. This navigation menu renders the navigation items using a tree component that enables users to expand or collapse sub items. Based on the available space, Side Navigation Menu displays a full menu (as show below), or collapses to a narrow icon bar. You can also configure a navigation menu to be completely hidden (default) or to display a narrow icon bar when collapsed using Template Options.

The following example shows an application with a Side Navigation Menu.



Top Navigation Menu

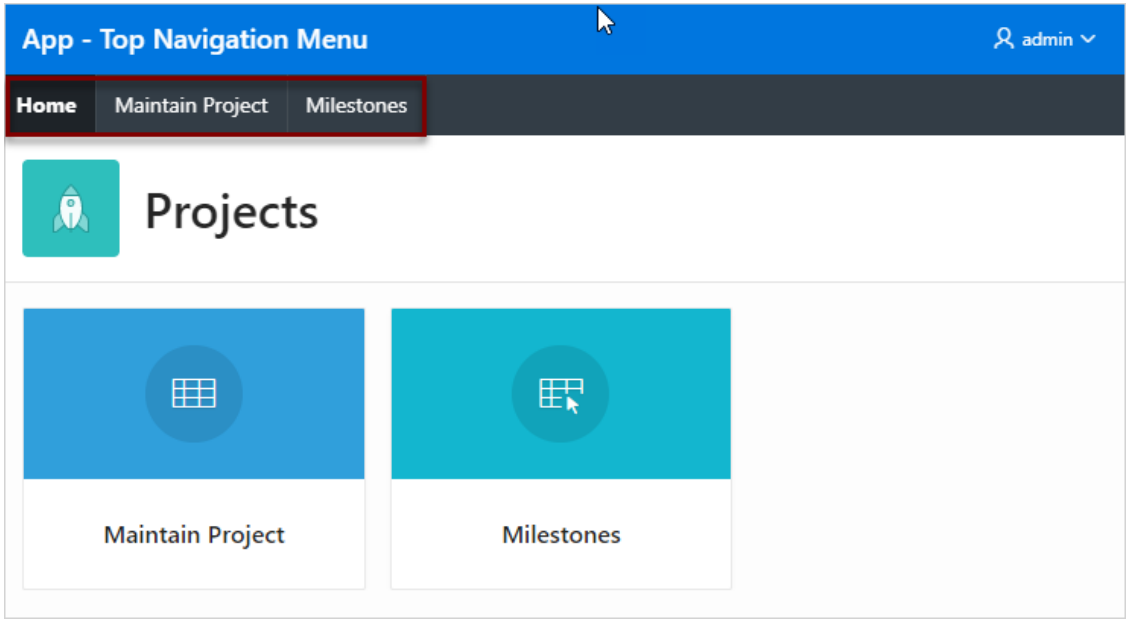
A **Top Navigation Menu** displays at the top of the application. You can control how the menu displays by choosing between two templates:

- **Top Navigation Tabs** - The **Top Navigation Tabs** template renders your application navigation as tabs and is ideally suited for simple applications where you have six or fewer

tabs. This template automatically positions to the bottom of the screen for small screen or mobile devices.

- **Top Navigation Menu** - The **Top Navigation Menu** template renders your application navigation as a menu bar, similar to what you find on most desktop applications. Developers commonly use this option for more complex applications with several layers of hierarchy within the navigation.

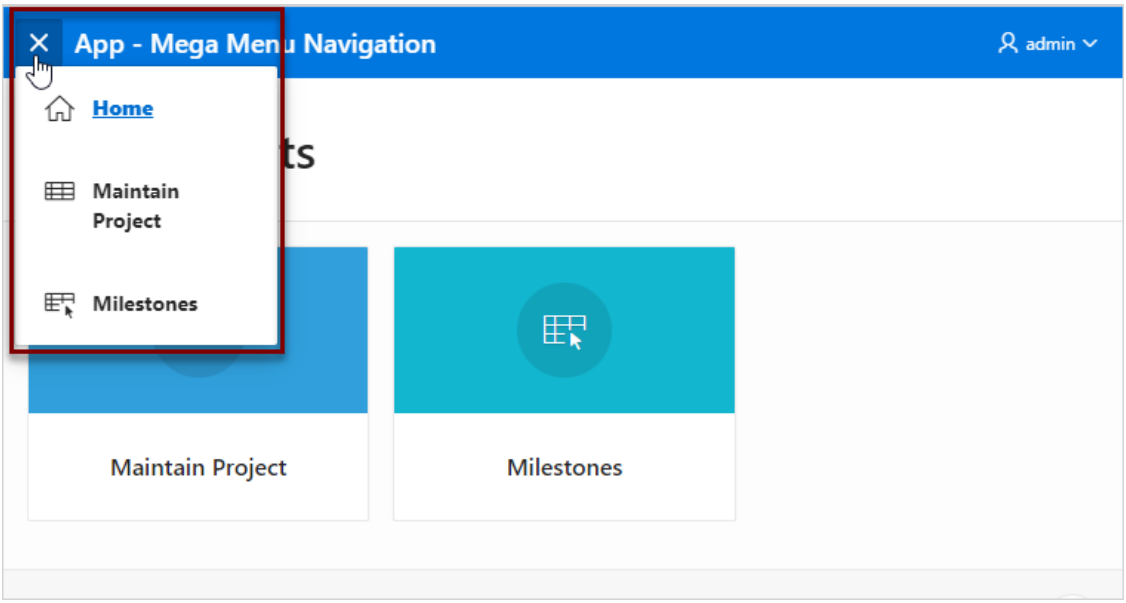
The following example shows an application with a Top Navigation Menu that uses the Top Navigation Menu template.



Mega Menu Navigation

The **Top Navigation Mega Menu** template renders your application navigation in a pop-up panel that can be opened or closed from the header menu button. Users can expand or collapse a Mega Menu by clicking on the menu icon from the header. Mega menus are especially useful when you want to display all navigation items at once to your user.

The following example shows an application with a Mega Menu using the Top Navigation Mega Menu template.



16.8.10.2 About Editing Navigation Menu Lists

Since a Navigation menu is a list, you create and edit in the same way as any other list.

You can access Navigation menus in Shared Components in the Navigation region in two ways:

- Select the application and then click **Shared Components**. Under Navigation and Search, select **Navigation Menu**.
- Select the application and then click **Shared Components**. Under the Navigation region, select **Lists**.

Once you select the Navigation Menu, you edit it as you would any other list.

> **See Also:**
>
> Creating Lists

16.8.10.3 Changing Where and How Navigation Menus Display

Change how and where a navigation menu displays by editing Navigation Menu attributes on the User Interface page.

Navigation menus are controlled at the theme and user interface level.

> **Tip:**
>
> You can access the User Interface page from the Application Definition as described in the following procedure, or by navigating to **Shared Components**, **User Interface** and selecting **User Interface Attributes**.

To edit the Navigation menu attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.

   The Application home page appears.
3. Click the **Edit Application Definition** button.

   The Edit Application Definition page appears.
4. Click the **User Interface** tab.
5. Edit the attributes under **Navigation Menu**:
   - **Display Navigation** - Turn navigation on or off by selecting **On** or **Off**.
   - **Navigation Menu List** - Select the list utilized for the navigation menu for the application.
   - **Position** - Select the position where you would like your navigation menu to be placed in this application.
     - **Side** - Setting the position to **Side** renders the navigation menu list template in the `#SIDE_GLOBAL_NAVIGATION_LIST#` position on your page template.
     - **Top** - Setting the position to **Top** renders the navigation menu list template in the `#TOP_GLOBAL_NAVIGATION_LIST#` position on your page template.

       For example, if you choose **Side**, your navigation could be rendered as a tree on the left side of your pages. In contrast, choosing **Top** can render your navigation as a menu bar in the header of your pages.
   - **List Template** - Select the List Template used to render the navigation menu for the application.
   - **Template Options** - Set Template Options for the List Template used for the navigation menu list for the application.
6. Click **Apply Changes** to save your changes.
7. Run the application to view your changes.

See Also:

- Editing User Interface Attributes
- List Templates
- Understanding Template Options

16.8.10.4 About Switching from Tabs to Navigation Menus

You can switch from a theme using tabs to a theme using navigation menus if the existing application used one level of tabs was used.

If an existing applications uses two levels of tabs, you must convert it to one level before changing the theme.

> **See Also:**
>
> Editing Themes

16.8.11 Managing Navigation Bar Lists

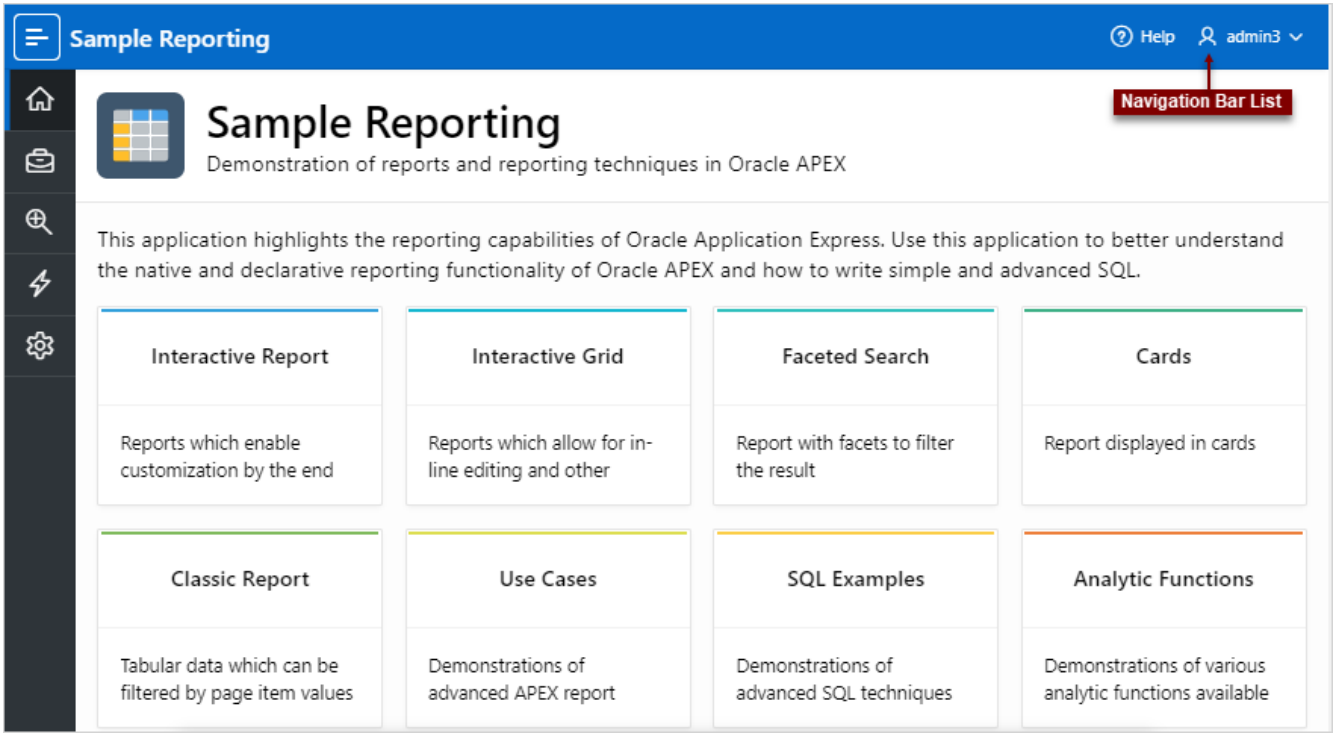
Applications using newer themes, such as *Universal Theme - 42,* include navigation bar lists.

- About Navigation Bars
  A navigation bar list displays with a list template in the `#NAVIGATION_BAR#` position on your page template.
- Editing Navigation Bar Lists
  Since a navigation bar is a list, you create and edit in the same way as any other list.
- Editing Navigation Bar Attributes
  Change how navigation bar displays by editing Navigation Menu attributes on the User Interface page.

16.8.11.1 About Navigation Bars

A navigation bar list displays with a list template in the `#NAVIGATION_BAR#` position on your page template.

The example below shows the *Sample Reporting* Sample App with a navigation bar displaying in the upper right corner.



16.8.11.2 Editing Navigation Bar Lists

Since a navigation bar is a list, you create and edit in the same way as any other list.

You can access navigation bars in Shared Components in the Navigation region in two ways:

- Select the application and then click **Shared Components**.
- Under Navigation and Search, select one of the following:
  - **Navigation Bar List**
  - **Lists**

Once you select the navigation bar list, you edit it as you would any other list.

> **See Also:**
>
> Creating Lists

16.8.11.3 Editing Navigation Bar Attributes

Change how navigation bar displays by editing Navigation Menu attributes on the User Interface page.

Selecting the Implementation **Classic** uses tabs instead of a list.

To edit the navigation bar attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.

   The Application home page appears.
3. Access the User Interfaces page:
   a. Click **Shared Components**.
   b. Under User Interface, click **User Interface Attributes**.

   The User Interfaces page appears.
4. Under **Navigation Bar**, edit the following attributes:
   - **Implementation** - Select how to you wish to implement the navigation bar in this application. Options include:
     - **List** renders the navigation bar as a list, using the selected list and list template in the `#NAVIGATION_BAR#` position on your page template.
     - **Classic** renders the navigation bar as a classic navigation bar in the `#NAVIGATION_BAR#` position on your page template.
   - **Navigation Bar List** - Select the list utilized for the navigation bar on this application.
   - **List Template** - Select the List Template used to render the navigation menu for the application.
   - **Template Options** - Set Template Options for the List Template used for the navigation menu list for the application.
5. Click **Apply Changes** to save your changes.

6. Run the application again to view your changes.

> **Note:**
>
> Editing User Interface Attributes

16.9 Adding Search to an Application

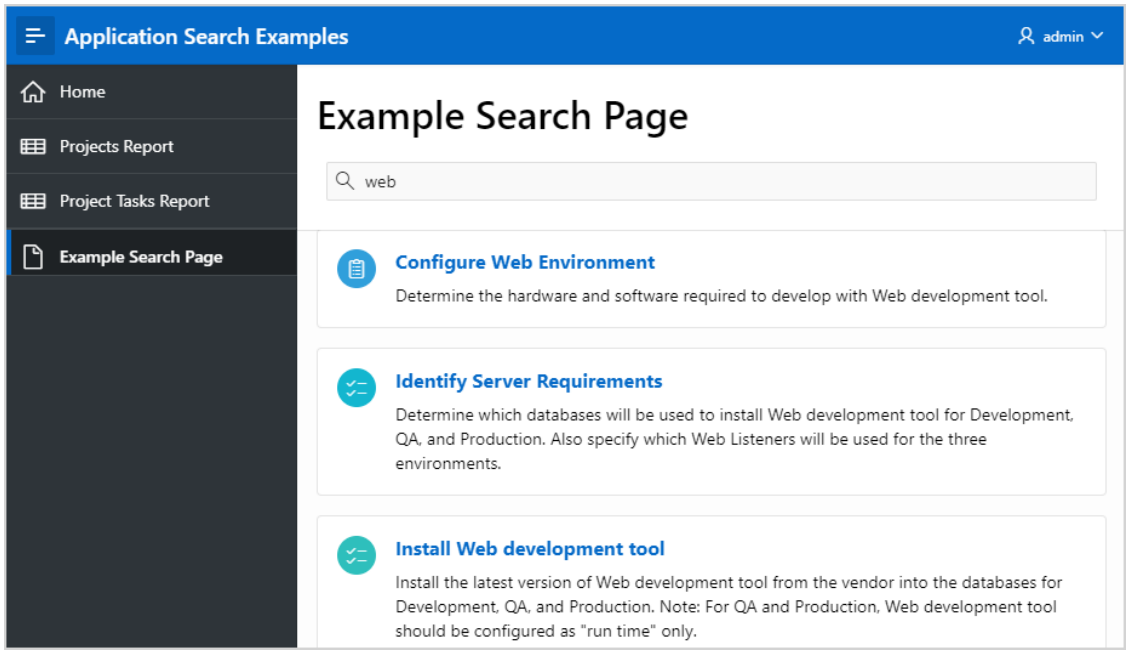
Add declarative searches to an application by creating a search configuration and then a search page that displays the search results.

- Example Searches
  View search examples.
- About Creating Application Searches
  Create a declarative search for an application by creating a search configuration on a data source and then creating a page that displays the search results.
- Creating a Standard Search
  Learn how to create a Standard search.
- Creating an Oracle Text Search
  Create an Oracle TEXT search on table that has an Oracle TEXT index.
- Creating a List Search
  Create a List search of an APEX List.
- Creating a Ubiquitous Search
  Search across multiple objects within an entire schema by creating a ubiquitous database search.
- Creating an Oracle Vector Search
  Create an Oracle Vector Search to search both structured and unstructured data by semantics or meaning, and by values, enabling ultra-sophisticated AI search applications.
- Creating a Search Page Manually
  Learn how to create a Search page manually in Page Designer.
- Editing a Search Configuration
  Edit an existing search configuration in Shared Components on the Search Configurations, Edit page.
- Copying or Subscribing to a Search Configuration
  Copy a search configuration from the current application or from another applications. When copying a shortcut from another app, you can also subscribe to it.
- Subscribing to a Search Configuration
  Subscribe to a search configuration on the Search Configurations, Edit page.
- Editing a Search Page
  Learn how attributes can change search page behavior
- Accessing Search Configuration Reports
  View the search configurations Utilization and History reports by clicking the appropriate tab on the Search Configurations page.

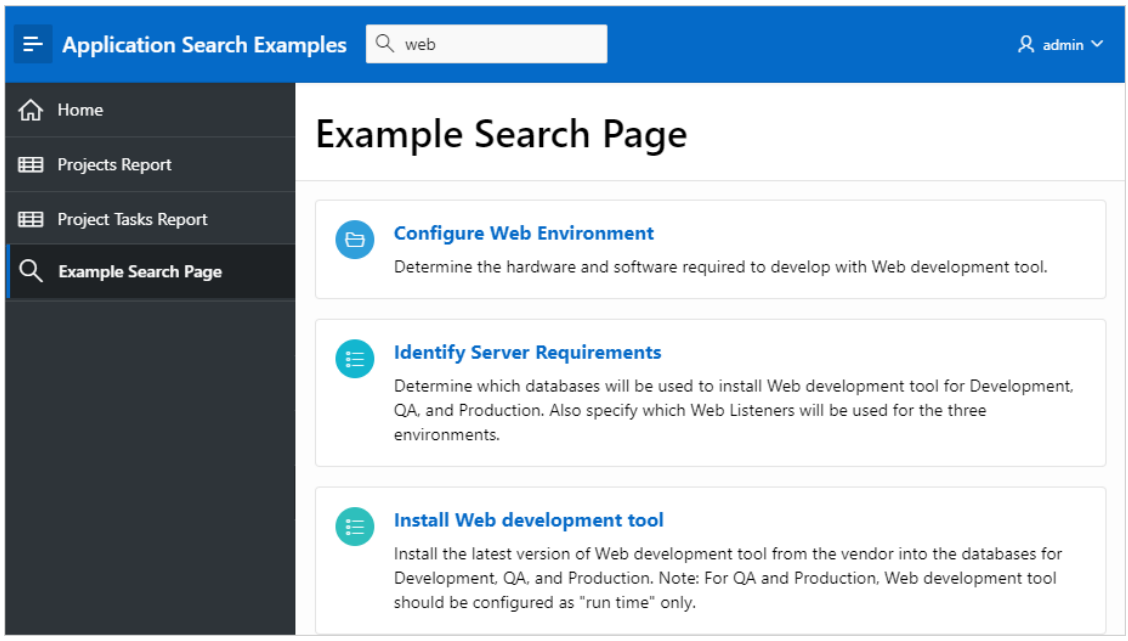
16.9.1 Example Searches

View search examples.

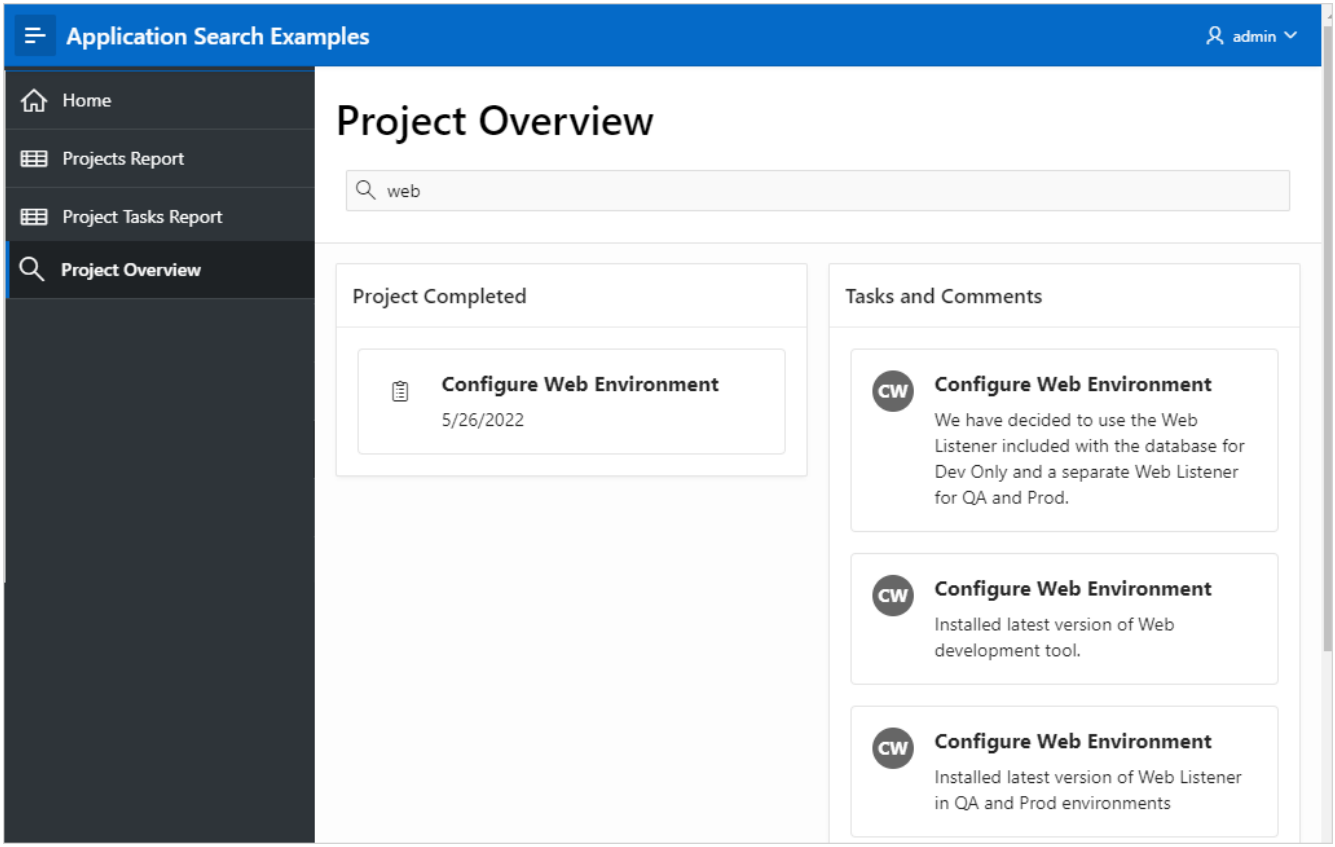
The following example shows a sample search page created with the Create Page Wizard. Notice a search field displays after the page title and is followed by a Search Results region. In Page Designer, the search field is positioned in the Breadcrumb region and the Search Results region displays in the Body Position. This example has two search configurations that map to two separate tables, `EBA_PROJECTS` and `EBA_PROJECT_TASKS`.



The next example is the same page but with the search field is in a different position. Notice that the search field displays in the Position, After Logo, which is between the logo and the navigation bar. As in the previous example, the Search Results region displays in the Body of the page. However, the icons are different.



You can also create a search components by adding them manually to a page in Page Designer. The following example shows a search page named Project Overview. Note that the search field displays after the page title and contains the search term, `web`. The page includes two search result regions: **Projects Completed** and **Tasks and Comments** which use the Standard Template so the region Title appears.



> **See Also:**
>
> Creating a Search Page Manually

16.9.2 About Creating Application Searches

Create a declarative search for an application by creating a search configuration on a data source and then creating a page that displays the search results.

Adding search to an application involves two basic steps:

1. Create a search configuration in Shared Components.

   A **search configuration** contains information about searchable data source. Supported search types include:

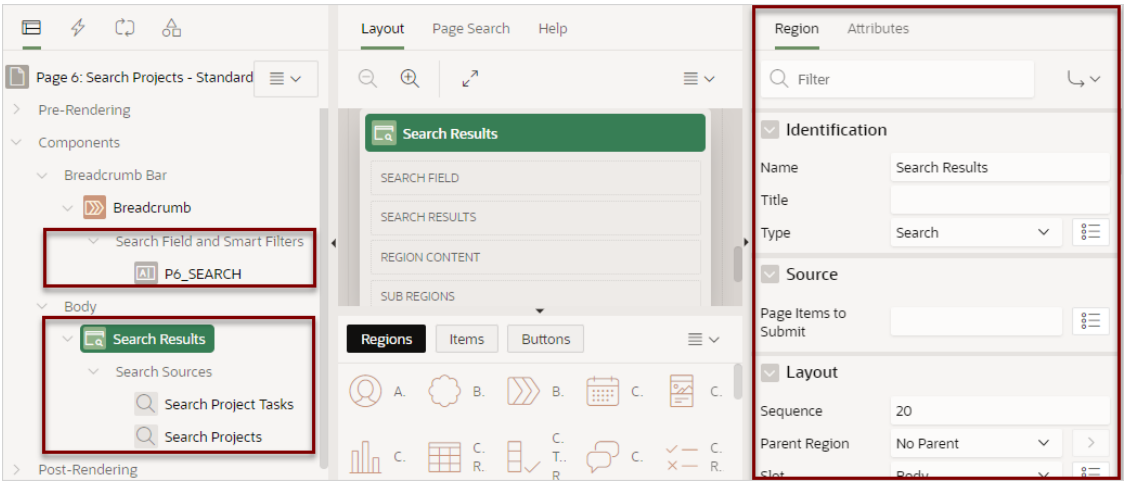
   - **Standard** - Defines a searchable data source such as a table, SQL Query, or REST Data Source. Searches is executed by a SQL Query using `LIKE` expressions.

- **Oracle TEXT** - Define a searchable table or SQL Query which already has an Oracle TEXT index created. Search is executed by using Oracle TEXT functionality, which provides linguistic and fuzzy search capabilities.
- **Oracle Ubiquitous Search** - Search using an Oracle ubiquitous search index (requires Oracle Database 23ai or later). The index must be located in the parsing schema of the application.
- **List** - Search within a selected List from Shared Components.

2. Create a Search page.

   You can create a search page by running the Create Page Wizard and selecting **Search Page**, or by adding search components manually. A Search page features a search field (or page item) which enables users to submit search terms and a Search Results region. The Search Results region contains Search Sources which map to search configurations.

   The following example shows a search page in Page Designer. In the Rendering tab, notice the page item, **P6_SEARCH** and the **Search Results** region. **P6_SEARCH** is the search field which enables users to submit search terms. The Search Results region contains both the search results and two **Search Sources** which map to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.



See Also:

- Creating a Search Page Manually
- Editing a Search Page

16.9.3 Creating a Standard Search

Learn how to create a Standard search.

You can create Standard search on a searchable data source which can be a table, a SQL query, or REST Data Source.

This topic demonstrates how to create a search page on two different tables available in the sample dataset, Project Data. This example assumes you have already created an application and two interactive reports built on `EBA_PROJECTS` and `EBA_PROJECT_TASKS` tables.

- Creating Standard Search Configurations
  Create two Standard search configurations based on two separate tables.
- Editing Search Configurations to Include Links
  Learn how to edit search configurations to include links to existing pages.
- Creating a Standard Search Page
  Create a Search page by running the Create Page Wizard.

> **See Also:**
>
> Using Sample Datasets in *Oracle APEX SQL Workshop Guide*

16.9.3.1 Creating Standard Search Configurations

Create two Standard search configurations based on two separate tables.

A **search configuration** contains information about a searchable data source. In this task, you create two search configurations and then a search page using the Create Page Wizard. The first search configuration maps to the `EBA_PROJECTS` table. The second search configuration maps to the `EBA_PROJECT_TASKS` table. Both tables are available in the Project Data sample dataset.

> **Note:**
>
> This task assumes you have already created an application and two interactive reports built on `EBA_PROJECTS` and `EBA_PROJECT_TASKS` tables available in the sample dataset, Project Data.

To create two Standard search configurations:

1. Create an application.
2. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

      The Search Configurations page appears.

Create the first search configuration that maps to the `EBA_PROJECTS` table:

3. On the Search Configurations page, click **Create**.

   The Create Search Configuration Wizard appears.
4. Detail:
   a. Name - Enter a name for the search configuration (for example, `Search Projects`).
   b. Search Type - Select a type of search. Options include:
      - **Standard** - Defines a searchable data source such as a table, SQL Query, or REST Data Source. Searches is executed by a SQL Query using `LIKE` expressions.

- **Oracle TEXT** - Define a searchable table or SQL Query which already has an Oracle TEXT index created. Search is executed by using Oracle TEXT functionality, which provides linguistic and fuzzy search capabilities.
- **Oracle Ubiquitous Search** - Search using an Oracle ubiquitous search index (requires Oracle Database 23ai or later). The index must be located in the parsing schema of the application.
- **List** - Search within a selected List from Shared Components.

For this example, select **Standard**.

c. Click **Next**.

5. Source:

a. Data Source - Select the Data Source for the search configuration. Options include:

- **Local Database** - Data is sourced from a local database.
- **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
- **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
- **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
- **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

The UI changes depending upon the Data Source you select. For this example, select **Local Database**.

b. Source Type - Specify the source type. UI changes depending on selected Source Type. For this example, select **Table**.

c. Table / View Owner - Select the owner of the table on which you are building the search configuration.

d. Table / View Name - Select the table or view on which the search configuration is based (for example, select `EBA_PROJECTS)`.

e. Click **Next**.

6. Column Mapping:

a. Primary Key Column - Select the primary key column for the search configuration data source (for example, **ID**).

b. Title Column - Select the column to be used as title (for example, **NAME**).

c. Description Column - Select the column to be used as the description (for example, **DESCRIPTION**).

d. Icon Source - Configure the icon that displays in the header of the search result. Options include:

- **Initials** - Use the Title column to display the icon as initials.
- **CSS Classes** - Use the entered CSS classes to display the icon.

- **Class Column** - Use the selected column containing the icon CSS class to display the icon.

For this example, accept the default, **Initials**.

7. Click **Create Search Configuration**.

   The search configuration is created and the Search Configuration, Edit page appears and displays the message `Search configuration created`.

Create a second search configuration that maps to the `EBA_PROJECT_TASKS` table:

8. Click the **Search Configurations** breadcrumb to return to the Search Configurations page.
9. Click **Create** on the Search Configurations page.

   The Create Search Configuration Wizard appears.

10. Detail:
    a. Name - Enter a name for the search configuration (for example, `Search Project Tasks`). This name displays in the search result.
    b. Search Type - Select a type of search.

       Since this example is built on a table, select **Standard**.
    c. Click **Next**.
11. Source:
    a. Data Source - Select a Data Source for the search configuration.

       The UI changes depending upon the Data Source you select. For this example, select **Local Database**.
    b. Source Type - Specify the source type. UI changes depending on selected Source Type. For this example, select **Table**.
    c. Table / View Owner - Select the owner of the table on which you are building a search configuration.
    d. Table / View Name - Select the table or view on which the search configuration is based. For this example, select the `EBA_PROJECT_TASKS` table.
    e. Click **Next**.
12. Column Mapping:
    a. Primary Key Column - Select the primary key column for the search configuration Data Source (for example, **ID**).
    b. Title Column - Select the column to be used as title (for example, **NAME**).
    c. Description Column - Select the column to be used as the description (for example, **DESCRIPTION**).
    d. Icon Source - Select the type for the icon that is displayed in the header of the search result.

       For this example, accept the default, **Initials**.
13. Click **Create Search Configuration**.

    The search configuration is created and the Search Configuration, Edit page appears.
14. Click the **Search Configurations** breadcrumb to return to the Search Configurations page.

16.9.3.2 Editing Search Configurations to Include Links

Learn how to edit search configurations to include links to existing pages.

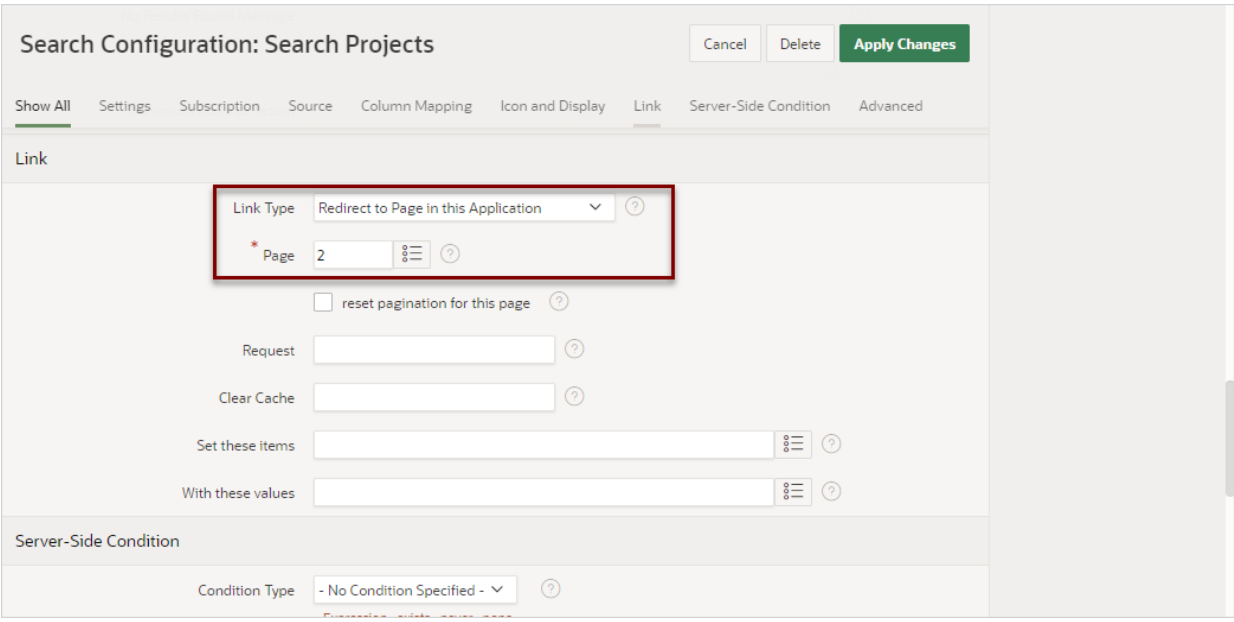
> **Note:**
>
> This example assumes you have previously created the search configurations described in Creating Standard Search Configurations.

To edit search configurations to include links:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configuration**.

      The Search Configurations page appears.
2. Edit the Search Projects search configuration to link to a report on another page:
   a. Select the search configuration (for example, **Search Projects**).
   b. Link - Select the type of link. The UI changes depending upon the **Link Type** you select.

      For this example, select **Redirect to Page in this Application**.
   c. Page - Select or enter the page to redirect to (for example, **2**).



   d. Configure the remaining attributes as needed. For example, you can use **Set these items** in conjunction with **With these values** to pass values from the search page to the target page.

- reset pagination for this page - Select this option to reset the pagination that is linked to.
- Request - Enter text that you would like to set the built-in application item `REQUEST`. This is also the item that is set with the name of a button that was clicked.
- Clear Cache - Enter the page numbers, separated by commas, for which you would like the user's session state to be cleared (for example: `10,11,12`). You can also reset a collection by passing including a collection name.
- Set these items - Enter the list of application item names, separated by a comma, for which you would like to set the session state.
- With these values - Enter a comma delimited list of values for the items specified in Set these items. You can specify static values or substitution syntax. Consider the following examples:
  - `&APP_ITEM_NAME.`
  - `&COL_NAME.`
  - `&COL1_NAME.,&COL2_NAME.`

  Item values passed to `f?p=` in the URL must never contain a colon (:). Item values may also not contain commas unless you enclose the entire value in backslash characters. This is useful when decimal values must be passed where the NLS decimal separator character is a comma. For example:

  ```
  \1234,56\
  ```

e. Click **Apply Changes**.

   The Search Configurations page appears with the message `Search Configuration updated`.

3. Edit the **Search Project Tasks** search configuration to link to a report on another page:

   a. Select the search configuration (for example, **Search Project Tasks**).

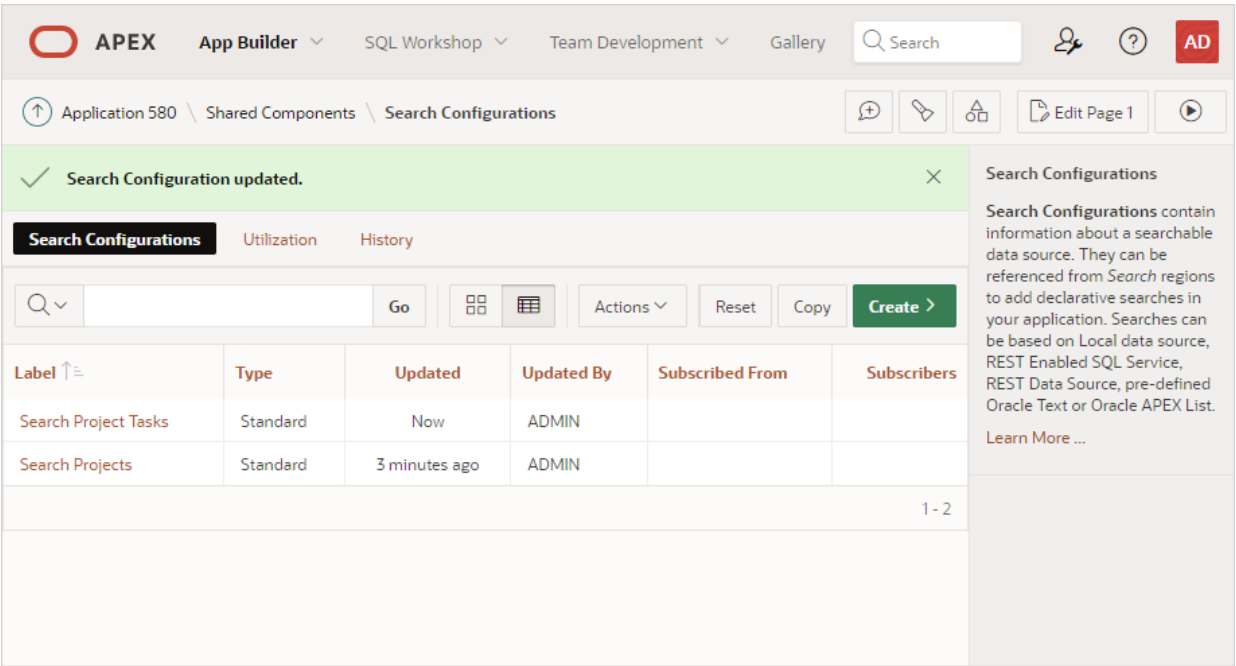
   b. Link - Select the type of link. The UI changes depending upon the **Link Type** you select.

      For this example, select **Redirect to Page in this Application**.

   c. Page - Select or enter the page to redirect to (for example, **4**).

   d. Click **Apply Changes**.

      The Search Configurations page appears with the message `Search Configuration updated`.



4. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.3.3 Creating a Standard Search Page

Create a Search page by running the Create Page Wizard.

Whether you are creating a Standard search or Oracle TEXT search, creating a search page by running the Create Page Wizard involves the same steps. The Create Page Wizard creates a Search field for you and prompts you to select search configurations. Note that you can also add additional search configurations later in Page Designer.

A Search page features a Search field and a Search Results region. In Page Designer, the Search Results region contains **Search Sources** which map to the search configurations defined in Shared Components.

To create a search page by running the Create Page Wizard:

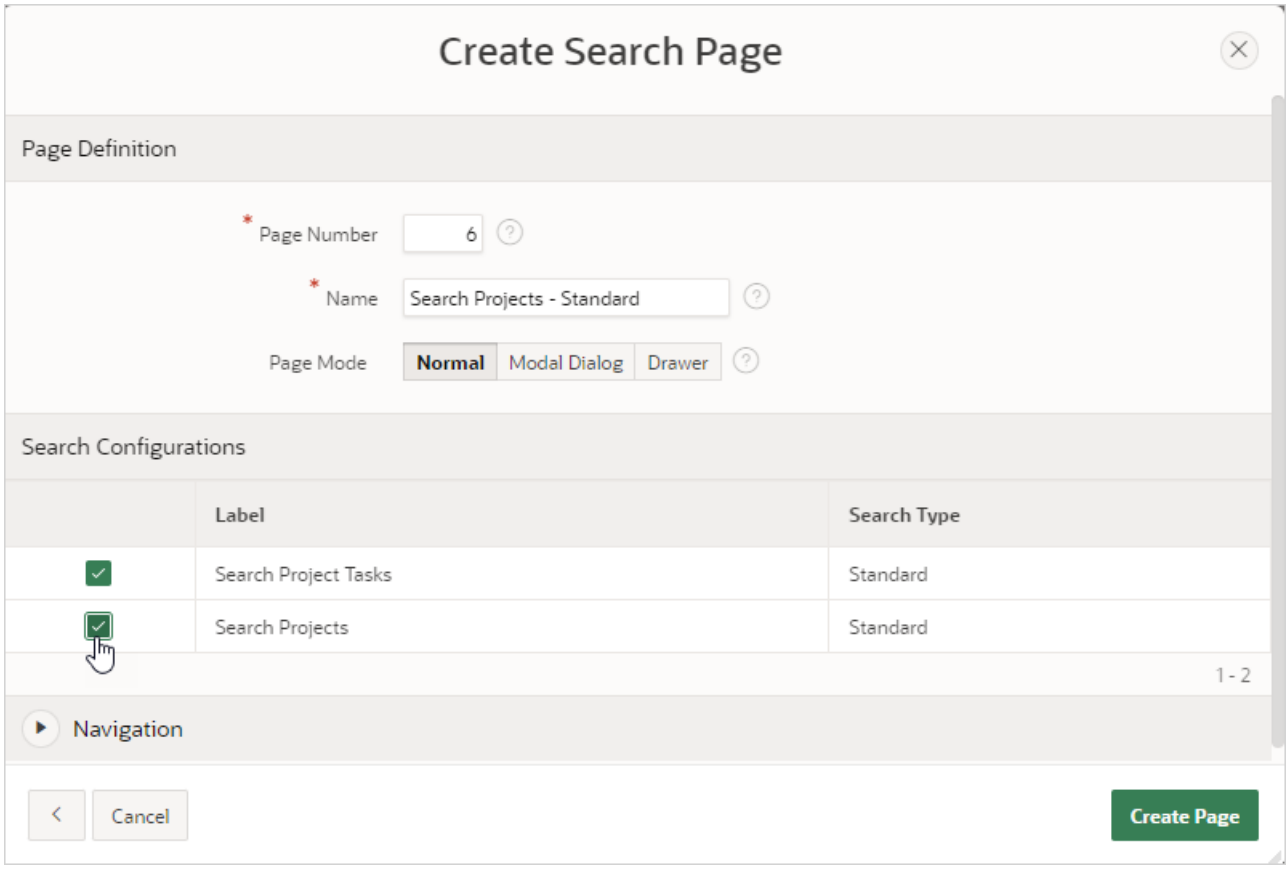
1. Navigate to the Application home page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
2. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
3. Under Component, select **Search Page**.
4. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page (for example, `Search Projects - Standard`).

      This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

c. Page Mode - Identify the page mode. Accept the default, **Normal**.

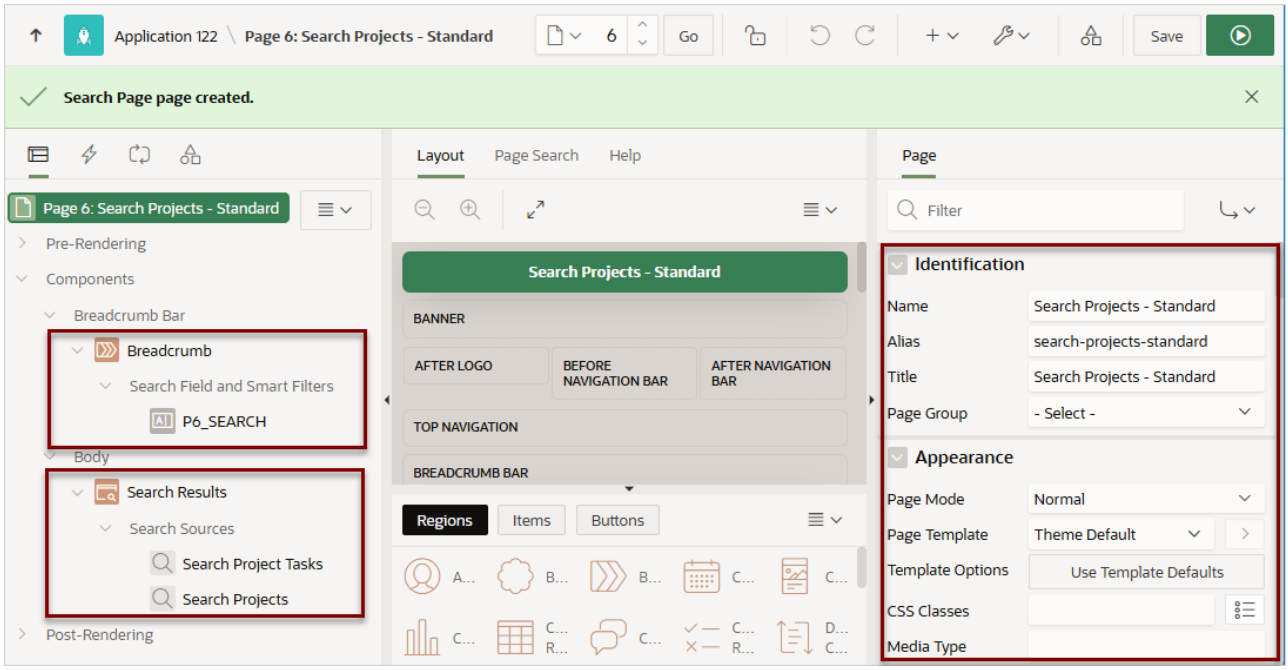
5. Search Configurations - Select the search configurations to be used by this page (for example, **Search Project Tasks** and **Search Projects**).



6. Navigation - Accept the default and enable **Use Breadcrumb** and **Use Navigation**.

7. Click **Create Page**.

   Page Designer appears with the message, `Search Page page created`



In the Rendering tab, notice the page item, **P6_SEARCH** and the **Search Results** region. **P6_SEARCH** is the search field which enables users to submit search terms. The **Search Results** region contains both the search results and two Search Sources which map to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.

Also notice that if you select a search source, Search Source attributes display in the Property Editor.

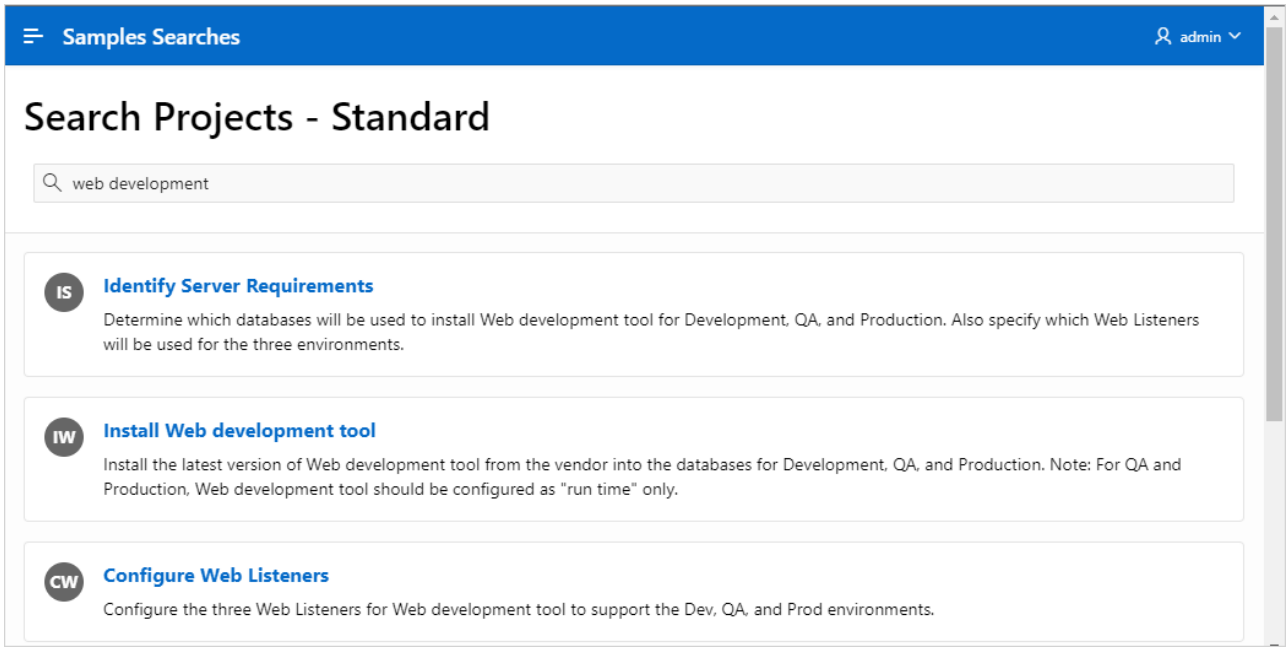
8. Run and test the page:

   a. Click **Save and Run Page**.

      The Search Projects page appears.

   b. In the search field, enter a search (for example, `web development`) and press **Enter**.

   The search results appear.



16.9.4 Creating an Oracle Text Search

Create an Oracle TEXT search on table that has an Oracle TEXT index.

This topic demonstrates how create a search page on a table that has an Oracle TEXT index. In this task, you create a simple Oracle TEXT index, then a search configuration using that Oracle TEXT index, and then a search page using the Create Page Wizard.

> **Note:**
>
> This topic assumes you have already created an application and two interactive reports built on `EBA_PROJECTS` and `EBA_PROJECT_TASKS` tables available in the sample dataset, Project Data.

- Creating an Oracle Text Index
  Learn how to create a simple Oracle Text index on a table.
- Creating an Oracle TEXT Search Configuration
  Create a search configuration on an existing Oracle TEXT index,
- Creating an Oracle TEXT Search Page
  Create a Search page by running the Create Page Wizard.

16.9.4.1 Creating an Oracle Text Index

Learn how to create a simple Oracle Text index on a table.

This topic describes how to create a very simple Oracle Text index on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

> **Tip:**
>
> Oracle Text provides indexing, word and theme searching, and viewing capabilities for text in query applications and document classification applications. To learn more, see *Oracle Text Application Developer's Guide*

To create a simple Oracle Text index on a table:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **SQL Commands**.

   The SQL Commands home page appears.
3. Enter the following SQL command in the command editor:

```
CREATE INDEX eba_project_tasks_ctx ON eba_project_tasks(DESCRIPTION)
INDEXTYPE IS CTXSYS.CONTEXT;
```

4. Click **Run** (**Ctrl+Enter**) to execute the command.

   The results appear in the Results pane.

16.9.4.2 Creating an Oracle TEXT Search Configuration

Create a search configuration on an existing Oracle TEXT index,

A **search configuration** contains information about searchable data source. This task creates a search configuration using an existing Oracle TEXT index created on the `EBA_PROJECT_TASKS` table.

To create an Oracle TEXT search configuration:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

      The Search Configurations page appears.
2. On the Search Configurations page, click **Create**.

   The Create Search Configuration Wizard appears.
3. Detail:
   a. Name - Enter a name for the search configuration (for example, `Projects Tasks - TEXT`).
   b. Search Type - Select a type of search. Options include:
      - **Standard** - Defines a searchable data source such as a table, SQL Query, or REST Data Source. Searches is executed by a SQL Query using `LIKE` expressions.
      - **Oracle TEXT** - Define a searchable table or SQL Query which already has an Oracle TEXT index created. Search is executed by using Oracle TEXT functionality, which provides linguistic and fuzzy search capabilities.

- **Oracle Ubiquitous Search** - Search using an Oracle ubiquitous search index (requires Oracle Database 23ai or later). The index must be located in the parsing schema of the application.
- **List** - Search within a selected List from Shared Components.

For this example, select **Oracle TEXT**.

c. Click **Next**.

4. Source:

   a. Source Type - Specify the source of the search configuration. The Oracle Text search data source can be based on a Table or SQL Query.

      For this example, select **Table**.

   b. Table / View Owner - Select the owner of the table on which you are building a search configuration.

   c. Table / View Name - Select the table or view on which the search configuration is based. For this example, select `EBA_PROJECT_TASKS`.

   d. Click **Next**.

5. Column Mapping:

   a. Primary Key Column - Select the primary key column for the search configuration Data Source (for example, **ID**).

   b. Oracle Text Index Column - Select the Oracle Index enabled column (for example, **DESCRIPTION**).

   c. Title Column - Select the column to be used as title (for example, **NAME**).

   d. Description Column - Select the column to be used as the description (for example, **DESCRIPTION**).

   e. Icon Source - Configure the icon that displays in the header of the search result. Options include:

      - **Initials** - Use the Title column to display the icon as initials.
      - **CSS Classes** - Use the entered CSS classes to display the icon.
      - **Class Column** - Use the selected column containing the icon CSS class to display the icon.

      For this example, accept the default, **Initials**.

6. Click **Create Search Configuration**.

   The search configuration is created. The Search Configuration, Edit page appears and displays the message `Search configuration created`.

7. Edit the search configuration to link to a report:

   a. Link, Link Type - Select the type of link (for example, **Redirect to Page in this Application**).

   b. Page - Select or enter the page to redirect to (for example, **4**).

   c. Click **Apply Changes**.

      The Search Configurations page appears with the message `Search Configuration updated`.

8. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.4.3 Creating an Oracle TEXT Search Page

Create a Search page by running the Create Page Wizard.

Whether you are creating a Standard search or Oracle TEXT search, creating a search page by running the Create Page Wizard involves the same steps. The Create Page Wizard creates a Search field for you and prompts you to select search configurations. Note that you can also add additional search configurations later in Page Designer.

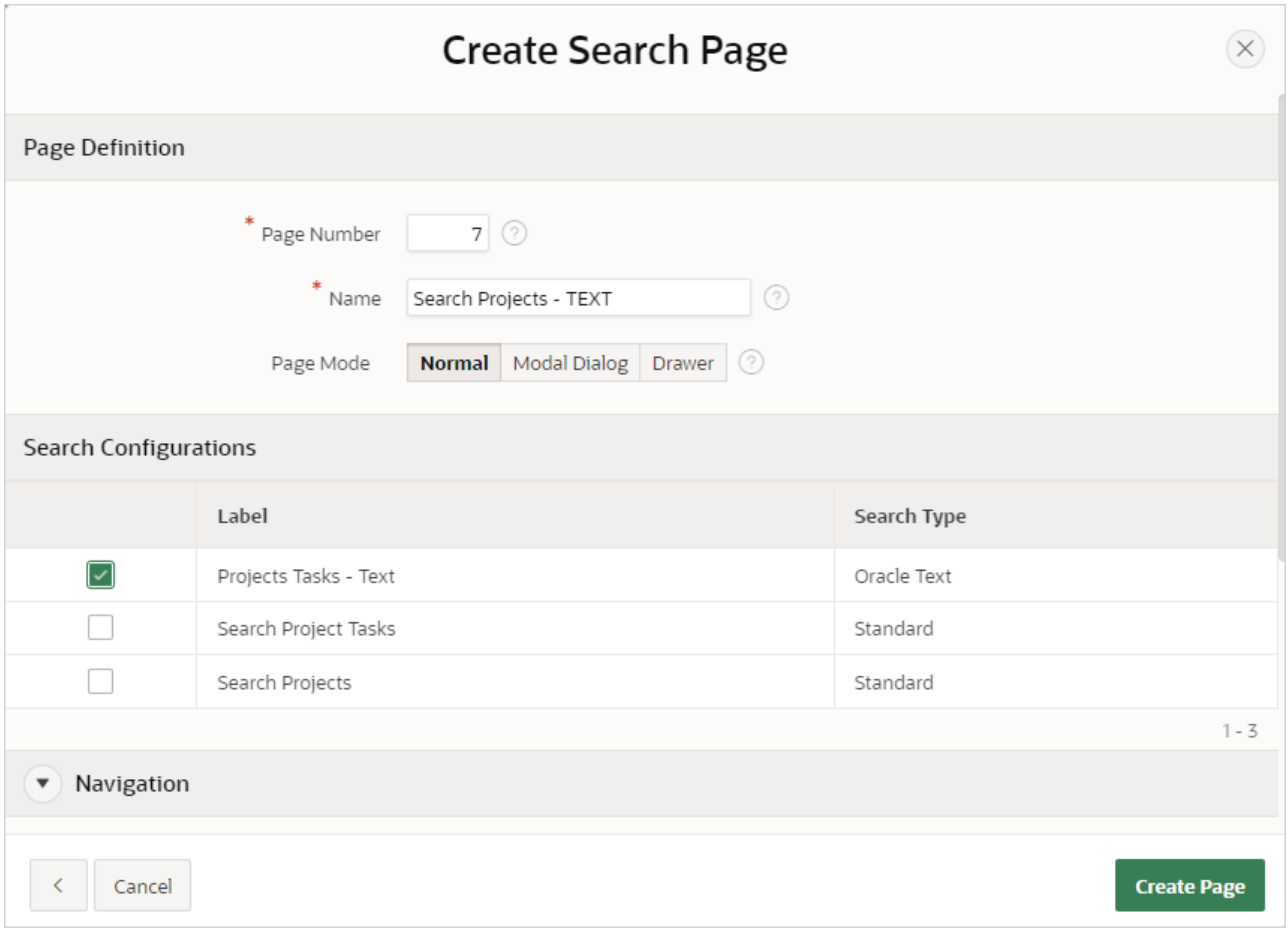
A Search page features a Search field and a Search Results region. In Page Designer, the Search Results region contains **Search Sources** which map to the search configurations defined in Shared Components.

To create a search page by running the Create Page Wizard:

1. Navigate to the Application home page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
2. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
3. Under Component, select **Search Page**.
4. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page (for example, `Search Projects - TEXT`).

      This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.
   c. Page Mode - Identify the page mode. Accept the default, **Normal**.
5. Search Configurations - Select the search configuration to be used by this page (for example, select **Projects Tasks - TEXT**).



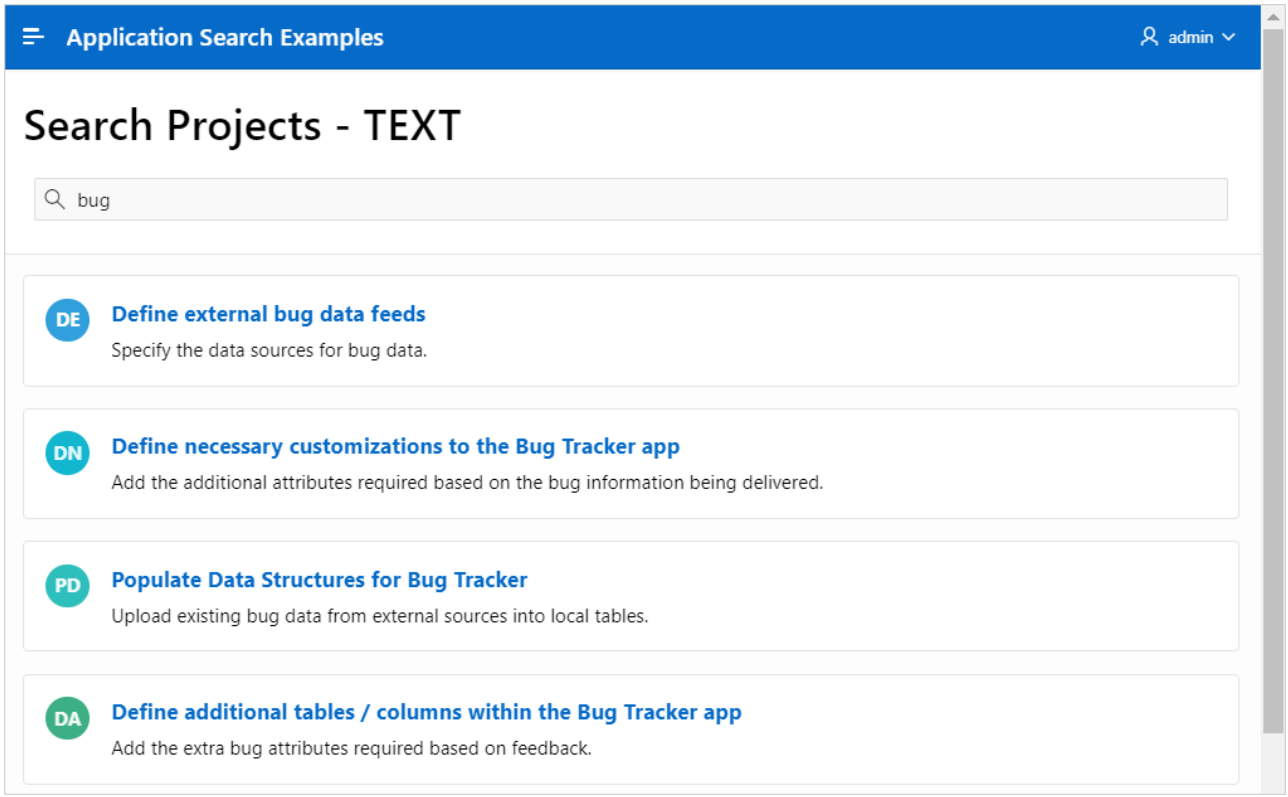
6. Navigation - Accept the default and enable **Use Breadcrumb** and **Use Navigation**.
7. Click **Create Page**.

   Page Designer appears.

   In the Rendering tab, notice the page item, `PX_SEARCH` (where *X* is the page number) and the Search Results region. `PX_SEARCH` is the search field which enables users to submit search terms. The Search Results region contains both the search results and the Search Source which maps to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.
8. Run and test the page:
   a. Click **Save and Run Page**.

      The search page appears.
   b. In the search field, enter a search (for example, `bug`) and press **Enter**.

   The search results appear.



16.9.5 Creating a List Search

Create a List search of an APEX List.

This topic demonstrates how create a search page based on an APEX list of search pages.

> **Note:**
>
> This example assumes you have already created the search pages in Creating a Standard Search Page and Creating an Oracle Text Search.

- Creating an APEX List
  Create an APEX list in Shared Components.
- Creating a List Search Configuration
  Create a List search configuration.
- Creating a List Search Page
  Create a List Search page by running the Create Page Wizard.

16.9.5.1 Creating an APEX List

Create an APEX list in Shared Components.

The following task demonstrates how to create a list on which to base a List search.

> **See Also:**
>
> Creating Static Lists

To create a static list:

1. Access the Create List Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Lists**.
   e. Click **Create**.

   The Create List Wizard appears.
2. For Name and Type:
   a. Name - Enter a numeric or alphanumeric name for the list (for example, `Project Pages List`).
   b. Type - Select **Static**.
   c. Click **Next**.
3. For Query or Static Values - Enter the labels and target pages. For example:
   a. Row 1:
      - List Entry Label - **Search Projects - Standard**
      - Target Page ID or custom URL - **6**
   b. Row 2:
      - List Entry Label - **Search Tasks - TEXT**
      - Target Page ID or custom URL - **7**
   c. Click **Next**.
4. Click **Create List**.
5. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.5.2 Creating a List Search Configuration

Create a List search configuration.

A search configuration contains information about searchable data source. This task creates a search configuration using the list you created in Creating an APEX List.

To create a List search configurations:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

   The Search Configurations page appears.

2. On the Search Configurations page, click **Create**.

   The Create Search Configuration Wizard appears.

3. Detail:

   a. Name - Enter a name for the search configuration (for example, `Search Project Page - List`).

   b. Search Type - Select a type of search. Options include:

      - **Standard** - Defines a searchable data source such as a table, SQL Query, or REST Data Source. Searches is executed by a SQL Query using `LIKE` expressions.
      - **Oracle TEXT** - Define a searchable table or SQL Query which already has an Oracle TEXT index created. Search is executed by using Oracle TEXT functionality, which provides linguistic and fuzzy search capabilities.
      - **Oracle Ubiquitous Search** - Search using an Oracle ubiquitous search index (requires Oracle Database 23ai or later). The index must be located in the parsing schema of the application.
      - **List** - Search within a selected List from Shared Components.

      For this example, select **List**.

   c. List - Select this list you previously created (for example, **Project Pages List**).

   d. Click **Next**.

4. Column Mapping:

   a. Primary Key Column - Select the primary key column for the search configuration Data Source (for example, **LIST_ENTRY_ID**).

   b. Title Column - Select the column to be used as title (for example, **ENTRY_TARGET**).

   c. Description Column - Select the column to be used as the description (for example, **ENTRY_TEXT**).

   d. Icon Source - Configure the icon that displays in the header of the search result. Options include:

      - **Initials** - Use the Title column to display the icon as initials.
      - **CSS Classes** - Use the entered CSS classes to display the icon.
      - **Class Column**

      For this example, accept the default, **Initials**.

5. Click **Create Search Configuration**.

   The search configuration is created and the Search Configuration, Edit page appears and displays the message `Search configuration created`.

6. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.5.3 Creating a List Search Page

Create a List Search page by running the Create Page Wizard.

A Search page features a Search field and a Search Results region. In Page Designer, the Search Results region contains **Search Sources** which map to the search configurations defined in Shared Components.

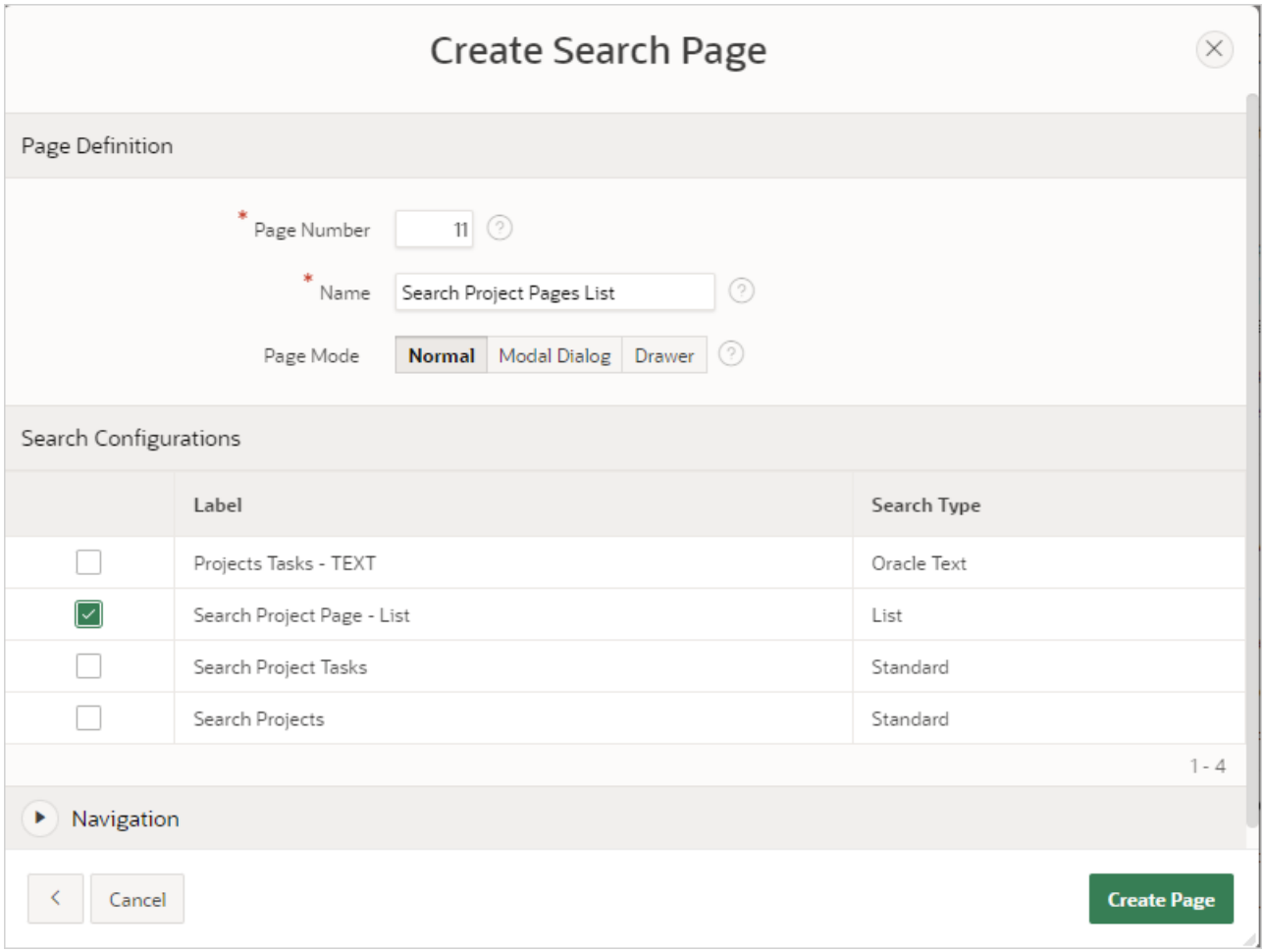
When you create a search page with the Create Page Wizard, the wizard creates a Search field for you and prompts you to select search configurations. You can also add additional search configurations later in Page Designer.

To create a search page by running the Create Page Wizard:

1. Navigate to the Application home page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
2. On the Application home page, click **Create Page**.

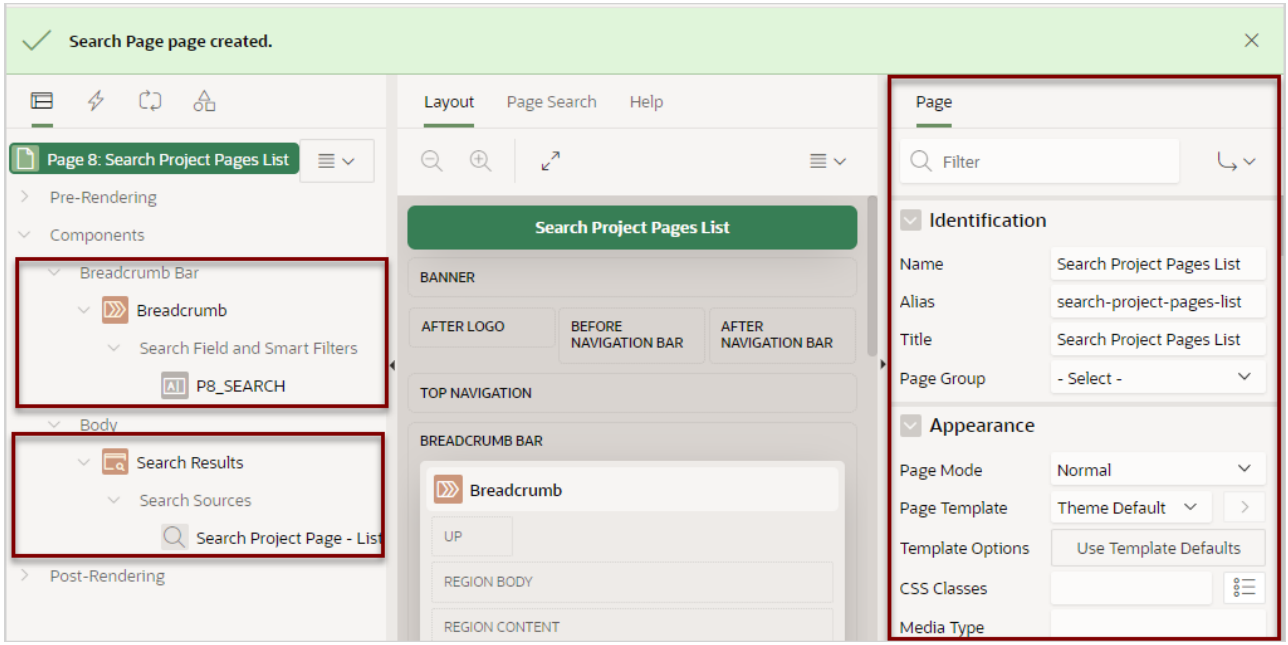
   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
3. Under Component, select **Search Page**.
4. Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page (for example, `Search Project Pages List`).

      This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.
   c. Page Mode - Identify the page mode. Accept the default, **Normal**.
5. Search Configurations - Select the search configuration to be used by this page (for example, select **Search Project Page - List**).



6. Navigation - Accept the default and enable **Use Breadcrumb** and **Use Navigation**.
7. Click **Create Page**.

   Page Designer appears.

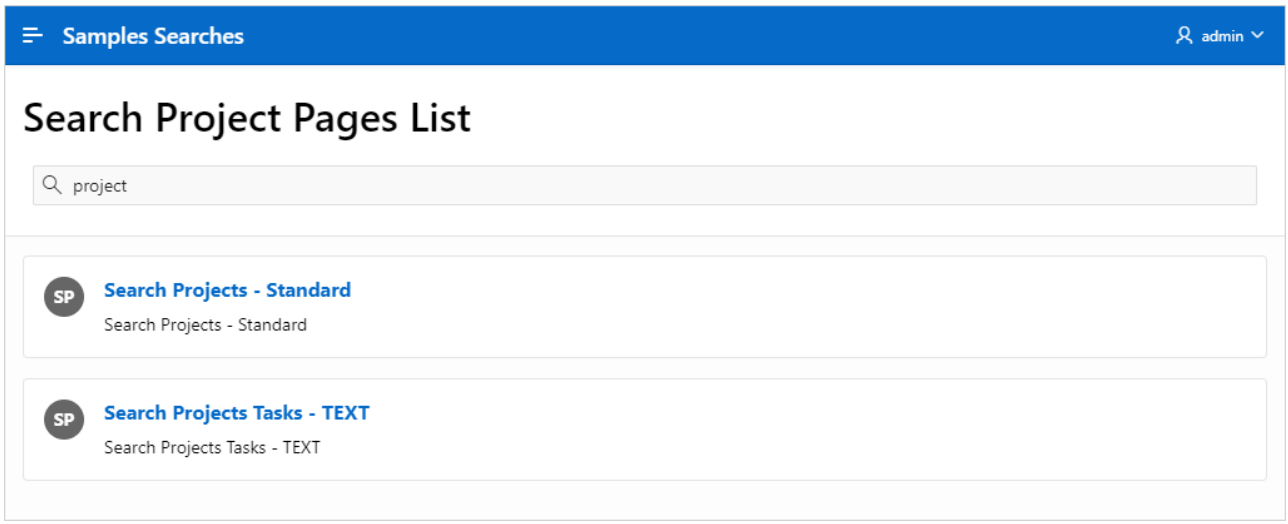


In the Rendering tab, notice the page item, **P8_SEARCH** and the **Search Results** region. **P8_SEARCH** is the search field which enables users to submit search terms. The **Search Results** region contains both the search results and the Search Source (**Search Project Page - List**) which maps to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.

8. Run and test the page:
   a. Click **Save and Run Page**.

      The search page appears.
   b. In the search field, enter a search (for example, `bug`) and press **Enter**.

   The search results appear.



16.9.6 Creating a Ubiquitous Search

Search across multiple objects within an entire schema by creating a ubiquitous database search.

> **Note:**
>
> Searching using an Oracle ubiquitous search index requires Oracle Database 23ai or later. Also, the index must be located in the parsing schema of the application.

This topic demonstrates how to create a search page on multiple tables that have a ubiquitous search index. In this topic, you create a simple ubiquitous search index, a search configuration using that ubiquitous search index, and then a search page using the Create Page Wizard.

- Creating a Ubiquitous Search Index
  Learn how to create a ubiquitous search index on multiple tables.
- Creating a Ubiquitous Search Configuration
  Create a search configuration on an existing ubiquitous search index.
- Creating a Ubiquitous Search Page
  Create a Search page by running the Create Page Wizard.

16.9.6.1 Creating a Ubiquitous Search Index

Learn how to create a ubiquitous search index on multiple tables.

> **Note:**
>
> Searching using an Oracle ubiquitous search index requires Oracle Database 23ai or later. Also, the index must be located in the parsing schema of the application.

Starting with Oracle Database 23ai, you can use the `DBMS_SEARCH PL/SQL` package for indexing of multiple schema objects in a single index. To learn more, see Performing Ubiquitous Search with DBMS_SEARCH APIs in *Oracle Text Application Developer's Guide*.

A ubiquitous search index (or `DBMS_SEARCH` index) is a JSON search index with predefined set of preferences and settings enabled for performing full text search on multiple columns, tables, or views. You use the `DBMS_SEARCH PL/SQL` package to create, manage, and query these indexes.

This topic describes how to create a very simple ubiquitous search index on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

To create a ubiquitous search index:

1. Click **SQL Commands**.

   The SQL Commands home page appears.

2. Enter the following SQL command in the command editor:

```
BEGIN
    DBMS_SEARCH.CREATE_INDEX( 'MY_SEARCH_INDEX' );
    DBMS_SEARCH.ADD_SOURCE ( 'MY_SEARCH_INDEX', 'EBA_PROJECT_TASKS' );
END;
/
```

3. Click **Run** (**Ctrl+Enter**) to execute the command.

   The results appear in the Results pane.

16.9.6.2 Creating a Ubiquitous Search Configuration

Create a search configuration on an existing ubiquitous search index.

A **search configuration** contains information about searchable data source. This task creates a search configuration using an existing ubiquitous search index created on the `EBA_PROJECT_TASKS` tables.

> **Note:**
>
> This task assumes you have already created an application and an interactive report built on the `EBA_PROJECT_TASKS` table available in the sample dataset, Project Data.

To create a ubiquitous search configuration:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

      The Search Configurations page appears.
2. On the Search Configurations page, click **Create**.
   a. Name - Enter a name for the search configuration (for example, `Projects Tasks - Ubiquitous`).
   b. Search Type - Select a type of search. Options include:
      - **Standard** - Defines a searchable data source such as a table, SQL Query, or REST Data Source. Searches is executed by a SQL Query using `LIKE` expressions.
      - **Oracle TEXT** - Define a searchable table or SQL Query which already has an Oracle TEXT index created. Search is executed by using Oracle TEXT functionality, which provides linguistic and fuzzy search capabilities.
      - **Oracle Ubiquitous Search** - Search using an Oracle ubiquitous search index (requires Oracle Database 23ai or later). The index must be located in the parsing schema of the application.
      - **List** - Search within a selected List from Shared Components.

      For this example, select **Oracle Ubiquitous Search** and click **Next**.
3. Source:
   a. Search Index Name - Select an Oracle ubiquitous search index, this index must already exist in your parsing schema.

      For this example, select **MY_SEARCH_INDEX**.
   b. Search Source Name - Choose an Oracle Ubiquitous Search index source from your parsing schema which is part of your selected Oracle Ubiquitous Search index. For this example, select `EBA_PROJECT_TASKS`.
   c. Click **Next**.
4. Column Mapping:
   a. Primary Key Column - Select the primary key column for the Search Configuration Data Source (for example, **ID**).
   b. Description Column - Select the column to be used as description.
   c. Oracle Ubiquitous Index Column - Select the Oracle Ubiquitous enabled column (for example, **NAME**).
   d. Description Column - Select the column to be used as the description (for example, **DESCRIPTION**).
   e. Icon Source - Configure the icon that displays in the header of the search result. Options include:
      - **Initials** - Use the Title column to display the icon as initials.
      - **CSS Classes** - Use the entered CSS classes to display the icon.
      - **Class Column** - Use the selected column containing the icon CSS class to display the icon.

For this example, accept the default, **Initials**.

5. Click **Create Search Configuration**.

   The search configuration is created. The Search Configuration, Edit page appears and displays the message `Search configuration created`.

6. Edit the search configuration to link to a report:

   a. Link, Link Type - Select the type of link (for example, **Redirect to Page in this Application**).

   b. Page - Select or enter the page to redirect to (for example, **2**).

   c. Click **Apply Changes**.

      The Search Configurations page appears with the message `Search Configuration updated`.

7. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.6.3 Creating a Ubiquitous Search Page

Create a Search page by running the Create Page Wizard.

Whether you are creating a **Standard** search or **Oracle Ubiquitous** search, creating a search page by running the Create Page Wizard involves the same steps. The Create Page Wizard creates a Search field for you and prompts you to select search configurations. Note that you can also add additional search configurations later in Page Designer.

A Search page features a Search field and a Search Results region. In Page Designer, the Search Results region contains **Search Sources** which map to the search configurations defined in Shared Components.

To create a search page by running the Create Page Wizard:

1. Navigate to the Application home page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select the application.

2. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

3. Under Component, select **Search Page**.

4. Page Definition:

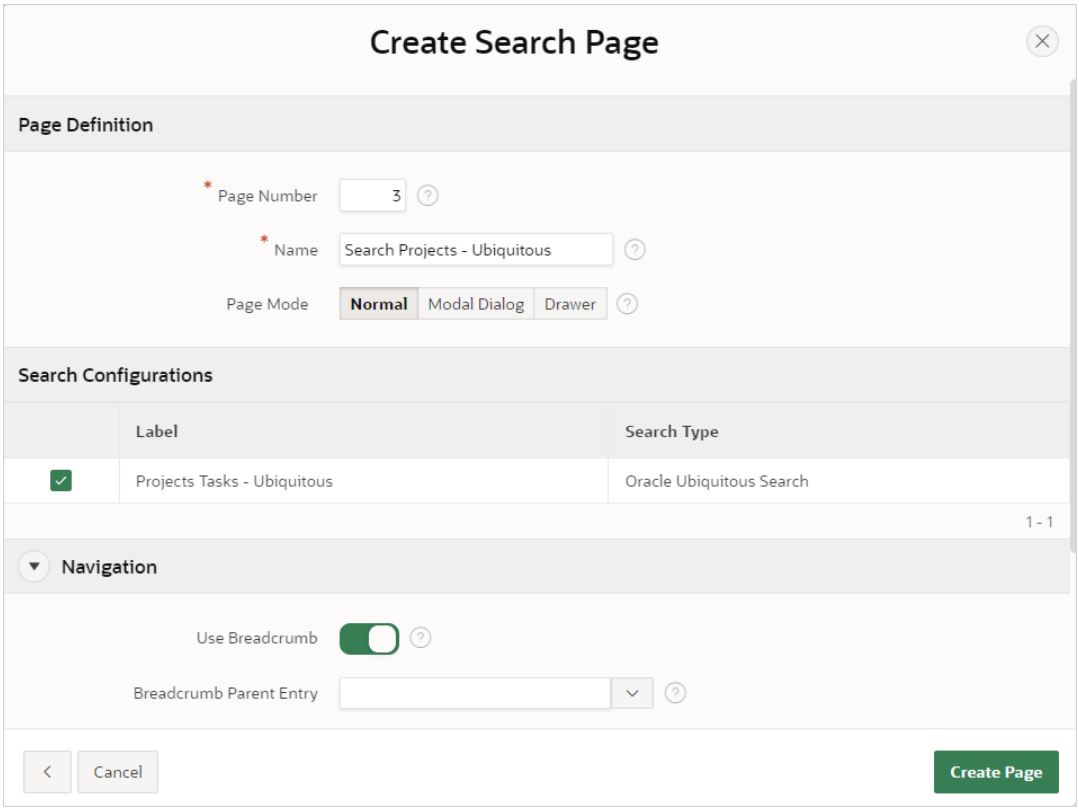
   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page (for example, `Search Projects - Ubiquitous`).

      This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. Accept the default, **Normal**.

5. Search Configurations - Select the search configuration to be used by this page (for example, select **Projects Tasks - Ubiquitous**).



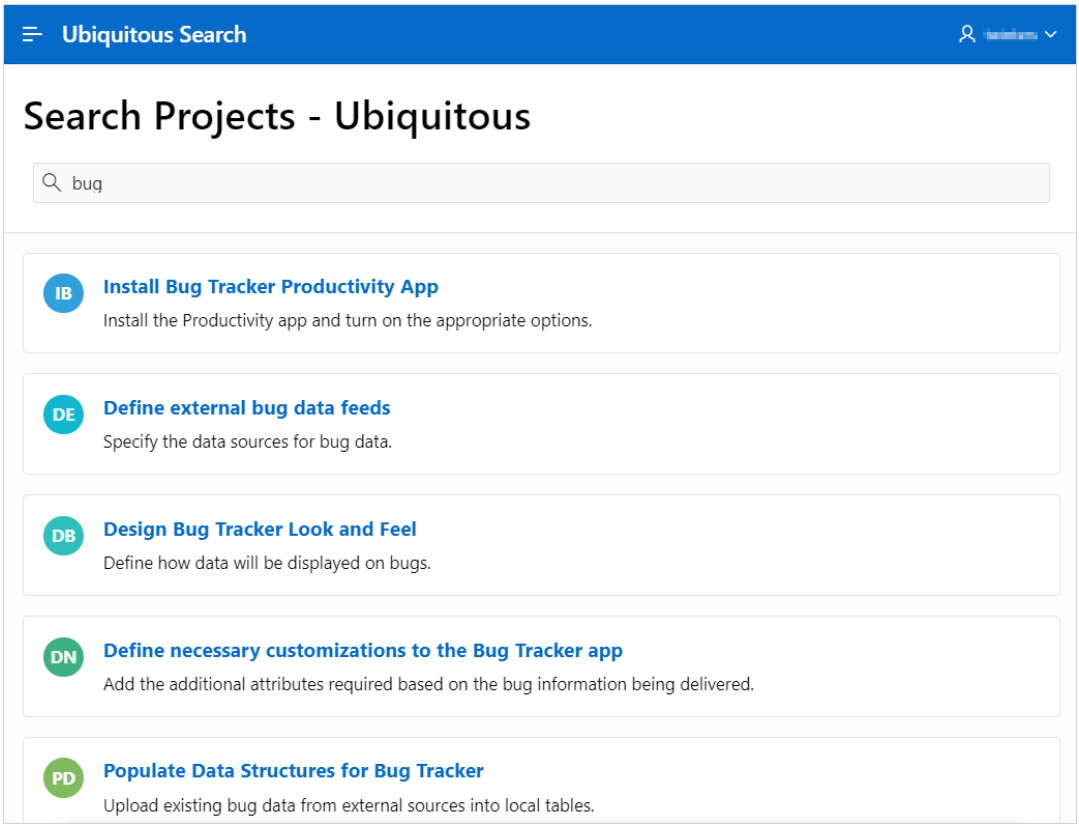
6. Navigation - Accept the default and enable **Use Breadcrumb**.
7. Click **Create Page**.

   Page Designer appears.

   In the Rendering tab, notice the page item, `PX_SEARCH` (where *X* is the page number) and the Search Results region. `PX_SEARCH` is the search field which enables users to submit search terms. The Search Results region contains both the search results and the Search Source which maps to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.
8. Run and test the page:
   a. Click **Save and Run Page**.

      The search page appears.
   b. In the search field, enter a search (for example, `bug`) and press **Enter**.

   The search results appear.



16.9.7 Creating an Oracle Vector Search

Create an Oracle Vector Search to search both structured and unstructured data by semantics or meaning, and by values, enabling ultra-sophisticated AI search applications.

> **Note:**
>
> Oracle Vector Search requires Oracle Database 23ai or later.

This topic demonstrates how to create a Oracle Vector Search on the `EBA_PROJECT_MILESTONES` table available in the sample dataset, Project Data. This example assumes you have already installed the sample, dataset, Project Data.

- About Oracle Vector Search Configurations
  Learn about Oracle Vector Search configurations.
- Adding a Vector Column
  Learn how to add a `VECTOR` column to a table.
- Retrieving the Vector Embeddings from a Vector Provider
  Retrieve vector embeddings from a Vector Provider.
- Creating an Oracle Vector Search Configuration
  Create an Oracle Vector Search configuration.
- Creating an Oracle Vector Search Page
  Create a Search page by running the Create Page Wizard.

> **See Also:**
>
> Using Sample Datasets in *Oracle APEX SQL Workshop Guide*

16.9.7.1 About Oracle Vector Search Configurations

Learn about Oracle Vector Search configurations.

Starting with Oracle Database 23ai, you can use the `VECTOR` data type to store vector embeddings alongside business data in the database. Using embedding models, you can transform unstructured data into vector embeddings that can then be used for semantic queries on business data.

The process of creating an Oracle Vector Search configuration involves the following general steps:

1. Add a column with a data type of `VECTOR` to your table.
2. Create a Vector Provider to create vector embeddings for text strings. See Managing Vector Providers.
3. Retrieve the embeddings from a Vector Provider using the `GET_VECTOR_EMBEDDINGS` function in `APEX_AI` package. See Retrieving the Vector Embeddings from a Vector Provider
4. Create a Search Configuration based on your Vector Provider and your `VECTOR` column. See Creating an Oracle Vector Search Configuration.
5. Create an Oracle Vector Search page. Retrieving the Vector Embeddings from a Vector Provider.

> **See Also:**
>
> - `GET_VECTOR_EMBEDDINGS` function in APEX_AI in *Oracle APEX API Reference*
> - Overview in *Oracle Database AI Vector Search User's Guide*

16.9.7.2 Adding a Vector Column

Learn how to add a `VECTOR` column to a table.

The following examples uses the `EBA_PROJECT_MILESTONES` table which is available in sample dataset, Project Data.

To add a `VECTOR` column to a table:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **SQL Commands**.

   The SQL Commands home page appears.
3. Enter the following SQL command in the command editor:

```
ALTER TABLE eba_project_milestones ADD (vector_column vector);
```

4. Click **Run** (**Ctrl+Enter**) to execute the command.

   The results appear in the Results pane.

16.9.7.3 Retrieving the Vector Embeddings from a Vector Provider

Retrieve vector embeddings from a Vector Provider.

A Vector provider enables you to convert text into an embedding. The functionality can be covered by an AI Service, but also locally by an ONNX model in the database or a user-defined PL/SQL function.

The following example assumes you have created Vector Provider and are retrieving the embeddings from the `EBA_PROJECT_MILESTONES` table using the `GET_VECTOR_EMBEDDINGS` function in `APEX_AI` package.

To retrieve vector embeddings from a Vector Provider:

1. On the Workspace home page, click **SQL Workshop**.
2. Click **SQL Commands**.

   The SQL Commands home page appears.
3. Enter the following SQL command in the command editor:

```
UPDATE eba_project_milestones
   SET vector_column = apex_ai.get_vector_embeddings(
                           p_value             => name || ' ' ||
description,
                           p_service_static_id => 'VP_OCI_EMBEDDING' )
```

4. Click **Run** (**Ctrl+Enter**) to execute the command.

   The results appear in the Results pane.

> **See Also:**
>
> `GET_VECTOR_EMBEDDINGS` functions in APEX_AI in *Oracle APEX API Reference*

16.9.7.4 Creating an Oracle Vector Search Configuration

Create an Oracle Vector Search configuration.

> **Note:**
>
> An Oracle Vector Search configuration requires Oracle Database 23ai or later. Before you can create an Oracle Vector Search configuration, you must complete all the steps described in About Oracle Vector Search Configurations.

A **search configuration** contains information about searchable data source.

To create a Oracle Vector Search configuration:

1. Access the Search Configurations page:

a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

c. On the Application home page, click **Shared Components**.

d. Under Navigation and Search, click **Search Configurations**.

The Search Configurations page appears.

2. On the Search Configurations page, click **Create**.

a. Name - Enter a name for the search configuration (for example, `Search Milestones - Vector`).

b. Search Type - Select a type of search. Select **Oracle Vector Search**:

c. Click **Next**.

3. Source:

a. Vector Provider - Select a Vector Provider used to get an embedding for a given search term. The search term must be converted using the same embedding model as the dataset in your table.

b. Source Type - Specify the source of the search configuration. Oracle Vector search configuration can be based on a Table or SQL Query.

c. Table / View Owner - Select the owner of the table that contains the vector column to be used for the search configuration (for example, **Table**).

d. Table / View Name - Select the table that contains the vector column to be used for the search configuration (for example, `EBA_PROJECT_MILESTONES`).

e. Click **Next**.

4. Column Mapping:

a. Primary Key Column - Select the primary key column for the Search Configuration Data Source (for example, **ID Number**).

b. Vector Column - Select the vector column. For similarity searches through the approximate method, there must be a vector index for this column (for example, **VECTOR_COLUMN (Vector)**).

c. Title Column - Select the column to be used as title (for example, **NAME (Varchar2)**).

d. Description Column - Select the column to be used as the description (for example, **DESCRIPTION (Varchar2)**).

e. Icon Source - Configure the icon that displays in the header of the search result. Options include:

- **Initials** - The Title column to be used to display the icon as initials.
- **CSS Classes** - The entered CSS classes to be used to display the icon.
- **Class Column** - The selected column containing the icon CSS class to be used to display the icon.

For this example, accept the default, **Initials**.

5. Click **Create Search Configuration**.

The search configuration is created. The Search Configuration, Edit page appears and displays the message `Search configuration created`.

6. Edit the search configuration to link to a report:

a. Link, Link Type - Select the type of link (for example, **Redirect to Page in this Application**).

b. Page - Select or enter the page to redirect to (for example, **2**).

c. Click **Apply Changes**.

The Search Configurations page appears with the message `Search Configuration updated`.

7. Click the **Application ID** breadcrumb to return to the Application home page.

16.9.7.5 Creating an Oracle Vector Search Page

Create a Search page by running the Create Page Wizard.

The Create Page Wizard creates a Search field for you and prompts you to select search configurations. Note that you can also add additional search configurations later in Page Designer.

A Search page features a Search field and a Search Results region. In Page Designer, the Search Results region contains **Search Sources** which map to the search configurations defined in Shared Components.

To create a search page by running the Create Page Wizard:

1. Navigate to the Application home page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select the application.

2. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

3. Under Component, select **Search Page**.

4. Page Definition:

   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page (for example, `Search Projects – Vector`).

   This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. Accept the default, **Normal**.

5. Search Configurations - Select the search configuration to be used by this page (for example, select **Search Page - Vector**).

6. Navigation - Accept the default and enable **Use Breadcrumb**.

7. Click **Create Page**.

   Page Designer appears.

   In the Rendering tab, notice the page item, `PX_SEARCH` (where *X* is the page number) and the Search Results region. `PX_SEARCH` is the search field which enables users to submit search terms. The Search Results region contains both the search results and the Search Source which maps to search configurations maintained in Shared Components. You can further refine and control the search display and behavior by editing attributes in the Property Editor.

8. Run and test the page:

   a. Click **Save and Run Page**.

The search page appears.

b. In the search field, enter a search (for example, `bug`) and press **Enter**.

The search results appear.

16.9.8 Creating a Search Page Manually

Learn how to create a Search page manually in Page Designer.

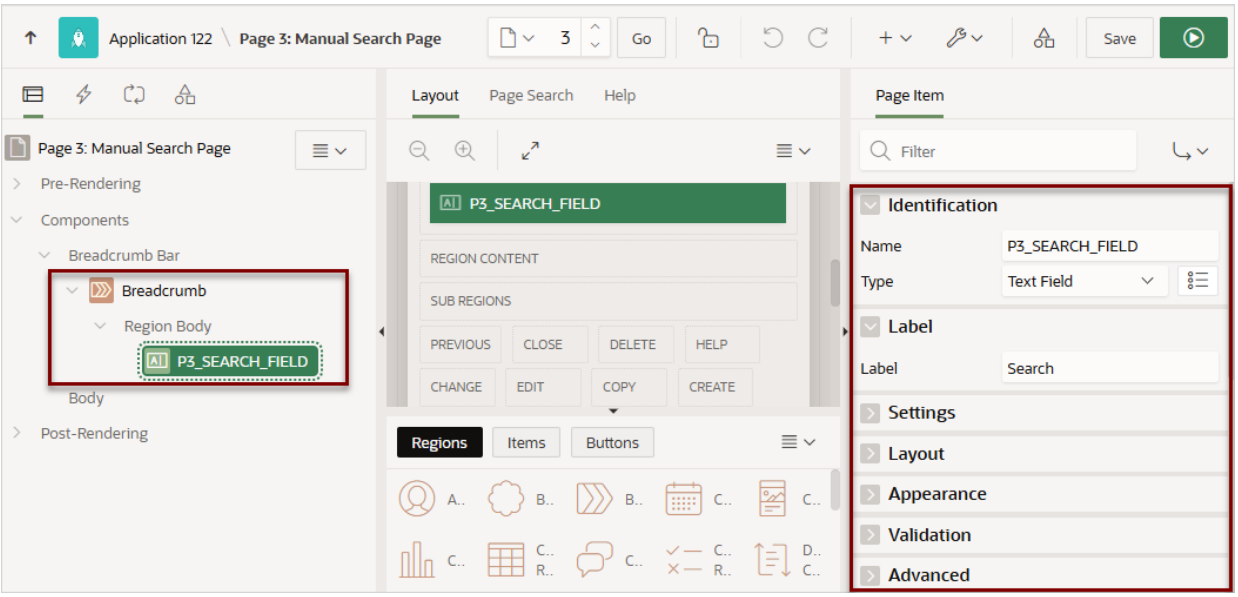
In previous examples, you create a search page by running the Create Page Wizard. You can also create a blank page and add search components to it manually.

> **Tip:**
>
> This task uses the tables and search configurations described in Creating Standard Search Configurations. Once you create your new application, access Shared Components and click **Copy** to copy them from another application.

To create a search page manually:

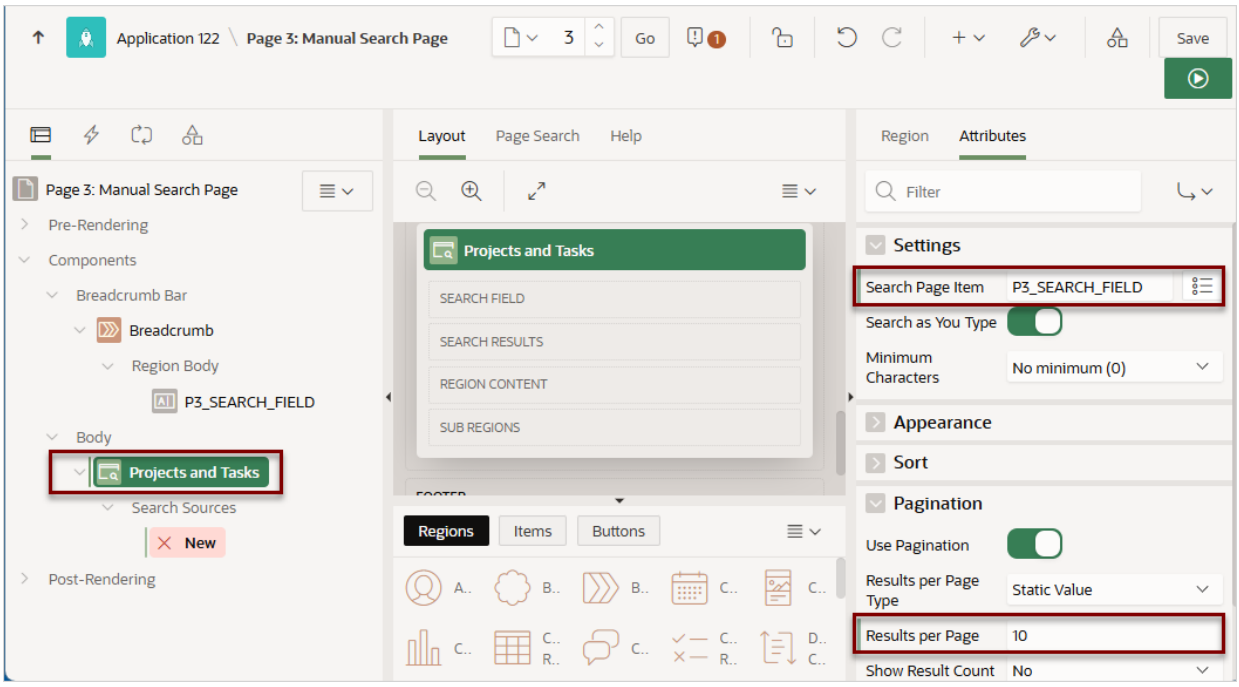
1. Create an application and then your search configurations.
2. Run the Create Page Wizard and create a blank page.
   a. On the Application home page click the **Create Page** button.
   b. Under Component, select **Blank Page**.
   c. Page Definition:
      - Page Number - The page number is an integer value that identifies a page within an application.
      - Name - Specify a text name for this page (for example, `Manual Search Page`). This text is also used for page Title. After page creation, you can modify the Title in Page Designer.
      - Page Mode - Identify the page mode. For this example, select **Normal**.
   d. Navigation - Accept the defaults and enable navigation.
   e. Click **Create Page**.
3. Create a text field for the search input:
   a. In the Rendering tab, right-click **Breadcrumb** and select **Create Page Item**.

      The New item appears.
   b. Select the new item and edit following attributes in the Property Editor:
      - Identification Name - Enter the name of this item (for example, `PX_SEARCH_FIELD`, where X is the page number).
      - Label, Label - Enter the field label for the page item (for example, `Search`).
      - Advanced, Warn on Unsaved Changes - Select **Ignore**.
      - Click **Save**.



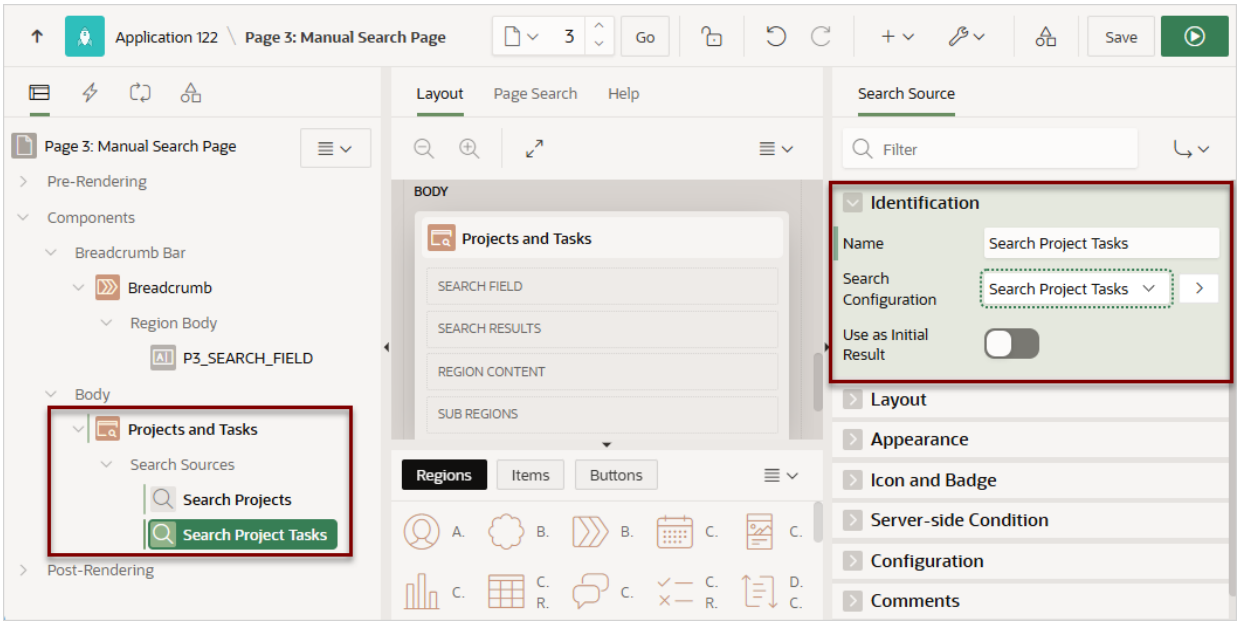
4. Create a search region:
   a. In the Rendering tab, right-click **Body** and select **Create Region**.

      The New region appears.
   b. Select the **New** region and edit the following Region attributes:
      - Identification, Name - Enter a region title (for example, `Projects and Tasks`).
      - Identification, Type - Select **Search**.
   c. Click the **Attributes** tab and edit the following:
      - Settings, Search Page Item - Select or enter the page item in to which users type their search (for example, **PX_SEARCH_FIELD** where *X* is the page number).
      - Pagination, Results per Page - Specify the results to display on the page (for example, **10**).



5. Define the first Search Source:
   a. In the Rendering tab under Search Sources, select **New**.
   b. Edit the following attributes in the Property Editor:
      - Identification, Name - Enter a descriptive name for this search source (for example, `Search Projects`).
      - Identification, Search Configuration - Select a search configuration to search within (for example, **Search Projects**).
6. Define the second Search Source:
   a. In the Rendering tab, right-click **Search Sources** and select **Create Search Source**.

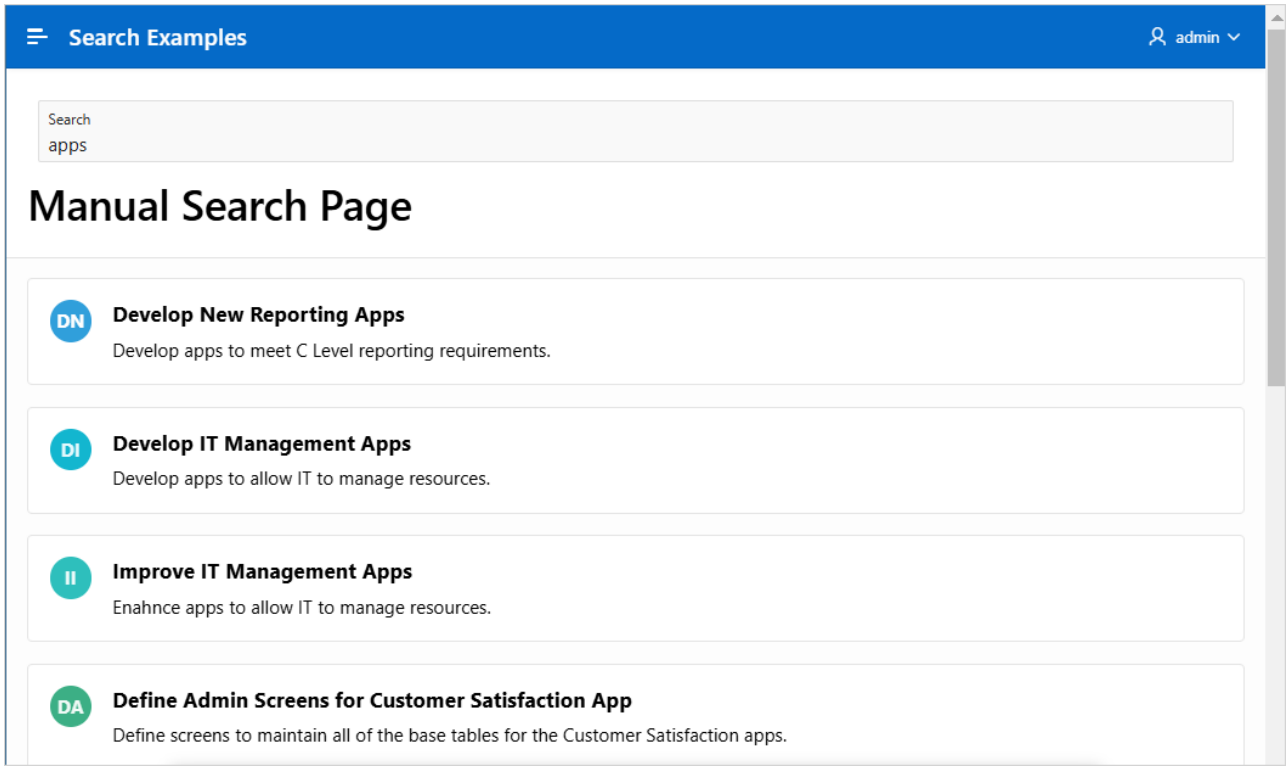
      The **New** Search Source appears.
   b. In the Rendering tab under Search Sources, select **New** and edit the following Region attributes in the Property Editor:
      - Identification, Name - Enter a descriptive name for this search source (for example, `Search Project Tasks`).
      - Identification, Search Configuration - Select a Search Configuration to search within (for example, **Search Project Tasks**).



7. Run and test the page:
   a. Click **Save and Run Page**.

      The Search Projects page appears.
   b. In the search field, enter a search (for example, `apps`).

   The search results appear.



16.9.9 Editing a Search Configuration

Edit an existing search configuration in Shared Components on the Search Configurations, Edit page.

To access the Search Configurations, Edit page:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

      The Search Configurations page appears.
2. Select the appropriate search configuration label.

   The Edit page appears.
3. On the Edit page, configure the attributes.

   To learn more about an attribute or view examples, see field-level Help.
4. Settings:
   a. Search Type - Displays the search type (Standard, Oracle TEXT, Oracle Ubiquitous Search, or List).
   b. Label - Enter the label to be displayed in Search results. When using a search configuration within a search region, the label can be overridden in Page Designer.
   c. Search Query Prefix - Enter a prefix to be used by end users in search queries. A search term which is prefixed will only be used for search configurations with the same

prefix. If a prefix is used in a search query, but not defined in any search configuration, the prefix is ignored and the whole term is used for searching

d. Static ID - Use the Static ID to reference the search configuration definition in API calls (for example, when using the `APEX_SEARCH.SEARCH` function).

5. Subscription - Select a master search configuration from which to subscribe. See Subscribing to a Search Configuration

6. Source:

   a. Source Type - Specifies the source of the search configuration. Search configuration can be based on a Table, SQL Query, or Function Body returning a SQL Query.

      The UI changes depending upon the Source Type you select. This topic assumes Source Type is set to **SQL Query**. To learn more about an attribute, see field-level Help.

   b. Order By Clause - Enter a valid SQL ORDER BY expression, used to sort the records retrieved.

   c. Index Column Name - Select the column which is indexed with an Oracle Text index. Note that an Oracle Text index is always created in one specific column, but can actually index multiple columns or even multiple tables. See "Multicolumn Datastore" or "User Datastore" within the Oracle Text documentation for more details.

   d. Text Query Function - Select the Text Query Function type to use.

      The Oracle Text query syntax is very complex and powerful and provides a rich set of query operators to deal with wildcards, stemming, fuzzy searches, and so forth. For end users, this syntax is often hard to understand. You can select a function to convert simple end user input to an Oracle Text query with the advanced search features automatically applied.Text Query Function options include:

      - **None** - Oracle APEX encloses each query term with "{" and "}", except the keywords `and` and `or`.
      - **Search Engine** - This function performs a query relaxation search to retrieve relevant results beyond just exact matches. The goal is to get as many related results as possible, allowing users to easily find related information without needing to input precise queries. The search process begins with an exact match search and then progressively relaxes the query using techniques like stemming and fuzzy matching. While maximizing the number of results, the search engine function ensures that the most exact and relevant matches are always ranked at the top of the search results.
      - **Expert Search** - This function supports more advanced search operators for precise queries. It allows exact matching, Boolean operators (AND, OR, NOT), precedence with parentheses and proximity search. Additional parameters allow customizing search behavior. This is best when requiring highly accurate results from very specific queries.

        See APEX_SEARCH, `QUERY_EXPERT_SEARCH` in *Oracle APEX API Reference*
      - **Custom** - Provide a custom function in the **Custom Function Name** attribute.

7. Column Mapping - Select the appropriate columns To learn more or view an example, see field-level Help.

8. Icon and Display - Configure icon and icon display options. The UI changes depending upon the **Icon Source** you select. To learn more, see field-level Help.

9. Link - Select the type of link. The UI changes depending upon the **Link Type** you select. To learn more, see field-level Help.

> **Tip:**
>
> To see an example of linking to another page, see Editing Search Configurations to Include Links.

10. Server-Side Condition - Select a condition type from the list that must be met in order for this component to be rendered processed. The UI changes depending upon the **Condition Type** you select.
11. Advanced:
    - Authorization Scheme - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
    - Build Option - Select a build option for this component. Build options are predefined settings that determine whether or not components within an application are enabled.
    - Comments - Enter developer comments or notes.
12. Click **Apply Changes** to save your changes.

16.9.10 Copying or Subscribing to a Search Configuration

Copy a search configuration from the current application or from another applications. When copying a shortcut from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a search configuration:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Search Configurations**.

   The Search Configurations page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
3. To copy a search configuration within the current application:
   a. Search for the search configuration and click **Copy** in the Copy column.

      The Copy Search Configuration Wizard appears.
   b. Copy From - Accept the default, **This Application**.
   c. Copy Search Configuration - Identify the search configuration to copy.
   d. New Search Configuration Label - Provide a label of the search configuration you would like to create.

   e. Click **Copy**.

4. To copy and/or subscribe to copy a search configuration from another application:
   a. From the Tasks region on the Email Templates page, click **Copy from another app**.

      The Copy Search Configuration Wizard appears.
   b. Copy From - Accept the default, **Other Application**.
   c. Application - Select the application from which you want to copy search configuration.
   d. Copy Search Configuration - Identify the search configuration to copy.
   e. New Search Configuration Label - Provide a label of the search configuration you would like to create.
   f. Subscribe - Set Subscribe to **On** to subscribe to the search configuration..
   g. Click **Copy**.

16.9.11 Subscribing to a Search Configuration

Subscribe to a search configuration on the Search Configurations, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a search configuration by copying it and then subscribing. See Copying or Subscribing to a Search Configuration.

To subscribe to a search configuration:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Search Configurations**.

   The Search Configurations page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
3. Select a search configuration.

   The Search Configurations, Edit page appears.
4. To subscribe to a search configuration:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master search configuration to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.

5. To refresh a search configuration:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Shortcut - If the shortcut is already subscribed, click **Refresh**.
6. To unsubscribe from a search configuration:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.9.12 Editing a Search Page

Learn how attributes can change search page behavior

> **Tip:**
>
> The examples in this section use the tables and search configurations described in Creating Standard Search Configurations.

- Reviewing Search Page Components
  Review the components that comprise a search page by viewing attributes in Page Designer.
- Editing Search Results Region Attributes
  Edit Search Results region attributes to change region behavior.
- Editing Search Results Attributes
  Edit Search Results Attributes to control how a search works.
- Editing Search Source Attributes
  Edit the behavior of a search source by editing Search Source attributes in Page Designer.

16.9.12.1 Reviewing Search Page Components

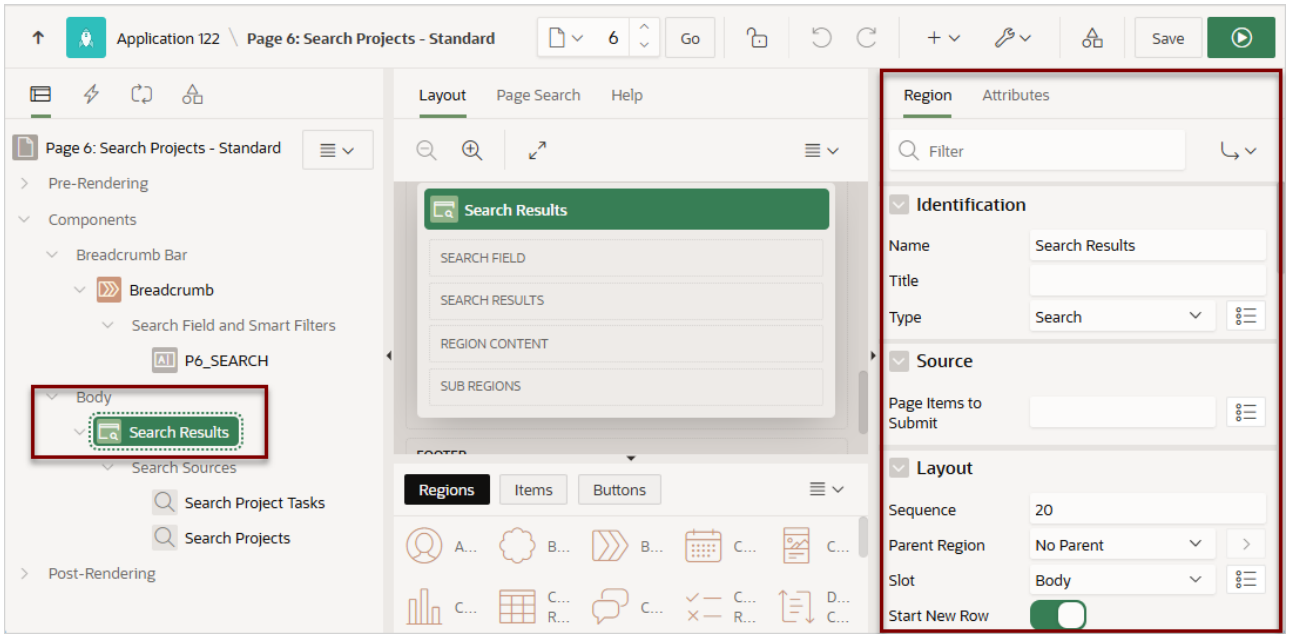
Review the components that comprise a search page by viewing attributes in Page Designer.

A search page features a **Search Results** region which contains **Search Sources** that map to search configurations defined in Shared Components.

To review the components of a search page:

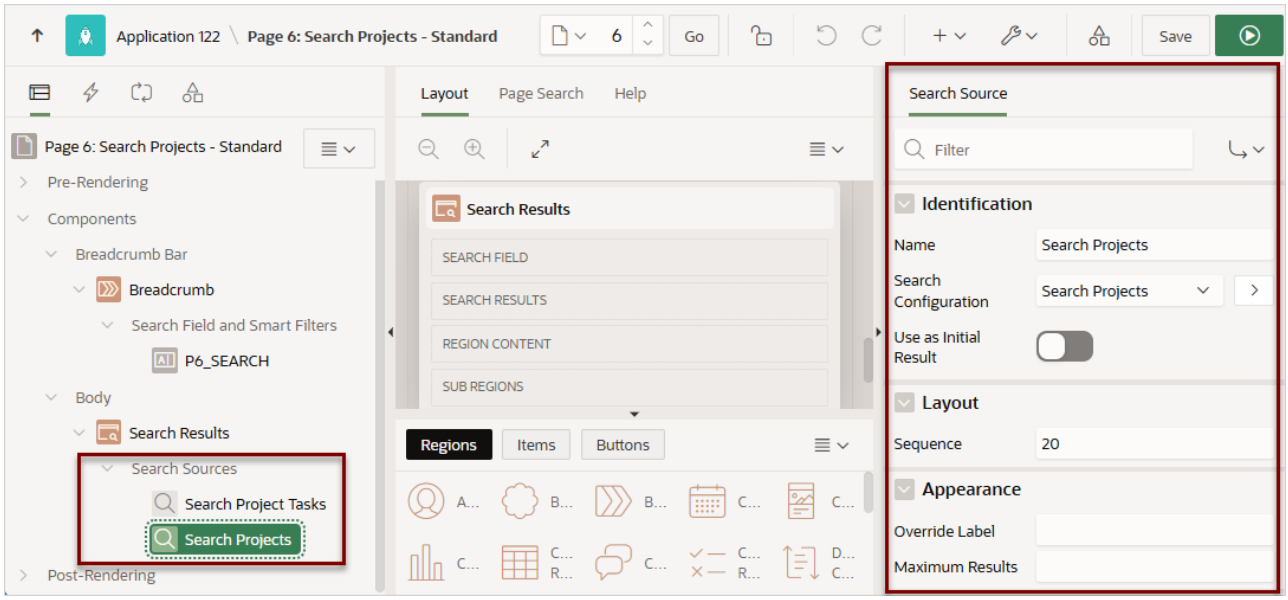
1. View the search page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Review the Search Results. In the Rendering tab under **Body**, locate and select the **Search Results** region.



Note that the Property Editor contains two tabs: **Region** and **Attributes**. Under Region, note that the region Type is **Search**.

3. In the Rendering tab, select a Search Source (for example, **Search Projects**).



4. In the Property Editor, review the Search Source attributes.

   In the previous example, note that the attributes indicate that the Search Source maps to the Search Configuration, **Search Projects**.

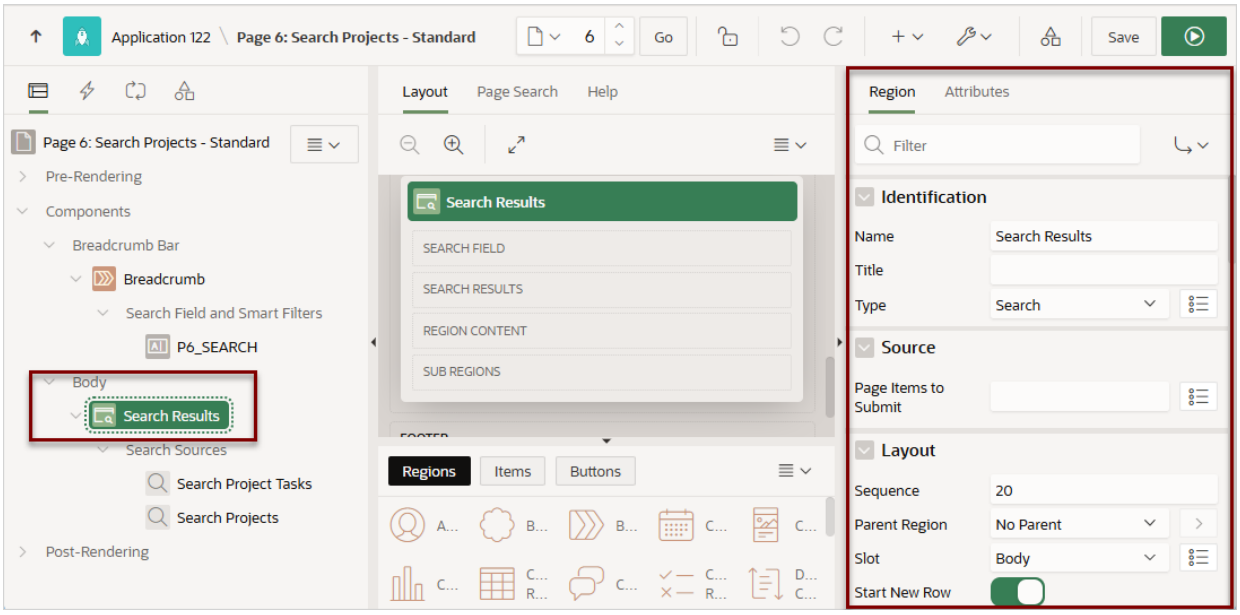
16.9.12.2 Editing Search Results Region Attributes

Edit Search Results region attributes to change region behavior.

To edit Search Results Region attributes:

1. View the search page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. View the Search Results Region attributes:
   a. In the Rendering tab (or left pane), select the **Search Results** region.
   b. In the Property Editor, select the **Region** tab (if not already selected).



3. Edit the attributes.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

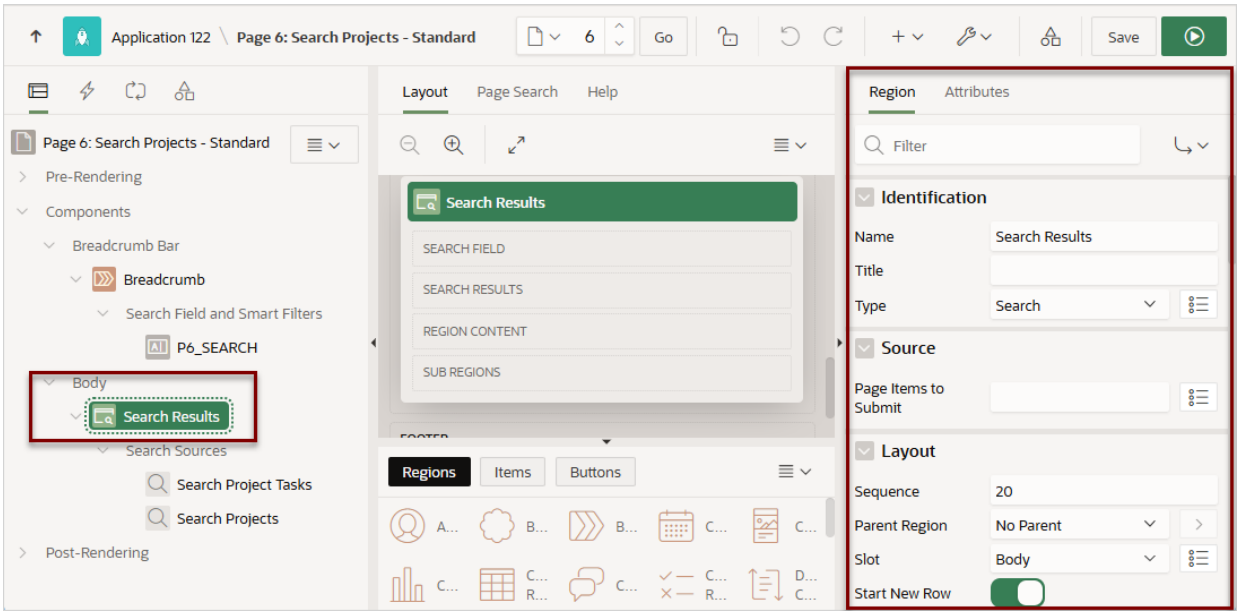
16.9.12.3 Editing Search Results Attributes

Edit Search Results Attributes to control how a search works.

To edit Search Results Attributes:

1. View the search page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. View the Search Results Attributes:
   a. In the Rendering tab (or left pane), select the **Search Results** region.
   b. In the Property Editor, select the **Attributes** tab.



3. Edit the Setting attributes.

   To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

   - Settings, Search Page Item - Reference the page item for end users to type their searches in (for example, **P6_SEARCH**).
   - Settings, Search as You Type - When enabled, the search is executed as you type in the search field. Otherwise the search is not executed until the user presses Enter or a button.
   - Settings, Minimum Characters - Specify the minimum number of characters required in a search term before the search executes. This setting can be useful to reduce the amount of searching done when queries are expensive.
   - Pagination, Use Pagination - Specify whether to use pagination for displaying search results.
   - Pagination, Results per Page Type - Select how the number of results per page is determined. Options include: **Static Value** or **Based on Item Value**.
   - Pagination, Results per Page - Determine the amount of results that display on each page.
   - Pagination, Show Result Count - Specify where to display the search results row count. See Help in the central pane for more details.
   - Appearance, Custom Layout - Specify whether to use a custom template for search results, instead of the display templates specified within the search configuration.
   - Appearance, Last Modified Format Mask - Provide the format mask to use for displaying the Last Modified attribute of a search result. Values are returned as the `TIMESTAMP WITH LOCAL TIME ZONE` data type.
   - Sort, Overall Sort - Search results are returned in the order they appear on the Search Configurations page. The order within each search configuration is defined on the Shared Components, Edit page. To override the default order, select a column from the list.
     See the Help tab in the central pane for more details.

- Messages, No Query Entered Message - Enter a message to display when no query is entered and no Search Source has the **Use as Initial Result** switch enabled.
  See the Help tab in the central pane for examples.
- Messages, No Results Found Message - Enter a message to display when no results are found. If specified, messages defined in the search configurations are ignored.
- Performance, Lazy Loading - Specify whether to load the region independently of the data for the region.
  When lazy loading is enabled, the page renders immediately, showing an empty region until the data loads. Generally, the page does not display until the entire page loads and is ready to be rendered. For example, if it takes 5 seconds to load all of the data for a region, without lazy loading, the end user would have to wait 5 seconds before the page started to render.

4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

16.9.12.4 Editing Search Source Attributes

Edit the behavior of a search source by editing Search Source attributes in Page Designer.

A search configuration contains information about searchable data source. When you add a search configuration to a search page, it displays under Search Sources in the Rendering tab in Page Designer. You can further refine search behavior by selecting the search source in Page Designer and editing attributes in the Property Editor.

The following task demonstrates how to edit the search source so that the each result is prefaced by an icon.

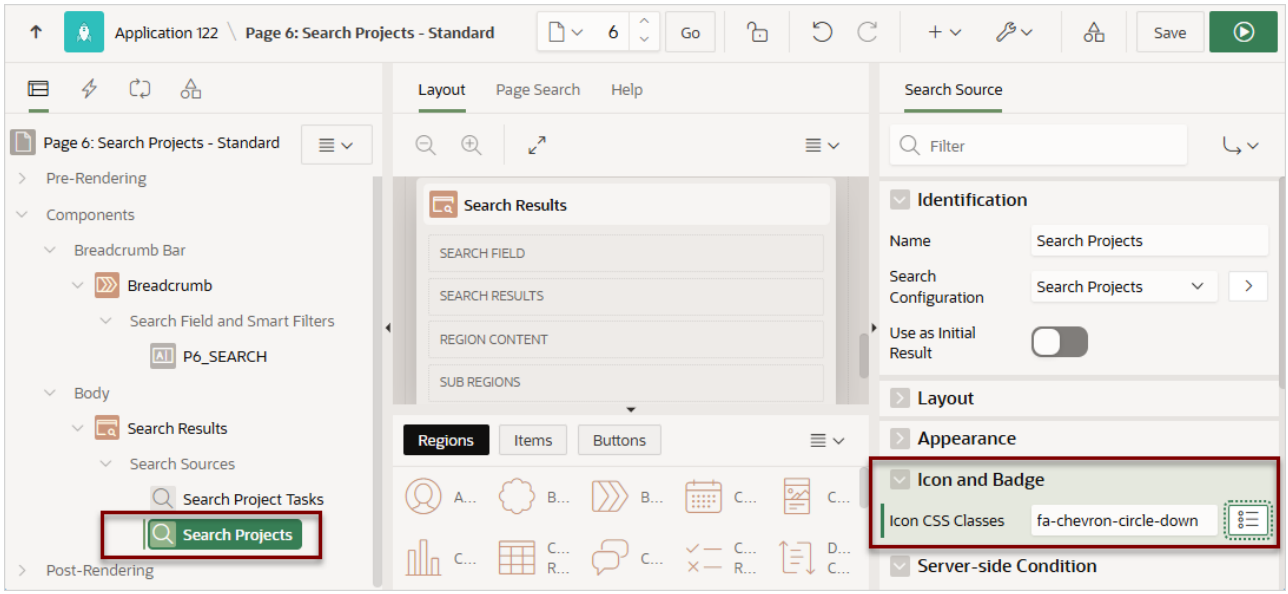
To edit Search Source attributes:

1. View the search page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Edit the first Search Source:
   a. In the Rendering tab (or left pane), select the Search Source (for example, **Search Project Tasks**).

      The Search Source attributes display in the Property Editor.
   b. Icon and Badge - Select an icon to display with the search result (for example, **fa-tasks-alt**),
3. Edit the second Search Source:
   a. In the Rendering tab (or left pane), select the Search Source (for example, **Search Projects**).

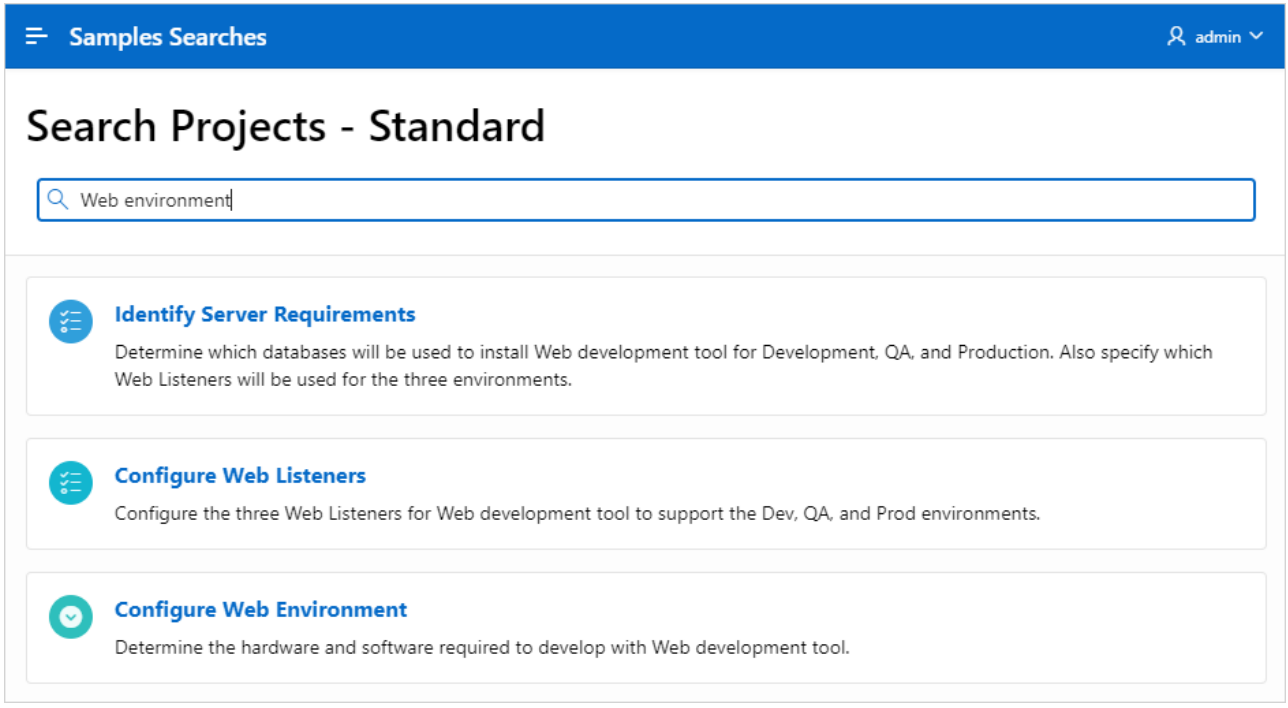
      The Search Source attributes display in the Property Editor. To view help for an attribute, select the attribute in the Property Editor and click the Help tab in the central pane.
   b. Icon and Badge - Select an icon to display with the search result (for example, **fa-chevron-circle-down**).



4. Run and test the change:
   a. Click **Save and Run Page**.

      The Search Projects page appears.
   b. In the search field, enter a search (for example, `Web environment`) and press **Enter**.

   The search results appear.



Note that each search result is prefaced by a new icon.

16.9.13 Accessing Search Configuration Reports

View the search configurations Utilization and History reports by clicking the appropriate tab on the Search Configurations page.

To view search configurations reports:

1. Access the Search Configurations page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Navigation and Search, click **Search Configurations**.

      The Search Configurations page appears.
2. Click the appropriate tab:
   - **Utilization** - Displays where search configurations are used within the application.
   - **History** - Displays recent modifications made to search configurations in this application.

16.10 Using Shortcuts

Use shortcuts to avoid repetitive coding of HTML or PL/SQL functions.

- What is a Shortcut?
  Learn about shortcuts and when to use them.
- Supported Shortcut Types
  Oracle APEX supports six types of shortcuts.
- Defining a Shortcut from Scratch
  Define a shortcut on the Shortcuts page.
- Editing Existing Shortcuts
  Once you create a shortcut, you can alter it by editing attributes on the Edit Shortcut page.
- Copying an Existing Shortcut
  Copy existing shortcuts from the Shortcut repository.
- Copying or Subscribing to a Shortcut
  Copy a shortcuts from the current application or from another applications. When copying a shortcut from another app, you can also subscribe to it.
- Subscribing to a Shortcut
  Subscribe to a shortcut on the Shortcuts, Edit page.
- Accessing Shortcut Reports
  View the shortcut Subscription and History reports.

16.10.1 What is a Shortcut?

Learn about shortcuts and when to use them.

Use a shortcut to define frequently used code once and then reference it in many places thus reducing code redundancy. For example, you can create a shortcut to define a page control such as a button, HTML text, a PL/SQL procedure, or HTML.

You can use a shortcut within the following locations:

- The Region Source attribute of regions defined as HTML Text (with shortcuts).
- Region Header and Footer Text attribute.
- Item Label attributes, Pre Element Text, Post Element Text, and Default Value attribute.
- Region Templates attributes.

Referencing a Shortcut

Once you define a shortcut, you can invoke it using specific syntax unique to the location in which the shortcut is used.

You reference shortcuts using the following syntax:

```
"MY_SHORTCUT"
```

The shortcut name must be capitalized and enclosed in quotation marks.

> **See Also:**
>
> - About Regions
> - Specifying a Region Header and Footer
> - Creating Custom Themes

16.10.2 Supported Shortcut Types

Oracle APEX supports six types of shortcuts.

When you create a shortcut you select a shortcut type. Supported shortcut types include:

- PL/SQL Function Body
- HTML Text
- HTML Text with Escaped Special Characters
- Image
- Text with JavaScript Escaped Single Quotes
- Message
- Message with JavaScript Escaped Special Quotes

Text with JavaScript Escaped Single Quotes

Use this type of shortcut to reference a shortcut inside of a JavaScript literal string. This shortcut defines a text string. When the shortcut is referenced, it escapes the single quotation marks required for JavaScript.

Message

Use this type of shortcut to reference a translatable message at runtime. Since this shortcut does not have a shortcut body, the name of the shortcut must match the corresponding

message name. At runtime, the name of the shortcut expands to the text of the translatable message for the current language.

Message with JavaScript Escaped Single Quotes

Use this type of shortcut to reference a shortcut inside of JavaScript literal string and reference a translatable message at runtime.

> **See Also:**
>
> Understanding Application Translation and Globalization Support

16.10.3 Defining a Shortcut from Scratch

Define a shortcut on the Shortcuts page.

To define a shortcut:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Click **Create**.
4. For Create Shortcut, select **From Scratch** and click **Next**.
5. On Shortcut Attributes:
   a. Name - Enter the name by which the shortcut will be referenced.
   b. Type - Select a shortcut type. Options include:
      - PL/SQL Function Body
      - HTML Text
      - HTML Text with Escaped Special Characters
      - Image
      - Text with JavaScript Escaped Single Quotes
      - Message
      - Message with JavaScript Escaped Special Quotes
   c. Shortcut - Enter the text of the shortcut. The shortcut must be less then 30,000 characters in length.
   d. Error Text - Enter the text to display if an error should occur during the processing of a shortcut.
   e. Build Option - Select a build option for this component.
6. Click **Create**.

New shortcuts are added to the Shortcuts page (also referred to as the Shortcut repository).

16.10.4 Editing Existing Shortcuts

Once you create a shortcut, you can alter it by editing attributes on the Edit Shortcut page.

To edit an existing shortcut:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Select a shortcut.

   The Edit Shortcut page appears.
4. Edit the attributes.

   To learn more about an attribute, see field-level Help.
5. Click **Apply Changes**.

> **See Also:**
>
> Viewing Field-Level and Page Designer Help

16.10.5 Copying an Existing Shortcut

Copy existing shortcuts from the Shortcut repository.

To copy an existing shortcut:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.

   The Shortcuts page appears. You can copy an existing shortcut from the Create Shortcut Wizard or by clicking Copy icon in report view.
3. To copy an existing shortcut the Create Shortcut Wizard:
   a. On the Shortcut page, click **Create**.
   b. On Creation Method, select **As a Copy of an Existing Shortcut** and click **Next**.
   c. From Copy From Application, select the application you want to copy shortcuts from and click **Next**.

d. Copy From Application - Select the application from which you want to copy shortcuts.

e. New Shortcuts:

- To Name - Optionally enter the new shortcut name.
- Copy - Select an action:
  - **Yes** - Copy the shortcut.
  - **No** - Do not copy this shortcut.
  - **Copy and Subscribe** - Copy and subscribe to this shortcut. Subscribing to a shortcut copies it to application and sets the plug-in master reference.

f. Click **Copy Shortcut**.

4. To copy an existing shortcut by clicking the Copy icon:

   a. On the Shortcut page verify report view is enabled.

   b. In the Copy column, click **Copy**.

   c. Copy Shortcut - Select the shortcut to copy.

   d. Shortcut Name - Enter the new name of the shortcut to be copied.

   e. Click **Copy**.

16.10.6 Copying or Subscribing to a Shortcut

Copy a shortcuts from the current application or from another applications. When copying a shortcut from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a shortcut:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

2. Under Other Components, click **Shortcuts**.

   The Shortcuts page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

3. From the Tasks list, click **Copy from another app**.

4. Copy From:

   a. Copy From Application - Select an application from which to copy.

   b. Click **Next**.

5. New Shortcuts:

a. To Name - Optionally enter the new shortcut name.

b. Copy - Select an action:

- **Yes** - Copy the shortcut.
- **No** - Do not copy this shortcut.
- **Copy and Subscribe** - Copy and subscribe to this shortcut. Subscribing to a shortcut copies it to application and sets the plug-in master reference.

6. Click **Copy Shortcut**.

16.10.7 Subscribing to a Shortcut

Subscribe to a shortcut on the Shortcuts, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a shortcut by copying it and then subscribing. See Copying or Subscribing to a Shortcut.

To subscribe to a shortcut:

1. Navigate to the Shared Components page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

2. Under Other Components, click **Shortcuts**.

   The Shortcuts page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

3. Select a shortcut.

   The Shortcuts, Edit page appears.

4. To subscribe to a shortcut:

   a. Find the **Subscription** region.

   b. Subscription, Subscribe From - Select a master shortcut to subscribe to and click **Apply Changes**.

   c. Confirm your selection and click **Subscribe**.

   The subscribed component is automatically refreshed from the selected master.

5. To refresh a shortcut:

   a. Find the **Subscription** region.

   b. Subscription, Refresh Shortcut - If the shortcut is already subscribed, click **Refresh Shortcut**.

   c. Click **Apply Changes**.

6. To unsubscribe from a shortcut:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.10.8 Accessing Shortcut Reports

View the shortcut Subscription and History reports.

To access shortcut reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, select **Shortcuts**.
3. Click one of the following tabs:
   - **Subscription** - Displays the Subscribed Shortcuts page which displays all subscribed shortcuts in your application.
   - **History** - Display the Shortcut History page which displays a history of recently changed shortcuts by date.

> **Note:**
>
> The Subscription and History tabs only appear after you create a shortcut.

16.11 Using Component Groups

Use component groups to easily copy or subscribe to a collection of shared components.

- About Component Groups
  Learn about component groups.
- Creating a Component Group
  Create a component group by assigning components to the group.
- Copying and Subscribing to a Component Group
  Copy a component group from the current application or from another application. When copying a component group from another application, you can also subscribe to it.
- Refreshing a Subscribed Component Group
  Access changes in a master component group by refreshing the subscribed component group.
- Publishing a Component Group
  Publish a component group to push changes to all subscribing component groups.
- Unsubscribing from a Component Group
  Unsubscribe from a component group on the Component Group Details page.
- Deleting a Component Group
  Learn how to delete a component group.

16.11.1 About Component Groups

Learn about component groups.

A **Component Group** is a collection of other shared components. Use component groups to bulk copy, subscribe, and refresh the components in the group. You can copy and/or subscribe component groups from one application to another.

The following shared components can be part of a component group:

- Application Items
- Application Processes
- Application Computations
- Application Settings
- Build Options
- Lists
- Search Configurations
- Data Load Definitions
- REST Data Sources
- Authentication Schemes
- Authorization Schemes
- Application Access Control roles
- Email Templates
- Lists of Values
- Plug-ins
- Component Settings (native components only)
- Shortcuts
- Map Backgrounds
- Report Layouts
- Text Messages

16.11.2 Creating a Component Group

Create a component group by assigning components to the group.

The components you plan to add to the component group must already exist in the application.

To create a component group:

1. Access the Component Groups page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Other Components, click **Component Groups**.
   e. Click **Create**.

2. For Name and Comments:
   a. Name - Enter the name of the component group. The component group name must be unique within the application.
   b. Comments - Enter a brief description of this component group.
   c. Click **Create**.

   The **Component Group Details** screen appears.

To add components to the component group:

3. Click **Assign Components**.
4. Select the **Component Type** from the available options.
5. From the available components, select components for the component group using **Reset**, **Move All**, **Move**, **Remove**, and **Remove All**.

> **WARNING:**
>
> Once you select components for the component group, you **must** assign them to the group before switching the desired component type. If you select components and then switch the component type without assigning the selected components, the components are **not** added to the group.

6. Assign the components to the component group:
   - Click **Assign and Assign More** to assign the selected components, and then continue selecting more components of a different component type.
   - Click **Assign** to assign the selected components and return to the component group details screen.
   - Click **Cancel** to return to the component group details screen.

To remove components from the component group:

7. Click **Unassign from Component Group**.

To save the component group:

8. Click **Apply Changes**.

16.11.3 Copying and Subscribing to a Component Group

Copy a component group from the current application or from another application. When copying a component group from another application, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

1. Navigate to the Shared Components, Component Groups page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.

c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.

d. Under Other Components, select **Component Groups**.

   The Component Groups page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

To copy a component group within the current application:

2. Search for the component group and click **Copy** in the Copy column.

   The Copy Component Group Wizard appears.

3. New Component Group Name - Enter the name of the component group. The component group name must be unique within the application.

4. Click **Copy Component Group**.

To copy a component group from another application:

5. From the Tasks region on the Component Groups page, click **Copy from another app**.

   The Copy Component Group Wizard appears.

6. Application - Select the application from which you want to copy.

7. Component Group - Select the component group to copy.

8. Subscribe - Set this flag to **On** to subscribe to the component group.

> **Tip:**
>
> You cannot subscribe to a component group once it is copied to the new application. You can only subscribe during this step of the copy process.

9. Click **Next**.

10. Review the actions column for each component.

    - Add: A new component will be added to the target application.
    - Refresh: An existing component will be reused after refresh.
    - Replace: An existing component will be replaced with the master component.
    - Reuse: An existing component will be reused.

11. Click **Copy Component Group**.

16.11.4 Refreshing a Subscribed Component Group

Access changes in a master component group by refreshing the subscribed component group.

This procedure discusses refreshing an entire component group. To refresh an individual item in a component group, see Refreshing a Subscribed Shared Component.

If a master component group or any of the components that are part of the component group have changes, subscribed component groups have a Subscription Status of `Needs refresh`, and a **Refresh** button appears on the component group edit page. When you refresh a component group, all the components assigned to the component group with an individual component subscription status of `Needs refresh` are refreshed. The **Refresh** button does not appear if the subscription status is `Up to date`.

You can refresh a subscribed component group on the component group edit page.

To refresh a subscribed component group:

1. Navigate to the Shared Components, Component Groups page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Other Components, select **Component Groups**.

      The Component Groups page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Refresh the component group subscription:
   a. Find and access the edit page of the component group.

      If there are new changes in the master component group, the Subscription Status is `Needs refresh`.

      To access the edit page, click the component group name.
   b. On the component group edit page, find the **Subscription** region and click the **Refresh** button.
3. Review the actions column for each component.
   - Add: A new component will be added to the target application and assigned to the component group.
   - Refresh: An existing component will be refreshed.
   - Refresh and Assign: An existing component will be refreshed and assigned to the component group.
   - Replace: An existing component will be replaced with the master component and assigned to the component group.
   - Reuse: An existing component will be assigned to the component group.
   - Unassign: This component will be unassigned from the component group.
4. Click **Refresh** to update the component group.

The component group refreshes.

16.11.5 Publishing a Component Group

Publish a component group to push changes to all subscribing component groups.

In order to publish a component group, you must have already made changes to one or more of the components in the component group.

To publish a component group:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application containing the master component group.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Component Groups**.

3. Find and access the edit page of the master component group.

   By default, each component group displays as a report. To access the edit page, click the component group name.

4. (Optional) Assign or unassign components from the component group.

   For more on assigning and unassigning components, see Creating a Component Group.

To push the changes to subscribed component groups:

5. Under Subscription, click **Publish**.
6. Review the actions column for each component.
   - Add: A new component will be added to the target application and assigned to the component group.
   - Refresh: An existing component will be refreshed.
   - Refresh and Assign: An existing component will be refreshed and assigned to the component group.
   - Replace: An existing component will be replaced with the master component and assigned to the component group.
   - Reuse: An existing component will be assigned to the component group.
   - Unassign: This component will be unassigned from the component group.
7. In the Confirm Publish dialog, click **Publish** again.

   The following confirmation message appears:
   `All subscribing component groups are refreshed.`

16.11.6 Unsubscribing from a Component Group

Unsubscribe from a component group on the Component Group Details page.

Unsubscribing from a component group automatically unsubscribes all the individual components in that group.

To unsubscribe from a component group:

1. Navigate to the Component Groups page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Other Components, click **Component Groups**.

      The Component Groups page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select a component group that is currently subscribed from an application.

> **Note:**
>
> When a component group is subscribed, the **Subscribed From** column indicates the application number of the master component group.

3. In the **Subscription** region, click **Unsubscribe**.

16.11.7 Deleting a Component Group

Learn how to delete a component group.

Deleting a component group does not delete the individual shared components that make up the group. Deleting a component group deletes the group and the assignment of components to the group.

To delete a component group:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Other Components**, select **Component Groups**.

   The Component Groups page appears.
3. Click the name of the component group to be deleted.
4. On the Edit page, click **Delete**.
5. Click **Delete** to confirm.

16.12 Managing Email Templates

Enable users to send emails from your application by creating email templates.

> **Note:**
>
> Before you can send email from an application, your Instance administrator must sign in to Oracle APEX Administration Services, navigate to the Instance Settings page and configure Email attributes.

- Defining an Email Template
  Define email templates for your application. Email templates include both an HTML Format and a Plain Text Format.
- Viewing Sample Email Templates
  View the sample email templates: Load Order Details, Load Event Reminder, or Load Scheduled Outage.
- About Including Dynamic Text
  Provide dynamic text in your email templates using existing substitution modifiers.
- Editing or Deleting Email Templates
  Edit or delete existing email templates.
- Copying or Subscribing to an Email Template
  Copy an email template from the current application or from another application. When copying an email template from another application, you can also subscribe to it.

- Subscribing to Email Templates
  Subscribe to email templates on the Email Templates, Details page.
- Accessing Email Template Reports
  View the email template Utilization and History reports by clicking the appropriate tab on the Email Templates page.

> **See Also:**
>
> - Sending Email from an Application
> - Configuring Email in *Oracle APEX Administration Guide*
> - APEX_MAIL in *Oracle APEX API Reference*

16.12.1 Defining an Email Template

Define email templates for your application. Email templates include both an HTML Format and a Plain Text Format.

To define an email template:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, select **Email Templates**.

   The Email Templates page appears.
3. On the Email Templates page, click **Create Email Template**.

   The Details page appears.

   > **Tip:**
   >
   > To get started, review the sample email templates available under **Sample Templates** on the right side of the page. To load a sample email template, simply select it. See Viewing Sample Email Templates.
4. Next, define your template.

   > **Tip:**
   >
   > For **Email Subject**, **HTML Format** or **Plain Text Format**, insert substitution strings using the format `#STRING_NAME#`. You can pass in values for these strings using the Placeholder Values dialog for the process in Page Designer or the `APEX_MAIL` API.
5. Under Identification:

a. Template Name - Enter a descriptive name for this template.

b. Static Identifier - Static string identifier used to refer to the template when calling the `APEX_MAIL` APIs.

c. Email Subject - Enter the text to display for the email subject. You may use substitution strings with the `#STRING_NAME#` format. You can pass in values for these strings using Placeholder Values dialog for the process in Page Designer or the `APEX_MAIL` API.

6. Under HTML Format:

> **Tip:**
>
> You may use substitution strings with the `#STRING_NAME#` format. You can pass in values for these strings using the Placeholder Values dialog for the process in Page Designer or the `APEX_MAIL` API.

a. Header - Define the Header.

b. Body - Define the Body. You can use basic markup to format the body of your HTML email.

c. Footer - Define the Footer.

d. Advanced, HTML Template - See item Help for a list of supported substitution strings.

e. (Optional) Click **Load Default HTML**. The default HTML appears in the HTML Template. If needed, edit the default HTML Template.

7. Plain Text Format - Enter the appropriate template defaults. You may use substitution strings with the `#STRING_NAME#` format. You can pass in values for these strings using the Placeholder Values dialog for the process in Page Designer or the `APEX_MAIL` API.

8. (Optional) Comments - Enter comments that describe this template.

9. Click **Create Email Template**.

Next, create a button and process to call the `APEX_MAIL` API.

10. Add a **Send Mail** button:

a. Return to the application from which you want to send email.

b. Navigate to the appropriate page or create a new page.

c. In Page Designer, add a button to send the email.

d. Create a Process of the Identification, Type: **Send E-Mail**.

The Settings field displays configuration fields for sending email.

e. Complete the required fields. Address fields (To, Cc, Bcc, Reply To) can contain substitution strings in the format `&P2_CUST_EMAIL.` (include the period).

11. Do one of the following:

- **Apply the Email Template**

  Under Settings, Email Template, select the template you defined earlier.

  (Optional) Click the **Placeholder Values** button to open the Set Placeholder Values dialog. This dialog shows all placeholders defined in the Email Template and enables you to assign a static value or the value of a page or application item to it. In the dialog, click the LOV icon right to the "Item or Value" input field to pick a page item.

- **Apply a PL/SQL Process**

  Create a PL/SQL process which calls the `APEX_MAIL` API.

  Consider the following example:

```
begin
    apex_mail.send (
        p_to                 => 'steven.king@example.com',
        p_template_static_id => 'ORDER_CONFIRMATION',
        p_placeholders       => q'~
        {
            "CUSTOMER_NAME": "Steven King",
            "ORDER_NUMBER": 1234,
            "ORDER_DATE": "02-Feb-2018",
            "SHIP_TO": "Steven King",
            "SHIPPING_ADDRESS_LINE_1": "2004 Charade Rd",
            "SHIPPING_ADDRESS_LINE_2": "Seattle, Washinton",
            "ITEMS_ORDERED": 3,
            "ORDER_TOTAL": "$ 1,200.99",
            "ORDER_URL": "http://domain/apex/f?p=&APP_ID"
         }~' );
    apex_mail.push_queue;
end;
```

12. **Save** the page.
13. Run the page and click the **Send Mail** button.

> **See Also:**
>
> - Managing Pages in an Application
> - Managing Buttons
> - Understanding Page Processes
> - APEX_MAIL in *Oracle APEX API Reference*

16.12.2 Viewing Sample Email Templates

View the sample email templates: Load Order Details, Load Event Reminder, or Load Scheduled Outage.

To view sample email template:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, select **Email Templates**.

   The Email Templates page appears.

3. On the Email Templates page, click **Create Email Template**.

   The Details page appears.

4. On the right side of the page, find the **Sample Templates** region.

   The Sample Templates region contains three templates:

   - Load Order Details
   - Load Event Reminder
   - Load Scheduled Outage

5. Select a template. For example, select **Load Order Details**.

   The template appears.

6. Expand the **Sample API Usage** region at the bottom of the page to see an example. The following is the **Load Order Details** API example.

```
begin
    apex_mail.send (
        p_to                 => email_address_of_user,
        p_template_static_id => '',
        p_placeholders       => '{'
        '    "CUSTOMER_NAME":'           ||
apex_json.stringify( some_value )
        '   ,"ITEMS_ORDERED":'           ||
apex_json.stringify( some_value )
        '   ,"MY_APPLICATION_LINK":'     ||
apex_json.stringify( some_value )
        '   ,"ORDER_DATE":'              ||
apex_json.stringify( some_value )
        '   ,"ORDER_NUMBER":'            ||
apex_json.stringify( some_value )
        '   ,"ORDER_TOTAL":'             ||
apex_json.stringify( some_value )
        '   ,"ORDER_URL":'               ||
apex_json.stringify( some_value )
        '   ,"SHIPPING_ADDRESS_LINE_1":' ||
apex_json.stringify( some_value )
        '   ,"SHIPPING_ADDRESS_LINE_2":' ||
apex_json.stringify( some_value )
        '   ,"SHIP_TO":'                 ||
apex_json.stringify( some_value )
        '}' );
end;
```

7. See Defining an Email Template for information edting attributes, creating a template, and creating a **Send Mail** button.

16.12.3 About Including Dynamic Text

Provide dynamic text in your email templates using existing substitution modifiers.

> **Note:**
>
> Template directives are supported by specific attributes of email templates. See Using Template Directives.

For security reasons, the values substituted for the placeholders (`#NAME#`) in the email templates are automatically escaped based on the context. In other words, HTML formatted templates include HTML escaping and Plain Text templates do not. If this is not the desired behavior for your environment, you can provide dynamic text which contains HTML tags using the existing substitution modifiers:

- `#NAME!RAW#` - Text is substituted as is.
- `#NAME!STRIPHTML#` - All HTML tags are removed.

Extending the Email Templates

For example, you can extend the **HTML Format - Body** template with:

```
<tr>
        <th align="left">Additional Information</th>
        <td>#ADDITIONAL_INFO!RAW#</td>
</tr>
```

Similarly, you can extend the **Plain Text Format** template with:

```
  Additional Info:  #ADDITIONAL_INFO!STRIPHTML#
```

Sample PL/SQL

The PL/SQL code would look similar to this:

```
begin
    apex_mail.send (
        p_to                 => 'steven.king@example.com',
        p_template_static_id => 'ORDER_CONFIRMATION',
        p_placeholders       => q'~
        {
            "CUSTOMER_NAME": "Steven King",
            "ORDER_NUMBER": 1234,
            "ORDER_DATE": "02-Feb-2018",
            "SHIP_TO": "Steven King",
            "SHIPPING_ADDRESS_LINE_1": "2004 Charade Rd",
            "SHIPPING_ADDRESS_LINE_2": "Seattle, Washinton",
            "ITEMS_ORDERED": 3,
            "ORDER_TOTAL": "$ 1,200.99",
            "ORDER_URL": "http://domain/apex/f?p=&APP_ID"
               "ADDITIONAL_INFO": "We plan to deliver your ordered items
<strong>tomorrow morning between 08:00 and 09:00</strong>
         }~' );
    apex_mail.push_queue;
end;
```

16.12.4 Editing or Deleting Email Templates

Edit or delete existing email templates.

To edit or delete an email template:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, select **Email Templates**.

   The Email Templates page appears.
3. To edit an email template:
   a. Click the email template name.

      The Template Details appear.
   b. Edit the appropriate attributes.
   c. Click **Apply Changes**.
4. To delete an email template:
   a. Click the email template name.

      The Template Details appear.
   b. Click **Delete**.
   c. Confirm your selection and click **Delete**.

16.12.5 Copying or Subscribing to an Email Template

Copy an email template from the current application or from another application. When copying an email template from another application, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy an email template:

1. Navigate to the Shared Components, Email Templates page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under User Interface, select **Email Templates**.

The Email Templates page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. To copy an email template within the current application:
   a. Search for the email template and click **Copy** in the Copy column.

      The Copy Email Template Wizard appears.
   b. New Name - Provide a name for the copied email template.
   c. New Static Identifier - Provide a unique static identifier for the copied email template.
   d. Click **Copy Email Template**.
3. To copy an email template from another application:
   a. From the Tasks region on the Email Templates page, click **Copy from another app**.

      The Copy Email Template Wizard appears.
   b. Application - Select the application from which you want to copy.
   c. Copy Email Template - Select the email template to copy.
   d. Subscribe - Enable this flag to subscribe to the email template.
   e. Click **Copy Email Template**.

16.12.6 Subscribing to Email Templates

Subscribe to email templates on the Email Templates, Details page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an email template by copying it and enabling the Subscribe option. See Copying or Subscribing to an Email Template.

To subscribe to email templates in the current application:

1. Navigate to the Email Templates page:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Navigation and Search, click **Email Templates**.

      The Email Templates page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an email template.

   The Email Templates, Details page appears.
3. To subscribe to an email template:
   a. Find the **Subscription** region.

   b. Subscription, Subscribe From - Select a master email template to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.

4. To refresh a subscription:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Email Template - If the email template is already subscribed, click **Refresh Email Template**.
   c. Click **Apply Changes**.
5. To unsubscribe from an email template:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

16.12.7 Accessing Email Template Reports

View the email template Utilization and History reports by clicking the appropriate tab on the Email Templates page.

To view email template reports:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, select **Email Templates**.

   The Email Templates page appears.
3. Click the appropriate tab:
   - **Utilization** - Displays where email templates are used within the application.
   - **History** - Displays recent modifications made to email templates in this application.

16.13 Managing Static Application Files

Manage static application files in the Oracle database on the Shared Components, Static Application Files page or in remote storage solution such as OCI Object Storage.

Static application files include images, CSS files, and any other files associated with an application that must be managed independently.

- Specifying the Location of Static Application Files
  Specify the location of static application files.
- Referencing Static Application Files
  Reference a static application file in your application using `#APP_FILES#` substitution string.
- Managing Static Application Files in the Database
  Manage static application files stored locally in the database in Shared Components.

- Managing Static Application Files in Remote Storage
  Manage static application files in Oracle Cloud Infrastructure Object Storage service.

> **See Also:**
>
> - Managing Static Workspace Files
> - About Managing Images

16.13.1 Specifying the Location of Static Application Files

Specify the location of static application files.

Application files are stored with your application definition in the Oracle database. For performance reasons, you can also store your application files on your Web server. Edit the #APP_FILES# Path attribute to change the location.

To edit the #APP_FILES# Path attribute:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under User Interface, select **User Interface Attributes**.

   The User Interface page appears.
3. Click the **Advanced** tab:
   - #APP_FILES# Path - Determines the virtual path the Web server uses to point to the static files when using the `#APP_FILES#` substitution string.

     Do not specify anything to reference files which are stored with your application definition in the database.

     For performance reasons you can also store your application files on your Web server or a Content Delivery Network (CDN). Use any valid URL to reference them.

     Examples:

     `/myFiles/`

     `https://www.example.com/myFiles/`
4. Click **Apply Changes**.

16.13.2 Referencing Static Application Files

Reference a static application file in your application using `#APP_FILES#` substitution string.

Static application files are only available to the currently selected application. To view sample syntax for uploaded images, see the Reference column on the Static Application Files report.

> **See Also:**
>
> Using Substitution Strings

16.13.3 Managing Static Application Files in the Database

Manage static application files stored locally in the database in Shared Components.

- Adding Static Application Files to the Database
  Add static application files to the Oracle Database from Shared Components.
- Viewing Static Application Files
  View static application files uploaded to the database on the Static Application Files page.
- Downloading Static Application Files
  Download individual static application files added to the database, or download all static application files in a Zip.
- Deleting Static Application Files
  Delete static application files uploaded to the database.

16.13.3.1 Adding Static Application Files to the Database

Add static application files to the Oracle Database from Shared Components.

To add static application files to the Oracle database, navigate to Shared Components, Static Application Files, and click **Create File**.

To create a static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.
3. On Static Application Files page, click **Create File**.

   The Create page appears.
4. On the Create page:
   a. Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`.

      If no directory is specified, the file is stored in the root directory.
   b. File Name - If creating a blank file (for example, `script.js`), enter the name of the file.
   c. Content - If uploading a file, do one of the following:
      - Drag and drop the file to the Content region.
      - Click the Content region and select the file.
   d. Click **Create** or **Create and Create Another**.

16.13.3.2 Viewing Static Application Files

View static application files uploaded to the database on the Static Application Files page.

To view static application files added to the database:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. If not already selected, click **View Report** icon.

   A report appears. To customize the report, use the Search bar at the top of the page.

   The Static Application Files report includes the following columns: Name, Mime Type, Size, Reference, File, Updated, and Updated By.
4. To view or edit file details, click the file Name. On the Edit page:
   a. Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`.

      If no directory is specified, the file is stored in the root directory.
   b. File Name - Edit the name of the file.
   c. To save your changes, click **Save Changes**.

16.13.3.3 Downloading Static Application Files

Download individual static application files added to the database, or download all static application files in a Zip.

- Downloading a Static Application File
  Download an uploaded static application file.
- Downloading All Static Application Files in a Zip
  Learn how to download all static application files in a zip.

16.13.3.3.1 Downloading a Static Application File

Download an uploaded static application file.

To download an uploaded static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.

3. If not already selected, click **View Report** icon.

   A report appears. To customize the report, use the Search bar at the top of the page.
4. Locate the file to be downloaded and click the **Download** link.

16.13.3.3.2 Downloading All Static Application Files in a Zip

Learn how to download all static application files in a zip.

To download all static application files in a Zip:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. Click **Download as Zip**.

16.13.3.4 Deleting Static Application Files

Delete static application files uploaded to the database.

- Deleting a Static Application File
  Delete an uploaded static application file on the Static Application Files, Edit page.
- Deleting Multiple or All Static Application Files
  Delete multiple or all uploaded static application files.

16.13.3.4.1 Deleting a Static Application File

Delete an uploaded static application file on the Static Application Files, Edit page.

To delete an uploaded static application file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. If not already selected, click **View Report** icon. To customize the report, use the Search bar at the top of the page.
4. Select the file to be deleted and click **Delete**.

> **Tip:**
>
> You can also click the name of the file to be deleted. On the Edit page, click **Delete**.

16.13.3.4.2 Deleting Multiple or All Static Application Files

Delete multiple or all uploaded static application files.

To delete multiple files or all static application files:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. If not already selected, click **View Report** icon. To customize the report, use the Search bar at the top of the page.
4. To delete multiple files at once, select the files to be deleted and click **Delete Checked**.
5. To delete all files, click **Delete All**.

16.13.4 Managing Static Application Files in Remote Storage

Manage static application files in Oracle Cloud Infrastructure Object Storage service.

About OCI Object Storage Service

Oracle Cloud Infrastructure (OCI) offers high-performance compute capabilities (such as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network. If you have OCI account, you can store static application files in the OCI Object Storage service. OCI Object Storage stores objects in logical containers called **buckets**. Each bucket is associated with a single **compartment** that has policies that determine the actions users can perform on all objects in the bucket.

> **Tip:**
>
> To learn more about OCI Object Storage, see Object Storage.

- Uploading Files to Remote Storage
  Store static application files in Oracle Cloud Infrastructure (OCI) Object Storage service.
- Viewing or Downloading Files in Remote Storage
  View or download static application files loaded to Object Storage on the Object Storage Files page.
- Synchronizing Files in Remote Storage
  Synchronize static application files stored in OCI Object Storage with those uploaded to the database.

16.13.4.1 Uploading Files to Remote Storage

Store static application files in Oracle Cloud Infrastructure (OCI) Object Storage service.

Prior to uploading static application files to OCI Object Storage, you must:

1. **Signup for an OCI account**. If you have OCI account, you can store static application files in the OCI Object Storage service. Begin the signup process, by reviewing Oracle Cloud Infrastructure Free Tier. To start your signup, go to https://signup.oraclecloud.com/.
2. **Create OCI Credentials**. See Managing Web Credentials.

To upload objects to OCI Object Storage:

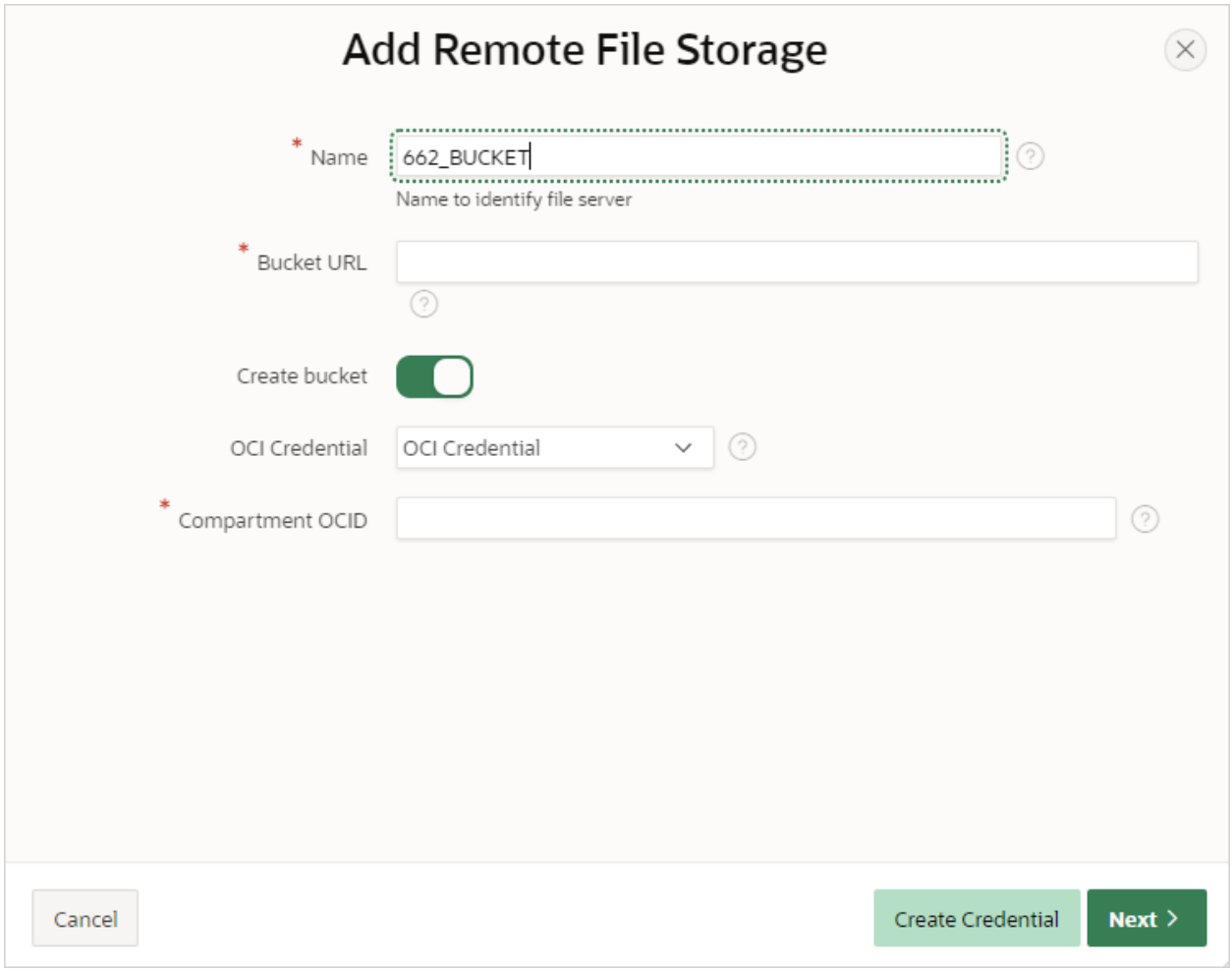
1. Access the Application Definition page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.

      The Application home page appears.
   c. Click the **Edit Application Definition** button.

      The Edit Application Definition page appears.
2. On the Edit Application Definition page, click **Add Object Storage**.

> **Note:**
>
> If you have not yet created an OCI Credential, a warning appears. You must create an OCI Credentials before you can proceed. To learn more, see Managing Web Credentials

The Add Remote File Storage Wizard appears.



3. In the Add Remote File Storage Wizard:
   a. Name - Enter a name to identify the file server.
   b. Bucket URL - Enter the endpoint URL of the file storage bucket.

      Use the following format:

      `https://objectstorage.region.oraclecloud.com/n/tenancy_namespace/b/bucket_name`
   c. Create Bucket - If enabled, the wizard creates a bucket for you.
   d. OCI Credential - Select an OCI Web Credential from the list.

      If you have not yet created the appropriate OCI Credential, click **Create Credential**.
   e. Compartment OCID - Enter the OCID of the compartment for this bucket. The compartment OCID can be found in the Oracle Cloud Infrastructure (OCI) console under Identity, Compartments.
   f. Click **Next**.

   A Success message appears.
4. On the Add Remote File Storage Success message:
   - **Migrate Files to Object Storage** - If enabled, existing static application files in the database are automatically migrated to OCI Object Storage.

If **Migrate Files to Object Storage** is disabled, you can migrate static application files manually using the **Synchronize Files** option on the Shared Components, Static Application Files page. See Synchronizing Files in Remote Storage.

- Click **Finish**.

The Edit Application Definition page appears. If you enabled **Migrate Files to Object Storage**, the following message displays:

```
Files migrated to Object Storage.
```

The Edit Application Definition, Application File Storage region displays the following:

- File Storage - Displays **Oracle Cloud Object Storage**.
- Remote File Server - Displays the name of the file server you specified in the Add Remote File Storage Wizard.

16.13.4.2 Viewing or Downloading Files in Remote Storage

View or download static application files loaded to Object Storage on the Object Storage Files page.

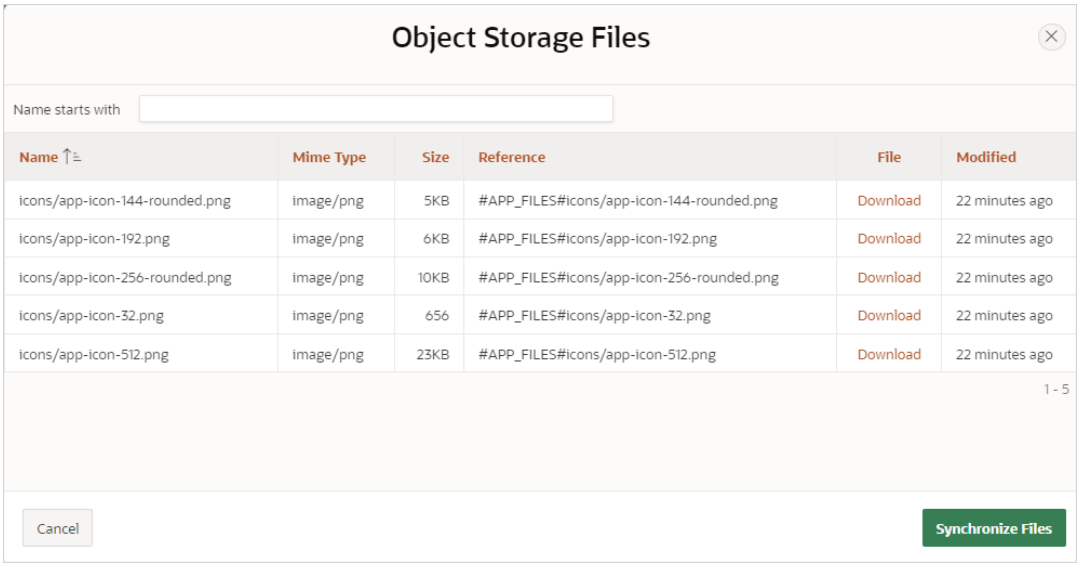
To view static application files uploaded to OCI Object Storage:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. Under Tasks, click **View Object Storage Files**.

   Object Storage Files page appears.



4. To narrow the display, enter keywords in the **Name starts with** field.

To show all files again, delete any text entered in the **Use Name starts** field and press **Enter**.

> **Tip:**
>
> The first 1000 files in Object Storage are retrieved for this report. If the file or files you want to synchronize does not appear in this report, or you want to narrow the number of files considered for the synchronize operation, use **Name starts with** to narrow the results.

5. To download a file, click the Download link adjacent to the appropriate file name.
6. To synchronize the files, click **Synchronize Files**. See Synchronizing Files in Remote Storage.

16.13.4.3 Synchronizing Files in Remote Storage

Synchronize static application files stored in OCI Object Storage with those uploaded to the database.

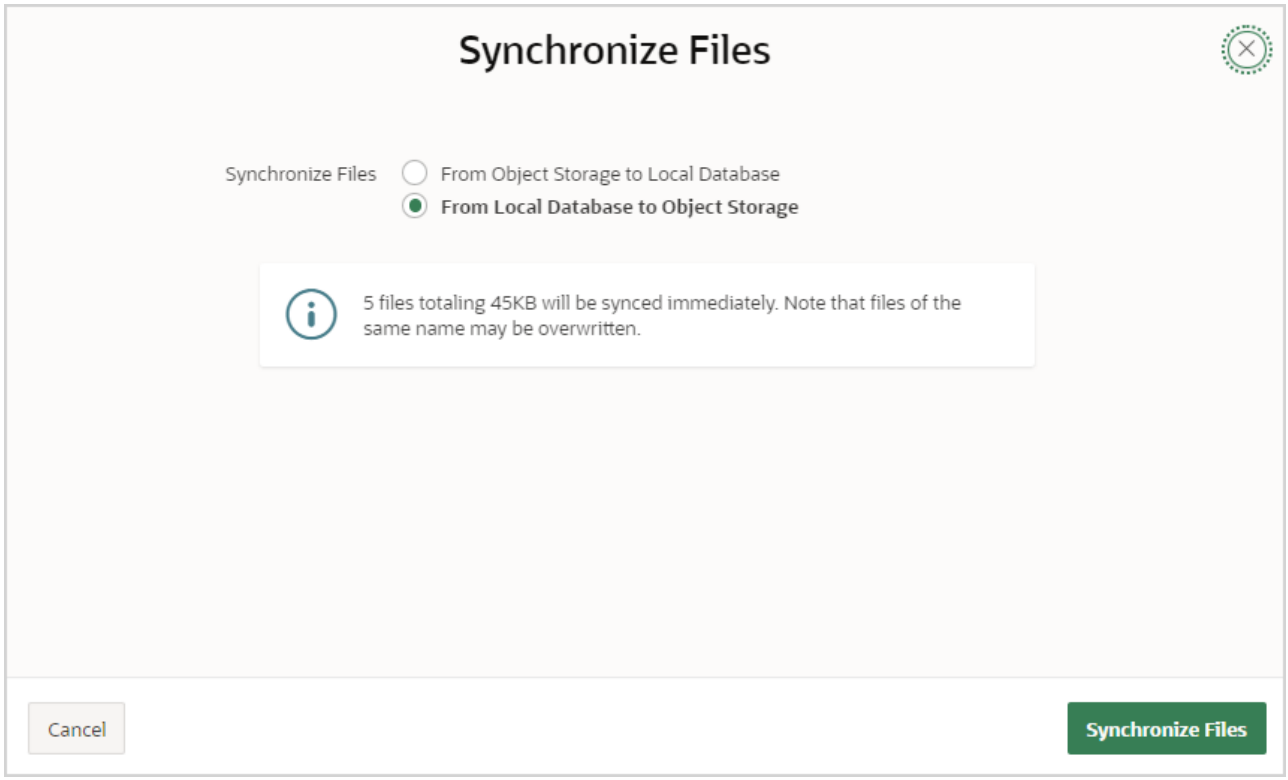
To synchronize static application files uploaded to OCI Object Storage:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Application Files**.

   The Static Application Files page appears.
3. Under Tasks, click **Synchronize Files**.

   The Synchronize Files page appears.



4. For Synchronize Files - Select the synchronization direction. Options include:
   - **From Object Storage to Local Database**
   - **From Local Database to Object Storage**
5. Click **Synchronize Files**.

16.14 Managing Static Workspace Files

Navigate to Shared Components to upload, view, download, and delete static workspace files (including images, CSS files, and other files which must be managed independently).

- About Referencing Static Workspace Files
  Reference a static workspace file in your application using `#WORKSPACE_FILES#` substitution string.
- Creating Static Workspace Files
  Create static workspace files from Shared Components.
- Viewing the Static Workspace File Report
  View the Static Workspace Files report.
- Downloading Static Workspace Files
  Download an uploaded static workspace file, or download all static application files in a Zip.
- Deleting Static Workspace Files
  Delete uploaded static workspace files.

> **See Also:**
>
> - Managing Static Application Files
> - About Managing Images

16.14.1 About Referencing Static Workspace Files

Reference a static workspace file in your application using `#WORKSPACE_FILES#` substitution string.

You can reference static workspace files in any application in the current workspace. To view sample syntax for uploaded images, see the Reference column on the Static Workspace Files report.

> **See Also:**
>
> Using Built-in Substitution Strings

16.14.2 Creating Static Workspace Files

Create static workspace files from Shared Components.

To create a static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.
3. To upload a file, click **Create File**.
4. On the Create page:
   a. Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`.

      If no directory is specified, the file is stored in the root directory.
   b. File Name - If creating a blank file (for example, `script.js`), enter the name of the file.
   c. Content - If uploading a file, do one of the following:
      - Drag and drop the file to the Content region.
      - Click the Content region and select the file.
   d. Click **Create** or **Create and Create Another**.

16.14.3 Viewing the Static Workspace File Report

View the Static Workspace Files report.

To view the static workspace file as a report:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.
3. If not already selected, click **View Report** icon.

   A report appears. To customize the report, use the Search bar at the top of the page.

   The Static Application Files report includes the following columns: Name, Mime Type, Size, Reference, File, Updated, and Updated By.
4. To view or edit file details, click the file Name. On the Edit page:
   a. Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`.

      If no directory is specified, the file is stored in the root directory.
   b. File Name - Edit the name of the file.
   c. To save your changes, click **Save Changes**.

16.14.4 Downloading Static Workspace Files

Download an uploaded static workspace file, or download all static application files in a Zip.

- Downloading a Static Workspace File
  Learn how to download a static workspace file.
- Downloading All Static Workspace Files in a Zip
  Learn how to download all static workspace files in a Zip.

16.14.4.1 Downloading a Static Workspace File

Learn how to download a static workspace file.

To download an uploaded static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.

3. If not already selected, click **View Report** icon.

   A report appears.
4. Locate the file to be downloaded and click the **Download** link.

16.14.4.2 Downloading All Static Workspace Files in a Zip

Learn how to download all static workspace files in a Zip.

To download all static workspace files in a Zip:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.
3. Click **Download as Zip**.

16.14.5 Deleting Static Workspace Files

Delete uploaded static workspace files.

- Deleting a Static Workspace File
  Learn how to delete a static workspace file.
- Deleting Multiple or All Static Workspace Files
  Learn how to delete multiple or all static workspace files.

16.14.5.1 Deleting a Static Workspace File

Learn how to delete a static workspace file.

To delete an uploaded static workspace file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.
3. If not already selected, click the **View Report** icon.

   A report appears. To customize the report, use the Search bar at the top of the page.
4. Click the name of the file to be deleted.
5. On the Edit page, click **Delete**.

16.14.5.2 Deleting Multiple or All Static Workspace Files

Learn how to delete multiple or all static workspace files.

To delete multiple files or all static workspace files:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, select **Static Workspace Files**.

   The Static Workspace Files page appears.
3. To delete multiple files at once, select the files to be deleted and click **Delete Checked**.
4. To delete all files, click **Delete All**.

16.15 Printing a Report Region by Defining a Report Query

Print a report region by defining a report query.

- About Report Queries
  Learn about report queries.
- Creating a Report Query
  Create a report query in Shared Components.
- Editing a Report Query
  Edit a report query in Shared Components.
- Copying a Report Query
  Copy a report query in Shared Components.
- Viewing Report Query Utilization
  View the Report Query Utilization report.
- Viewing Report Query History
  View the Report Query History report.

16.15.1 About Report Queries

Learn about report queries.

In Oracle APEX, a **report query** is a SQL query that is used to fetch data to be displayed in a report. The data source can be a table, a SQL query, or a REST Data Source. The report query defines the columns to be included in the report, the data source, and any conditions or sorting to be applied to the data. Note that report queries must be SQL statements. Functions returning SQL statements are not supported.

You can print a report region by defining a report query as a Shared Component.

To download a report query as a formatted document, you associate it with a **report layout**. If no report layout is selected, APEX uses a generic layout. You can use the generic layout to test and verify a report query. When using the generic layout option, and when multiple source queries are defined, only the first result set is included in the print document. The reports can include session state of the current application.

The following Print Server Types do not support generic report layouts:

- Oracle Document Generator Pre-built Function
- APEX Office Print

To make these reports available to users, you then integrate them with an application. For example, you can associate a report query with a button, list item, branch, or other navigational component that enables you to use URLs as targets. Selecting that item then initiates the printing process.

16.15.2 Creating a Report Query

Create a report query in Shared Components.

The availability of the report query options depends on how your service administrator configures report printing for your instance. All options described in these steps may not be available to you.

To create a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, click **Report Queries**.
3. Click **Create**.
4. In Create Report Query, specify the following:
   a. Report Query Name - Enter a name to identify the report query. When referencing a report query as a link target, the report query name is part of the request string.
   b. Report Layout - Select the Report Layout. If you have not yet created a Report Layout, you can specify one later.
   c. Click **Next**.
5. In Report Query Definition, specify the following:
   a. Output Format - Select the report output format. Valid options include:
      - Derive from Item - The output format is derived from an item at runtime.
      - PDF - Adobe Portable Document Format.
      - Word - Microsoft Word Rich Text Format.
      - Excel - Microsoft Excel format.
      - HTML
      - XML - Extensible Markup Language.

      Additional options may be available depending on the print server type. For example, AOP also supports Powerpoint, Markdown, CSV, Open Document Text, Open Document Spreadsheet, and Open Document Presentation. Document Generator supports PDF, DOCX, and XLSX formats.
   b. (Optional) Item - If using Derive from Item, select the item that determines the output format at runtime.

c. View File As - Select how the web browser opens the report query document:
   - **Attachment** - Displays a File Open/Save dialog box when the user clicks the report query download button.
   - **Inline** - Displays the report query document inside the browser window.

d. Include Application and Session Information - Enable this option to reference session state values, such as the values of page items, in the output of your report query.

e. Click **Next**.

6. In Report Query Source, select the source:
   - **Local Database** - Data is sourced from a local database.
   - **REST Enabled SQL** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to **Shared Components**, **REST Enabled SQL Databases**.
   - **REST Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to **Shared Components**, **REST Data Sources**.
   - **Duality View** - Data is sourced from a Duality View Source defined using **Duality Views** within Shared Components.
   - **JSON Source** - Data is sourced from a JSON Source defined using **JSON Sources** within Shared Components. JSON Sources include tables with JSON columns or JSON Collection Tables.

   The UI changes based on the data source you select.

7. Set the remaining options. For example, if you selected **Local Database** in the previous step, set the following:

   a. Source Type - Specify the source type. For example, Table.

   b. Table/View Owner - Select the owner of the table that you are building the search configuration on.

   c. Table/View Name - Select the table or view that the search configuration is based on.

8. Click **Next**.

9. (Optional) Click **Add Query** to add additional source queries.

10. Click **Next**.

11. Confirm displays the following information:

    a. Query Name - Identifies the query.

    b. Report Layout - Identifies the report layout you selected.

    c. Output Format - Identifies the format for this report query.

    d. Derive from Item - Identifies the item that holds the format information.

    e. URL - Report queries can be integrated with an application by using the URL shown as the target for buttons, navigation list entries, list items, or any other type of link. Click into the URL field to copy the URL.

12. Click **Create**.

    The Report Query is created and saved to Shared Components.

> **Caution:**
>
> If you are using **Document Generator** or **APEX Office Print**, it is important to edit the report query to specify the **Data Loop Name**. The data loop name and the report query source must match.

> **See Also:**
>
> About Report Printing

16.15.3 Editing a Report Query

Edit a report query in Shared Components.

To edit a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, click **Report Queries**.
3. Select the appropriate report query.
4. On the Edit page, edit the appropriate attributes.

   > **Caution:**
   >
   > If you are using **Document Generator** or **APEX Office Print**, it is important to specify the **Data Loop Name**. The data loop name and the report query source must match.
5. Click **Apply Changes**.

16.15.4 Copying a Report Query

Copy a report query in Shared Components.

To copy a report query:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under **Files and Reports**, click **Report Queries**.
3. On the Report Queries page, click **Copy**.
4. On the Copy Report Query, select the query you want to copy, enter a name for the report query, and click **Copy**.

   The copy appears in the query list.

16.15.5 Viewing Report Query Utilization

View the Report Query Utilization report.

To view the Report Query Utilization report:

1. Navigate to the Report Query page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Files and Reports, select **Report Queries**.
2. On the Report Queries page, click **Utilization**.

   The Utilization page displays page processes and dynamic actions that print the select Report Query in the current application.

16.15.6 Viewing Report Query History

View the Report Query History report.

To view the Report Query History report:

1. Navigate to the Report Query page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Files and Reports, select **Report Queries**.
2. On the Report Queries page, click **History**.

   The History page displays recent modifications made to Report Queries in the current workspace.

16.16 Formatting a Report Region or Report Query Using Report Layouts

Create a report layout to format a report region or report layout.

- About Report Layouts
  Learn about report layouts.
- About Report Layout Options
  Learn about report layout options.
- Creating a Report Layout
  Create a report layout in Shared Components.

- Editing a Report Layout
  Edit a report layout in Shared Components.
- Copying or Subscribing to a Report Layout
  Copy a report layout from the current application or from another application. When copying a report layout from another application, you can also subscribe to it.
- Subscribing to Email Templates
  Subscribe to email templates on the Email Templates, Details page.

16.16.1 About Report Layouts

Learn about report layouts.

To format either a classic report region, interactive report region, or report query, you associate it with a report layout. Using report layouts renders the data in a printer-friendly format. If you do not select a report layout, a default XSL-FO layout is used.

Depending on the print server you use, various report layouts are acceptable.

Print Server Type	Layout Types
Oracle Document Generator Pre-built Function	DOCX, XLSX
Oracle Analytics (BI) Publisher	RTF, XSL-FO
APEX Office Print	DOCX, XLSX, PPTX, HTML, Markdown, CSV, TXT, ODS, ODT, ODP
External (Apache FOP)	XSL-FO

Note that Oracle Document Generator Pre-built Function and APEX Office Print do not support XSL-FO layouts.

> **See Also:**
>
> Configuring Report Printing in *Oracle APEX Administration Guide*

16.16.2 About Report Layout Options

Learn about report layout options.

You can create a report layout based on one of these options:

- **Generic Columns** - A generic report layout works with most query result sets. With this layout, the number of columns is automatically adjusted when generating the printable document.

  Many report layout attributes can be defined declaratively for report regions using the built-in XSL-FO default layout. This step allows for creating customizable copies of the built-in default XSL-FO layout, if additional control over the report layout is needed.
- **Named Columns** - A named column report layout is a query-specific report layout designed to work with a defined list of columns in the query result set. This type of layout is used for custom-designed layouts when precise control of the positioning of page items and query columns is required.

Note that the availability of the Report Layout options depends on how your site administrator configured the report printing settings at your site. All options described in these steps may not be available to you. The following Print Server Types do not support Generic Columns:

- Oracle Document Generator Pre-built Function
- APEX Office Print

> **See Also:**
>
> Configuring Report Printing in *Oracle APEX Administration Guide*

16.16.3 Creating a Report Layout

Create a report layout in Shared Components.

To create a report layout:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, click **Report Layouts**.
3. Click **Create**.

   The Create Report Layout wizard appears.
4. For Layout Name, enter a name for the Report Layout.
5. For Report Layout Type, select an option and click **Next**:
   - Generic Columns (XSL-FO) - Uses the default template to populate the report. In the next step, you can customize the default code.
   - Named Columns (XSL-FO) - Requires that you upload an XSL-FO or RTF file in the next step.

   The options that appear on the next page depend on the layout type you select.
6. If you selected **Generic Columns**:
   a. Report Layout - The report layout is the XSL-FO based definition of the page formatting. All attributes defining page size, orientation, fonts, styles, and so on, are defined in this section.
   b. Report Column Heading - Defines the look of each cell in the report heading row.
7. If you selected **Named Columns**, upload the file containing the report layout.
8. Click **Create Layout**.

16.16.4 Editing a Report Layout

Edit a report layout in Shared Components.

You can edit a generic column report layout directly in Oracle APEX. However, to edit a named column report layout, you must download the current file, edit it, and then upload it again.

To edit a report layout:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under **Files and Reports**, click **Report Layouts**.
3. On the Report Layouts page, select the layout you want to edit.
4. For **generic column layouts**, edit the layout directly on the Edit Report Layout page and click **Apply Changes**.
5. For **named column layouts**:
   a. Click **Download** and save the file to your computer.
   b. Edit the file.
   c. Upload the updated version of the file as a new report layout.
   d. For Oracle Document Generator Pre-built Function and APEX Office Print, specify the **Data Loop Name**.

      The data loop name is used as the name for the array when APEX prepares the JSON data. It is only used when printing Classic Report regions and Interactive Report regions, and only for named column layouts that are not RTF or XSL-FO.
   e. Click **Apply Changes**.

16.16.5 Copying or Subscribing to a Report Layout

Copy a report layout from the current application or from another application. When copying a report layout from another application, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy a report layout:

1. Navigate to the Shared Components, Report Layouts page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

d. Under Files and Reports, select **Report Layouts**.

The Report Layouts page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. To copy a report layout within the current application:

a. Search for the report layout and click **Copy** in the Copy column.

The Copy Report Layout Wizard appears.

b. New Report Layout Name - Enter the name of the report layout. Report layout name must be unique within the application.

c. Click **Copy Report Layout**.

3. To copy a report layout from another application:

a. From the Tasks region on the Report Layouts page, click **Copy from another app**.

The Copy Report Layout Wizard appears.

b. Application - Select the application from which you want to copy.

c. Copy Report Layout - Select the report layout to copy.

d. Subscribe - Set this flag to **On** to subscribe to the report layout.

e. Click **Copy Report Layout**.

16.16.6 Subscribing to Email Templates

Subscribe to email templates on the Email Templates, Details page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an email template by copying it and enabling the Subscribe option. See Copying or Subscribing to an Email Template.

To subscribe to email templates in the current application:

1. Navigate to the Email Templates page:

a. Navigate to the Workspace home page.

b. Click the **App Builder** icon.

c. Select an application.

d. On the Application home page, click **Shared Components**.

e. Under Navigation and Search, click **Email Templates**.

The Email Templates page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. Select an email template.

The Email Templates, Details page appears.

3. To subscribe to an email template:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master email template to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscription:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Email Template - If the email template is already subscribed, click **Refresh Email Template**.
   c. Click **Apply Changes**.
5. To unsubscribe from an email template:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

17
Upgrading Oracle APEX Applications

Installing a new release of APEX updates existing applications to the latest release, but does not alter application user interface or application components.

Once you upgrade an APEX instance from a previous release, existing applications will work without modification. However, to keep applications maintainable, up-to-date, and to leverage new functionality, perform the following steps after an upgrade or after an application has been imported which was exported from a previous release.

> **Tip:**
>
> At the completion of each task described in this section, Oracle recommends running and thoroughly testing your application.

- Reviewing Release Notes
  After you upgrade from a previous APEX release, review the *Oracle APEX Release Notes*.
- Refreshing the Universal Theme
  Update an application to use the latest Universal Theme to take advantage of new features, enhancements, and bug fixes as well as styling for new APEX components.
- Updating App Compatibility Mode
  The Compatibility Mode attribute controls the backward compatibility of the APEX runtime engine when executing an application.
- Running Application Upgrade
  After upgrading to a new release, the Application Upgrade wizard can help you to bulk update your application to leverage new features instead of deprecated ones or to automatically enable new functionality for existing components.
- Reviewing Old JavaScript
  For applications using jQuery Migrate or legacy JavaScript, review the User Interface, JavaScript settings.

> **See Also:**
>
> Upgrading from a Previous Oracle APEX Release in *Oracle APEX Installation Guide*

17.1 Reviewing Release Notes

After you upgrade from a previous APEX release, review the *Oracle APEX Release Notes*.

Oracle APEX Release Notes contain important information not included in the APEX documentation. Key sections to review include:

- Changed Behavior - Describes changes in application and component behavior compared to previous releases.

- Deprecated Features - Lists features which Oracle plans to desupport or remove in a future APEX release. If a feature is related to application metadata or an API, existing applications can still use the feature, but Oracle strongly recommends that developers start to modify their applications as described in this section. Use APEX Advisor to scan existing applications for deprecated attributes.
- Deprecated in Previous Releases - Lists features which were deprecated in a previous release but have not been fully removed or desupported from APEX. Oracle strongly recommends that developers start to modify their applications as described in this section. Run APEX Advisor to scan existing applications for deprecated attributes.
- Desupported Features - Lists features that are no longer available. If a desupported feature is related to application metadata or APIs, then existing applications may not work as they did previously. Oracle recommends modifying the application to replace the feature.
- Desupported in Previous Releases- Lists features that were desupported in a previous release.

> **See Also:**
>
> Running Advisor to Check Application Integrity

17.2 Refreshing the Universal Theme

Update an application to use the latest Universal Theme to take advantage of new features, enhancements, and bug fixes as well as styling for new APEX components.

> **Tip:**
>
> Before starting this procedure, Oracle recommends backing up your application by either exporting it or creating a copy. See Exporting an Application and Copying an Application.

To update an application to use the latest Universal Theme

1. Navigate to the App Builder home page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

      If the Universal Theme is not up-to-date, the App Builder home page displays a banner at the top of the page that reads:

      ```
      There is a new version of Universal Theme available! You can refresh your
      theme to take advantage of new features, enhancements, and bug fixes.
      ```
2. Click **Refresh Theme**.
3. Run and test your application.

> **See Also:**
>
> Migration Guide page in the Universal Theme Reference app.

17.3 Updating App Compatibility Mode

The Compatibility Mode attribute controls the backward compatibility of the APEX runtime engine when executing an application.

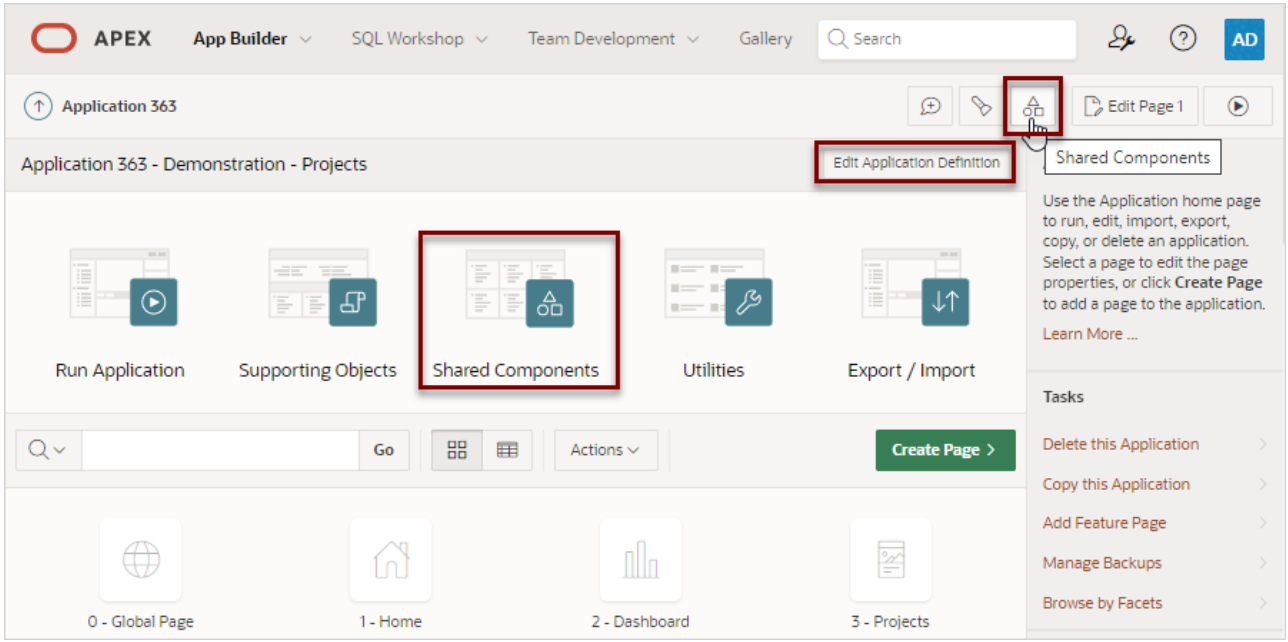
Certain runtime behaviors change from release to release. To not break or change the behavior of existing applications, this attribute is automatically set during upgrade if necessary.

To keep applications maintainable, set **Compatibility Mode** to the latest release after reviewing the changed behavior in the Release Notes and making the necessary changes.

To update the Compatibility Mode within an application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.



3. From Application home page, you can access the Edit Application Definition page in two ways:
   - Click the **Edit Application Definition** button.
   - From Shared Components:
     - Click **Shared Components**.
     - Under Application Logic, click **Application Definition**.

   The Edit Application Definition page appears.
4. Click the **Definition** tab.

5. **Properties**, **Compatibility Mode** - To use standard behavior in an application, set the compatibility mode of the application to the current version.
6. Click **Apply Changes** to save your changes.
7. Run and test your application.

> **See Also:**
>
> **Changed Behavior**, Compatibility Mode in *Oracle APEX Release Notes*

17.4 Running Application Upgrade

After upgrading to a new release, the Application Upgrade wizard can help you to bulk update your application to leverage new features instead of deprecated ones or to automatically enable new functionality for existing components.

The Upgrade Application page displays the available upgrade types and the number of objects that can be upgraded. Click on the number under **Candidate Object** to review the candidates and upgrade.

To upgrade an application to use new components:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.

   The Application home page appears.
4. Click **Utilities**.
5. Click **Upgrade Application**.

   The Upgrade Application page appears displaying the available upgrade types and the number of objects that can be upgraded.
6. To review the candidate objects, click the number link under the **Candidate Objects** column.
7. On the Upgrade page, select the objects to upgrade and click **Upgrade**.
8. Run and test your application.

17.5 Reviewing Old JavaScript

For applications using jQuery Migrate or legacy JavaScript, review the User Interface, JavaScript settings.

If your application uses jQuery Migrate or legacy JavaScript, review this logic to determine if it is still necessary. Removing this dependency benefits the application by not only ensuring you are only reliant on current, supported technology, but also a performance benefit because APEX will not need to load the corresponding library code.

jQuery 3.x breaks compatibility with earlier 2.x versions. For applications that still rely on removed 2.x functionality, you can use the jQuery Migrate plug-in. To include the jQuery Migrate plug-in, go to the User Interface page and enable **Include jQuery Migrate** as described in the task that follows.

If your application relies on removed 1.x jQuery APIs, that functionality no longer works as of Oracle Application Express release 18.1. You must update the JavaScript to only use jQuery 3.5 or later APIs. To learn more, see Known Issues for jQuery Upgrade in *Oracle APEX Release Notes*.

To review the User Interface, JavaScript settings:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From Application home page, you can access the Edit Application Definition page in two ways:
   - Click the **Edit Application Definition** button.
   - From Shared Components:
     - Click **Shared Components**.
     - Under Application Logic, click **Application Definition**.

   The Edit Application Definition page appears.
4. Click the **User Interface** tab.
5. Edit the following settings:
   a. **JavaScript**, **Include Deprecated or Desupported Javascript Functions** - Specifies if deprecated or desupported JavaScript functions are included on every page in the application by selecting the appropriate releases. See Oracle APEX Release Notes for a list of deprecated and desupported functions.

   > **Tip:**
   >
   > If you are confident your application does not contain any references to deprecated or supported functions, do not make a selection and reduce the overall size of the JavaScript files loaded.

   b. **JavaScript**, **Include jQuery Migrate** - Specifies if the jQuery Migrate plug-in should be included on every page in the application.

   jQuery Migrate plug-in restores deprecated features and behaviors of jQuery so that old JavaScript code and jQuery plug-ins will still run properly with the jQuery version loaded by APEX. This plug-in also helps find where you are using deprecated or removed jQuery APIs by logging warnings to the browser console.

   > **Tip:**
   >
   > If you are confident your application does not use a jQuery plug-in and does not contain any references to deprecated jQuery features, disable **Include jQuery Migrate** to reduce the overall size of the JavaScript files loaded.
6. Click **Apply Changes** to save your changes.
7. Run and test your application.

18
Managing Application Data

Learn how to manage application data such as adding data loading capability to applications, using collections to temporarily store information, or how to work with data from different local and remote data sources, including tables, SQL queries or JSON documents.

- Creating Applications with Data Loading Capability
  Create applications with data loading capability to enable end users to dynamically import data into a table within any schema to which they have access.
- About Using Collections
  Use collections to temporarily capture one or more nonscalar values. Collections enable you to store rows and columns currently in session state so they can be accessed, manipulated, or processed during a user's specific session.
- Working with Duality Views and JSON Sources
  Learn about creating Duality View and JSON Sources to retrieve data from a local or a remote database.
- Managing REST Data Sources
  Oracle APEX REST Data Sources enable developers to access to REST services or generic JSON data feeds in applications.
- Managing REST Enabled SQL References
  Execute SQL queries or PL/SQL defined at the component-level on a remote database using REST Enabled SQL references.
- Managing Remote Servers
  Store REST Data Source server information as a Remote Server object.
- Managing Web Credentials
  Store authentication credentials for external REST services or REST Enabled SQL services.
- Managing Generative AI in APEX
  Create Generative AI Services to include AI features in App Builder and enable support for developers to include Generative AI in their applications.
- Accessing Data with Database Links
  Access data remotely by creating a database link.

18.1 Creating Applications with Data Loading Capability

Create applications with data loading capability to enable end users to dynamically import data into a table within any schema to which they have access.

- About Creating a Page with Data Loading Capability
  Add data loading capability to an application by creating a Data Load Definition and then a Data Load page.
- Creating a Data Load Definition
  Create a Data Load Definition in Shared Components.
- Creating a Data Loading Page Using the Create Page Wizard
  If a Data Load Definition exists, run the Create Page Wizard to add a Data Loading page.

- Editing a Data Load Definition
  Edit a Data Load Definition in Shared Components.
- Copying or Subscribing to a Data Load Definition
  Copy a Data Load Definition from another application. When copying a Data Load Definition from another app, you can also subscribe to it.
- Example: Log Errors to a DML Error Log
  Edit the Data Load Definition so that errors are logged into a DML error log.

> **See Also:**
>
> Managing Applications with Legacy Data Loading Capability

18.1.1 About Creating a Page with Data Loading Capability

Add data loading capability to an application by creating a Data Load Definition and then a Data Load page.

> **Tip:**
>
> To see Data Load Definition examples, install the *Sample Data Loading* app from the Gallery. See Using the App Gallery.

A Data Load Definition is comprised of a Data Load Definition, Data Profile, and Data Profile Columns. Data can be loaded either to an existing table in your schema or to a collection. You can define SQL Expression, SQL Query, Lookups, or Transformation Rules for each data profile column. These definitions are used in the Native Data Loading page process type.

When you create a Data Load Definition, the wizard prompts you to add a Data Loading page in your application. However, if a Data Load definition exists, you can also add a Data Loading page later by running the Create Page Wizard. A Data Loading page consists of a single page with a Native Data Loading page process which enables users to upload data from a file or by copy and paste, preview the data, and then upload the data.

Key features of application new data loading include:

- New application data loading supports `CSV`, `XLSX`, `XML`, and `JSON` formats.
- Column mapping occurs at design time, removing the burden for end users.
- Flexible column mappings based on simple names or regular expressions.
- Data conversion with transformation rules or lookup queries.
- Easy work flow for end users: upload the file, verify the preview, and load data.
- `CSV`, `XLSX`, `XML`, and `JSON` data formats can be loaded to tables or collections.
- Configure data loading to Append, Merge or Replace data, with or without Error Handling.
- Simple new Process Type Data Loading: Customize Data Loading pages as you wish.
- `APEX_DATA_LOADING` PL/SQL API available for custom processing.
- Maximum number of columns to load is 300. Note that Legacy data load supports up to 45.

> **Note:**
>
> A Data Load Wizard is not designed or intended to load hundreds of thousands of rows of data. While it is possible to use a Data Load Wizard to load this high volume of data, you may encounter performance issues with both transmitting and loading large data files. Tools like Oracle SQL Developer and Oracle SQL*Loader are better suited to loading large volumes of data.

Supported Data Types

New Data Load supported data types:

- `VARCHAR2`
- `NUMBER`
- `DATE`
- `TIMESTAMP`
- `TIMESTAMP WITH LOCAL TIMEZONE`
- `TIMESTAMP WITH TIMEZONE`
- `CLOB`

> **See Also:**
>
> Creating a Legacy Data Load Page

18.1.2 Creating a Data Load Definition

Create a Data Load Definition in Shared Components.

When you create a Data Load Definition, the wizard prompts you to create a Data Loading page. You can also create it later by running the Create Page Wizard later as described in Creating a Data Loading Page Using the Create Page Wizard.

The Create Data Load definition wizard supports uploading of the following sample file formats: `CSV`, `XLSX`, `XML` or `JSON`. The uploaded file format is saved as data load definition data profile format. This format dictates the type of file end users can upload. For example, if you create a data load definition using a CSV sample file, only `TXT` or `CSV` file types can be uploaded from a page using the data load definition. Similarly, if you create a data load definition using a `XLSX` sample file, only `XLSX` file types can be uploaded from a page using the data load definition. The following task includes sample images in which the Data Load Definition is created on the `EMP` table using `EMP .csv` file.

> **Note:**
>
> You can load this sample table by installing the `EMP/ DEPT` Sample Dataset. See Using Sample Datasets in *Oracle APEX SQL Workshop Guide*

To create a Data Load Definition:

1. Create a new application.
2. Navigate to the Data Load Definitions page:
   a. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

   b. Under **Other Components**, click **Data Load Definitions**.
3. Click **Create**.
4. On Method:
   a. Create Data Load - Select the method by which you would like to create the Data Load Definition. Options include: **From Scratch** and **As a copy of an existing Data Load Definition**.

      This task assumes you select **From Scratch**.

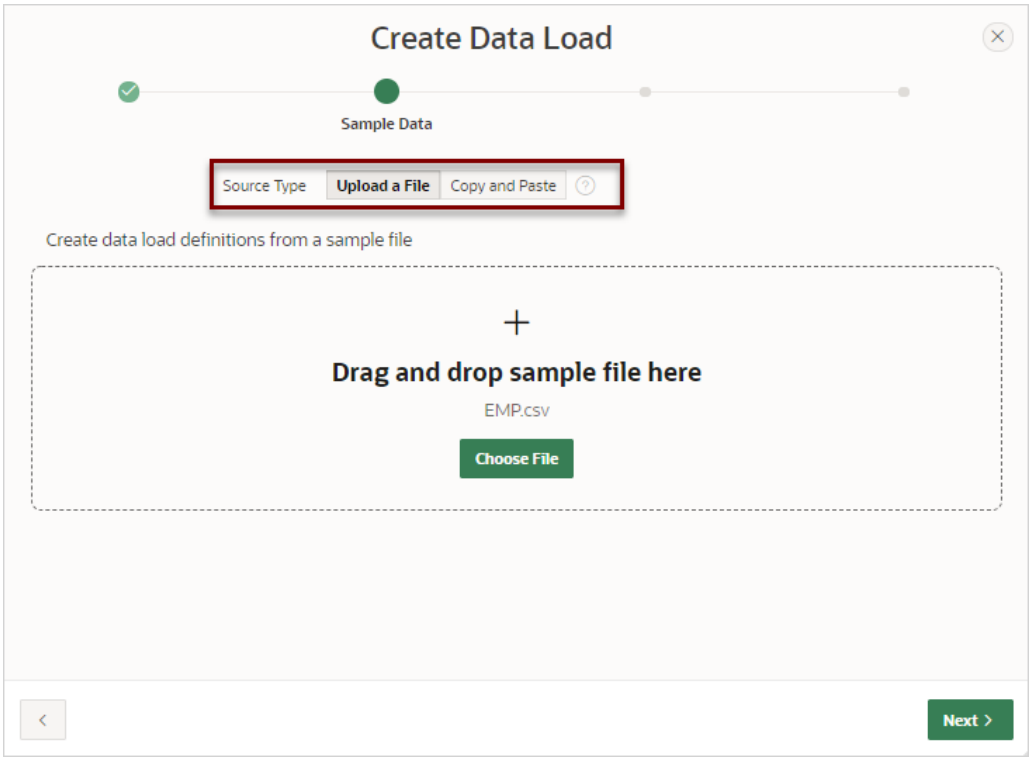
   b. Click **Next**.
5. On Target:
   a. Name - Enter a name for the Data Load Definition.
   b. Target Type - Specify whether the data will be uploaded to a table or an APEX collection (for example, select **Table**).
   c. Table Owner - Specify the owner of the table to load data. If nothing is selected, the application parsing schema is used.
   d. Table Name - Select the table to which the data will be loaded (for example, select `EMP`).

      If entering the table name manually, the table name is case sensitive.

   e. Click **Next**.
6. On Sample Data:
   a. Source Type - Select how the sample data will be uploaded from a file. The Source Type you select determines what appears next.

      Options include:

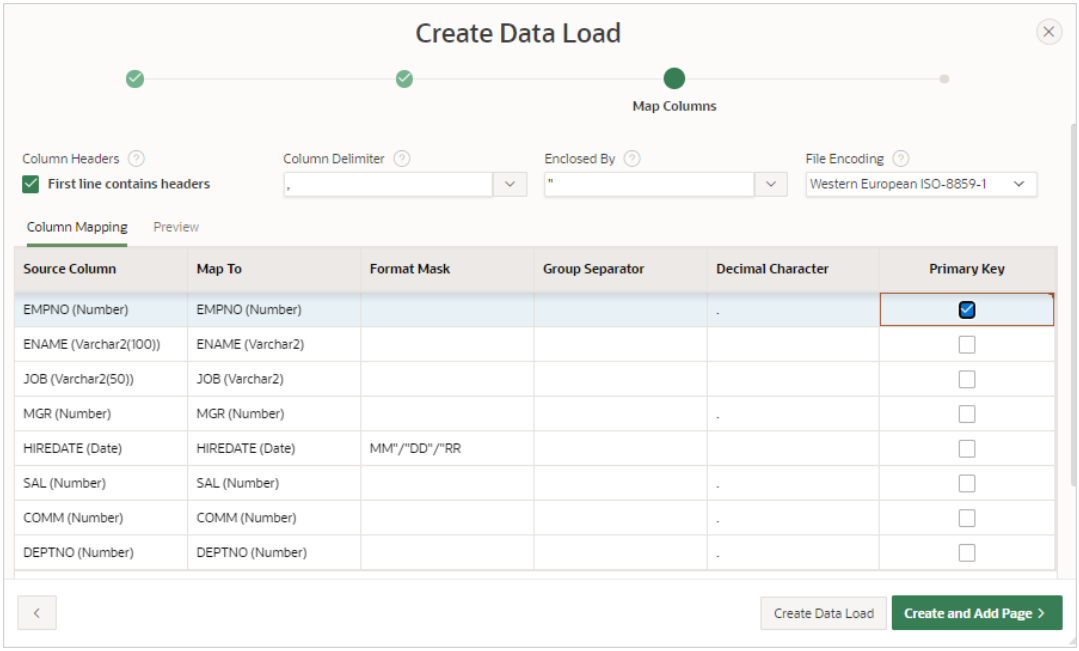
      - If **Upload a File** - Drag and drop or click Choose File to navigate to the file. Supported formats include CSV, XLSX, XML or JSON.
      - If **Copy and Paste** - Copy and paste delimited text data.



Note that `EMP.csv` displays in the center of region indicating the `EMP.csv` sample file has been uploaded.

b. Click **Next**.

7. On Map Columns - Column mapping is done automatically by matching the Sample Data columns with the table column. Accept the defaults.



In the previous example, note that **Primary Key** is selected for the `EMPNO` column.

Determine whether to create just the Data Load Definition or the Data Load Definition and the Data Load page. Click one of the following buttons:

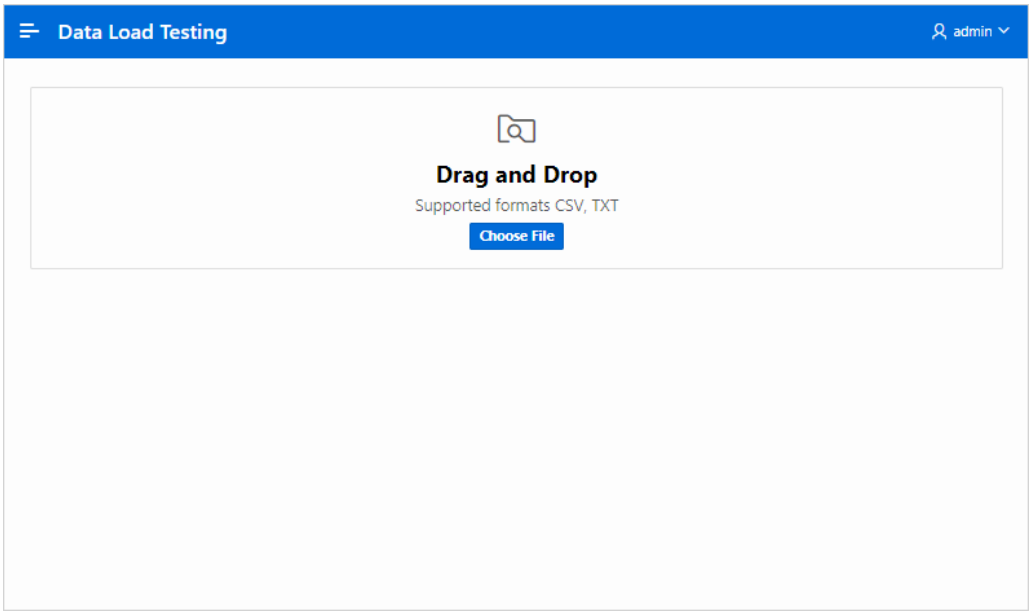
- **Create Data Load** - Creates just the Data Load Definition. If you select this option, you can create a Data Load page later by running the Create Page Wizard.
- **Create and Add Page** - Creates the Data Load Definition and then prompts you to create the Data Load page.

  This task assumes you select **Create and Add Page**.

8. For Page Definition:
   a. Page Number - Displays the page number which is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page. This name is visible only during the application development process.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
9. For Data Load Attributes:
   a. Data Load - Select the data load definition to use.
   b. Upload Data From - Select if application users will provide data by uploading files or by using copy and paste of delimited text data.
   c. Maximum File Size (MB) - Enter the maximum file size (in MB) that should be accepted for the file upload.
10. For Navigation:
    a. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page and which breadcrumb navigation control you want to use.
    b. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
11. Click **Create Page**.

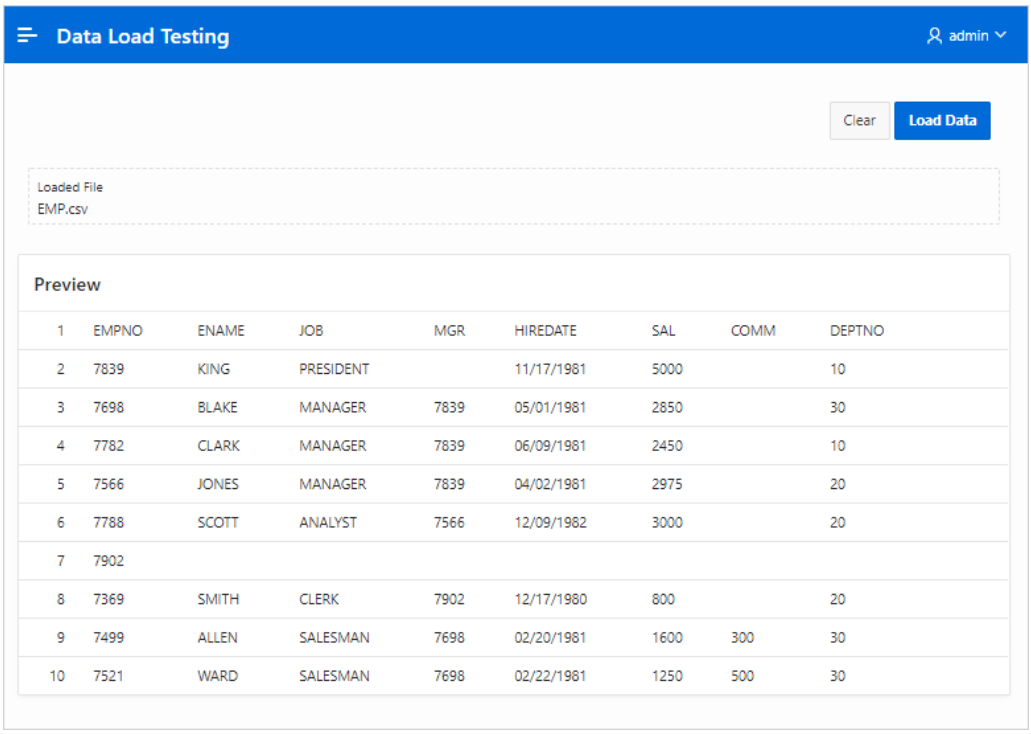
    Page Designer appears displaying the new page.
12. Test the new page:
    a. Click **Save and Run Page**.

       The new Data Load page appears.



b. Upload a sample file. Drag and Drop a file or click **Choose File**.

   A Preview page appears.



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1								
2	7839	KING	PRESIDENT		11/17/1981	5000		10
3	7698	BLAKE	MANAGER	7839	05/01/1981	2850		30
4	7782	CLARK	MANAGER	7839	06/09/1981	2450		10
5	7566	JONES	MANAGER	7839	04/02/1981	2975		20
6	7788	SCOTT	ANALYST	7566	12/09/1982	3000		20
7	7902							
8	7369	SMITH	CLERK	7902	12/17/1980	800		20
9	7499	ALLEN	SALESMAN	7698	02/20/1981	1600	300	30
10	7521	WARD	SALESMAN	7698	02/22/1981	1250	500	30

c. Click **Load Data**. The sample data is loaded into the target table with Merge loading method.

> **See Also:**
>
> Creating Applications

18.1.3 Creating a Data Loading Page Using the Create Page Wizard

If a Data Load Definition exists, run the Create Page Wizard to add a Data Loading page.

When you create a Data Load Definition, the wizard prompts you create Data Loading page. You can also create it later by running the Create Page Wizard.

The Create Page Wizard creates a single page with a Native Data Loading page process to provide the ability to upload data from a file or copy and paste. The copy and paste option is available only if selected data load definition data profile format is `CSV`. The type of file users can upload depends on the data load definition data profile format.

To create a Data Loading page by running the Create Page Wizard:

1. Create a Data Load Definition as described in Creating a Data Load Definition.
2. Run the Create Page Wizard:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. Click **Create Page**.

Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

3. Under Component, select **Data Loading**.
4. Page Definition:
   a. Page Number - Displays the page number which is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page. This name is visible only during the application development process.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
5. Data Load Attributes:
   a. Data Load - Select the data load definition to use.
   b. Upload Data From - Select if application users will provide data by uploading files or by using copy and paste of delimited text data.
   c. Maximum File Size (MB) - Enter the maximum file size (in MB) that should be accepted for the file upload.
6. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
7. Click **Create Page**.

   Page Designer appears displaying the new page.
8. Test the new page:
   a. Click **Save and Run Page**.

      The new Data Load page appears.
   b. Upload a sample file. Drag and Drop a file or click **Choose File**.

      A Preview page appears.
   c. Click **Load Data**. The sample data is loaded into the target table with Merge loading method.

18.1.4 Editing a Data Load Definition

Edit a Data Load Definition in Shared Components.

A Data Load Definition is comprised of a target data load table, loading method, error handling attributes, and data profile.

To edit a Data Load Definition:

1. Select the application associated with the Data Load Definition.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
2. Navigate to the Data Load Definitions page:
   a. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   b. Under **Other Components**, click **Data Load Definitions**.
3. Click the Data Load Definition you want to edit.

   The Data Load Definition Details page appears.
4. For Data Load:
   a. Name - Defines the name for the data load definition.
   b. Static ID - Use the Static ID to reference the Data Load definition in API Calls.
5. For Target:
   a. Target Type - Select whether data is to be loaded into a table or an APEX collection.
   b. Table Owner - Defines the owner of the table into which the data is loaded.
   c. Table Name - Defines the name of the table into which the data is loaded.
6. For Settings:
   a. Loading Method - Specifies the loading method to use. Options include:
      - **Append** - Append rows to the table or collection. If one or more primary key columns are defined in the Data Profile, existing rows are not changed and new rows are appended. If the data profile has no primary key definition, all rows will be simply appended.
      - **Merge** - Merge rows into the table. To use this option, the data profile must have a Primary Key defined. If a row for the given primary key value exists, the row is updated. Otherwise the row will be created. This option is only available for loading data into tables.
      - **Replace** - Existing data in the target table or collection is removed before new data is loaded.
   b. Commit Interval - Specify the commit interval to use during data load. If no commit interval is specified, the whole data load is one transaction, with a commit after the last row. If an interval is specified, APEX commits each time after the specified amount of rows have been processed.
7. For Error Handling:
   - When on Error - Specify how to handle errors during data load. Options include:
     - **Ignore** - The error rows are silently ignored and Data Loading simply continues.
     - **Stop** - Data loading stops on the first error and raises an error message.
     - **Log Error into Collection** - Error rows are logged into an APEX collection.
     - **Log into Error Log** - Error rows are logged into a DML error logging table using the DML Error Logging database functionality. This is only available when loading into a table and only if the Append method is used without a primary key.
8. Data Profile - Displays the Data Profile format and number of columns. Click **Edit Profile** to view and edit the Data Profile.

Data Profiles describe how Data Source format are being parsed and converted to rows and columns. For XML or JSON data format, the Row Selector attribute stores an XML or JSON path expression pointing to the node containing the collection of rows. Data Profile Columns determine how one row is parsed and converted to multiple columns.

9. Subscription - Lists the Data Load Definitions that subscribe to this definition.
10. Comments - Add comments that describe this Data Load Definition. Comments only display within App Builder and never display when running the application.
11. Click **Apply Changes**.

18.1.5 Copying or Subscribing to a Data Load Definition

Copy a Data Load Definition from another application. When copying a Data Load Definition from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy copy or subscribe to Data Load Definition:

1. Navigate to the Data Load Definitions page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under **Other Components**, click **Data Load Definitions**.

      The Data Load Definitions page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. From the Tasks list, select **Copy from another app**.
3. Copy From:
   a. Copy From Application - Select an application from which to copy the data load definition.
   b. Click **Next**.
4. Copy:
   a. To Name - Optionally edit the name of the Data Load Definition.
   b. Action - Select an Action:
      - **Yes** - Copy the Data Load Definition.
      - **No** - Do not copy the Data Load Definition.
      - **Copy and Subscribe** - Copy and subscribe to the Data Load Definition. Subscribing to a Data Load Definition adds a copy of it in your application and sets the Data Load Definition master reference.
5. Click **Copy Data Load Definition**.

18.1.6 Example: Log Errors to a DML Error Log

Edit the Data Load Definition so that errors are logged into a DML error log.

The following example demonstrates how to edit a Data Load Definition that loads data into the `EMP` table. If the data load fails, errors are logged into a DML error log named `ERR$_EMP`.

To edit the Data Load Definition and log errors are logged into a DML error log:

Create a DML error log:

1. Create a DML error log table in SQL commands:
   a. On the Workspace home page, click **SQL Workshop** and then **SQL Commands**.
   b. In the Command Editor, enter the SQL command to create a DML error log named `ERR$_EMP`. For example:

   ```
   begin
       dbms_errlog.create_error_log( dml_table_name => 'EMP' );
   end;
   ```

   c. Click **Run** (Ctrl+Enter) to execute the command.

   The results appear in the Results pane.

   > **Tip:**
   >
   > The table `ERR$_EMP` now appears in Object Browser.

Edit the Data Load Definition and change the Loading Method and define the Error Handling:

2. Navigate to the Data Load Definitions page:
   a. Select the application associated with the Data Load Definition.
   b. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   c. Under **Other Components**, click **Data Load Definitions**.
3. Click the Data Load Definition you want to edit.

   Data Load Definition Details page appears.
4. Change the Loading Method and specify the Error Handling:
   a. Settings, Loading Method - Click **Replace**.
   b. Error Handling, When on Error - Select **Log into Error Log**.
   c. Error Handling, Error Logging Table - Select the DML error log table (for example, `ERR$_EMP (TABLE)`.
   d. Click **Apply Changes**.

Create another Data Loading page:

5. Run the Create Page Wizard:
   a. Click **Edit Page X** in the toolbar in the header region (where **X** represents the page number).

      Page Designer appears.

b. Select the **Create** menu in the Page Designer toolbar and select **Page**.

   The Create Page Wizard appears.

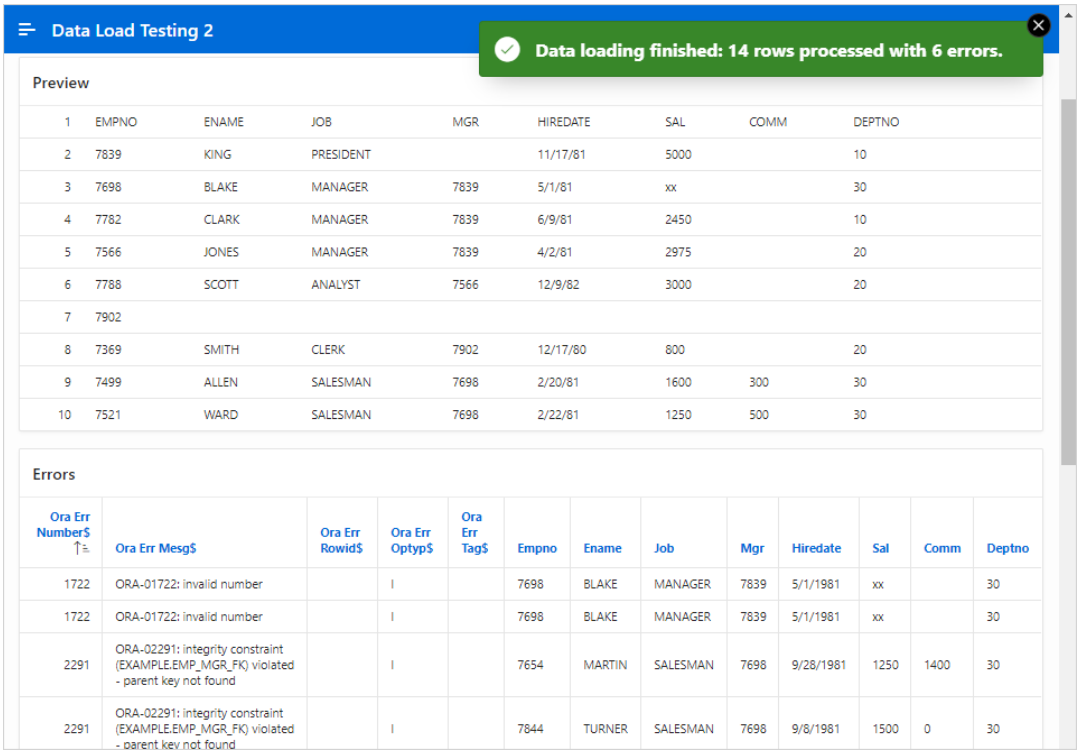
6. On Create a Page, select **Data Loading**.
7. For Page Definition:
   a. Page Number - Displays the page number which is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page. This name is visible only during the application development process.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
8. For Data Load Attributes:
   a. Data Load - Select the data load definition you edited previously.
   b. Upload Data From - Select if users will provide data by uploading a file, or by using copy and paste of delimited text data. For this example, select **File**.
   c. Maximum File Size (MB) - Enter the maximum file size (in MB) that should be accepted for the file upload.
9. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
10. Click **Create Page**.

    Page Designer appears displaying the new page.
11. Test the new page:
    a. Click **Save and Run Page**.

       The new Data Load page appears.
    b. Upload a sample file that fails loading to see the error log report. For example, load string value for the `SAL` column where a `NUMBER` column type is expected. Drag and drop a file, or click **Choose File**.

       A Preview page appears.
    c. Click **Load Data**.

    An error message appears and the errors display at the bottom of the page as shown in the following example.



1	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
2	7839	KING	PRESIDENT		11/17/81	5000		10
3	7698	BLAKE	MANAGER	7839	5/1/81	xx		30
4	7782	CLARK	MANAGER	7839	6/9/81	2450		10
5	7566	JONES	MANAGER	7839	4/2/81	2975		20
6	7788	SCOTT	ANALYST	7566	12/9/82	3000		20
7	7902							
8	7369	SMITH	CLERK	7902	12/17/80	800		20
9	7499	ALLEN	SALESMAN	7698	2/20/81	1600	300	30
10	7521	WARD	SALESMAN	7698	2/22/81	1250	500	30

Ora Err Number$	Ora Err Mesg$	Ora Err Rowid$	Ora Err Optyp$	Ora Err Tag$	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
1722	ORA-01722: invalid number		I		7698	BLAKE	MANAGER	7839	5/1/1981	xx		30
1722	ORA-01722: invalid number		I		7698	BLAKE	MANAGER	7839	5/1/1981	xx		30
2291	ORA-02291: integrity constraint (EXAMPLE.EMP_MGR_FK) violated - parent key not found		I		7654	MARTIN	SALESMAN	7698	9/28/1981	1250	1400	30
2291	ORA-02291: integrity constraint (EXAMPLE.EMP_MGR_FK) violated - parent key not found		I		7844	TURNER	SALESMAN	7698	9/8/1981	1500	0	30

18.2 About Using Collections

Use collections to temporarily capture one or more nonscalar values. Collections enable you to store rows and columns currently in session state so they can be accessed, manipulated, or processed during a user's specific session.

Think of a collection as a bucket in which you temporarily store and name rows of information.

The following are examples of when you might use collections:

- When you are creating a data-entry wizard in which multiple rows of information first need to be collected within a logical transaction. You can use collections to temporarily store the contents of the multiple rows of information, before performing the final step in the wizard when both the physical and logical transactions are completed.
- When your application includes an update page on which a user updates multiple detail rows on one page. The user can make many updates, apply these updates to a collection and then call a final process to apply the changes to the database.
- When you are building a wizard where you are collecting an arbitrary number of attributes. At the end of the wizard, the user then performs a task that takes the information temporarily stored in the collection and applies it to the database.

You insert, update, and delete collection information using the PL/SQL API `APEX_COLLECTION`.

> **See Also:**
>
> - APEX_COLLECTION in *Oracle APEX API Reference*
> - To view an example, install the sample app, *Sample Collections*. To learn more, see Installing Apps from the Gallery.

18.3 Working with Duality Views and JSON Sources

Learn about creating Duality View and JSON Sources to retrieve data from a local or a remote database.

- Managing Duality Views
  Create Duality View objects to retrieve data sourced from a Duality View in a local or a remote database. Duality Views are only available in Oracle Database 23ai or later.
- Managing JSON Sources
  Create a JSON Source to retrieve data from a JSON Collection Table or table with JSON columns in a local or a remote database.

18.3.1 Managing Duality Views

Create Duality View objects to retrieve data sourced from a Duality View in a local or a remote database. Duality Views are only available in Oracle Database 23ai or later.

- About Duality Views
  Learn about Duality Views in Oracle APEX.
- Creating a Duality View Source
  Create a Duality View Source from Shared Components.
- Editing and Deleting a Duality View Source
  Edit or delete a Duality View Source on the Edit Duality page.
- Copying or Subscribing to a Duality View
  Copy a Duality View from the current application or from another application. When copying a Duality View from another application, you can also subscribe to it.
- Viewing Duality Views Utilization
  View the Duality Views Utilization report.
- Viewing Duality Views History
  View the Duality Views History report.

18.3.1.1 About Duality Views

Learn about Duality Views in Oracle APEX.

> **Note:**
>
> Duality Views are only available in Oracle Database 23ai or later.

Duality views combine the advantages of using JSON documents with those of the relational model, while avoiding the limitations of each.

> **Tip:**
>
> To learn more about JSON-Relational Duality Views, see Overview of JSON-Relational Duality Views in *JSON-Relational Duality Developer's Guide*.

In Oracle APEX, a Duality View Source consists of information about the owner and name of the Duality View object in the Oracle Database and a Data Profile.

The Data Profile maps JSON attributes to columns with data types which APEX components can consume. You can define a duality view using the a local database or using REST Enabled SQL reference. After selecting a duality view, Oracle APEX rerieves a JSON schema from the database and builds a data profile.

Once created, the Duality View is available to page components, such as reports, charts, forms, and so on, and shared components, such as list of values and automations. Developers can also use the `APEX_EXEC` package to access the Duality View programmatically.

> **See Also:**
>
> APEX_EXEC in *Oracle APEX API Reference*

18.3.1.2 Creating a Duality View Source

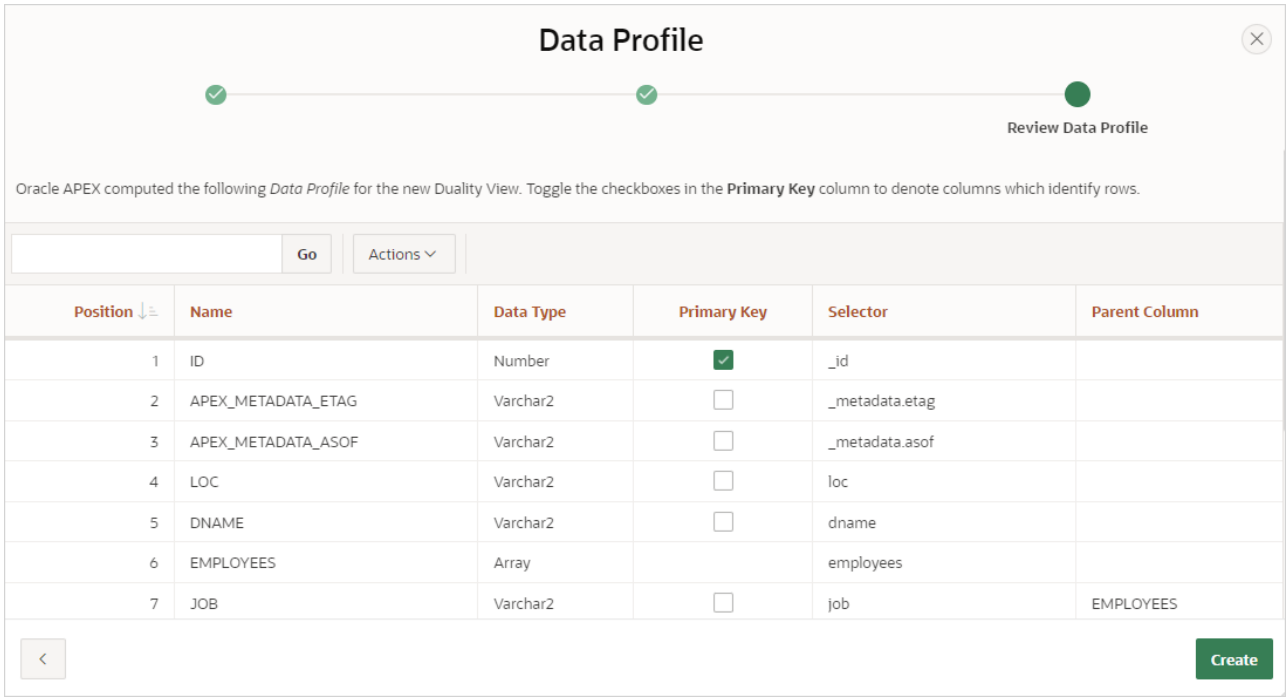
Create a Duality View Source from Shared Components.

Duality Views require Oracle Database Duality Views require Oracle Database 23ai or later. This topic also assumes you have created a Duality View and an application.

To create a Duality Views Source:

1. Navigate to the Duality Views page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Duality Views**.
2. On the Duality Views page, click **Create**.
3. On General, specify:
   a. Name - Enter a name for the new Duality View.
   b. Location - Specify whether data is to be sourced, that is from a **Local Database** or from a **REST Enabled SQL Service**. This example assumes you are using a local database.
   c. Owner - Select the owner of the Table or View.
   d. Duality View - Select a duality view to use.
   e. Click **Next**.

   Oracle APEX displays a Data Profile for the new duality view.



4. Data Profile - Toggle the checkboxes in the **Primary Key** column to denote columns which identify rows.
5. Click **Create**.

   The following message appears:

   ```
   Duality View created.
   ```

   The duality view appears on the Duality Views page.

18.3.1.3 Editing and Deleting a Duality View Source

Edit or delete a Duality View Source on the Edit Duality page.

To edit or delete a Duality Views Source:

1. Navigate to the Duality Views page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Duality Views**.
2. On the Duality Views page, click the duality view Name.

   The Edit Duality View page appears. Use this page to edit the Duality View Source or delete it.
3. To delete the Duality View, click **Delete**. When prompted, confirm your request and click **Delete** again.
4. Under **Identification**:
   a. Name - Enter a name for the new Duality View.
   b. Static ID - Enter a Static ID of the Duality View. The static ID is used to programmatically access the Duality View in PL/SQL code.

5. Under **Source**:
   a. Location - Specify whether the data is sourced from a **Local Database** or from a **REST Enabled SQL Service**.
   b. Owner - Select the owner of the Table or View.
   c. Duality View - Select the Duality View to use.
6. Under **Data Profile**:
   a. Click **Edit Data Profile** to change how the Data Source format is being parsed and converted to rows and columns.
   b. Under **Settings**:
      - Name - Enter a name for the data profile.
      - Format - Choose the data format.
      - Row Selector - For a data profile returning a table, enter the expression to select the collection of rows within the Data Source.
        To learn more and view examples, see item Help.
   c. Under **Columns** - Determines how one row is parsed and converted to multiple columns. Use the Edit icon or the available buttons to edit or add the columns.
   d. Rediscovery:
      - Click **Rediscover Data Profile** to refresh the Data Profile.
      - JSON Schema - Provide a JSON Schema file fromwhich to generate the Data Profile. If no JSON Schema is provided, Oracle APEX generates the data profile by sampling existing data in the table. To learn more, see item Help.
   e. Under **Advanced**:
      - Use Raw Selectors - If enabled, Oracle APEX will not sanitize row and column selectors when generating the SQL query for JSON parsing. All selectors will be used exactly as specified.
      - Profile Comment - Enter any developer comments or notes.
   f. Click **Apply Changes** to save your changes.
7. Advanced - In **Comments**, enter any developer comments or notes.
8. Click **Apply Changes** to save your changes.

18.3.1.4 Copying or Subscribing to a Duality View

Copy a Duality View from the current application or from another application. When copying a Duality View from another application, you can also subscribe to it.

To copy a Duality View:

1. Navigate to the Duality Views page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Duality Views**.
2. On the Duality Views page, click **Copy**.
3. Copy From - Choose where to copy the Duality View from. Options include:

- **This Application**
- **Other Application**

4. To copy from the current application:
   a. Copy From - Select **This Application**.
   b. Copy Duality View - Select the Duality View to copy.
   c. New Duality View Name - Enter the name of the Duality View you would like to create..
   d. Click **Copy**.
5. To copy from another application:
   a. Copy From - Select **Other Application**.
   b. Application - Select the application from which you want to copy.
   c. Copy Duality View - Select the Duality View to copy.
   d. Subscribe - Enable **Subscribe** to subscribe to the Duality View.
   e. New Duality View Name - Enter the name of the Duality View you would like to create..
   f. Click **Copy**.

18.3.1.5 Viewing Duality Views Utilization

View the Duality Views Utilization report.

To view Duality Views Utilization report:

1. Navigate to the Duality Views page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Duality Views**.
2. On the Duality Views page, click **Utilization**.

   The Utilization page displays where a Duality Views are used within the current application.

18.3.1.6 Viewing Duality Views History

View the Duality Views History report.

To view Duality Views History report:

1. Navigate to the Duality Views page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Duality Views**.
2. On the Duality Views page, click **History**.

   The History page displays recent modifications made to Duality Views in the current application.

18.3.2 Managing JSON Sources

Create a JSON Source to retrieve data from a JSON Collection Table or table with JSON columns in a local or a remote database.

- About JSON Sources
  Learn about creating and using JSON Sources in Oracle APEX.
- Creating a JSON Source
  Create a JSON Source from Shared Components.
- Editing and Deleting a JSON Source
  Edit or delete a JSON Source on the Edit JSON Source page.
- Copying or Subscribing to a JSON Source
  Copy a JSON Source from the current application or from another application. When copying a JSON Source from another application, you can also subscribe to it.
- Viewing JSON Sources Utilization
  View the JSON Sources Utilization report.
- Viewing JSON Sources History
  View the JSON Sources History report.

18.3.2.1 About JSON Sources

Learn about creating and using JSON Sources in Oracle APEX.

JSON Sources in APEX consist of information about the owner and name of a database table which can be a plain table with columns containing JSON data or a JSON collection table. Each JSON Source also includes Data Profile which maps JSON attributes to columns, to be consumed by APEX components.

Plain tables with JSON columns are supported in Oracle Database 19c and 21c. JSON collection tables are only available in Oracle Database 23ai or higher.

> **Note:**
>
> To learn more, see Tables With JSON Columns and JSON Collections in *Oracle Database JSON Developer's Guide*.

You can define a JSON Source using a local database or a remote database by using a REST Enabled SQL service. After picking the table and column to use, APEX either determines the data profile from the actual data, or from an uploaded JSON Schema file.

Once created, the JSON Source is available to page components, such as reports, charts, forms, and so on, and shared components, such as list of values and automations. Developers can also use the `APEX_EXEC` package to access the JSON Source programmatically.

> **See Also:**
>
> APEX_EXEC in *Oracle APEX API Reference*

18.3.2.2 Creating a JSON Source

Create a JSON Source from Shared Components.

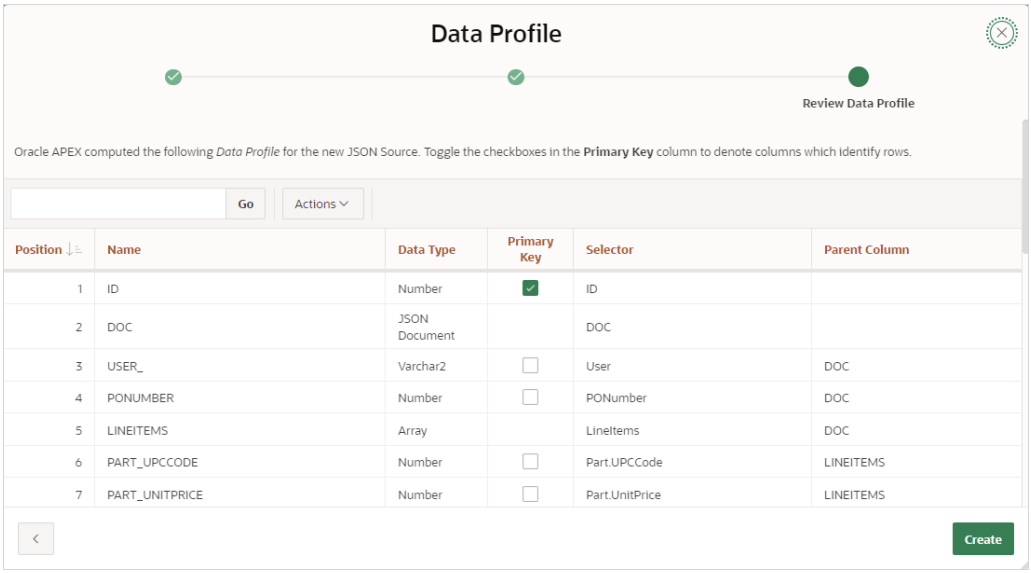
To create a JSON Source:

1. Navigate to the JSON Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **JSON Sources**.
2. On the JSON Sources page, click **Create**.
3. On General, specify:
   a. Name - Enter a name for the new JSON Source.
   b. JSON Source Type - Select the JSON Source type. Options include:
      - **JSON Collection Table** - Denotes JSON collection tables created using the `CREATE JSON COLLECTION TABLE` SQL syntax.
      - **Table with JSON columns** - Denotes normal tables with one or more columns containing JSON documents. After choosing the table, Oracle APEX will ask for up to three columns which contain JSON documents.
   c. Location - Specify whether data is to be sourced from the **Local Database** or from a **REST Enabled SQL Service**.
   d. Owner - Select the owner of the Table or View.
   e. Table with JSON Columns - Select a table with JSON columns.
   f. Click **Next**.
4. JSON Columns - Select up to three columns from the selected table. Selected columns must contain valid JSON documents.
   a. JSON Column 1 - Select a column containing JSON documents, to use for this JSON Source. Columns can be of `VARCHAR2`, `CLOB`, `BLOB`, or `JSON` data type.
   b. JSON Schema for Column 1 - Provide a JSON Schema file to generate the Data Profile from.

      If no JSON Schema is provided, Oracle APEX generates the data profile by sampling existing data in the table. Providing a JSON schema is particularly useful when the table contains no or no representative data.

      Oracle APEX currently only supports the full definition being in one self-contained JSON Schema. Multiple JSON schemas, using imports or being bundled in one file, are not supported; the $id attribute is ignored. Also, Oracle APEX only extracts information from the JSON Schema which is relevant for the Data Profile, which is attribute names, data types and the JSON hierarchy.
   c. Click **Next**.

      The Data Profile appears. Oracle APEX computes the Data Profile for the new JSON Source.



5. Data Profile - Review the data profile and check the Primary Key columns for each hierarchy level.
6. Click **Create**.

18.3.2.3 Editing and Deleting a JSON Source

Edit or delete a JSON Source on the Edit JSON Source page.

To edit or delete a JSON Source:

1. Navigate to the JSON Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **JSON Sources**.
2. On the JSON Source page, click the JSON Source Name.

   The Edit JSON Source page appears. Use this page to edit the JSON Source or delete it.
3. To delete the JSON Source, click **Delete**. When prompted, confirm your request and click **Delete** again.
4. Under **Identification**:
   a. Name - Name of the JSON Source.
   b. Static ID - Static ID of the JSON Source. The static ID is used to programmatically access the JSON Source in PL/SQL code.
   c. JSON Source Type - Select JSON Source type:
      - **JSON Collection Table** - Denotes JSON collection tables created using the `CREATE JSON COLLECTION TABLE` SQL syntax.
      - **Table with JSON columns** - Denotes normal tables with one or more columns containing JSON documents.
5. Under **Source**:

a. Location - Specify whether data is sourced from a **Local Database** or from a **REST Enabled SQL Service**.

b. Table with JSON Columns - Select a table with JSON columns to use.

c. Where Clause - Provide a `WHERE` clause to restrict rows from the data source.

6. Under **Data Profile**:

   a. Click **Edit Data Profile** to change how the Data Source format is being parsed and converted to rows and columns.

   b. Under **Settings**:

      - Name - Enter a name for the data profile.
      - Format - Choose the data format.
      - Row Selector - For a data profile returning a table, enter the expression to select the collection of rows within the Data Source.
        To learn more and view examples, see item Help.

   c. Under **Columns** - Determines how one row is parsed and converted to multiple columns. Use the Edit icon or the available buttons to add or edit the columns.

   d. Rediscovery:

      - Click **Rediscover Data Profile** to refresh the Data Profile.
      - JSON Schema - Provide a JSON Schema file fromwhich to generate the Data Profile. If no JSON Schema is provided, Oracle APEX generates the data profile by sampling existing data in the table. To learn more, see item Help.

   e. Under **Advanced**:

      - Use Raw Selectors - If enabled, Oracle APEX will not sanitize row and column selectors when generating the SQL query for JSON parsing. All selectors will be used exactly as specified.
      - Profile Comment - Enter any developer comments or notes.

   f. Click **Apply Changes** to save your changes.

7. Advanced - In **Comment**, enter any developer comments or notes.

8. Click **Apply Changes** to save your changes.

18.3.2.4 Copying or Subscribing to a JSON Source

Copy a JSON Source from the current application or from another application. When copying a JSON Source from another application, you can also subscribe to it.

To copy a JSON Source:

1. Navigate to the JSON Sources page:

   a. On the Workspace home page, click **App Builder**.

   b. Select an application.

   c. On the Application home page, click **Shared Components** in the center of the page.

   d. Under Data Sources, select **JSON Sources**.

2. On the JSON Sources page, click **Copy**.

3. Copy From - Choose where to copy the JSON Source from. Options include:

   - **This Application**

- **Other Application**

4. To copy from the current application:
   a. Copy From - Select **This Application**.
   b. Copy JSON Source - Select the JSON Source to copy.
   c. New JSON Source Name - Enter the name of the JSON Source you would like to create.
   d. Click **Copy**.
5. To copy from another application:
   a. Copy From - Select **Other Application**.
   b. Application - Select the application from which you want to copy.
   c. Copy JSON Source - Select the JSON Source to copy.
   d. Subscribe - Enable **Subscribe** to subscribe to the JSON Source.
   e. New JSON Source Name - Enter the name of the JSON Source you would like to create.
   f. Click **Copy**.

18.3.2.5 Viewing JSON Sources Utilization

View the JSON Sources Utilization report.

To view JSON Sources Utilization report:

1. Navigate to the JSON Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **JSON Sources**.
2. On the JSON Sources page, click **Utilization**.

   The Utilization page displays where a JSON Sources are used within the current application.

18.3.2.6 Viewing JSON Sources History

View the JSON Sources History report.

To view JSON Sources History report:

1. Navigate to the JSON Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **JSON Sources**.
2. On the JSON Sources page, click **History**.

   The History page isplays recent modifications made to JSON Sources in the current application.

18.4 Managing REST Data Sources

Oracle APEX REST Data Sources enable developers to access to REST services or generic JSON data feeds in applications.

- Understanding REST Data Sources
  REST Data Sources enable developers to access to Representational State Transfer (REST) services or generic JSON data feeds in applications and use the data in Oracle APEX components.
- Creating a REST Data Source
  Create a REST Data Source from Shared Components.
- Example: Creating a REST Data Source on Top of a SQL Query
  Review an example of creating a REST Data Source on top of a SQL Query executed on a REST Enabled SQL service reference.
- Editing or Deleting a REST Data Source
  Edit or delete a REST Data Source from Shared Components.
- Copying or Subscribing to a REST Data Source
  Copy a REST Data Source from another application. When copying a REST Data Source from another app, you can also subscribe to it.
- Copying a REST Data Source
  Copy a REST Data Source by selecting Copy on the REST Data Sources page.
- Working with REST Data Sources for Oracle Cloud SaaS Applications
  Learn about using APEX REST Data Sources with Oracle Cloud SaaS applications.
- Working with Oracle Cloud Applications BOSS REST Services
  Learn about using APEX REST Data Sources for Oracle Cloud Applications (BOSS) REST Services.
- Managing REST Source Catalogs
  Create a catalog of REST Data Source metadata that can be used by all applications in a workspace to create REST Data Sources.
- Managing Data Synchronization
  Enable Data Synchronization to keep the contents of a local table automatically in sync with the data from a REST service.
- Viewing REST Data Source Utilization
  View the REST Data Sources Utilization report.
- Viewing REST Data Sources History
  View the REST Data Sources History report.

18.4.1 Understanding REST Data Sources

REST Data Sources enable developers to access to Representational State Transfer (REST) services or generic JSON data feeds in applications and use the data in Oracle APEX components.

- About REST Data Source Types
  Use REST Data Sources to access to REST services or generic JSON data feeds in applications and use the data Oracle APEX components.
- Understanding Data Profiles
  A Data Profile describes how a Data Source format is being parsed and converted to rows and columns.

- How REST Data Sources Differ from Legacy Web Service References
  Learn how REST Data Sources differ from Legacy Web Service References.
- About Authentication and Credentials in REST Data Sources
  Learn about supported authentication types and credentials in REST Data Sources.

18.4.1.1 About REST Data Source Types

Use REST Data Sources to access to REST services or generic JSON data feeds in applications and use the data Oracle APEX components.

Oracle APEX provides direct integration of REST Data Sources in classic reports, interactive reports, CSS Calendar, and JET Charts. A REST Data Source can contain one or many Operations which are the references to a concrete external web service. Configurations at the REST source level are shared across all contained operations.

Oracle APEX supports the following built-in REST Data Sources types:

- **Simple HTTP** - Select this option for a simple HTTP data feed. Oracle APEX assumes that all data is returned with the first request and that the server does not support server-side filtering, ordering, or other advanced REST service feature. All invocation details and parameters must be configured manually.
- **Oracle REST Data Services** - Denotes Oracle REST Data Services (ORDS). These REST services follow the Oracle REST standard, which means that `GET`, `POST`, `PUT`, or `DELETE` operations are standardized. Server-side filtering and ordering are supported. Oracle APEX can leverage these features by delegating report order-bys and filters to the REST service.
- **REST Enabled SQL Query** - Denotes a REST Source on top of a SQL Query executed on a REST Enabled SQL reference. Oracle APEX supports filtering and pagination, but no DML on these endpoints.

  Advantages of the REST Enabled SQL Query type include:

  – Provides a REST Source for a dedicated SQL query to all components in the APEX application.

  – Components can simply use the REST Source and do not have to replicate the remote SQL query.

  – REST Source Synchronization enables you to download data from a REST Enabled SQL Source to a local table.
- **Oracle Cloud Applications (SaaS) REST Service** - Denotes RESTful web services provided by Oracle Cloud SaaS Applications. Use this web service type to integrate Oracle SaaS REST Services into your Oracle APEX application. Oracle Oracle APEX supports pagination for these services.

  See Working with REST Data Sources for Oracle Cloud SaaS Applications
- **Oracle Cloud Infrastructure (OCI) REST Service** - Denotes RESTful web services provided by Oracle Cloud Infrastructure. Use this web service type to integrate OCI REST Services (for example, accessing the Object Store) into your Oracle APEX application. Oracle APEX supports pagination for these services.
- **OData REST Service** - Denotes a REST service which implements the Open Data (OData) protocol, which is an ISO/IEC approved, OASIS standard that defines a set of best practices for building and consuming RESTful APIs. Oracle APEX supports OData version 4.0.1 or later.

> **Note:**
>
> Oracle APEX currently only supports the Fetch APIs (read data); DML operations for OData are not yet supported.

- **Oracle Cloud Applications (BOSS) REST Service** - Denotes a REST service used to build Oracle APEX apps that query, insert, update, and delete data from Business Object Spectra Service (BOSS) REST endpoints.

  See About Oracle Cloud Applications BOSS REST Services.

> **See Also:**
>
> Managing Data Synchronization

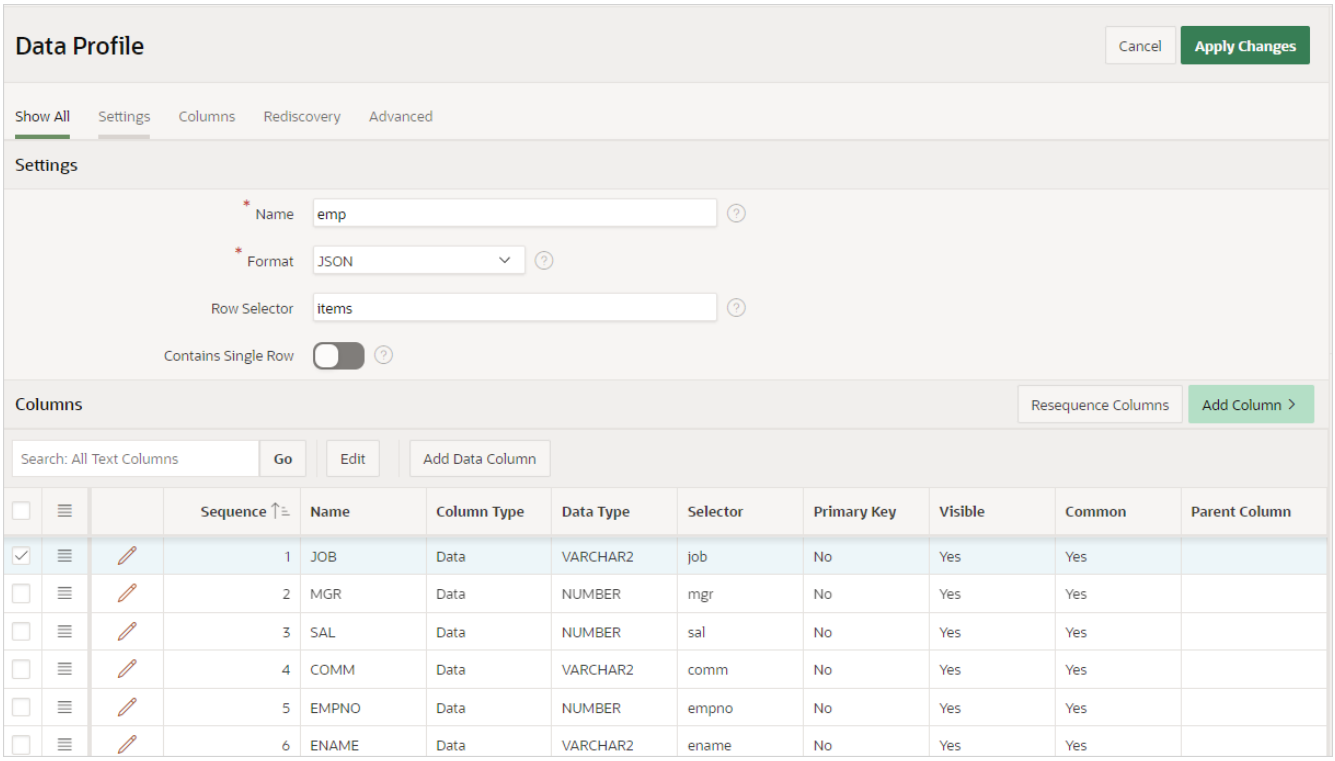
18.4.1.2 Understanding Data Profiles

A Data Profile describes how a Data Source format is being parsed and converted to rows and columns.

The Data Profile of a REST Data Source contains information about how the response of a REST API must be parsed in order to extract data. A Data Profile stores the following information:

- The Response Format, that is, whether a REST Source returns data in JSON, XML or CSV formats.
- Information about Array Columns. An **Array Column** within the response contains a set of rows.
- Data Profile Columns store information about its name, the data type and the name of the attribute within the response from the REST API.

The example that follows shows a sample Data Profile named `emp` with a Format of `JSON`.



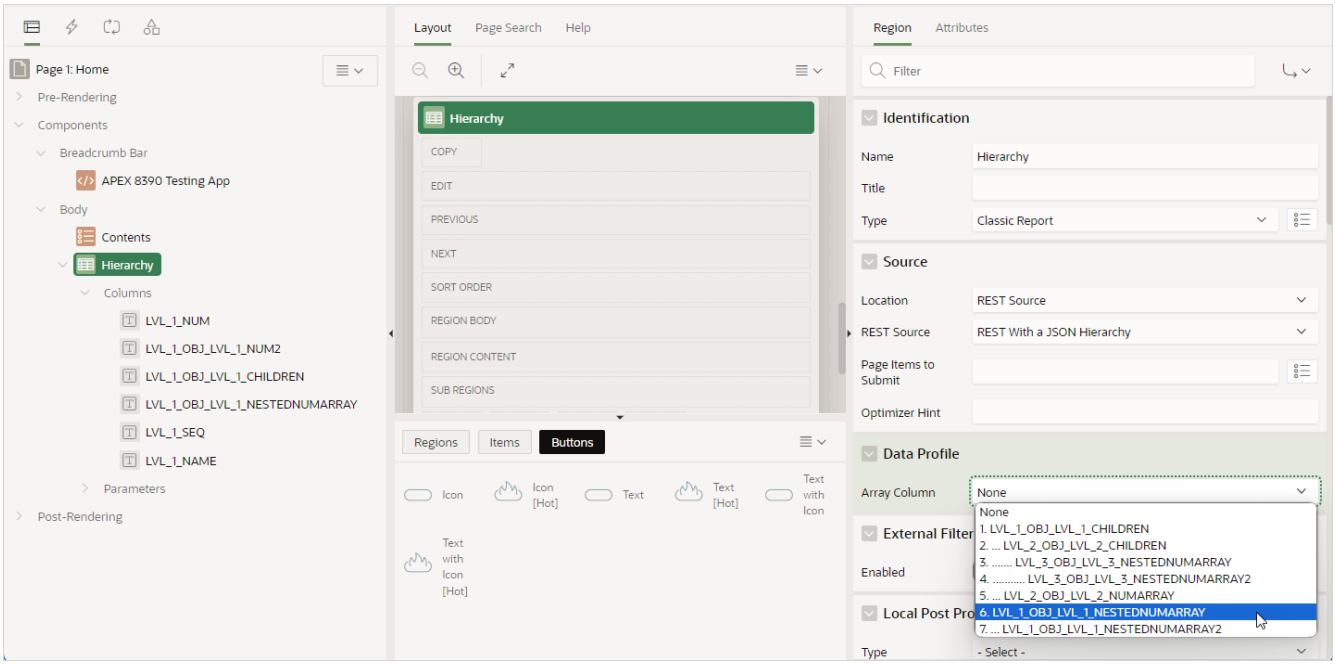
APEX uses the information within the data profile to automatically parse the response and to provide the data in rows and columns, using the configured data types so that APEX components, such as reports, charts, calendars or others, can use them.

The **Row Selector** of the data profile determines which XML or JSON attribute contains the set of rows. APEX makes that collection of rows available to the consuming component. Each Data Profile column represents an attribute of the response data, within the row set denoted by the Row Selector. For XML or CSV formats, APEX always treats the response as a flat, table-like data structure.

JSON Data Profiles

APEX supports defining **Array Columns** to represent a nested array within a row set. The nested array can have one or many Data Profile columns which reference the array column as their parent. Note that cyclical references are not allowed. In other words, a column cannot reference an Array Column as a parent if that column is a child or grandchild of the current column.

If an APEX component is created based on a REST Data Source and the Data Profile contains array columns, Page Designer enables you to choose an array column. The direct children of that chosen array column, and its parent columns are then available to the APEX component. In the example that follows, the Data Profile, Array Columns select list is selected.



18.4.1.3 How REST Data Sources Differ from Legacy Web Service References

Learn how REST Data Sources differ from Legacy Web Service References.

Unlike Legacy Web Service References, a REST Data Source contains metadata about the Web service which can be used by APEX Components or PL/SQL processes to invoke the service and to process the responses.

REST Data Sources contain multiple operators that differ depending upon the Web service target. For a REST services, an operation is a specific service handler (such as, `GET`, `PUT`, `POST`, or `DELETE`). Developers assign Operation a Database Action such as `Fetch Multiple Rows`, `Fetch Single Row`, `Insert Row`, `Update Row`, and `Delete Row`. However, you can assign each Database Operation only once to a REST Data Source Operation.

> **See Also:**
>
> Managing Legacy Web Services

18.4.1.4 About Authentication and Credentials in REST Data Sources

Learn about supported authentication types and credentials in REST Data Sources.

REST Data Sources supports various authentication types. Authentication credentials can be specified at the Remote Server-level for all REST Data Sources using the Remote Server). If credentials are set at the REST Data Source-level, that setting supersedes credentials stored at the Remote Server level.

A Credential denotes the Authentication method, a Client ID (or user name) and a Client Secret (or password). Credentials are stored as a named entity within Shared Components and can be re-used across multiple REST Data Sources or Remote Servers.

> **See Also:**
>
> - About Supported Authentication Types in Protecting Web Credentials by Defining Valid URLs
> - Managing Web Credentials

18.4.2 Creating a REST Data Source

Create a REST Data Source from Shared Components.

To create a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Create**.
3. Create REST Data Source - Select **From Scratch** and click **Next**.

   > **Tip:**
   >
   > If you select **As copy of an existing REST Data Source**, select the application to copy from and follow the on-screen instructions.

4. General:
   a. REST Data Source Type - Select the REST Data Source type, which is either a built-in type or a type defined by a REST Source Plug-In. To learn more, review item Help for more details or see About REST Data Source Types.

      The **REST Data Source Type** determines what appears next. This example assumes, REST Data Source Type is set to **Simple HTTP**.
   b. Name - Enter a descriptive name for this REST Data Source.
   c. URL Endpoint - Enter the Endpoint URL (starting with `http://` or `https://`) for this REST Data Source.

      > **Tip:**
      >
      > Do not add query string parameters by appending `?name1=value&name2=value`. These are configured on the Advanced wizard page after you configure Authentication.

   d. Open API URL - Enter the URL where the OpenAPI information is available. If populated, the REST Data Source uses OpenAPI/Swagger for discovery.

e. HTTPS Host Name - The host name to be matched against the common name (CN) of the remote server's certificate for an HTTPS request. It can also be a domain name like `*.example.com`. If NULL, the host name in the given URL is used.

f. Click **Next**.

5. Remote Server:

   a. Remote Server - Choose an existing Remote Server object. The HTTP Endpoint URL of a REST Data Source consists of the remote server (Base URL) and the service specific part. Remote Servers are stored as separate Entities. When multiple REST Data Sources share the same remote server, they can easily be redirected to another physical location (for example, from test to production) by just changing the base URL within the Remote Server definition..

   b. Base URL - This is the Base URL of the chosen Remote Server object. The HTTP Endpoint URL of this REST Data Source is made up from the Base URL and the service-specific path suffix below.

      For example:

      - `http://api.example.com:8080/myservices/test/`
      - `https://api.example.com/ords/scott/`

   c. Service URL Path - Enter the service-specific URL Path for this REST Data Source. The HTTP Endpoint URL is built by appending this to the Base URL specified in the Remote Server object.

      For example:

      - `/services/customers/`
      - `/products/4711/details`

   d. Click **Next**.

6. If **REST Data Source Type** is **Simple HTTP**, under **Settings**, select one of the following:

   a. **Pagination Type** - Select one of the following:

      - **No Pagination**
      - **Page Size and Fetch Offset** - Flexible pagination where the REST API allows the request a specific set of rows, using page size and offset ("start at row") parameters.

        The following shows typical URL. In this example, the URL requests page number 2 with a page size of 100, which leads to rows 101 to 200 being returned.

        ```
        http://hostname.domain.com/path/to/rest_api?limit=100&offset=20
        ```

        In the previous example, note that the parameters named `limit` and `offset` can be arbitrary and depend upon the REST API being used.

      - **Page Size and Page Number** - Pagination where the REST API allows the request a set of rows, using page size and page number parameters.

        URLs typically look like the following example which requests page number 2 with a page size of 100, which leads to rows 101 to 200 being returned.

        ```
        http://hostname.domain.com/path/to/rest_api?size=100&page=2
        ```

      - **Page Number** - Pagination where the REST API allows to request a set of rows, using a page parameter. The Page Size is fixed and determined by the REST API.

        URLs typically look like the following example which requests page number 2. If the REST API uses a fixed page size of 20, rows 21 to 40 will be returned.

```
http://hostname.domain.com/path/to/rest_api?page=2
```

- **Page Size and Page Token** - Pagination where the REST API allows to request a set of rows, using a combination of a **page token** and **page size** parameters. The token is used to retrieve the next set of rows, and is returned in the response body of the previous request. The page size determines the number of rows to return in each request.

  URLs typically look like the following example, which requests the next set of rows using the token `ApEx=` and a page size of `100`.

  ```
  http://hostname.example.com/path/to/rest_api?token=ApEx=&limit=100
  ```

The UI changes based on Pagination Type you select. See field-level Help for more information on specific attributes.

b. Click **Next**.

7. Authentication Required - Choose whether the new REST Data Source requires Authentication.

> **Tip:**
>
> To learn more about authentication, see About Supported Authentication Types in Protecting Web Credentials by Defining Valid URLs

8. To proceed, select one of the following:
   - **Create REST Source Manually** - This option does not include Discovery mode. Based on the previously entered information, a REST Data Source is created with a few sample operations and a sample data profile. This option creates an *incomplete* REST Data Source that is unusable with an APEX component. To correctly reflect the external web service, you will need to edit the operations and the data profile.
   - **Advanced** - Presents advanced parameters for the Data Profile. Use **Response Sample** to upload a sample JSON (or XML) response. Once a file is uploaded, APEX performs a Data Profile Discovery on the uploaded file.
   - **Discover** - Starts Discovery mode. APEX executes multiple HTTP requests in order to get information about the external Web service. One of these multiple HTTP requests can be an HTTP request to get the OpenAPI information about the Web service. APEX samples the data (JSON or XML) returned by the Web service and derives a Data Profile (consisting of metadata about the JSON attributes). Discovery results then display and confirms whether to create the REST Data Source.
9. If you click **Advanced**, the Parameter page appears.

   Use the Parameters page to configure initial REST Data Source parameters which might be required for discovery. Some REST Data Sources may require you set specific HTTP Headers (for example, `User-Agent`). After setting advanced parameters, select either **Create REST Source Manually** or **Discover**.
10. If you click **Discover**, APEX invokes the REST service URL and a Preview appears. From the JSON response, APEX derives attributes and their data types and uses this to create a Data Profile.

    The information that displays depends upon the complexity of the target service. For simple HTTP services or ORDS services which do not provide metadata about their JSON response attributes, APEX investigates the JSON or XML response, samples the data, and creates a proposal for a data profile. The Data tab displays the sample response data, formatted as a report.

The columns tab shows information about the data profile columns APEX has found during discovery. Use this tab to verify whether the column data types are correct.

If you wish to remove a named column you can navigate with the TAB key to the remove column and press Enter, or simply by using the mouse by pressing the **X** icon.

The Response Body tab shows information about the response body structure APEX has found during discovery. Use this tab to verify the complexity of the structure in the response body that will be returned depending on the type of response received in discovery.

Click **More Detail** to view additional detailed information about the REST Data Source response, such as the response body, response headers and the REST Data Source operations.

11. Click **Create REST Data Source**.

    The REST Data Source is created.

18.4.3 Example: Creating a REST Data Source on Top of a SQL Query

Review an example of creating a REST Data Source on top of a SQL Query executed on a REST Enabled SQL service reference.

REST Enabled SQL service references to a remote database include support for executing a SQL query on REST Enabled SQL reference. The following example creates a REST Data Source with a MySQL query using the fictional reference, MySQL REST Enabled SQL Example, described in Example: Creating REST Enabled SQL Service Reference on MySQL.

To create a REST Source on top of a SQL query executed on a REST Enabled SQL reference:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Create**.
3. Method - Select **From Scratch** and click **Next**.
4. General:
   a. REST Data Source Type - Select **REST Enabled SQL Query**.
   b. Name - Enter a descriptive name for this REST Data Source (for example, **MySQL REST Data Source**).
   c. REST Enabled SQL Reference - Select a previously created REST Enabled SQL Reference on which to execute a SQL query (for example, **MySQL REST Enabled SQL Example**).
   d. Click **Next**.
5. REST Enabled SQL Query:
   a. SQL Query - Enter a query to execute on the REST Enabled SQL Service for this REST Data Source. In the following example, you create a MySQL query using the fictional reference, MySQL REST Enabled SQL Example, described in Example: Creating REST Enabled SQL Service Reference on MySQL.

```
select * from `film`
```

Result discovery and data type detection is done based on the results of this query. On MySQL databases, the query executes within the default database, which is configured within the REST Enabled SQL reference. Tables in other databases can be referenced by simply prefixing them.

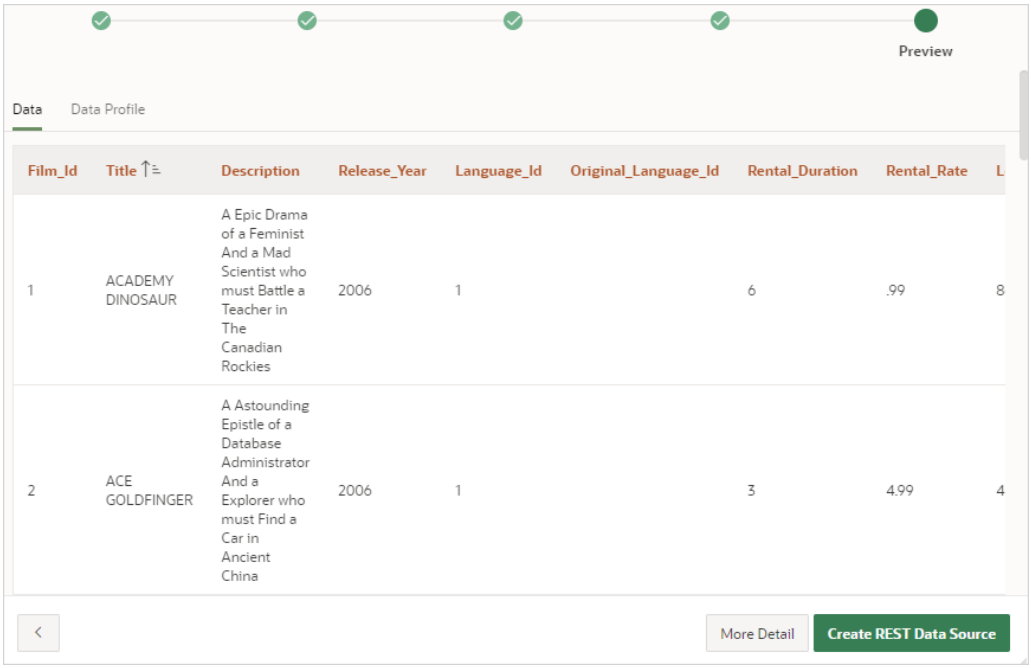
b. Click **Next**.

6. Authentication:

   a. Credentials - Accept the default or choose existing Credentials to be used for authentication.

   b. Click **Discover**.

   APEX invokes the REST service URL and a Preview appears. From the REST Enabled SQL response, APEX derives attributes and their data types and uses this to create a Data Profile. Consider the following example:



7. Click **Create REST Data Source**.

   The REST Data Source is created.

   The data from the `film` table is now available as a REST Data Source which can be used in APEX components, REST Source Catalogs or for REST Synchronizations.

> **Note:**
>
> When used in interactive reports or interactive grids, end user filters are converted to MySQL or Oracle specific SQL and pushed down to the REST Enabled SQL Service. You can also use **Post Processing SQL** which enables support for joins between a MySQL Query and a local Oracle table.

18.4.4 Editing or Deleting a REST Data Source

Edit or delete a REST Data Source from Shared Components.

- Editing a REST Data Source
  Edit a REST Data Source from Shared Components.
- Deleting a REST Data Source
  Delete a REST Data Source from Shared Components.

18.4.4.1 Editing a REST Data Source

Edit a REST Data Source from Shared Components.

To edit a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click the **REST Source Name**.

   The REST Data Source page appears.
3. Edit the REST Data Source attributes.
4. Under **REST Data Source**:
   a. Name - Defines descriptive name for this REST Data Source.
   b. REST Data Source Type - Select the REST Data Source type, which is either a built-in type or a type defined by a REST Source Plug-In. To learn more, review item Help for more details or see About REST Data Source Types.

      The **REST Data Source Type** determines what appears next. This example assumes, REST Data Source Type is set to **Simple HTTP**.
   c. Remote Server - Choose a Remote Server object for this REST Data Source. To learn more, see field-level Help.
   d. Base URL - Displays the Base URL.
   e. URL Path Prefix - The URL Path Prefix is specific to this REST Data Source. The URL Endpoints of all operations within the REST Data Source share the same URL Path Prefix. To learn more, see field-level Help.
5. If **REST Data Source Type** is **Simple HTTP**, under **Settings**, select a **Pagination Type**:
   - **No Pagination**
   - **Page Size and Fetch Offset** - Flexible pagination where the REST API allows the request a specific set of rows, using page size and offset ("start at row") parameters.

     The following shows typical URL. In this example, the URL requests page number 2 with a page size of 100, which leads to rows 101 to 200 being returned.

     ```
     http://hostname.domain.com/path/to/rest_api?limit=100&offset=20
     ```

     In the previous example, note that the parameters named `limit` and `offset` can be arbitrary and depend upon the REST API being used.
   - **Page Size and Page Number** - Pagination where the REST API allows the request a set of rows, using page size and page number parameters.

     URLs typically look like the following example which requests page number 2 with a page size of 100, which leads to rows 101 to 200 being returned.

```
http://hostname.domain.com/path/to/rest_api?size=100&page=2
```

- **Page Number** - Pagination where the REST API allows to request a set of rows, using a page parameter. The Page Size is fixed and determined by the REST API.

  URLs typically look like the following example which requests page number 2. If the REST API uses a fixed page size of 20, rows 21 to 40 will be returned.

  ```
  http://hostname.domain.com/path/to/rest_api?page=2
  ```

- **Page Size and Page Token** - Pagination where the REST API allows to request a set of rows, using a combination of a **page token** and **page size** parameters. The token is used to retrieve the next set of rows, and is returned in the response body of the previous request. The page size determines the number of rows to return in each request.

  URLs typically look like the following example, which requests the next set of rows using the token `ApEx=` and a page size of `100`.

  ```
  http://hostname.example.com/path/to/rest_api?token=ApEx=&limit=100
  ```

The UI changes based on Pagination Type you select. See field-level Help for more information on specific attributes.

6. If **REST Data Source Type** is **REST Enabled SQL Query**, under **Settings**, **REST Enabled SQL Query**:
   - SQL Query - Displays the SQL Query to execute on the REST Enabled SQL Reference.
7. Under **Authentication**:
   - Credentials - Select stored credentials for authentication.
   - Authentication Server - If authentication has to be done on a (different) *Authentication Server*, create a **Remote Server** object for this and choose it here. See field-level Help for more information.
8. Under **Data Profile**:
   a. Click **Edit Data Profile** to change how REST Data Source responses (in XML or JSON) format are being parsed and converted to rows and columns.

      The Data Profile page appears.

   b. Under **Data Profile** and **Columns**, edit the attributes.

      Row Selector stores an XML or JSON path expression pointing to the node containing the collection of rows. **Columns** determine how one row is parsed and converted to multiple columns.

   c. Click **Apply Changes** to save your changes.
9. Under **Operations** - **Operations** contains the actual information about the external web service handler. Operations can be mapped to Database Operations which enables APEX components to pick up the correct operation.
   a. To add or edit an operation:
      - Click **Add Operation** to add a REST Data Source Operation.
      - Click the **Edit** icon to edit an existing operation.

      The REST Source Operation page appears.

   b. Under **Operation**:

- URL Pattern - The **URL Pattern** is the last part of the Web Service endpoint URL. The combination of URL pattern and HTTP Operation (GET, POST, PUT, DELETE) identifies the operation within a REST Data Source.
- HTTP Method - Defines the HTTP Operation or **Method** (GET, POST, PUT, DELETE, PATCH) to be executed on the REST Web Service.
- Database Operation - Assign a Database Operation to this HTTP operation. The database operation is important for APEX components to work with the REST Data Source.

To learn more about these attributes and see examples, see field-level Help.

c. Under **Caching**:

- Caching - Select how the APEX engine can cache REST Source responses on the server side. If caching can be used, the component is rendered based on cached results instead of fetching from the web service again. Only HTTP "GET" requests will be cached, which will cut down expensive HTTP requests. This feature is unrelated to the browser's caching capabilities.
- Invalidate When - If you enable Caching, specify when cached content is to be invalidated. Provide either the amount of minutes or a "DBMS_SCHEDULER Calendaring String". With the latter, you determine cached content to be invalidated at the top of the hour, at midnight, or another specific recurring point in time.

To learn more about these attributes and see examples, see field-level Help.

d. Under **Operation Parameters**, click **Add Parameter** to configure REST Data Source parameters.

Parameters enable support for dynamic URL parts or to pass additional HTTP request headers or a specific request body to the web service. Oracle APEX components assign values to the parameters before invoking the web service. To learn more about these attributes and see examples, see field-level Help.

e. Under **Advanced**:

- Fixed Page Size - Enter the size of the REST service's pagination window. The pagination for a resource handler HTTP operation GET method equates to the number of rows to return on each page of a JSON format result set based on a database query.
- Static ID - Provide a static ID for programmatic access to this REST Operation. The static ID will be converted to lower case, and all non-alphanumeric or special characters will be replaced with an underscore ("_").
- Operation Comment - Enter any developer comments

To learn more about these attributes, see field-level Help.

f. Click **Apply Changes**.

10. Under **Parameters** - **Parameters** enable support for dynamic URL parts or to pass additional HTTP request headers or a specific request body to the web service. APEX components assign values to the parameters before invoking the web service.

a. Click **Add Parameter**.

The Parameter page appears.

b. On Edit REST Data Source Parameter:

- Edit the attributes. To learn more about an attribute, see field-level Help.
- Click **Add Parameter** or **Add and Add Another**.

   c. On Edit REST Data Source Parameter, edit the attributes and then click Add Parameter. To learn more about an attribute, see field-level Help.

11. **Subscription** lists REST Data Sources that subscribe to this module and indicates if this REST Data Source is a copy.
12. Under **Open API**:
    a. Open API - Enable or disable **Use OpenAPI**. If enabled, the REST Data Source uses OpenAPI/Swagger for discovery.
    b. Endpoint is relative to Base URL - Choose whether the OpenAPI Endpoint is relative to the Base URL of the Remote Server.
    c. OpenAPI URL - Enter the URL where the OpenAPI information is available. This can be an absolute URL or an URL relative to the Base URL of the service.
13. **REST Source Catalog** - Click **Save to Catalog** to save the REST Data Source to a REST Source Catalog. See Managing REST Source Catalogs.
14. Under **Advanced**:
    a. Static ID - Use the Static ID to reference the REST Data Source in API Calls.
    b. Pass ECID - Indicates whether to pass the Execution Context ID (ECID) to the external web services for end-to-end tracing.

       See Advanced to learn more about the application setting in Shared Components, Security Attributes, Advanced, Pass ECID.
    c. HTTP Transfer Timeout - Specify the transfer timeout for all future HTTP requests that are attempted while reading the HTTP response from the Web server or proxy server.

       This timeout value can be used to avoid the application from being blocked by busy Web servers or heavy network traffic while retrieving responses. See SET_TRANSFER_TIMEOUT Procedure in *Oracle Database PL/SQL Packages and Types Reference*
    d. Comments - Enter Comments or notes.
15. To save your changes, click **Apply Changes**.

18.4.4.2 Deleting a REST Data Source

Delete a REST Data Source from Shared Components.

> **Tip:**
>
> A REST Data Source can only be deleted if no Oracle APEX component is referencing it.

To edit or delete a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click the **REST Source Name**.

The REST Data Source page appears.

3. To delete the REST Data Sources, click **Delete**.
4. Confirm your selection and click **OK**.

18.4.5 Copying or Subscribing to a REST Data Source

Copy a REST Data Source from another application. When copying a REST Data Source from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy a REST Data Source from another application:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. From the Tasks list, select **Copy from another app**.
3. Copy From Application - Select an application from which to copy the REST Data Source and click **Next**.
4. Copy:
   a. To Name - Optionally edit the name of the REST Data Source.
   b. Action - Select an Action:
      - **Yes** - Copy the REST Data Source.
      - **No** - Do not copy the REST Data Source.
      - **Copy and Subscribe** - Copy and subscribe to the REST Data Sources. Subscribing to a REST Data Source adds a copy of it in your application and sets the REST Data Source master reference.
5. Click **Copy REST Data Source**.

18.4.6 Copying a REST Data Source

Copy a REST Data Source by selecting Copy on the REST Data Sources page.

To copy a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.

   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Copy**.
3. On Copy REST Data Source:
   a. Copy REST Data Source - Select an existing REST Data Source.
   b. New REST Data Source Name - Enter a new name.
   c. Click **Copy**.

18.4.7 Working with REST Data Sources for Oracle Cloud SaaS Applications

Learn about using APEX REST Data Sources with Oracle Cloud SaaS applications.

- About REST Data Source Support for Oracle Cloud SaaS Applications
  Learn about REST Data Source support for Oracle Cloud SaaS applications.
- Creating REST Data Source for Oracle Cloud SaaS Apps
  Learn about creating a REST Data Source for an Oracle Cloud SaaS application.
- REST Data Source Definitions for Oracle Cloud SaaS Apps
  Learn about viewing REST Data Source definitions for an Oracle Cloud SaaS application.
- REST Data Source Runtime Features for Oracle Cloud SaaS Apps
  Learn about Oracle Cloud SaaS application runtime features.
- Filter Parameters for Query By Example and Parent and Child Use Cases
  Create filter parameters for Query By Example and parent and child use cases.
- Working Against a Sandbox
  Associate an APEX application with a sandbox by defining a sandbox name on the Shared Components, Component Settings, Oracle Cloud Applications (SaaS) REST Service page.
- Best Practices for Configuring REST Service Base URL
  Learn about best practices for configuring REST Service Base URL.
- Overriding Default Runtime Headers If Necessary
  Define a REST Data Source parameter to use a later REST framework version.
- Data Profile Column Annotations
  Add Data Profile Column annotations using the Additional Info attribute.
- Example: Creating a REST Data Source Against a Fusion Applications Endpoint
  Learn how avoid an error when creating a REST Data Source against a Fusion Applications end point which uses selectively required attributes.

18.4.7.1 About REST Data Source Support for Oracle Cloud SaaS Applications

Learn about REST Data Source support for Oracle Cloud SaaS applications.

Determining the REST Data Source Type

For example, if the Fusion Applications REST API you want to work with contains `RestApi/resources/`, you would create a REST Data Source with a Type of **Oracle Cloud Applications (SaaS) REST Service**. Consider the following examples: `/hcmRestApi/resources/`,`/fscmRestApi/resources/`, or `/crmRestApi/resources/`.

In contrast, if the URL contains `/api/boss/`, you would create a REST Data Source with a Type of **Oracle Cloud Applications (BOSS) REST Service**.

About SaaS) REST Services

Oracle APEX simplifies the process of building APEX applications that query, insert, update, and delete data from Oracle Cloud applications (SaaS) REST Service endpoints. This support includes all REST APIs for:

- Oracle Fusion Cloud Applications business objects (possibly customized by Fusion Application business administrators)
- Oracle Visual Builder application business objects
- Custom apps that use Oracle Application Development Framework (ADF) business components

> **See Also:**
>
> - Oracle Fusion Cloud Applications Suite
> - Working with Business Objects in *Developing Applications with Oracle Visual Builder*
> - Oracle Application Development Framework (ADF)

18.4.7.2 Creating REST Data Source for Oracle Cloud SaaS Apps

Learn about creating a REST Data Source for an Oracle Cloud SaaS application.

To create a REST Data Source for an Oracle Cloud SaaS application, run the Create REST Data Source Wizard. When prompted, select the REST Data Source Type, **Oracle Cloud Applications (SaaS) REST Service**. This topic describes key decision points when running the Create REST Data Source Wizard.

> **Tip:**
>
> For more details on all wizard options, see item Help or see the generic discussion, Creating a REST Data Source.

To create a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Create**.

   The Create REST Data Source Wizard appears.
3. Create REST Data Source - Select **From Scratch** and click **Next**.
4. General:
   a. REST Data Source Type - Select **Oracle Cloud Applications (SaaS) REST Service**.

b. Name - Enter a descriptive name for this REST Data Source.

c. URL Endpoint - Enter the Endpoint URL (starting with `http://` or `https://`) for the REST Data Source.

d. Click **Next**.

5. Remote Server - Accept the defaults.

   This step shows how APEX manages the REST API's endpoint URL as the combination of a **Base URL** and a **Service URL Path**.

6. Settings, Pagination Type - Accept the default and click **Next**.

7. Authentication Required - Choose an existing credential if appropriate, or choose an Authentication Type and enter the relevant details for defining a new credential.

8. Click **Create REST Source Manually**.

   The REST Data Source is created.

18.4.7.3 REST Data Source Definitions for Oracle Cloud SaaS Apps

Learn about viewing REST Data Source definitions for an Oracle Cloud SaaS application.

You can view a REST Data Source on the REST Data Source page as described in Editing a REST Data Source. A newly created Oracle Cloud SaaS application data source has all the operations the end point supports, a Data Profile reflecting all of its attributes, and a root resource name assigned. Once defined, you can use your REST Data Source in any APEX region while building pages. You can also use it programmatically using the appropriate procedures and functions available in the `APEX_EXEC` package.

> **See Also:**
>
> APEX_EXECin *Oracle APEX API Reference*

Operations

If the endpoint supports them all, the maximum set of wizard created operations includes:

- GET rows with filtering, ordering, and pagination support
- GET row by resource key
- POST (insert)
- PATCH (update)
- DELETE

Data Profile

The Data Profile contains a primary key column named `APEX$RESOURCEKEY` mapped to the REST service's resource key that uniquely identifies each row. This ensures that all REST services work in a consistent way against any endpoint, including ones whose underlying business object has a single-attribute primary keys, a multiple-attribute primary key, or defines an alternative unique key as a resource key. The data profile also has appropriately defined columns for all other business object attributes. Note that some Fusion Application business objects contain hundreds of standard attributes, and some can be customized to increase that number even further. Chances are very high that your APEX application does not need to work with all of these attributes, but the Create REST Data Source Wizard defines a Data Profile

column for all of the attributes. You can streamline the amount of data that will be exchanged between the APEX engine and a particular Oracle Cloud REST Data Source app as follows:

- Delete Data Profile columns you are sure you will never need.
- Mark Data Profile columns as not Visible to the APEX engine, without having to delete them.
- Ensure each region uses only the necessary columns by marking the others as Commented Out, by setting their Server-side Condition to **Never**, or by deleting them from the region's column list or the page items representing them.

APEX only requests the necessary field data for each region. There is no precise upper-limit on the maximum amount of REST endpoint business object attributes you can work with in a single request, but in practice trying to query the data of hundreds of attributes for a single region, you may encounter runtime exceptions. The actual limit will depend on the length of the attribute names involved and the total number of attributes participating in the query.

To ensure the best experience of working with REST endpoints for Fusion Application business objects containing hundreds of attributes, the Create REST Data Source Wizard defines all the data profile columns for the REST service, and then limits the ones marked Visible to a maximum of 150 columns. If your REST endpoint's business object contains custom fields (whose names are suffixed by " `_c` "), the wizard tries to give priority to include as many of those as possible in the 150 visible columns that it chooses by default. After the REST Data Source is created, at any time you can adjust which data profile columns are visible by editing the data source and updating its Data Profile.

Root Resource Name Setting

Each REST Data Source has a **Root Resource Name** setting whose value gets automatically inferred by the Create REST Data Source Wizard, but which can be set manually for Oracle Cloud SaaS Apps REST data sources upgraded from previous APEX releases. This setting identifies the case-sensitive name of the root resource for the endpoint.

When working with an endpoint such as `.../latest/employees` that defines a "top level business object" like an *employee*, the Root Resource Name will match the name of the REST resource (for example, `employees`). However, when defining an REST service for a collection of related child objects such as Vacation Requests that are owned by an employee using an endpoint URL such as `.../latest/employees/:empid/child/VacationRequests`, the Root Resource Name of this Vacation Requests data source would also be `employees` since it is part of a tree of data that is owned by the root Employee object. The setting value is optional for top-level REST endpoints since APEX can infer the correct root resource name in that case, but for data sources based on nested child resources, the setting is mandatory.

18.4.7.4 REST Data Source Runtime Features for Oracle Cloud SaaS Apps

Learn about Oracle Cloud SaaS application runtime features.

At runtime, a REST Data Dource based on a REST endpoint supports pagination, data filtering, ordering, and batch lost update protection. It supports enabling total results computation and disabling batch DML when needed.

Pagination and Total Results Computation

Both for user interface purposes and REST synchronization, a REST Data Source supports paging through rows returned by the endpoint.

To support showing the end user a pagination display such as *1-20 of **200***, go to the **Settings** tab and set **Use Total Results?** to **Yes**. This setting causes your application to show the total

number of results without having to retrieve them all. If the particular REST endpoint your are working with does not support pagination, configure its corresponding data source's **Use Pagination?** to **No**.

Data Filtering

The APEX engine delegates nearly all of the data filtering operations an end user can perform in your application to the REST endpoint for server-side execution. This includes case insensitive tokenized row search. In addition, as necessary, when defining a region you can specify an `External Filter` using a Service filter expression. This is a SQL-like predicate using the case sensitive business object attribute names from the REST payload. For reference, the attribute names are what appears in the **Selector** field of each Data Profile column's definition. The filter language supports `AND` and `OR` conjunctions, grouping using parentheses, and operations. For example, a filter might look like the following:

```
(category like 'SALES%' or (purchaseDate between '2023-07-15' and '2023-07-31'))
```

You can perform case-insensitive searches by wrapping the attribute name with `upper()`. For example:

```
upper(lastName) = 'STAR'
```

External filters do not support bind variables, but they do support APEX substitution parameters (such as `&P3_NAME.`). These variables must be used as the right-hand-side of a filter predicate and their values will be automatically single-quoted by the APEX engine when the substituted value is replaced at runtime. For example, an external where clause might resemble the following:

```
upper(lastName) = &P3_NAME.
```

Assuming that the `P3_NAME` page item contains the value `CHIP` , the REST service will see the external where clause of:

```
upper(lastName) = 'CHIP'
```

Data Ordering

The APEX engine delegates data ordering an end user can perform to the REST endpoint for server-side execution. In addition, as necessary, when defining a region based on an REST Data Source, you can specify an **External Order Bys** clause. The syntax is a comma-separated list of one or more case sensitive attribute names, each suffixed by `:asc` for ascending order or `:desc` for descending order. For example, the following external order bys expression would sort ascending by department number and then descending by date hired:

```
departmentNumber:asc,dateHired:desc
```

Batch Lost Update Protection

When an end user modifies one or more rows of data from a REST Data Source, APEX enforces lost update protection as an efficient batch operation. For example, if a user inserts four new rows in an interactive grid region, modifies three rows, and deletes two rows, clicking **Save** causes the APEX engine to perform the lost update protection checks for all nine rows in a single round trip to the REST endpoint.

Batch DML

When an end user modifies one or more rows of data from a REST Data Source, by default the APEX engine saves all the modified rows in a single, efficient batch operation. For example, suppose a user inserts four new rows in an interactive grid region, modifies three rows, and deletes two rows, and then clicks **Save**. After passing the lost update protection checks

mentioned above, the APEX engine sends all nine rows in a single round trip to the REST endpoint. This means that all row modifications succeed as a unit, or else they all fail as a unit. This simplifies application development by avoiding any custom business logic to undo a partially successful transaction.

If you notice the REST endpoint you are working with does not behave as expected using batch DML, you can disable this on the **Settings** tab. Set **Use Batch DML?** to **No** . When batch DML is disabled for a REST Data Source, each delete, insert, and update is done using a separate call to the REST endpoint.

Validation Errors Display as APEX Errors Automatically

When an end user saves changes to a page, any region based on a REST Data Source will automatically report any validation errors as an APEX error. The user therefore sees validation failures raised by server-side business object validation rules or triggers in the same, familar way they are used to seeing error messages defined inside the APEX application itself. Multiple error messages will each appear on a new line in the APEX error message. When working with the error message programmatically, use the four-character string "`<br>`" as the delimiter between error messages to split the single message into the possibly multiple, separate error messages it may contain.

18.4.7.5 Filter Parameters for Query By Example and Parent and Child Use Cases

Create filter parameters for Query By Example and parent and child use cases.

To easily build cascading select list and Query By Example (QBE) use cases, an APEX REST Data Source lets you define filter parameters on the GET operation related to the Fetch rows database action. These are URL Pattern parameters with a specially formulated parameter name that declaratively configures three aspects of runtime behavior:

- The case-sensitive name of the attribute to filter. For example:

  `SomeAttr`

- Supported filter operators to use:
  - equals (`eq`)
  - case insensitive contains (`contains`)
  - case-insensitive starts-with (`startswith`)
- The desired behavior when the parameter's *value* is null:
  - Ignore the filter if the parameter value is null (`ignoreifnull`)
  - Return no rows if the parameter value is null (`norowsifnull`)
  - Match a null value (`matchifnull`)

The filter parameter name has the following format:

`attrName_operator$behavior`

Cascading List Use Case

For example, on a data source named `SubcomponentsForComponent` suppose you want to filter on the parent attribute named `componentId` using the equals operator and return no rows if its value is null, the filter parameter name to define is:

```
componentId_eq$norowsifnull
```

When building a page with cascading select lists for `P3_COMPONENT_ID` and `P3_SUBCOMPONENT_ID`, the latter can be a select list page item based on a shared component LOV that uses the `SubcomponentsForComponent` REST Data Source. This LOV can assign the value of the data source's `componentId_eq$norowsifnull` parameter to the value of page item `P3_COMPONENT_ID`. Finally, configure `P3_COMPONENT_ID` as the parent item of the `P3_SUBCOMPONENT_ID` select list page item in Page Designer and your cascading lists will work as expected.

Query by Example Page Use Case

On a data source named `People` to perform case insensitive contains matching on `firstName` and `lastName` attributes, ignoring the respective filter if its value is null, the two filter parameter names to define are:

- `firstName_contains$ignoreifnull`
- `lastName_contains$ignoreifnull`

Back in the query by example page, you can define page items `P4_FIRST_NAME` and `P4_LAST_NAME` and configure the two REST Data Source parameters above to get their value from the respective page items. After making sure the `P4_FIRST_NAME` and `P4_LAST_NAME` are mentioned in the search results region's **Page Items to Submit** attribute, you have a working query by example page.

18.4.7.6 Working Against a Sandbox

Associate an APEX application with a sandbox by defining a sandbox name on the Shared Components, Component Settings, Oracle Cloud Applications (SaaS) REST Service page.

When Oracle Fusion Applications business administrators customize the SaaS application's business object data model, they makes their pending modifications in the context of a named, private development area called a **sandbox**. The changes made in the sandbox are not visible to end users using the production application. Oracle APEX lets you create apps whose Oracle Cloud Apps REST Data Sources can work against the REST endpoints for the business objects undergoing active customization in the sandbox.

Administrators simply configure the appropriate sandbox name in their APEX application using the Shared Components, Component Settings, Oracle Cloud Applications (SaaS) REST Service page as described in Configuring an Oracle Cloud Applications (SaaS) REST Service. After a sandbox name is defined, App Builder uses the REST endpoints reflecting the pending version of the business objects that exists in that sandbox when creating, editing, and executing any REST Data Sources in that application.

This allows you to iteratively develop your Fusion Applications business objects customizations in parallel with an APEX application that needs to work with the customized version of their business object data. If your add new custom objects in the Fusion Apps backend, you can just define a new REST Data Source to use that custom object from the sandbox. If you add new custom attributes to an existing object, APEX will see those custom attributes in the sandbox, too. If you previously defined an APEX REST Data Source for a particular Oracle SaaS REST endpoint, you can easily update it to reflect new customizations you made in the meantime. For example, if you add additional attributes in the sandbox, you can just perform the following steps to iteratively see the new custom attributes in her APEX application as well:

1. Edit the existing REST Data Source, edit the **Data Profile**, and click **Rediscover**. APEX presents any new attributes and lets you extend the data profile to include them.
2. Edit an existing region in a page that is using this REST Data Source and choose **Synchronize Columns** from the context menu. This will include the new columns into the region in question. When performing a **Synchronize Columns**, if you have marked

columns or page items in a region to be Commented Out, then they will remain commented out and only the new fields will be added as uncommented columns or page items as appropriate.

When it's time to publish the sandbox, the you just clear the sandbox name in your APEX app and it instantly begins using the now-published REST endpoint again.

18.4.7.7 Best Practices for Configuring REST Service Base URL

Learn about best practices for configuring REST Service Base URL.

When you build an APEX application, it's common to use distinct development and production environments. When that application makes use of a REST Data Source, you will often want it to use a different REST endpoint URL in DEV than in PROD. This lets you test your app in development without disturbing production end users or production data. The APEX Remote Server feature makes it easy to accommodate this usage pattern when working with REST Data Sources, including Oracle Cloud SaaS application REST Data Sources.

When defining a REST Data Source, you provide the Endpoint URL. APEX manages this URL value as a two-part string. The leading part is the **Base URL** and it comes from a particular Remote Server definition. The trailing part of the Endpoint URL is the **Service URL Path**. It's important to understand this concept because APEX makes it easy to change the value of the Remote Server **Base URL** when you deploy your app to a different environment. The value of the Remote Server **Base URL** stays sticky to the target deployment environment so you can easily deploy updated version of your app from DEV to PROD and the Remote Server **Base URL** definition in PROD stays how you configured it for PROD.

Study the URLs of the REST endpoints you need to work with, and how they need to change when switching between DEV and PROD environments. This lets you decide where to split the Endpoint URLs into a changeable Base URL part and the static Service URL Path that is the same in both environments. While running the Create REST Data Source Wizard, on the Remote Server step you can adjust the Remote Server and Service URL Path so the remote server's Base URL encapsulates the leading half of the REST service's endpoint URL that needs to change when deploying your application to another APEX instance. You can then adjust the Service URL Path to retain the trailing half of the service's endpoint URL that will always stay the same across deployments. You can also always adjust how a REST Data Source's endpoint URL is split between base URL and service path URL at a later time, too.

18.4.7.8 Overriding Default Runtime Headers If Necessary

Define a REST Data Source parameter to use a later REST framework version.

By default APEX uses version 6 of the REST framework in its communication with the endpoint. If a particuar REST Data Source you define needs to use a later REST framework version, you can achieve this by defining a module parameter (or operation parameter) of type `Header`. Name the header `REST-Framework-Version` and provide the unquoted, overridden version number you need to use as the value of the header. For example:

```
REST-Framework-Version = 7
```

By default APEX will configure the appropriate header on REST requests to use the sandbox name you configure in the Shared Components, Component Settings, Oracle Cloud Applications (SaaS) REST Service page of your application if it's value is non-null. If that app-level setting is null, then no header gets sent. If you need to override the app-level sandbox name setting for a particular REST Data Source, then you can define a module (or operation) parameter of type Header to achieve this. Name the header `Metadata-Context` and set the value to one of the following:

- `sandbox="YourOtherSandboxName"`, assuming the sandbox is named `YourOtherSandbox`
- Leave the header value blank (or null) as a signal that this particular data source (or operation) should avoid using a sandbox.

18.4.7.9 Data Profile Column Annotations

Add Data Profile Column annotations using the Additional Info attribute.

In addition to a properly configured data type, format mask (for dates), and REST payload selector, you can use the Data Profile Column **Additional Info** attribute to include a comma-separated list of one or more of the following case-sensitive tag names used to optimize how APEX communicates with the REST Data Source endpoint:

- `RemotePK` - Omit null remote primary keys from create payloads so the service defaults their values.
- `CreateOnly` - Exclude column from update payloads.
- `ReadOnly` - Exclude column from DML payloads.
- `HasDefault` - Omit a null-valued column from create payload to prioritize the serverside default.
- `Required` - Not currently enforced but useful to be aware of (and possibly enforced by APEX in the future).

18.4.7.10 Example: Creating a REST Data Source Against a Fusion Applications Endpoint

Learn how avoid an error when creating a REST Data Source against a Fusion Applications end point which uses selectively required attributes.

Some Fusion Applications end points identify a set of indexed attributes, at least one of which must be provided when retrieving the data. Since App Builder retrieves sample data from the endpoint when you define a REST Data Source, you may encounter an error when trying to use such an endpoint. This topic explains how to avoid the error during REST Data Source creation by configuring a filter that uses one of these selectively required attributes.

- Create a REST Data Source for an Endpoint with Selectively Required Attributes
  Create a REST Data Source against a Fusion Applications end point which uses selectively required attributes.
- Remove the Hard-Coded q Parameter
  Edit the REST Data Source to remove the hard-coded `q` parameter and add a filter parameter for selectively required attributes.

18.4.7.10.1 Create a REST Data Source for an Endpoint with Selectively Required Attributes

Create a REST Data Source against a Fusion Applications end point which uses selectively required attributes.

To create the REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

   d. Under Data Sources, select **REST Data Sources**.

2. On the REST Data Sources page, click **Create**.

   The Create REST Data Source Wizard appears.

3. Create REST Data Source - Select **From Scratch** and click **Next**.

4. General:

   a. REST Data Source Type - Select **Oracle Cloud Applications (SaaS) REST Service**.

   b. Name - Enter a descriptive name for this REST Data Source.

   c. URL Endpoint - Enter the endpoint of the service with Selectively Required attributes, without providing any Query String (that is, no question mark and no parameter values at this step). For example:

      `https://fa.com/fscmRestApi/resources/11.13.18.05/inventoryOnhandBalances`

   d. Click **Next**.

5. Remote Server - Accept the defaults and click **Next**.

6. Settings, Pagination Type - Accept the defaults and click **Next**.

7. Authentication, Authentication Required:

   a. Choose an existing credential if appropriate, or choose an Authentication Type and enter the relevant details for defining a new credential.

   b. Click **Discover**.

      The following error appears:

      `HTTP Status`

      `Discovery error: Oracle APEX cannot compute a data profile from the response data of type: application/vnd.oracle.adf.resourcecollection+json.`

      For this example, one of the following attributes is required: `ItemNumber`, `OrganizationCode`, `InventoryItemId`, `OrganizationId`

      To avoid this error, you need to create a filter for one for one attribute (for example, `OrganizationId`).

8. Click the **Back** arrow (<) to return to the previous Authentication step.

9. Authentication, Authentication Required:

   a. Re-enter your credentials.

   b. Click **Advanced**.

10. On Parameters:

    a. Parameter Type - Select **URL Query String**.

    b. Parameter Name - Enter `q`

    c. Value of - Enter `OrganizationId='0'`

> **Note:**
>
> Adjust the example attribute name (for example, `OrganizationId`) to instead use the name of one of the selectively required attributes required by the endpoint you want to work with. The value on the right hand side of the equals sign is not important.

d. Click **Discover**.

11. On the Preview step, click **Create REST Data Source**.

    The REST Data Source is created.

18.4.7.10.2 Remove the Hard-Coded q Parameter

Edit the REST Data Source to remove the hard-coded `q` parameter and add a filter parameter for selectively required attributes.

To remove the hard-coded `q` parameter:

1. On the REST Data Sources page, click the appropriate *REST Source Name*.

   The REST Data Source page appears.

2. Delete the `q` parameter:

   a. Find **Parameters**.

   b. Find the `q` parameter and click the **Edit** icon.

   c. Click **Delete**.

3. Add a Filter Parameter for selectively required attributes.

   a. Find **Operations**:

   b. Edit the `GET` operation with Database Action of `Fetch rows`.

   c. On the REST Source Operation page, find **Operation Parameters** and click **Add Parameter**.

   d. On Edit REST Data Source Parameter:

      - Type - Accept the default, **URL Pattern**.
      - Name - Enter the name of the parameter. For example, enter `OrganizationId_eq$norowsifnull`
        Be sure to enter the parameter name you specified when your created the REST Data Source.

   e. Click **Add Parameter**.

   f. On the REST Source Operation page, click **Apply Changes**.

   The REST Data Source page appears.

4. Click **Apply Changes** again to complete the REST Data Source editing task.

   The filter parameter you added lets you declaratively bind a page item or other value to the selectively required attribute in any region where you use this new REST Data Source. For more information on Filter Parameters, see Filter Parameters for Query By Example and Parent and Child Use Cases.

18.4.8 Working with Oracle Cloud Applications BOSS REST Services

Learn about using APEX REST Data Sources for Oracle Cloud Applications (BOSS) REST Services.

- About Oracle Cloud Applications BOSS REST Services
  Learn about REST Data Source support for Oracle Cloud Applications BOSS REST Services.
- Creating a BOSS REST Data Source
  Learn about creating a BOSS REST Data Source.

- About the BOSS REST Data Source Definition
  Learn about the BOSS REST Data Source definition.
- BOSS REST Data Source Runtime Features
  Learn about BOSS REST Data Source runtime features.
- Filter Parameters for Query By Example and Parent and Child Use Cases
  Learn about filter parameters for Query By Example (QBE) and parent and child use cases.
- Best Practices for Configuring BOSS REST Service Base URL
  Learn about best practices fo configuring BOSS REST Service Base URL
- Working with Parent and Child Data in Regions
  Learn about working with parent and child data in regions.
- Using BOSS REST Data Sources in PL/SQL
  Use BOSS REST Data Sources to show data in any APEX region
- Performing DML Operations on Objects with ETags
  Learn about performing DML operations on objects with `ETags`.
- Reading Business Objects with Nested Collections
  Define REST Data Source profile columns of type Array to work programmatically with nested collections of BOSS business objects.
- Writing Business Objects with Nested Collections
  Use the `APEX_EXEC` package to programmatically write a parent object and its nested child objects in a single payload.
- Invoking BOSS Business Object Custom Actions
  Learn about support for BOSS business object custom actions.

18.4.8.1 About Oracle Cloud Applications BOSS REST Services

Learn about REST Data Source support for Oracle Cloud Applications BOSS REST Services.

Determining the REST Data Source Type

The URL of the Fusion Applications service you want to work with identifies which type of REST Service you want to create.

For example, if the Fusion Applications REST API you want to work with contains `/api/boss/`, you would create a REST Data Source with a Type of **Oracle Cloud Applications (BOSS) REST Service**.

In contrast, if the URL contains `RestApi/resources/`, you would create a REST Data Source with a Type of **Oracle Cloud Applications (SaaS) REST Service**. Consider the following examples: `/hcmRestApi/resources/`, `/fscmRestApi/resources/`, or `/crmRestApi/resources/`.

About BOSS REST Services

To build Oracle APEX apps that query, insert, update, and delete data from Business Object Spectra Service (BOSS) REST endpoints, create a new REST Data Source with a Type of **Oracle Cloud Applications (BOSS) REST Service**. Oracle APEX supports key BOSS features including:

- Business object endpoints (both top-level and nested child collections related by composition)
- Custom business object view endpoints

- Search view endpoints for optimized full-text searching
- Extraction view endpoints with optional bind parameters

In addition, this REST Data Source type supports BOSS endpoints that use:

- Restriction types
- Enumeration types
- Composite types
- Polymorphic results

The goal of this REST Data Sources type makes using APEX against a BOSS endpoint as easy as working with a local table while delegating a maximal amount of functionality for filtering, sorting, attribute selection, and pagination to the BOSS backend.

18.4.8.2 Creating a BOSS REST Data Source

Learn about creating a BOSS REST Data Source.

To create a BOSS REST Data Source, run the Create REST Data Source Wizard. When prompted, select the REST Data Source Type, **Oracle Cloud Applications (BOSS) REST Service**. This topic describes key decision points when running the Create REST Data Source Wizard.

> **Tip:**
>
> For more details on all wizard options, see item Help or see the generic discussion, Creating a REST Data Source.

To create a REST Data Source:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Create**.

   The Create REST Data Source Wizard appears.
3. Create REST Data Source - Select **From Scratch** and click **Next**.
4. General:
   a. REST Data Source Type - Select **Oracle Cloud Applications (BOSS) REST Service**.
   b. Name - Enter a descriptive name for this REST Data Source.
   c. URL Endpoint - Enter the Endpoint URL (starting with `http://` or `https://`) for the REST Data Source.
   d. Click **Next**.
5. Remote Server - Accept the defaults.

   This step shows how APEX manages the REST API's endpoint URL as the combination of a **Base URL** and a **Service URL Path**.

6. Settings:
   a. Discover References - Set **Yes** to also include the first level of referenced objects as additional, read-only information for the BOSS endpoint when discovering and rediscovering the data profile. If set to **No**, APEX includes only the endpoint's root business object attributes. You can include others as needed by editing the Data Profile once you have created the initial REST Data Source.
   b. Use Bulk DML - Bulk DML improves performance by grouping multiple delete, update, and insert operations against this REST service into a single exchange with the remote server. Oracle recommends disabling this setting only if you encounter functional issues with a particular BOSS service endpoint while using bulk DML.
   c. Use Total Results - An Oracle Cloud Applications (BOSS) REST Service supports returning the total number of results, even if only a subset is returned (for example, the first page). In certain situations, this total row computation may add unwanted runtime overhead. You can disable APEX's use of this feature for this data source by setting the value to **No**.
   d. Use Pagination - In general, Oracle Cloud Applications (BOSS) REST services support retrieving results a page at a time. However, if this data source's endpoint does not support the "$offset" parameter, then set this property to **No** to disable pagination.
   e. Click **Next**.
7. Authentication, Authentication Required - Choose whether the new REST Data Source requires Authentication. If enabled, specify **Credentials** and **Authentication Type**.
8. To proceed, select one of the following:
   - **Advanced** - Presents advanced parameters for the Data Profile. Use **Advanced** to add additional named parameters that might be required for discovery.
   - **Discover** - Starts Discovery mode. APEX executes multiple HTTP requests in order to get information about the external Web service. One of these multiple HTTP requests can be an HTTP request to get the OpenAPI information about the Web service. APEX samples the data (JSON or XML) returned by the Web service and derives a Data Profile (consisting of metadata about the JSON attributes). Discovery results then display and confirms whether to create the REST Data Source.
9. If you click **Advanced**, the Parameter page appears.

   Use the Parameters page to configure initial REST Data Source parameters which might be required for discovery. Some REST Data Sources may require you set specific HTTP Headers (for example, `User-Agent`). After setting advanced parameters, click **Discover**.
10. After you click **Discover**, APEX invokes the REST service URL and a Preview appears. From the JSON response, APEX also retrieves the OpenAPI description of the BOSS endpoint, which it uses to derive attributes, data types, and additional information annotations to create a Data Profile.

    The information that displays depends on the attributes configured in the default view of the BOSS endpoint. It is expected that attributes not included in the default view will show as null during this preview. The Data tab displays the sample response data, formatted as a report.

    The Data Profile tab shows information about the data profile columns APEX has found during discovery. Use this tab to review the column names and data types.

    If you wish to remove a named column you can navigate with the TAB key to the remove column and press Enter, or simply by using the mouse by pressing the **X** icon.

    The Response Body tab shows information about the response body structure APEX has found during discovery.

Click **More Detail** to view additional detailed information about the REST Data Source response, such as the response body, response headers and the REST Data Source operations.

11. Click **Create REST Data Source**.

    The REST Data Source is created.

18.4.8.3 About the BOSS REST Data Source Definition

Learn about the BOSS REST Data Source definition.

You can view a REST Data Source on the REST Data Source page as described in Editing a REST Data Source. A newly created BOSS REST data source data source has the following:

- Operations defined for all those the endpoint supports.
- A data profile reflecting all its attributes.
  If you chose **Yes** for the **Discover References** setting during REST Data Source creation, the new REST Data Source's data profile will have:
  - Read-only attributes of the first level (1/many)-to-1 reference objects.
  - Where applicable, updateable accessor foreign key attributes your pages can use to assign or update the reference object by key.
- A root resource name assigned.
- A resource type properly configured.

Once defined, you can use a REST Data Source in any APEX region while building pages. You can also use it programmatically using the appropriate procedures and functions available in the `APEX_EXEC` package.

> **Tip:**
>
> For more information, see BOSS REST Data Sources in PL/SQL.

Operations

If the endpoint supports them all, the maximum set of wizard created operations includes:

- GET rows with filtering, ordering, and pagination support
- GET row by resource key
- POST (insert)
- PATCH (update)
- DELETE

Data Profile

For REST Data Sources of the resource type **Business Object/View** or **Search View** the Data Profile contains a primary key column named `APEX$RESOURCEKEY` mapped to the BOSS REST service's resource key that uniquely identifies each row. This ensures that all BOSS REST services work in a consistent way against any endpoint, including ones whose underlying business object has a single-attribute primary keys, a multiple-attribute primary key, or defines an alternative unique key as a resource key. These two business object endpoint data sources will also contain a column named `APEX$ETAG` mapped to the BOSS etag with selector `$context.etag`. Using this column is required to update or delete rows if the business

object defines an attribute of type `historyType='versionNumber'`. The Data Profile also has appropriately defined columns for all other business object attributes.

Data Profile Column Annotations

In addition to a properly configured data type, format mask (for dates), and REST payload selector, each data profile column's **Additional Information** field may include a comma-separated list of one or more of the following case-sensitive tag names used to optimize how APEX communicates with the BOSS REST endpoint. APEX's Create REST Data Source Wizard can automatically infer most of them, with the exception of `HasDefault` and `Subtype` which need to be added manually as appropriate.

- `RemotePK` - Omit null remote primary keys from create payloads so the service defaults their values.
- `ETag` - Identifies a column as the row version indicator used by BOSS as the ETag value.
- One of the following, or neither if the attribute is always updateable:
  - `CreateOnly` - Exclude column from update payloads.
  - `ReadOnly` - Exclude column from DML payloads.
- `HasDefault` - Omit a null-valued column from create payload to prioritize the server-side default.
- `Required` - Not currently enforced but useful to be aware of (and possibly enforced by APEX in the future).
- `Subtype` - Exclude field from $fields list and from create/update payloads if null.
- `Sortable` - indicates that the column is sortable.
- `AccessorFK` - Indicates that the column is the accessor foreign key for assigning a reference object.

Data Profile Considerations

Some Fusion Applications business objects contain hundreds of standard attributes. If you have opted to discover references, then adding in the first level of related objects can produce a data profile with many hundreds of attributes. The first 150 will be marked as `Visible = true` and the remaining attributes will have `false` for that property. The Visible property of a Data Profile column controls whether it appears in the column list in App Builder when creating regions. Chances are very high that your APEX application does not need to work with all of these attributes, but the Create REST Data Source Wizard defines a Data Profile column for all of the attributes. Developers have two easy ways to streamline the amount of data that will be exchanged between the APEX engine and a particular BOSS REST data source:

- You can delete Data Profile columns you are sure you will never need.
- You can mark Data Profile columns as not Visible to the APEX engine, without having to delete them.
- You can ensure each region uses only the necessary columns, either deleting the others from the region's column list, or alternatively commenting them out or setting their Server-side Condition to Never.

There is no precise upper-limit on the maximum amount of BOSS REST endpoint business object attributes you can work with in a single request, but in practice, trying to query the data of hundreds of attributes for a single region, you may encounter runtime exceptions. The actual limit will depend on the length of the attribute names involved and the total number of attributes participating in the query. After the BOSS REST Data Source is created, at any time you can

adjust which data profile columns are Visible by editing the data source and visiting its Data Profile.

Resource Name Setting

Each BOSS REST Data Source has a Root Resource Name setting whose value gets automatically inferred by the Create REST Data Source Wizard. This setting identifies the case-sensitive name of the root resource for the endpoint.

When working with an endpoint, such as `.../v1/employees`, that defines a top level business object, such as an employee, the root resource name will match the name of the REST resource (for example, employees). However, when defining a BOSS REST service for a collection of related child objects such as Vacation Requests that are owned by an employee using an endpoint URL, such as `.../v1/employees/:empid/vacationRequests`, the root resource name of this Vacation Requests data source would also be employees since it is part of a tree of data that is owned by the root Employee object.

18.4.8.4 BOSS REST Data Source Runtime Features

Learn about BOSS REST Data Source runtime features.

At runtime, a REST Data Source based on a BOSS REST endpoint supports pagination, data filtering, ordering, and lost update protection. In addition, it also supports enabling total results computation and disabling bulk DML when needed.

- Pagination and Total Results Computation
  Learn about BOSS REST Data Source Pagination and total results computations.
- Data Filtering
  Learn about data filtering operations.
- Data Ordering
  Learn about how the APEX delegates data ordering.
- Lost Update Protection
  Learn about BOSS REST Data Source lost update protection.
- ETag Support
  Learn about ETag support.
- Bulk DML
  Learn about support for Bulk DML.
- Validation Errors Display as APEX Errors Automatically
  Learn about the display of validation errors.

18.4.8.4.1 Pagination and Total Results Computation

Learn about BOSS REST Data Source Pagination and total results computations.

Both for user interface purposes and REST synchronization, a BOSS REST Data Source supports paging through rows returned by the endpoint.

To support showing the end user a pagination display such as *1-20 of 200*, set the data source's **Use Total Results** setting to `Yes`. This enables your application to easily show the total number of results without having to retrieve them all. If the particular BOSS REST endpoint your are working with does not support pagination, configure its corresponding data source's Use Pagination setting to **No**.

18.4.8.4.2 Data Filtering

Learn about data filtering operations.

The APEX engine delegates nearly all of the data filtering operations an end-user can perform in your application to the BOSS REST endpoint for server-side execution. This includes case-insensitive tokenized row search. When the Resource Type of the BOSS REST Data Source is **Business Object/View** or **Extraction View**, then the row search is translated into an appropriate combination of case-insensitive BOSS `like` filters. In contrast, when the Resource Type is **Search View**, the row search is translated into appropriate BOSS keyword search for faster full-text searching.

In addition, when defining a region, you can specify an **External Filter** using a BOSS REST Service filter expression. This is a SQL-like predicate using the case-sensitive business object attribute names from the REST payload. For reference, the attribute names are what appears in the `Selector` field of each data profile column's definition. The filter language supports `AND` and `OR` conjunctions, grouping using parentheses, and operations.

If a BOSS REST Data Source defines an URL Pattern parameter named `additionalExternalFilter`, then this parameter can be assigned a value at runtime to handle more dynamic filtering use cases. The value must be a syntactically legal BOSS filter predicate with a balanced set of left and right parenthesis, or an `invalid filter` error will result at runtime.

The following is example of what a filter might look like:

```
(category like 'SALES%' or (purchased >= '2023-07-15' and purchased
<='2023-07-31'))
```

You can perform case-insensitive searches by using the tilde-modified operators like this:

- `lastName ~= 'STAR'`
- `lastName ~like '%STAR%'`

External filters do not support bind variables, but they do support APEX substitution parameters (for example, `&P3_NAME.`). Substitution parameters must be used as the right side of a filter predicate and their values will be automatically single-quoted by the APEX engine when the substituted value is replaced at runtime. For example, an external WHERE clause might look similar to this:

```
lastName ~= &P3_NAME.
```

This example assumes the P3_NAME page item contains the value `CHIP`, the BOSS REST service will see the external where clause of:

```
lastName ~= 'CHIP'
```

This also works well for numerical values since the BOSS server's filter processing allows numerical values to also be quoted on the right-hand side of a filter expression like:

```
numberOfDependents > &P3_NUM_DEPS.
```

This example assumes the value of the page item is 2 will turn into:

```
numberOfDependents > '2'
```

18.4.8.4.3 Data Ordering

Learn about how the APEX delegates data ordering.

The APEX engine delegates data ordering an end-user can perform to the BOSS REST endpoint for server-side execution.

In addition, as necessary when defining a region based on a BOSS REST Data Source, you can specify an **External Order Bys** clause. Its syntax is a comma-separated list of one or more case-sensitive attribute names, each suffixed by `:asc` for ascending order or `:desc` for descending order. For example, the following external order bys expression would sort ascending by department number and then descending by date hired:

```
departmentNumber:asc,dateHired:desc
```

Note that an attempt to order by a BOSS attribute that is not configured to be sortable will result in a runtime error from the REST API. If you need to sort on a non-sortable attribute, this can be done by using the APEX Local Post-Processing feature that all REST Data Sources have available to use.

18.4.8.4.4 Lost Update Protection

Learn about BOSS REST Data Source lost update protection.

When an end user modifies one or more rows of data from a BOSS REST Data Source, APEX enforces lost update protection. If the user inserts four new rows in an interactive grid region, modifies three rows, and deletes two rows, when they click **Save**, the APEX engine performs the lost update protection checks for the five updated and deleted rows by retrieving each one using its resource key from the BOSS REST endpoint. If the BOSS REST Data Source has an `ETag`, you can use the `APEX$ETAG` column as the row version column for more efficient lost update detection checks by configuring the **Lost Update Type** on a form or grid to be **Row Version Column** and configuring `APEX$ETAG` as the row version column name.

> **Note:**
>
> When working with a BOSS object that has an attribute of type `historyType='versionNumber'`, then it is mandatory to always include the `ETag` column (by default named `APEX$ETAG`) in the list of columns used by a form region, grid region, or in programmatic use with `APEX_EXEC` package. Furthermore, it is important when defining the column to work with programmatically to mark the `ETag` column using the `p_is_checksum => true parameter` to `apex_exec.add_column()`. If the business object does not have an attribute of type `historyType='versionNumber'`, the the object will not have an `ETag`. In that case, you can delete the `APEX$ETAG` column from the Data Profile as its value will always be null and will not be used during updates or deletes.

18.4.8.4.5 ETag Support

Learn about ETag support.

When a BOSS business object defines a history attribute of type `historyType='versionNumber'` then the object will have an ETag returned in the response payload in the selector `$context.etag`. The BOSS adapter defines an `APEX$ETAG` column in the data profile mapped to this `$context.etag` selector, and including the **Additional Information**

annotation tag of `ETag`. At runtime, the BOSS adapter will include a non-null ETag value with update and delete operations as expected by the BOSS runtime. When configuring APEX regions like form or interactive grid based on a REST Data Source for a BOSS object with an ETag, developers must set the **Lost Update Type** to the value **Row Version Column** and choose `APEX$ETAG` for the row version column name. Alternatively, to have BOSS perform the lost update protection only on the server-side, you can disable APEX's lost update protection on a form or grid region when it's using a REST Data Source for a BOSS object that has an ETag.

18.4.8.4.6 Bulk DML

Learn about support for Bulk DML.

When an end user modifies one or more rows of data from a BOSS REST Data Source, by default the APEX engine saves all the modified rows in a single, efficient bulk operation. For example, suppose a user inserts four new rows in an interactive grid region, modifies three rows, and deletes two rows, and then clicks **Save**. After passing the lost update protection checks mentioned above, the APEX engine sends all nine rows in a single round trip to the BOSS REST endpoint.

If you notice the BOSS REST endpoint you are working with does not behave as expected using bulk DML, you can disable this by setting the **Use Bulk DML** data source setting to **No**. When bulk DML is disabled for a BOSS REST Data Source, each delete, insert, and update is done using a separate call to the BOSS REST endpoint. In addition, Bulk DML only works for top level business object endpoints. When using REST Data Sources for nested child resources, set **Use Bulk DML** to **No**.

18.4.8.4.7 Validation Errors Display as APEX Errors Automatically

Learn about the display of validation errors.

When an end user saves changes to a page, any region based on a BOSS REST Data Source will automatically report any validation errors as an APEX error. The user therefore sees validation failures raised by server-side business object validation rules or triggers in the same, familar way they are used to seeing error messages defined inside the APEX application itself. Multiple error messages will each appear on a new line in the APEX error message.

18.4.8.5 Filter Parameters for Query By Example and Parent and Child Use Cases

Learn about filter parameters for Query By Example (QBE) and parent and child use cases.

About Building Query By Example Use Cases

To easily build a cascading select list and Query By Example (QBE) use cases, a BOSS REST Data Source enables you define filter parameters on the `GET` operation related to the `Fetch rows` database action. These are URL Pattern parameters with a specially formulated parameter name that declaratively configures three aspects of runtime behavior:

- The case-sensitive name of the attribute to filter. For example:

  `someAttr`

- The filter operator to use:
  - equals (`eq`)
  - case-insensitive contains (`contains`)
  - case-insensitive starts-with (`startswith`)

- The desired behavior when the parameter's value is null:
  - Ignore the filter if the parameter value is null (`ignoreifnull`)
  - Return no rows if the parameter value is null (`norowsifnull`)
  - Match a null value (`matchifnull`)

The filter parameter name has this format:

```
attrName_operator$behavior
```

Cascading List Use Case

For example, on a data source named `SubcomponentsForComponent` to filter on the parent attribute named `componentId` using the `equals` operator and `return no rows` if its value is null, the filter parameter name to define is:

```
componentId_eq$norowsifnull
```

When building a page with cascading select lists for `P3_COMPONENT_ID` and `P3_SUBCOMPONENT_ID`, the latter can be a select list page item based on a shared component LOV that uses the `SubcomponentsForComponent` BOSS REST Data Source. This LOV can assign the value of the data source's `componentId_eq$norowsifnull` parameter to the value of page item `P3_COMPONENT_ID`. Then, configure `P3_COMPONENT_ID` as the Parent Item of the `P3_SUBCOMPONENT_ID` select list page item in the Page Designer and your cascading lists will work as expected.

Query by Example Page Use Case

Suppose you have a data source named `People`. To perform case-insensitive contains matching on the `firstName` and `lastName` attributes, ignoring the respective filter if its value is null, the two filter parameter names to define are:

- `firstName_contains$ignoreifnull`
- `lastName_contains$ignoreifnull`

On the query by example page, you can define page items `P4_FIRST_NAME` and `P4_LAST_NAME` and configure the two REST Data Source parameters above to get their value from the respective page items. After making sure the `P4_FIRST_NAME` and `P4_LAST_NAME` are mentioned in the search results region's **Page Items to Submit** property, you have a working query by example page.

18.4.8.6 Best Practices for Configuring BOSS REST Service Base URL

Learn about best practices fo configuring BOSS REST Service Base URL

When you build an APEX application, many developers use distinct development (DEV) and production (PROD) environments. When that application makes use of a REST Data Source, you will often want it to use a different REST endpoint URL in DEV than in PROD. This approach lets you test your application in DEV without disturbing PROD end users or PROD data. Oracle APEX's Remote Server feature makes it easy to accommodate this usage pattern when working with REST Data Sources, including BOSS REST Data Sources.

When defining a REST Data Source, you provide the Endpoint URL. APEX manages this URL value as a two-part string. The leading part is the **Base URL** and it comes from a particular **Remote Server** definition. The trailing part of the **Endpoint URL** is the **Service URL Path**. APEX makes it easy to change the value of the Remote Server Base URL when you deploy your app to a different environment. The value of the Remote Server Base URL stays "sticky"

to the target deployment environment so you can easily deploy updated version of your app from DEV to PROD and the Remote Server Base URL definition in PROD stays how you configured it for PROD.

Study the URLs of the REST endpoints you need to work with and determine how they need to change when switching between DEV and PROD environments. This lets you decide where to "split" the Endpoint URLs into a changeable Base URL part and the static Service URL Path that is the same in both environments. While using the Create REST Data Source Wizard, on the Remote Server step you can adjust the Remote Server and Service URL Path so the remote server's Base URL encapsulates the leading half of the REST service's endpoint URL that needs to change when deploying your application to another APEX instance. You can then adjust the **Service URL Path** to retain the trailing half of the service's endpoint URL that will always stay the same across deployments. You can always adjust how a REST Data Source's endpoint URL is split between base URL and service path URL at a later time.

18.4.8.7 Working with Parent and Child Data in Regions

Learn about working with parent and child data in regions.

A BOSS business object can be related to one or more collections of composed child objects. For example, an `Order` object can compose a related collection of `OrderLine` child business objects. A parent object, like `Order`, can have multiple composed child collections, and each collection can comprise business objects that, in turn, contain composed collections. For example, each `OrderLine` business object that is part of an `Order` can have a composed collection of `OrderLineFulfillment` child business objects.

When working with a root business object like `Order` that has a composed child collection, clients like APEX can read and write child objects using a nested resource name that includes the resource key of a specific parent object. For example, to list the `OrderLine` child objects for order 12345, the REST endpoint would end with the path:

```
/orders/12345/orderLines
```

In the previous example, `12345` is the resource key of a specific order, and `orderLines` is the name of the Order object's accessor attribute of type `object-collection`.

To work with the child objects of a specific `OrderLine` with resource key 67890, on a specific `Order` with resource key 12345, the REST endpoint URL would end with:

```
/orders/12345/orderLines/67890/orderLineFulfillments
```

The `orderLineFulfillments` is the name of the `OrderLine` object's accessor attribute of type `object-collection`.

You can define REST Data Sources for the nested resources, using a named parameter in the URL path like this for the order lines:

```
/orders/:orderId/orderLines
```

To access a second-level child, the REST Data Source's endpoint URL can contain multiple URL path parameters, for example:

```
/orders/:orderId/orderLines/:orderLineId/orderLineFulfillments
```

When using these parameterized REST Data Sources in a region on a page, you can then bind the value of the REST Data Source's URL path parameters under the Parameters node in the Rendering tab for the region to use an appropriate page item.

18.4.8.8 Using BOSS REST Data Sources in PL/SQL

Use BOSS REST Data Sources to show data in any APEX region

When using BOSS REST Data Sources based on business object endpoints with form and interactive grid regions, you can also insert, update, and delete data. None of these capabilities requires writing any code to achieve, however sometimes you may need to interact with a BOSS endpoint programmatically using the `APEX_EXEC` package. Use cases include:

- Creating pipelined table functions that programmatically reshape BOSS data.
- Working programmatically with BOSS endpoints in APEX business logic, workflows, automations, or background execution chain processes.
- Arbitrary scripting tasks involving interacting with BOSS endpoints.

Each BOSS REST data source has a unique identifier that APEX calls its static id. For example, an APEX developer creating a REST Data Source for a BOSS endpoint `../v1/opportunities` might configure its static id to be `opportunities`. This is the identifier you use when working with the functions in the `APEX_EXEC` package.

Retrieving Data from a BOSS Endpoint

To retrieve data from a BOSS endpoint, use the `apex_exec.open_rest_source_query` function to create an `apex_exec.t_context` object that works like a cursor, then use a while `apex_exec.next_row(ctx) loop` to iterate through the results. For each iteration, use the `apex_exec.get_xxx()` functions to access the attribute values from the current query result row. When done, call `apex_exec.close()` to close the context. For example:

```
declare
  l_ctx apex_exec.t_context;
  -- other declarations here
begin
  -- Process top five winnable opportunties for salesperson 12345 by amount
  -- get_opportunities_columns() returns apex_exec.t_columns for columns
  -- corresponding to fields: optyId,amount,customerName,closeDate
  l_ctx := apex_exec.open_rest_source_query(
             p_static_id              => 'opportunities',
             p_columns                => get_opportunities_columns,
             p_external_filter_expr   => 'resourceId = 12345 and winPct >
0.80',
             p_external_order_by_expr => 'amount:desc'
             p_max_rows               => 5);
  while apex_exec.next_row(l_ctx) loop
     l_amount := apex_exec.get_number(l_ctx,'AMOUNT');
     l_customer := apex_exec.get_varchar2(l_ctx,'CUSTOMERNAME');
     l_closes := apex_exec.get_date(l_ctx,'CLOSEDATE');
     -- Work with data here...
  end loop;
  apex_exec.close(l_ctx);
exception
  when others then
     apex_exec.close(l_ctx);
     raise;
end;
```

Inserting Data Using a BOSS Endpoint

To insert data using a BOSS endpoint, use the `apex_exec.open_rest_source_dml_context` function to create an `apex_exec.t_context` object that works like a temporary rowset. Then for each new row you want to add, call `apex_exec.add_dml_row()` and then use `apex_exec.set_value()` to set the value of each attribute necessary. Finally, to process the pending changes call `apex_exec.execute_dml()`. When done, call `apex_exec.close()` to close the context. For example:

```
declare
  l_ctx apex_exec.t_context;
  -- other declarations here
begin
  -- Insert two new opportunities
  -- get_opportunities_columns() returns apex_exec.t_columns for columns
  -- corresponding to fields:
APEX$ResourceKey,optyId,amount,customerName,closeDate
  l_ctx := apex_exec.open_rest_source_dml_context(
             p_static_id             => 'opportunities',
             p_columns               => get_opportunities_columns,
             p_lost_update_detection => apex_exec.c_lost_update_none);
  apex_exec.add_dml_row(l_ctx, apex_exec.c_dml_operation_insert);
  apex_exec.set_value(l_ctx,'AMOUNT',45000);
  apex_exec.set_value(l_ctx,'CUSTOMERNAME','Tesla');
  apex_exec.set_value(l_ctx,'CLOSEDATE','2024-05-11');
  apex_exec.add_dml_row(l_ctx, apex_exec.c_dml_operation_insert);
  apex_exec.set_value(l_ctx,'AMOUNT',1500000);
  apex_exec.set_value(l_ctx,'CUSTOMERNAME','Apple');
  apex_exec.set_value(l_ctx,'CLOSEDATE','2024-06-30');
  -- Save the changes
  apex_exec.execute_dml(l_ctx);
  apex_exec.close(l_ctx);
exception
  when others then
     apex_exec.close(l_ctx);
     raise;
end;
```

Updating Existing Row Using a BOSS Endpoint

To update data using a BOSS endpoint, use the `apex_exec.open_rest_source_dml_context` function to create an `apex_exec.t_context` object that works like a temporary rowset. Then for each row you want to update, call `apex_exec.add_dml_row()` with the update operation constant and then use `apex_exec.set_value()` to set the value of each attribute necessary. Finally, to process the pending changes call `apex_exec.execute_dml()`. Often you will preface this with a query to retrieve the row first to consider its existing values, before updating some of them. In that case you can call `apex_exec.set_values()` to copy all the row values from the queried row into the row to be updated. For example:

```
declare
  l_ctx apex_exec.t_context;
  l_dml_ctx apex_exec.t_context;
  -- other declarations here
begin
```

```
  -- Retrieve and update opportunity with resource key 98765
  -- to increase its amount by 10% and close date by one week
  -- get_opportunities_columns() returns apex_exec.t_columns for columns
  -- corresponding to fields: optyId,amount,customerName,closeDate
  l_ctx := apex_exec.open_rest_source_query(
           p_static_id              => 'opportunities',
           p_columns                => get_opportunities_columns,
           p_external_filter_expr   => 'optyId = 98765'
           p_max_rows               => 1);
  if not apex_exec.next_row(l_ctx) then
    raise_application_error(-20001,'Opportunity 98765 not found');
  end if;
  l_dml_ctx := apex_exec.open_rest_source_dml_context(
                p_static_id             => 'opportunities',
                p_columns               => get_opportunities_columns,
                p_lost_update_detection => apex_exec.c_lost_update_implicit);
  apex_exec.add_dml_row(l_ctx, apex_exec.c_dml_operation_update);
  apex_exec.set_row_version_checksum(
    p_context   => l_dml_context,
    p_checksum  => apex_exec.get_row_version_checksum(l_ctx));
  -- Copy the values of the fetched opportunity into the update row
  apex_exec.set_values(
      p_context         => l_dml_ctx,
      p_source_context  => l_ctx );
  -- Adjust the amount and closeDate appropriately
  apex_exec.set_value(l_dml_ctx,'AMOUNT',
                      apex_exec.get_number(l_dml_ctx,'AMOUNT') * 1.1);
  apex_exec.set_value(l_dml_ctx,'CLOSEDATE',
                      apex_exec.get_date(l_dml_ctx,'CLOSEDATE') + 7);
  -- Save the changes
  apex_exec.execute_dml(l_dml_ctx);
  apex_exec.close(l_dml_ctx);
  apex_exec.close(l_ctx);
exception
  when others then
    apex_exec.close(l_dml_ctx);
    apex_exec.close(l_ctx);
      raise;
end;
```

Deleting an Existing Row Using a BOSS Endpoint

To delete data using a BOSS endpoint, use use the `apex_exec.open_rest_source_dml_context` function to create an `apex_exec.t_context` object that works like a temporary rowset. Then for each row you want to delete, call `apex_exec.add_dml_row()` with the delete operation constant and then use **apex_exec.set_value()** to set the value of the resource key column `APEX$RESOURCEKEY`. Finally, to process the pending changes call `apex_exec.execute_dml()`. For example:

```
declare
  l_dml_ctx apex_exec.t_context;
  -- other declarations here
begin
  -- Delete opportunity with resource key 98765
  -- get_resource_key_columns returns apex_exec.t_columns for columns
```

```
  -- corresponding to fields: APEX$ResourceKey
  l_dml_ctx := apex_exec.open_rest_source_dml_context(
                 p_static_id => 'opportunities',
                 p_columns => get_resource_key_columns,
                 p_lost_update_detection => apex_exec.c_lost_update_implicit);
  apex_exec.add_dml_row(l_dml_ctx, apex_exec.c_dml_operation_delete);
  -- Set the resource key of the row to be deleted
  apex_exec.set_value(l_dml_ctx,'APEX$RESOURCEKEY', '98765');
  -- Save the changes
  apex_exec.execute_dml(l_dml_ctx);
  apex_exec.close(l_dml_ctx);
exception
   when others then
     apex_exec.close(l_dml_ctx);
      raise;
end;
```

> **Note:**
>
> If you need to first fetch the row to reason over its attributes before deciding to delete it, then follow the update example above but use the delete operation instead of the update operation when adding the DML row. Finally, to process the pending changes call `apex_exec.execute_dml()`.

18.4.8.9 Performing DML Operations on Objects with ETags

Learn about performing DML operations on objects with `ETags`.

When working with a BOSS object that has an attribute of type `historyType='versionNumber'` then it is mandatory to always include the `ETag` column (by default named `APEX$ETAG`) in the list of columns used by a form region, grid region, or in programmatic use with `APEX_EXEC` package. Furthermore, it is important when defining the column to work with programmatically to mark the `ETag` column using the `p_is_checksum => true` parameter to `apex_exec.add_column()`. If the business object does not have an attribute of type `historyType='versionNumber'`, the object will not have an `ETag`. In that case, you can delete the `APEX$ETAG` column from the data profile as its value will always be null and will not be used during updates or deletes.

18.4.8.10 Reading Business Objects with Nested Collections

Define REST Data Source profile columns of type Array to work programmatically with nested collections of BOSS business objects.

When defining an array column, use the relative path from the parent array for the **Selector**. In the **Remote Attribute Name** field, enter the full path from the root if that is different from the selector. Then, you can use the `APEX_EXEC` package to work with the nested arrays by using the `open_array()` and `close_array()` functions to set the context of the current array to work with. :

For example, suppose you are working with an `Order` business object, that has a composed collection of `OrderLineItem` business objects, each of which has a composed collection of `OrderLineFulfillment` objects. The following tables shows some example data profile columns from an Orders with Details REST Data Source with static id `orders_with_details`:

Column Name / *Parent Column*	Data Type	Selector / *Remote Attribute Name*
ORDER_STATUS	VARCHAR2	`orderStatus`
TOTAL_AMOUNT	NUMBER	`totalAmount`
ORDERLINES	*Array*	`orderLines.items`
ORDERLINES_QUANTITY *ORDERLINES*	NUMBER	`quantity` *`orderLines.quantity`*
ORDERLINES_PRODUCTNAME *ORDERLINES*	VARCHAR2	`product.productName` *`orderLines.product.productName`*
FULFILLMENTS *ORDERLINES*	*Array*	`fulfillments.items` *`orderLines.fulfillments.items`*
FULLFILLMENTS_QUANTITY *FULFILLMENTS*	NUMBER	`quantity` *`orderLines.fulfillments.quantity`*
FULLFILLMENTS_WAREHOUSE *FULFILLMENTS*	VARCHAR2	`warehousedProduct.warehouse.name` *`orderLines.fulfillments.warehousedProduct.warehouse.name`*

To retrieve the three-level hierarchy of data for an order whose id is passed to a `get_order()` function, write code like the following example function.

```
function get_order_exec(
   p_order_id in number)
   return t_order
is
   l_order t_order;
   l_ctx apex_exec.t_context;
   l_additional varchar2(255);
   l_cur_line t_order_line;
   l_ord_line pls_integer;
   l_ful_line pls_integer;
begin
   -- get_order_columns() returns apex_exec.t_columns for all fields
   l_ctx := apex_exec.open_rest_source_query(
      p_static_id               => 'orders_with_details',
      p_columns                 => get_order_columns,
      p_external_filter_expr    => 'orderId = '||p_order_id);
   if apex_exec.next_row(l_ctx) then
      l_order.order_id := apex_exec.get_number(l_ctx, 'ORDERID');
      l_order.tax_amount := apex_exec.get_number(l_ctx, 'TAXAMOUNT');
      l_order.total_amount := apex_exec.get_number(l_ctx, 'TOTALAMOUNT');
      l_order.order_date := apex_exec.get_date(l_ctx, 'ORDERDATE');
      l_order.order_status := apex_exec.get_varchar2(l_ctx, 'ORDERSTATUS');
      l_order.customer_first_name := apex_exec.get_varchar2(l_ctx,
'CUSTOMER_FIRSTNAME');
      l_order.customer_last_name := apex_exec.get_varchar2(l_ctx,
'CUSTOMER_LASTNAME');
      l_order.customer_phone := apex_exec.get_varchar2(l_ctx,
'CUSTOMER_PHONE');
      l_order.customer_email := apex_exec.get_varchar2(l_ctx,
```

```
'CUSTOMER_EMAILADDRESS');
      l_order.customer_loyalty_level := apex_exec.get_varchar2(l_ctx,
'CUSTOMER_LOYALTYLEVEL');
      l_additional := apex_exec.get_varchar2(l_ctx, 'SHIPTO_ADDITIONAL');
      l_order.ship_address := apex_exec.get_varchar2(l_ctx, 'SHIPTO_ADDRESS')
                              || case when l_additional is not null then ', '
end
                              || l_additional;
      l_order.ship_city := apex_exec.get_varchar2(l_ctx, 'SHIPTO_CITY');
      l_order.ship_state := apex_exec.get_varchar2(l_ctx, 'SHIPTO_STATE');
      l_order.ship_zip := apex_exec.get_varchar2(l_ctx, 'SHIPTO_ZIP');
      -- open & loop over Order Lines for current order
      apex_exec.open_array(l_ctx,'ORDERLINES');
      while apex_exec.next_array_row(l_ctx) loop
          l_ord_line := l_order.order_lines.count + 1;
          l_order.order_lines(l_ord_line).quantity
               := apex_exec.get_number(l_ctx, 'ORDERLINES_QUANTITY');
          l_order.order_lines(l_ord_line).list_price
               := apex_exec.get_number(l_ctx, 'ORDERLINES_LISTPRICE');
          l_order.order_lines(l_ord_line).amount
               := apex_exec.get_number(l_ctx, 'ORDERLINES_AMOUNT');
          l_order.order_lines(l_ord_line).product_name
               := apex_exec.get_varchar2(l_ctx, 'ORDERLINES_PRODUCTNAME');
          -- open & loop over Order Line Fulfillments for current order line
          apex_exec.open_array(l_ctx,'FULFILLMENTS');
          while apex_exec.next_array_row(l_ctx) loop
              l_ful_line :=
l_order.order_lines(l_ord_line).fulfillments.count + 1;

l_order.order_lines(l_ord_line).fulfillments(l_ful_line).quantity
                  := apex_exec.get_number(l_ctx, 'FULFILLMENTS_QUANTITY');

l_order.order_lines(l_ord_line).fulfillments(l_ful_line).warehouse_name
                  := apex_exec.get_varchar2(l_ctx, 'FULFILLMENTS_WAREHOUSE');
          end loop;
          apex_exec.close_array(l_ctx);
      end loop;
      apex_exec.close_array(l_ctx);
  end if;
  apex_exec.close(l_ctx);
  return l_order;
end get_order_exec;
```

The code above references a `get_order_columns()` function shown below. It returns the `apex_exec.t_columns` table of `apex_exec.t_column` records defining the columns participating in the query:

```
function get_order_columns
return apex_exec.t_columns
is
    l_columns apex_exec.t_columns;
begin
    apex_exec.add_column( /* orderId */
        p_columns         => l_columns,
        p_column_name     => 'ORDERID',
```

```
        p_data_type       => apex_exec.c_data_type_number);
    apex_exec.add_column( /* orderDate */
        p_columns         => l_columns,
        p_column_name     => 'ORDERDATE',
        p_data_type       => apex_exec.c_data_type_date);
    -- etc. (other top-level columns here)
    apex_exec.add_column( /* orderLines.items */
        p_columns         => l_columns,
        p_column_name     => 'ORDERLINES',
        p_data_type       => apex_exec.c_data_type_array);
    apex_exec.add_column( /* orderLines.product.productName */
        p_columns         => l_columns,
        p_column_name     => 'ORDERLINES_PRODUCTNAME',
        p_data_type       => apex_exec.c_data_type_varchar2,
        p_parent_column_path => '"ORDERLINES"');
    apex_exec.add_column( /* orderLines.quantity */
        p_columns         => l_columns,
        p_column_name     => 'ORDERLINES_QUANTITY',
        p_data_type       => apex_exec.c_data_type_number,
        p_parent_column_path => '"ORDERLINES"');
    apex_exec.add_column( /* orderLines.listPrice */
        p_columns         => l_columns,
        p_column_name     => 'ORDERLINES_LISTPRICE',
        p_data_type       => apex_exec.c_data_type_number,
        p_parent_column_path => '"ORDERLINES"');
    apex_exec.add_column( /* orderLines.amount */
        p_columns         => l_columns,
        p_column_name     => 'ORDERLINES_AMOUNT',
        p_data_type       => apex_exec.c_data_type_number,
        p_parent_column_path => '"ORDERLINES"');
    apex_exec.add_column( /* orderLines.fulfillments.items */
        p_columns         => l_columns,
        p_column_name     => 'FULFILLMENTS',
        p_data_type       => apex_exec.c_data_type_array,
        p_parent_column_path => '"ORDERLINES"');
    apex_exec.add_column( /*
orderLines.fulfillments.warehousedProduct.warehouse.name */
        p_columns         => l_columns,
        p_column_name     => 'FULFILLMENTS_WAREHOUSE',
        p_data_type       => apex_exec.c_data_type_varchar2,
        p_parent_column_path => '"ORDERLINES"."FULFILLMENTS"');
    apex_exec.add_column( /* orderLines.fulfillments.quantity */
        p_columns         => l_columns,
        p_column_name     => 'FULFILLMENTS_QUANTITY',
        p_data_type       => apex_exec.c_data_type_number,
        p_parent_column_path => '"ORDERLINES"."FULFILLMENTS"');
    return l_columns;
end get_order_columns;
```

18.4.8.11 Writing Business Objects with Nested Collections

Use the `APEX_EXEC` package to programmatically write a parent object and its nested child objects in a single payload.

> **Tip:**
>
> For all create, update, and delete use cases to work as expected, the top-level REST Data Source must configured so that **Use Bulk DML** is set to **Yes**.

Consider the following example:

```
function quick_create_order_exec(
    p_customer_id  in number,
    p_product_id1  in number,
    p_quantity1    in number,
    p_product_id2  in number,
    p_quantity2    in number)
    return            number
is
    l_ctx        apex_exec.t_context;
    l_order_id   number;
begin
    l_ctx := apex_exec.open_rest_source_dml_context(
                p_static_id             => 'orders_with_details',
                p_columns               => get_order_create_columns,
                p_lost_update_detection => apex_exec.c_lost_update_none);
    apex_exec.add_dml_row(l_ctx, apex_exec.c_dml_operation_insert);
    apex_exec.set_value(l_ctx, 'CUSTOMERID', p_customer_id);
    -- Set the context to the ORDERLINES array
    -- and add the first row in the array
    apex_exec.add_dml_array_row(l_ctx, 'ORDERLINES');
    apex_exec.set_value(l_ctx, 'ORDERLINES_PRODUCTID', p_product_id1);
    apex_exec.set_value(l_ctx, 'ORDERLINES_QUANTITY', p_quantity1);
    if     p_quantity2 is not null
        and p_product_id2 is not null
    then
        -- ORDERLINES array is already current. Add another row to array
        apex_exec.add_dml_array_row(l_ctx);
        apex_exec.set_value(l_ctx, 'ORDERLINES_PRODUCTID', p_product_id2);
        apex_exec.set_value(l_ctx, 'ORDERLINES_QUANTITY', p_quantity2);
    end if;
    -- Do the DML operation
    apex_exec.execute_dml(l_ctx);
    -- Retrieve system-assigned order number & return it
    apex_exec.set_current_row(l_ctx,1);
    l_order_id := apex_exec.get_varchar2(l_ctx,'APEX$RESOURCEKEY');
    apex_exec.close(l_ctx);
    return l_order_id;
end quick_create_order_exec;
```

The previous code example above references a `get_order_create_columns()`. The following example returns the `apex_exec.t_columns` table of `apex_exec.t_column` records defining the columns participating in the DML operation.

```
function get_order_create_columns
    return apex_exec.t_columns
    is
        l_columns apex_exec.t_columns;
    begin
        apex_exec.add_column( /* APEX$ResourceKey */
            p_columns         => l_columns,
            p_column_name     => 'APEX$RESOURCEKEY',
            p_data_type       => apex_exec.c_data_type_varchar2,
            p_is_primary_key  => true);
        apex_exec.add_column( /* customerId */
            p_columns         => l_columns,
            p_column_name     => 'CUSTOMERID',
            p_data_type       => apex_exec.c_data_type_number);
        apex_exec.add_column( /* orderLines.items */
            p_columns         => l_columns,
            p_column_name     => 'ORDERLINES',
            p_data_type       => apex_exec.c_data_type_array);
        apex_exec.add_column( /* orderLines.productId */
            p_columns         => l_columns,
            p_column_name     => 'ORDERLINES_PRODUCTID',
            p_data_type         => apex_exec.c_data_type_number,
            p_parent_column_path => '"ORDERLINES"');
        apex_exec.add_column( /* orderLines.quantity */
            p_columns         => l_columns,
            p_column_name     => 'ORDERLINES_QUANTITY',
            p_data_type     => apex_exec.c_data_type_number,
            p_parent_column_path => '"ORDERLINES"');
        return l_columns;
    end get_order_create_columns;
```

18.4.8.12 Invoking BOSS Business Object Custom Actions

Learn about support for BOSS business object custom actions.

To invoke BOSS business object custom actions, it is sufficient to use the **Simple HTTP REST** Data Source type, configuring the Input Parameters, Request Body Template, and Data Profile manually. To invoke the custom action using the declarative Invoke API page process, workflow activity, or the `APEX_EXEC.EXECUTE_REST_SOURCE` procedure, configure the Data Profile columns with appropriate selectors to identify which response payload data you want to return to the caller. Then, add OUT parameters to the REST Data Source operation of type Data Profile Column.

18.4.9 Managing REST Source Catalogs

Create a catalog of REST Data Source metadata that can be used by all applications in a workspace to create REST Data Sources.

REST Source Catalogs contain metadata about REST Data Sources. Developers can search and browse a catalog and then create REST Data Sources for their applications. You can

share REST Source Catalogs by exporting the REST Data Source catalog to a file and then importing it into another workspace or another instance.

- Creating a REST Source Catalog
  Create a REST Source Catalog in Workspace Utilities.
- Creating a REST Source Catalog from OpenAPI
  Create a REST Source Catalog from an OpenAPI file.
- Editing a REST Source Catalog
  Edit an existing REST Source Catalog on the REST Source Catalogs page.
- Managing Catalog Groups
  Use Catalog Groups to organize catalogs into categories..
- About Exporting and Importing a REST Source Catalog
  Move REST Source Catalog to another instance by exporting it to a file and then importing the file into the target instance.

18.4.9.1 Creating a REST Source Catalog

Create a REST Source Catalog in Workspace Utilities.

To create a REST Source Catalog:

1. Navigate to the REST Source Catalogs page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **REST Source Catalogs**.

      The REST Source Catalogs page appears. REST Source Catalogs are organized into categories called **groups**. First create a group and then create the catalog.
2. Create a group:
   a. From the Tasks list, **Manage Catalog Groups**.
   b. Click **Create Group**.
   c. In the Catalog Group Details dialog:
      - Name - Specify the name of the catalog group.
      - Description - Describe the purpose of the group.
      - Click **Create**.
3. Click the REST Source Catalogs breadcrumb to return to the REST Source Catalogs page.
4. Create a catalog:
   a. Click **Create Catalog**.
   b. In the REST Source Catalog Details dialog:
      - Group - Select a group used to organize the catalog and assist developers when creating services.
      - Name - Specify the name of the catalog. Catalog names must be unique within a workspace.
      - Internal Name - Specify an internal name of the catalog. The Internal Name is used to identify a catalog across multiple APEX instances. If a catalog is imported and a catalog with the same internal name already exists, then the existing catalog will be overwritten.

- Description - Describe the REST Source Catalog.
- Click **Create**.

5. Create the REST Data Sources to be included in the catalog. See Creating a REST Data Source.
6. Asssign the REST Data Source with the REST Source Catalog:
   a. Edit the REST Data Source. Click the **REST Source Name**.
   b. Under **REST Source Catalog**, click **Save to Catalog**.
   c. In the Save to REST Source Catalog dialog, verify the following:
      - Catalog Group - Select a Catalog Group. If the service will not be added to an existing catalog, select **Add a new Catalog Group** and follow in on-screen instructions.
      - Catalog - Select the Catalog to which the service will be added. Choose **Add a new Catalog** if the service will not be added to an existing catalog.
      - REST Source Name - Displays the current REST Source Name.
   d. Click **Save to Catalog**.
7. Repeat the previous step for each REST Data Source to be added to the catalog.

18.4.9.2 Creating a REST Source Catalog from OpenAPI

Create a REST Source Catalog from an OpenAPI file.

A single OpenAPI file can contain a full catalog of REST APIs. Upload an Open API fil and Oracle APEX generates a REST Catalog from it with a REST Source template for each of the contained API descriptions. You can then create REST Sources from a catalog.

To create a REST Source Catalog from an OpenAPI file:

1. Navigate to the REST Source Catalogs page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **REST Source Catalogs**.

      The REST Source Catalogs page appears. REST Source Catalogs are organized into categories called **groups**. First create a group and then create the catalog.
2. If needed, create a group:
   a. From the Tasks list, **Manage Catalog Groups**.
   b. Click **Create Group**.
   c. In the Catalog Group Details dialog:
      - Name - Specify the name of the catalog group.
      - Description - Describe the purpose of the group.
      - Click **Create**.
3. Click the **REST Source Catalogs** breadcrumb to return to the REST Source Catalogs page.
4. Click **Create from OpenAPI**.
5. On Upload OpenAPI:

a. Upload File - Upload the OpenAPI document. Options include:
   - **Drag and Drop** - Drag and drop the file to the Upload File region in the center of the page.
   - **Choose File** - Click **Choose File** and then navigate to and select the file.

b. OpenAPI Endpoint URL - As an alternative to uploading the JSON file, enter the URL where the OpenAPI information is available.

c. Click **Next**.

6. The Catalog from OpenAPI dialog appears with the following fields:

   a. Group - Select the group used to organize the catalog and assist developers when creating services.

   b. Catalog - Select an existing catalog to update all services within that catalog or select **Create New Catalog**.

   c. Name - The name of the catalog. Catalog names must be unique within a workspace.

   d. Internal Name - An internal name of the catalog. The Internal Name is used to identify a catalog across multiple APEX instances. If a catalog is imported and a catalog with the same internal name already exists, then the existing catalog will be overwritten.

   e. Description - Describes the REST Source Catalog.

   f. Default REST Source Type - Specifies the REST Data Source type.

   g. Default Authentication - Select a default authentication type. An empty credential of this type will be created when a REST Data Source is created from the catalog. The credential name and secret will need to be provided at that time. Options include:
      - **No Authentication**
      - **Basic Authentication**
      - **OAuth2 Client Credentials Flow**
      - **Oracle Cloud Infrastructure (OCI)**
      - **Certificate/Private Key Pair**
      - **HTTP Header**
      - **URL Query String**

   h. Default Token Server URL - Only displays if **Default Authentication** is set to **OAuth2 Client Credentials Flow**.

      Enter the URL of a token server. This token server is to be used for REST APIs within the OpenAPI file, which are protected by an OAuth2 Client Credentials authentication flow.

   i. Catalog Refresh URL - Provide a URL where the catalog export file can be downloaded from, in order to refresh the REST Source Catalog.

   j. If **Catalog Refresh URL** has a value, two more attributes appear:
      - Proxy Server - Proxy server to use when refreshing the REST Source Catalog. If not set, the Oracle APEX instance proxy will be used (if defined).
      - Web Credential - Identifies the Web Credentials to use when refreshing a catalog. Use this Web Credential if the refresh endpoint URL requires authentication.

   k. Click **Next**.

7. On REST Source Type Default Parameters - The parameters that appear depend upon the **Default REST Source Type** specified on the previous page.

a. Edit the fields. To learn more about a parameter, see field-level Help.

b. Click **Create**.

The catalog appears.

> **See Also:**
>
> About REST Data Source Types

18.4.9.3 Editing a REST Source Catalog

Edit an existing REST Source Catalog on the REST Source Catalogs page.

To edit a REST Source Catalog:

1. Navigate to the REST Source Catalogs page:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.

   b. Click the **Workspace Utilities** icon.

   c. On the Workspace Utilities page, click **REST Source Catalogs**.

      The REST Source Catalogs page appears. REST Source Catalogs are organized into categories called **groups**.
2. To edit a catalog details:
   a. Click the Catalog name.

      The REST Source Catalog Details dialog appears.

   b. Edit the appropriate attributes:
      - Group - Select a group to organize the catalog and assist developers when creating services.
      - Name - Edit the catalog name. Catalog names must be unique to a workspace.
      - Internal Name - Identifies a catalog across multiple APEX instances. If a catalog is imported, and a catalog with the same internal name already exists, then the existing catalog will be overwritten.
      - Description - Described the REST Source Catalog
      - Refresh Endpoint URL - Provide a URL where the catalog export file can be downloaded from, in order to refresh the REST Source Catalog.

   c. Click **Apply Changes**.
3. To delete a catalog.
   a. Click the Catalog name.

   b. On the REST Source Catalog Details dialog, click **Delete**.
4. To view catalog services:
   a. In the Contents column, click the **Services** button.

      The Catalog Services page appears.

   b. To a specific service, click the service name.

      The service details appear.

Although you may edit the descriptive data such as the catalog service Name or Description, edit other details such as data profiles, operations, parameters through the original REST Data Source definition.

5. To associate a new REST Data Source with a catalog:
   a. Navigate to REST Data Source.
   b. Under **REST Source Catalog**, click **Save to Catalog**.
   c. In the Save to REST Source Catalog dialog, verify the options and click **Save to Catalog**.

18.4.9.4 Managing Catalog Groups

Use Catalog Groups to organize catalogs into categories..

To manage Catalog Groups:

1. Navigate to the REST Source Catalogs page:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.
   b. Click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **REST Source Catalogs**.

      The REST Source Catalogs page appears.
2. From the Tasks menu on the right side of the page, click **Manage Catalog Groups**.
3. To add a Catalog Group:
   a. Click **Create Group**.
   b. In the Catalog Group Details dialog:
      - Name - Specify the name of the catalog group.
      - Description - Describe the purpose of the group.
      - Click **Create**.
4. To edit a catalog group.
   a. Click the Catalog Name.
   b. In the Catalog Group Details dialog:
      - Name - Specify the name of the catalog group.
      - Description - Describe the purpose of the group.
      - Click **Apply Changes**.
5. To delete a Catalog Group:
   a. Click the Catalog Name.
   b. Click **Delete**.

18.4.9.5 About Exporting and Importing a REST Source Catalog

Move REST Source Catalog to another instance by exporting it to a file and then importing the file into the target instance.

Exporting a REST Source Catalog to another instance involves the following steps:

1. Create the REST Source Catalog.
2. Export the REST Source Catalog. See Exporting a REST Source Catalog.
3. Import the exported file into the target Oracle APEX instance. See Importing a REST Source Catalog.
4. Install the exported file either directly after importing the file, or later from the Export Repository. See Installing Other Files from the Export Repository.

18.4.10 Managing Data Synchronization

Enable Data Synchronization to keep the contents of a local table automatically in sync with the data from a REST service.

- About Data Synchronization
  Data Synchronization enables developers to automatically sync the contents of a local table with the data from a REST service.
- About Importing and Exporting REST Source Synchronizations
  After importing an application, REST Source Synchronizations are disabled. Learn how to enable synchronization after importing.
- Configuring REST Synchronization
  Configure REST synchronization to automatically sync a local table with data from a REST service.
- Viewing REST Synchronizations Defined in an App
  View REST Synchronizations defined in the current application.
- Clearing Synchronization Settings
  Clear synchronization settings to stop synchronization processing and reset all REST Data Source attributes related to synchronizations.
- Viewing Synchronize Usage
  View a report of application components currently using the local synchronization table as their data source.
- Managing the REST Synchronization Log
  Learn about the REST Synchronization log.

> **See Also:**
>
> Configuring Background Jobs in *Oracle APEX Administration Guide*

18.4.10.1 About Data Synchronization

Data Synchronization enables developers to automatically sync the contents of a local table with the data from a REST service.

You can trigger **Data Synchronization** manually or on a regular schedule, using a scheduler job. Note that the `CREATE JOB` privilege must be granted to the application's parsing schema in order to execute REST source synchronizations on schedule.

Oracle APEX can create the local table based on the visible columns in the REST Data Source **Data Profile**. Since the table is a visible object in the database schema, developers can add indexes, change physical properties, or even add more columns.

If the table is not in sync with the Data Profile, Data Synchronization continues to work for columns present in both the table and the Data Profile (other columns are ignored). However, APEX generates SQL DDL statements to sync table columns back to the Data Profile. These statements can be added to the application as supporting objects.

Data Synchronization Uses Cases

Common use cases using REST Data Source Data Synchronization include:

- **Provide efficient reporting on large data sets coming from a REST service**

  Some REST Services can return large amounts of result data that includes pagination. If a reporting requirement needs larger data sets, this could result to multiple HTTP requests and poorer response times. Also no indexes or other database features can be used to speed up reporting queries. In this case, Oracle recommends enabling Data Synchronization to sync the remote data to a local tables and have any reports based on local tables.

- **Collect data from REST Services for consumption by PL/SQL packages or other logic**

  Having data from the REST service replicated in a local table enables developers to perform all sorts of processing and create a wider variety of reports on local tables.

- **Collect data from a REST API over a longer period of time**

  Some REST APIs only provide data for a recent time frame. For reporting on a larger time frame, Data Synchronization enables developers to fetch REST data and store it in a local table. Note that this use case also requires a "merge by primary key" capability since two subsequent REST requests will return an overlapping data set.

- **Utilize Intelligent caching for APEX components**

  REST Data Sources already provide a caching mechanism: APEX caches the JSON response (or responses) as CLOBs in a caching table. If an APEX component uses the REST Data Source and the cache is not stale, the HTTP request is saved and the cached JSON is used. However, JSON still needs to be parsed (that is, large data sets lead to a lot of JSON parsing overhead). Also, indexes or other database technology cannot be used to speed up queries. Caching within a named local table creates a better level of caching since JSON is no longer parsed and the table can be optimized to support reporting queries.

18.4.10.2 About Importing and Exporting REST Source Synchronizations

After importing an application, REST Source Synchronizations are disabled. Learn how to enable synchronization after importing.

Developers often import applications into a workspace multiple times, for versioning or other purposes. If synchronizations were enabled during these imports, this would lead to multiple instances of the synchronization running simultaneously which is not the desired result. Therefore, Oracle APEX disables synchronizations after you import an application.

You can activate synchronization directly in Shared Components, REST Data Sources, Manage Synchronizations, or in Runtime Only environment by using the following PL/SQL block as shown in the following examples:

Example 18-1 Enable synchronization and start execution schedule immediately

```
declare
    l_app_id number := {new application ID};
begin
```

```
    apex_session.create_session(
        p_app_id => l_app_id,
        p_page_id => {one of the application pages},
        p_username => '{workspace user}');

    apex_rest_source_sync.enable(
        p_application_id => l_app_id,
        p_module_static_id => '{REST Source static ID}');
end;
```

Example 18-2 Enable synchronization and set next execution timestamp explicitly

```
declare
    l_app_id number := {new application ID};
begin
    apex_session.create_session(
        p_app_id => l_app_id,
        p_page_id => {one of the application pages},
        p_username => '{workspace user}');

    apex_rest_source_sync.reschedule(
        p_application_id => l_app_id,
        p_module_static_id => '{REST Source static ID}');
        p_next_run_at => trunc( sysdate + 1 ), -- next midnight
end;
```

18.4.10.3 Configuring REST Synchronization

Configure REST synchronization to automatically sync a local table with data from a REST service.

To create a new REST Synchronization, first create a new REST Data Source and then click **Manage Synchronization** to configure the details.

To configure data synchronization:

1. Navigate to Shared Components:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
2. View the REST Data Source:
   a. On the Shared Components page, find Data Sources and select **REST Data Sources**.
   b. On the REST Data Sources page, select the REST Source Name.
3. Click **Manage Synchronization** .

   The REST Synchronization page appears.
4. Under Details:
   a. Local Table Owner - Select the schema of the local table to synchronize data to. By default, the application Parsing Schema is used.
   b. Synchronize to - Select either **New Table** or **Existing Table**.

- If Existing Table:
  - Table Name - Select an existing table name.
  - Nested Rows - Select to synchronize data to the local table.
  - Click **Save**.
- If **New Table**:
  - Table Name - Enter the name of a new table.
  - Nested Rows - Select Nested Rows to synchronize data to the local table. If creating a new table, the message `Table does not exist` appears. Click **Create Table**.
  - Click **Save**.

c. Table Name -

d. Synchronization Type - Select a synchronization type:

- **Append** - Append rows to the local table. This is typically used when no primary key has been defined in the data profile.
- **Merge** - Merge rows into the local table. To use this option, the data profile must have a Primary Key defined. If a row for the given Primary Key value exists, the row is updated.. Otherwise the row is created.
- **Replace** - Empty the local table before loading new data.

e. Synchronization Schedule - Use the Database **Scheduler Calendaring syntax** to define repeating synchronization schedules, such as `every Tuesday and Friday at 4:00 p.m.` or `the second Wednesday of every month`.

You can enter calendaring syntax manually or declaratively by using the Interval Builder to build simple expressions. To use the Interval Builder, click the **Wrench** icon.

For more complex execution schedules, review details about the Scheduler Calendaring Syntax in DBMS_SCHEDULER PL/SQL package, and provide your expression manually.

> **Note:**
>
> To learn more about scheduler calendaring syntax, see DBMS_SCHEDULER Operational Notes in *Oracle Database PL/SQL Packages and Types Reference*

5. Under Steps - Click **Add Step** to pass parameter values or specific external filters to the REST Service.

   If no step is defined, the REST Data Source will be invoked once, with default parameters. If steps are defined, APEX will invoke the REST Data Source once for each step, and pass the configured parameters.

6. Under Advanced Settings:

   a. Commit Interval - Specify the commit interval to use during data synchronization. If not specified, the whole synchronization will be one transaction, with a commit after the last row. If an interval is specified, APEX will commit each time after the specified amount of rows have been processed.

b. HTTP Request Limit - If required, provide a maximum number of HTTP requests to make for each synchronization step. If this limit is exceeded, synchronization will stop and logged as `Incomplete`.

   Note that due to the actual execution method, synchronization will exceed this limit by one request. For example, if you provide a limit of 10, synchronization will stop after the 11th request has been made.

c. Enable Simple Rate Limiting - When enabled, APEX ensures that only a limited amount of HTTP requests is executed within the given time frame. This rate limiting is purely based on the number of HTTP requests being executed within a time frame. It does not look into HTTP response headers returned by the REST service.

7. To save your changes, click **Save**.
8. To execute REST Synchronization, click **Save and Run**.

18.4.10.4 Viewing REST Synchronizations Defined in an App

View REST Synchronizations defined in the current application.

To view REST Synchronizations defined for the current application:

1. Select the application and navigate to Shared Components:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
2. Under Tasks, click **REST Synchronizations**.

   The Synchronization page appears displaying REST synchronizations defined in the current application.

> **See Also:**
>
> Configuring REST Synchronization

18.4.10.5 Clearing Synchronization Settings

Clear synchronization settings to stop synchronization processing and reset all REST Data Source attributes related to synchronizations.

If APEX components are using the local synchronization table, they will be reset to use the REST Service directly.

To clear synchronization settings:

1. Select the application and navigate to Shared Components:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**
2. On the The Shared Components page, select **REST Synchronizations** from the Tasks list.
3. Select the REST Synchronization.

4. Click **Clear Settings**.

18.4.10.6 Viewing Synchronize Usage

View a report of application components currently using the local synchronization table as their data source.

View the Synchronize Usage report:

1. Select the application and navigate to Shared Components:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**
2. On the The Shared Components page, select **REST Synchronizations** from the Tasks list.
3. Select the REST Synchronization.
4. Under Details, click **Synchronize Usage**.

   A report appears showing application components currently using the local synchronization table as their data source.

18.4.10.7 Managing the REST Synchronization Log

Learn about the REST Synchronization log.

- About the REST Synchronization Log
  The REST Synchronization log records details of previously executed synchronizations.
- Purging the Synchronization Log
  Purge the REST Synchronization log.

> **See Also:**
>
> Deleting REST Synchronization Log Entries in *Oracle APEX Administration Guide*

18.4.10.7.1 About the REST Synchronization Log

The REST Synchronization log records details of previously executed synchronizations.

Once you have successfully executed a synchronization, the REST Synchronization log displays under **Log** on the Synchronization page.

The REST Synchronization log contains details about previously executed synchronizations such as the status (that is, Success or Failure), the amount of processed rows, and the amount of HTTP requests being made. You can drill down from the Synchronization log to the Web Services log to review each individual HTTP request which was made for this synchronization.

18.4.10.7.2 Purging the Synchronization Log

Purge the REST Synchronization log.

To purge the REST Synchronization log:

1. Select the application and navigate to Shared Components:

   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**
2. On the The Shared Components page, select **REST Synchronizations** from the Tasks list.
3. Select the REST Synchronization.

   The Synchronization page appears.

   The REST Data Source Synchronization log displays under **Log** at the bottom of the page.
4. Under Log, click **Purge Log**.

18.4.11 Viewing REST Data Source Utilization

View the REST Data Sources Utilization report.

To view REST Data Sources Utilization report:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **Utilization**.

   The Utilization page displays where a REST Data Source is used within the application.

18.4.12 Viewing REST Data Sources History

View the REST Data Sources History report.

To view recent modifications to REST Data Sources:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **REST Data Sources**.
2. On the REST Data Sources page, click **History**.

   The History page displays recent modifications made to REST Data Sources in the current workspace.

18.5 Managing REST Enabled SQL References

Execute SQL queries or PL/SQL defined at the component-level on a remote database using REST Enabled SQL references.

- Understanding REST Enabled SQL Service References
  Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote database.

- Before You Begin: REST Enabled SQL Service Requirements
  Prior to creating a REST enabled SQL reference, developers must complete these requirements.
- Creating a REST Enabled SQL Service Reference
  Create REST Enabled SQL service references on the REST Enabled SQL page.
- Example: Creating a REST Enabled SQL Service Reference on Oracle
  Review an example of creating REST Enabled SQL service reference on a remote Oracle database.
- Example: Creating REST Enabled SQL Service Reference on MySQL
  Review an example of creating a REST Enabled SQL service reference on a remote MySQL database.
- Editing a REST Enabled SQL Service Reference
  Edit existing REST Enabled SQL references on the Edit page.
- Deleting a REST Enabled SQL Service Reference
  Delete REST Enabled SQL references on the Edit page by clicking **Delete**.
- Viewing REST Enabled SQL Service Utilization
  View where a REST Enabled SQL Service is being used within a workspace on the REST Enabled SQL Utilization report.
- Viewing REST Enabled SQL Service History
  View recent modifications made to REST Enabled SQL Services within a workspace on the REST Data Sources History report.
- Creating or Updating Components to Use a REST Enabled SQL
  Create or update reports (classic reports, interactive reports, interactive grids, List View report, Toggle Column reports, and Reflow Reports), forms, CSS calendars, JET charts, and Tree regions to point to data on a remote database using a REST Enabled SQL reference.

18.5.1 Understanding REST Enabled SQL Service References

Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote database.

- About REST Enabled SQL Service References
  Use REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote database.
- REST Enabled SQL Service Reference Requirements
  Review the minimum requirements for using REST Enabled SQL Service references.
- About MySQL Support
  MySQL only supports read-only APEX components.
- Differences between REST Enabled SQL Service References and Database Links
  Learn how REST Enabled SQL Service references differ from database links.
- Populating the Endpoint URL Dynamically
  Populate the server portion of the Endpoint URL dynamically using a callback procedure you specify in the Configuration Procedure attribute.
- Exporting and Importing REST Enabled SQL Services
  Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote Oracle database.

18.5.1.1 About REST Enabled SQL Service References

Use REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote database.

Oracle APEX installations that meet the minimum Oracle REST Data Services (ORDS) requirements can execute any SQL or PL/SQL through a REST endpoint.

Developers create REST Enabled SQL references by defining a name, the endpoint URL, and authentication information within Shared Components. APEX passes the SQL or PL/SQL query to ORDS over REST, and a self-describing JSON response is returned. The JSON object contains result set meta data, the result data, and pagination details.

Because REST Enabled SQL services are stored at the workspace-level within APEX components, they are available to all applications within a workspace. Developers can utilize REST Enabled SQL references for interactive reports, interactive grids, classic reports, forms, master detail forms, calendars, JET charts, trees, and PL/SQL processes. References can also be used with Calendars, JET Charts, Trees, and PL/SQL processes.

18.5.1.2 REST Enabled SQL Service Reference Requirements

Review the minimum requirements for using REST Enabled SQL Service references.

Requirements for using REST Enabled SQL Service references include:

- A remote Oracle Database that includes Oracle REST Data Services (ORDS) 23.3 or later on the front-end.
- A remote MySQL Database that includes Oracle REST Data Services (ORDS) 23.3 or later on the front-end.

  See Using the REST Enabled SQL Service with MySQL Database in *Oracle REST Data Services Installation and Configuration Guide* and REST Enabled SQL Service and MySQL Database in *Oracle REST Data Services Developer's Guide*.
- An administrator must enable RESTful Services either at the workspace or instance-level as described in Enabling RESTful Services and Defining a Path Prefix for a Workspace and Enabling RESTful Services for an Instance in *Oracle APEX Administration Guide*.

18.5.1.3 About MySQL Support

MySQL only supports read-only APEX components.

The following table lists supported APEX components and unsupported features when using a REST Enabled SQL Service references with a remote MySQL database.

APEX Component	Unsupported Features and Comments
Classic reports	BLOB column (see below).
Interactive reports	• Pivot View • Some aggregate functions (for example, Ratio To Report, Median, Approx Count Distinct) • Flashback • BLOB column (see below) • Column display based on LOV cannot use Static LOVs

APEX Component	Unsupported Features and Comments
Interactive grids	• Editing (DML support) • Some aggregate functions (for example, Ratio To Report, Median, Approx Count Distinct) • Control breaks require a Primary Key column to be defined • Filtering on multi-value columns • Flashback • Column display based on LOV cannot use Static LOVs
Faceted search and smart filters	Multi-value facets.
Calendars	Drag and drop.
Form regions	Only support for read-only forms. DML is not supported..
Charts	• Box plot charts • Some aggregate functions (for example, Ratio To Report, Median, Approx Count Distinct)
Cards	n/a
Column toggle reports	n/a
Reflow reports	n/a
Shared lists of values	n/a
Map regions	MySQL native Geometry type is not supported. Also, the map query must return GeoJSON.
Trees	Column chosen as **Order Siblings By** must be a `VARCHAR` data type (no numeric or date columns).
Execute Code page process	MySQL is not supported, and cannot be chosen from the list of remote servers when using REST Enabled SQL.
Automations	MySQL servers are supported for the Automation Query, but are not supported as a target within an Automation Action and the Execute Code action type.
BLOB column in interactive report and classic report	No support for BLOB columns in interactive reports and classic reports since these do not support REST Enabled SQL in general. BLOB columns are supported for the Cards regions and for Form Display.

18.5.1.4 Differences between REST Enabled SQL Service References and Database Links

Learn how REST Enabled SQL Service references differ from database links.

Both REST Enabled SQL Service references and database links enable developers to access data remotely. However, these features access remote data differently. Key differences between database links and REST Enabled SQL Service references include:

- **Database Link**
  - Functions at the SQL-level which enables developers to use remote tables and local tables in the same SQL query.

  - Works over SQL*Net, which can be problematic when connecting to a Cloud instance over the internet.
- **REST Enabled SQL Service References**
  - Functions at the workspace-level. Developers can create an APEX component with a query on a REST enabled SQL Service but cannot join it to a local table.
  - Works with JSON over HTTP(s) which makes it easy-to-use it in Cloud environments or over the internet.

Both Database Links and REST Enabled SQL fetch data over the network which is significantly slower than fetching data from a table in the local database. When evaluating the best approach for your environment, be sure to evaluate the impact on page view performance and always consider replicating remote data in local tables, with an appropriate refresh algorithm.

18.5.1.5 Populating the Endpoint URL Dynamically

Populate the server portion of the Endpoint URL dynamically using a callback procedure you specify in the Configuration Procedure attribute.

You can edit the Configuration Procedure attribute on the Edit Remote Server page. See Editing or Deleting a Remote Server.

Flexible Remote Server works as follows:

- When the REST Enabled SQL reference is invoked, the APEX Engine evaluates the Configuration Procedure attribute.
- If the Configuration Procedure attribute has a value, either stored in the database or in an anonymous block in the **PL/SQL Code** attribute, the APEX Engine executes the procedure to fetch a collection of substitution parameters that are then used to change the Endpoint URL.

Flexible Remote Server Use Cases

Common use cases for using a Flexible Remote Server include:

- **Environment dependencies** - A Flexible Remote Server is a good choice when the environment is dependent upon where the server or application is deployed (for example, in production, test or development instances). Instead of manually changing the Endpoint URL attribute for each environment, you can use an API to change it dynamically.

  > **Tip:**
  >
  > You can also use `APPLICATION_ADMIN.SET_REMOTE_SERVER` procedure to change the Endpoint URL in a deployment script. See SET_REMOTE_SERVER Procedure in *Oracle APEX API Reference*

- **Application data or the application user dependencies** - A Flexible Remote Server enables you to use the Configuration Procedure attribute to change the Endpoint URL at runtime depending on dynamic parameters. These dynamic parameters are flexible and can be based on a configuration table, items in Session State, Application Items or Substitution Items.

  The results of the Configuration Procedure are cached and reused within the same request (page view or page processing).

> **Note:**
>
> Flexible Remote Servers are only supported for **REST Data Source** and **Authentication** Server Types or Remote Servers used in REST Enabled SQL.

Configuring a Flexible Remote Server

To configure a Flexible Remote Server, edit the REST Enabled SQL service and enter a procedure name in the **Configuration Procedure** attribute. This procedure can be the name of a procedure stored in the database, stored in a database package procedure, or a procedure stored in the **PL/SQL Code** attribute.

This signature of the procedure must have two parameters of type:

- `p_info in apex_plugin.t_remote_server_info`
- `p_config in out apex_plugin.t_remote_server_config`

The parameter `p_config` has two optional attributes which can be changed in the configuration procedure:

- `p_config.base_url` - *`base_url`* changes the Endpoint URL.
- `p_config.substitutions` - *`substitutions`* assigns name/value pairs using `apex_t_varchar2`. Oracle APEX substitutes each `#NAME#` in the base url or Endpoint URL.

Together these two attributes determine the actual Endpoint URL the Remote Server uses.

> **Tip:**
>
> To view Configuration Procedure examples, see item Help for the **Configuration Procedure** attribute.

The following example changes the base URL if the application ID is 100:

```
procedure my_server_config(
  p_info    in  apex_plugin.t_remote_server_info,
  p_config  out apex_plugin.t_remote_server_config )
is
begin
  if v('APP_ID') = 100 then
    p_config.base_url := 'http://example100.com';
  else
    p_config.base_url := 'http://example.com';
  end if;
end;
```

You can also can use placeholders by enclosing the "*name*" with the number symbol (#) and use substitutions to replace the value in those placeholders. Consider the following example:

```
procedure my_server_config(
  p_info    in  apex_plugin.t_remote_server_info,
  p_config  out apex_plugin.t_remote_server_config )
is
begin
```

```
  if v('APP_ID') = 100 then
    p_config.base_url := 'http://example100.com';
  else
    p_config.base_url := 'http://example.com';
  end if;
end;
```

> **See Also:**
>
> Editing a REST Enabled SQL Service Reference

18.5.1.6 Exporting and Importing REST Enabled SQL Services

Create REST Enabled SQL Service references to execute SQL or PL/SQL defined on a remote Oracle database.

When you export an application, used REST Enabled SQL references are added to the export file. If you export an application and import it into another workspace, APEX checks whether the target workspace already contains REST Enabled SQL references with the same static ID. If a REST Enabled SQL reference already exists, the application uses the existing reference. If the reference does not exist, it is created in the target workspace.

18.5.2 Before You Begin: REST Enabled SQL Service Requirements

Prior to creating a REST enabled SQL reference, developers must complete these requirements.

Complete the following requirements before creating REST enabled SQL reference:

1. Set up a remote Oracle database.
2. On the remote Oracle database, install Oracle REST Data Services (ORDS) 20.x or later.

   > **Tip:**
   >
   > This ORDS installation is completely independent from the Oracle REST Data Services used as the Oracle APEX web server

3. Configure and enable the REST Enabled SQL service feature. See Installing and Configuring Oracle REST Data Services in *Oracle REST Data Services Installation and Configuration Guide*.
4. Activate REST Enabled SQL for the target schema on the remote database to be accessed by running `ORDS.ENABLE_SCHEMA`.

   Log in to the database schema to be enabled for REST Enabled SQL and execute the following:

   ```
   begin
       ords.enable_schema;
   end;
   /
   ```

```
commit
/
```

The REST Enabled SQL service is then available with a URL in the following format:

`http://host:port/ords/schema`

Where:

- *host* is the name of the system where Oracle REST Data Services is installed.
- *port* is the port number assigned when configuring Oracle REST Data Services. In a default installation, this number is `8080`.
- *ords* is the service name defined when configuring Oracle REST Data Services.
- *schema* is the target schema.

> **Tip:**
>
> When providing the URL for the target schema in APEX, do not append `/_/sql;`.

You will use a URL using the above format when creating the REST Enabled SQL service reference in APEX.

For example, the following is a REST Enabled SQL service for the `scott_obe` schema with ORDS running in standalone mode:

`http://server.example.com:8080/ords/scott_obe`

18.5.3 Creating a REST Enabled SQL Service Reference

Create REST Enabled SQL service references on the REST Enabled SQL page.

> **Tip:**
>
> Before creating a REST enabled SQL reference, developers must complete the requirements described in Before You Begin: REST Enabled SQL Service Requirements.

To create a REST Enabled SQL service reference:

1. Navigate to the REST Enabled SQL page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.

> **Tip:**
>
> You can also access the REST Enabled SQL page from Shared Components. On the App Builder home page, select the application and then click **Shared Components**. Find **Workspace Objects** and click **REST Enabled SQL Databases**.

2. On the REST Enabled SQL page, click **Create**.
3. On General, specify:
   a. Name - Enter a descriptive name for this REST Enabled SQL service.
   b. Endpoint URL - Enter the base URL for this REST Enabled SQL service.

   > **Tip:**
   >
   > The URL must be in the format `http://host:port/ords/schema`. You obtain the URL by enabling the target schema on the remote database to be accessed by running `ORDS.ENABLE_SCHEMA`. To learn more, see Before You Begin: REST Enabled SQL Service Requirements.

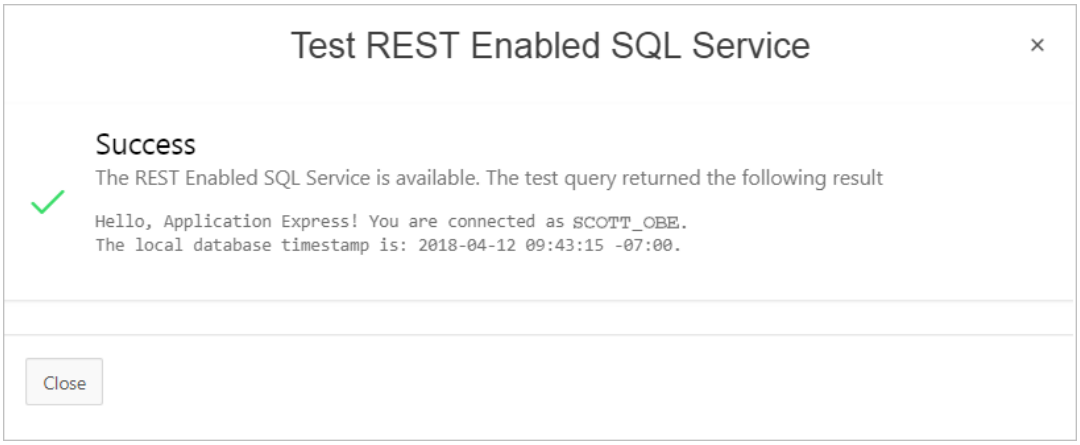
   c. Click **Next**.
4. For Authentication, specify the appropriate credentials:

   > **Note:**
   >
   > The credentials that appear change depending upon the target database. The example below assumes the target is an Oracle database.

   a. Credentials - Select an existing credential to authenticate against, or select **Enter new** and complete the remaining steps.
   b. Credential Name - Enter a descriptive name for the credentials.
   c. Authentication Type - Specify the Authentication type. Select either **Basic Authentication** or **OAuth2 Client**.
   d. Client ID:
      - For **Basic Authentication**, enter the database user name. User names for Basic Authentication are converted to upper case. To include a case-sensitive user name, prefix it with the exclamation mark (`!`).
      - For **OAuth2**, enter the client ID. This information will not be encrypted.
   e. Client Secret - Enter the password, or OAuth2 Client Secret. This information is stored encrypted and cannot be retrieved in clear text.
   f. Verify Client Secret - Enter the password or client secret again to verify your input. When the two values are not identical, an error message displays.
5. Click **Create**.

   APEX tests the REST Enabled SQL service. If everything has been done correctly (that is, you have included the correct URL, schema name, and password), the Test REST Enabled

SQL Service message appears as shown in the following illustration.



6. Click **Close**.

The REST Enabled SQL service appears on the REST Enabled SQL page.

18.5.4 Example: Creating a REST Enabled SQL Service Reference on Oracle

Review an example of creating REST Enabled SQL service reference on a remote Oracle database.

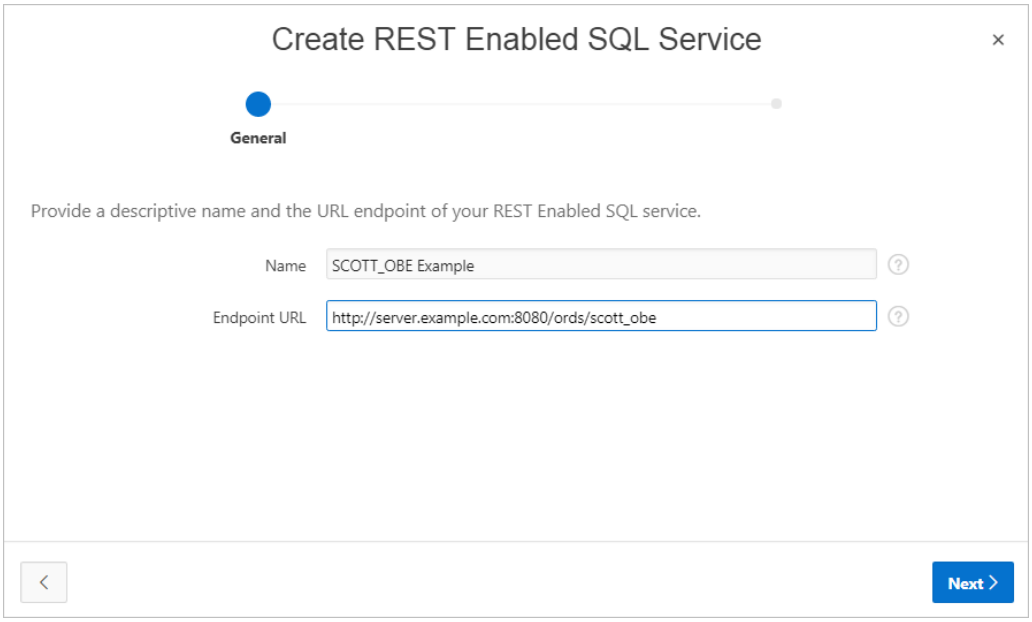
The following example builds a fictional REST Enabled SQL service reference named `SCOTT_OBE Example` using the Base URL `http://server.example.com:8080/ords/scott_obe`.

> **Tip:**
>
> Before creating a REST enabled SQL reference, developers must complete the requirements described in Before You Begin: REST Enabled SQL Service Requirements.

To create a REST Enabled SQL service reference:

1. Navigate to the REST Enabled SQL page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.
2. On the REST Enabled SQL page, click **Create**.
3. On General, specify:
   a. Name - Enter a descriptive name for this REST Enabled SQL service. For example: `SCOTT_OBE Example`.
   b. Endpoint URL - Enter the base URL for this REST Enabled SQL service. For example:

```
http://server.example.com:8080/ords/scott_obe
```



c. Click **Next**.

4. For Authentication, specify the appropriate credentials:

   a. Credentials - Select an existing credential to authenticate against. For example: To create new credentials, select **Enter New**.

   b. Credential Name - Enter a descriptive name for the credentials. For example: .

      ```
      SCOTT_OBE Credentials
      ```

   c. Authentication Type - Specify the Authentication type. For example: **Basic Authentication**.

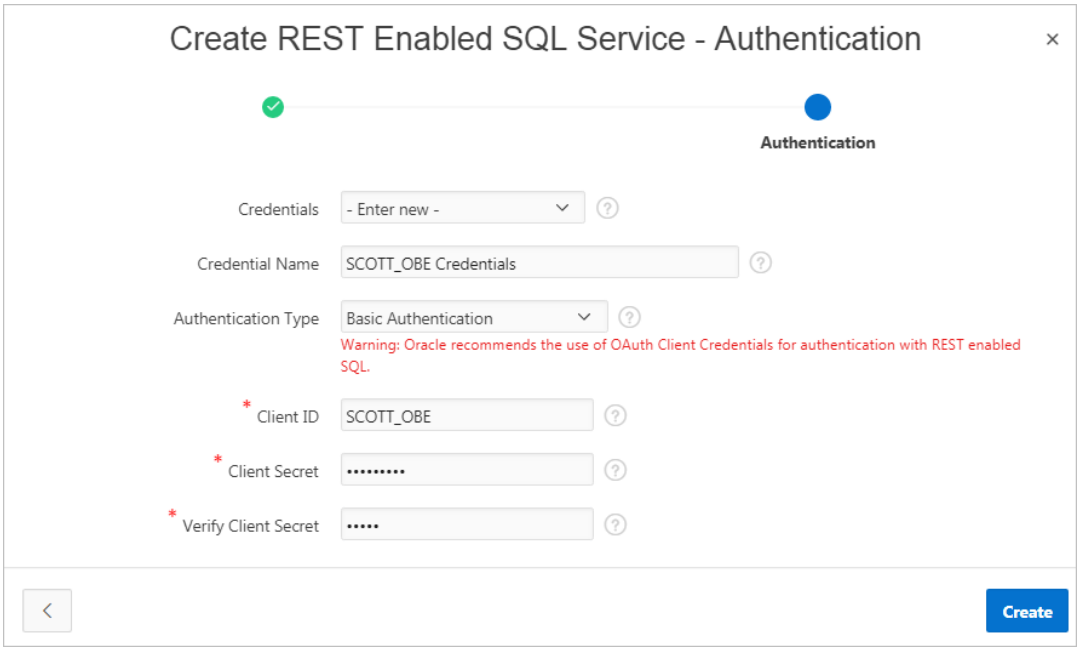
   d. Client ID - Enter the database user name. For example:

      ```
      SCOTT_OBE
      ```

      User names for Basic Authentication are converted to upper case. To include a case-sensitive user name, prefix it with the exclamation mark (`!`).

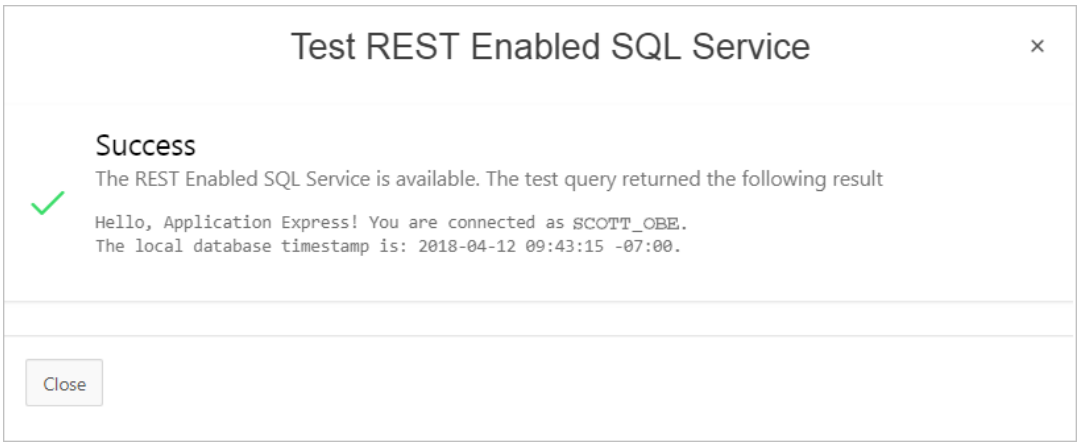
   e. Client Secret - Enter the database user name.

   f. Verify Client Secret - Enter the password.



5. Click **Create**.

   APEX tests the REST Enabled SQL service. If everything has been done correctly (that is, you have included the correct URL, schema name, and password), the Test REST Enabled SQL Service message appears as shown in the following illustration.



6. Click **Close**.

The REST Enabled SQL service reference appears on the REST Enabled SQL page.

18.5.5 Example: Creating REST Enabled SQL Service Reference on MySQL

Review an example of creating a REST Enabled SQL service reference on a remote MySQL database.

The following example builds a fictional REST Enabled SQL service reference named `MySQL REST Enabled SQL Example` using the Base URL `http://server.example.com:8080/ords/mysqldb`.

> **Tip:**
>
> Before creating a REST enabled SQL reference, developers must complete the requirements described in Before You Begin: REST Enabled SQL Service Requirements.

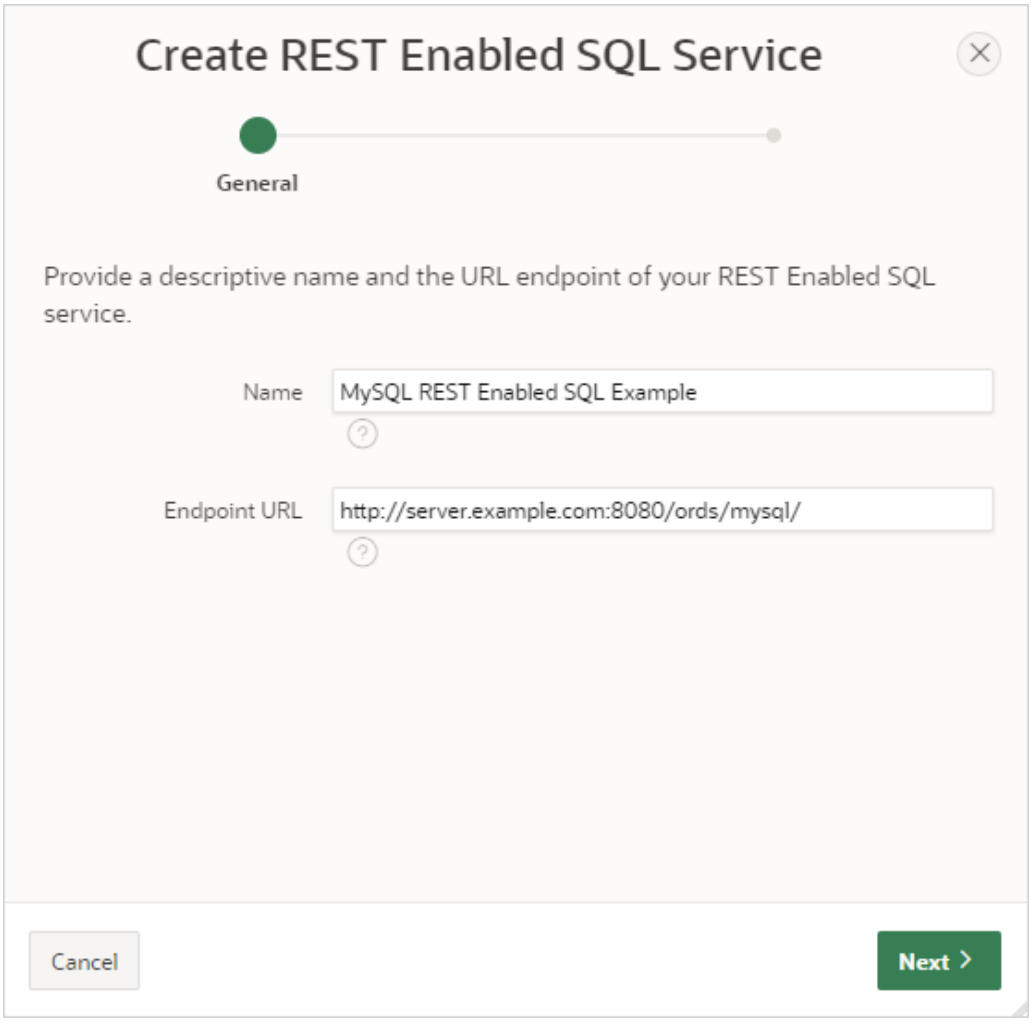
To create a REST Enabled SQL service reference to a MySQL database:

1. Navigate to the REST Enabled SQL page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.
2. On the REST Enabled SQL page, click **Create**.
3. For General:
   a. Name - Enter a descriptive name for this REST Enabled SQL service. For example:

      `MySQL REST Enabled SQL Example`

   b. Endpoint URL - Enter the base URL for this REST Enabled SQL service. For example:

      `http://server.example.com:8080/ords/mysql/`



c. Click **Next**.

4. For Authentication:

   a. Credentials - Select an existing credential to authenticate against. For example: To create new credentials, select **Enter New**.

   b. Credential Name - Enter a descriptive name for the credentials. For example:

      `Credentials for MySQL REST Enabled SQL Example`

   c. Store Credentials - Accept the default, enabled. When disabled, the named credential is created, with empty values stored as Username and Password. To use the REST Enabled SQL Reference, credentials must be set for the APEX Session, with the `APEX_CREDENTIAL.SET_SESSION_CREDENTIALS` procedure.

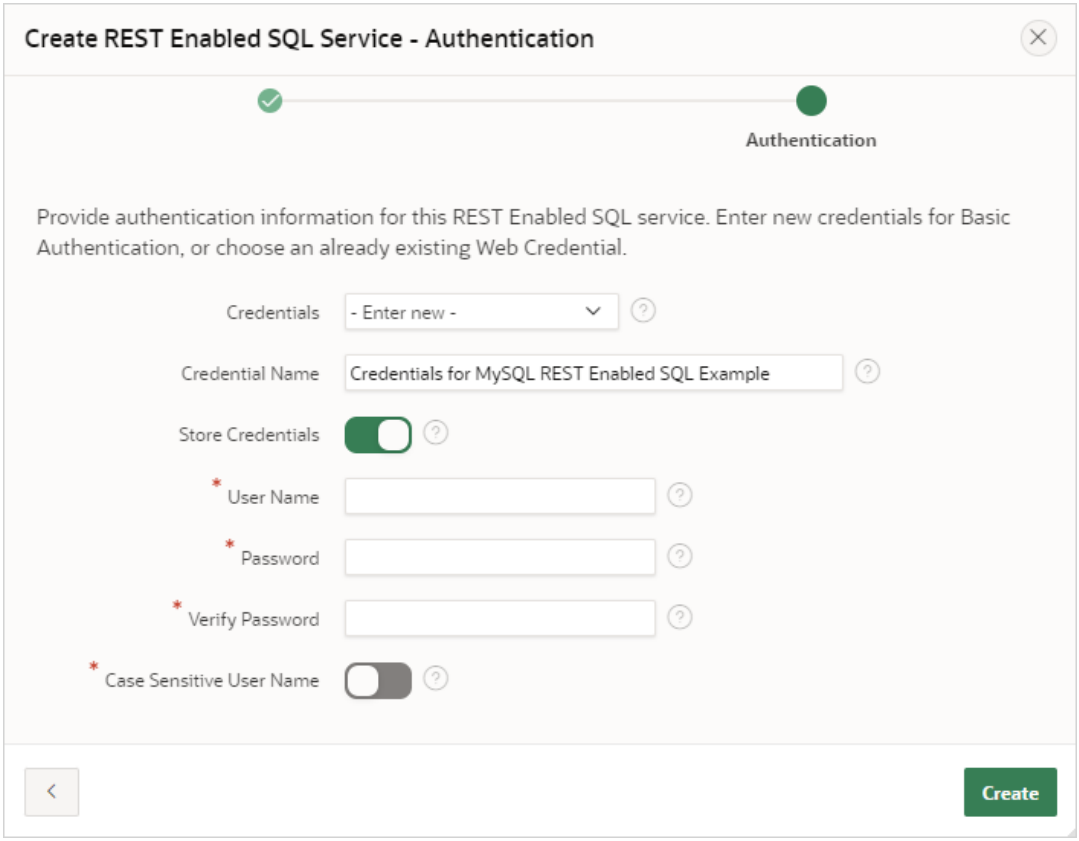
   d. Username - For Basic Authentication, enter the database user name.

      > **Tip:**
      >
      > Since MySQL user names are most often in lower case, enable the **Case Sensitive User Name** attribute.

   e. Password - Enter the password. This information is stored encrypted and cannot be retrieved in clear text.

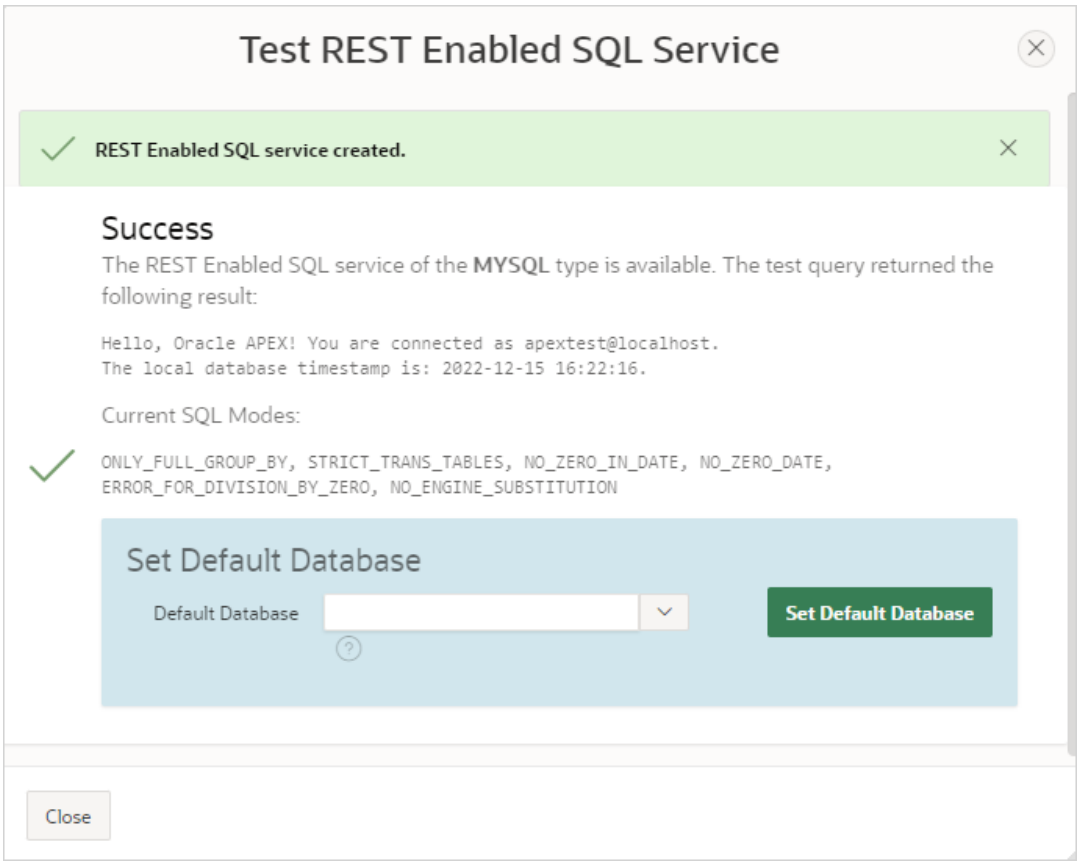
   f. Verify Password - Enter the password again to verify your input.

   g. Case Sensitive User Name - Enable this switch to store a case-sensitive User Name. Otherwise the user name will be converted to upper case. While user names are typically in upper case on Oracle databases, MySQL user names are most often in lower case.



5. Click **Create**.

   APEX tests the REST Enabled SQL service. If everything has been done correctly (that is, you have included the correct URL, schema name, and password), the Test REST Enabled SQL Service message appears as shown in the following image.



6. Set Default Database -
   a. From Default Database list, select the database to use for operations on this MySQL REST Enabled SQL Reference.
   b. Click **Set Default Database**.
7. Click **Close**.

The REST Enabled SQL service reference appears on the REST Enabled SQL page.

18.5.6 Editing a REST Enabled SQL Service Reference

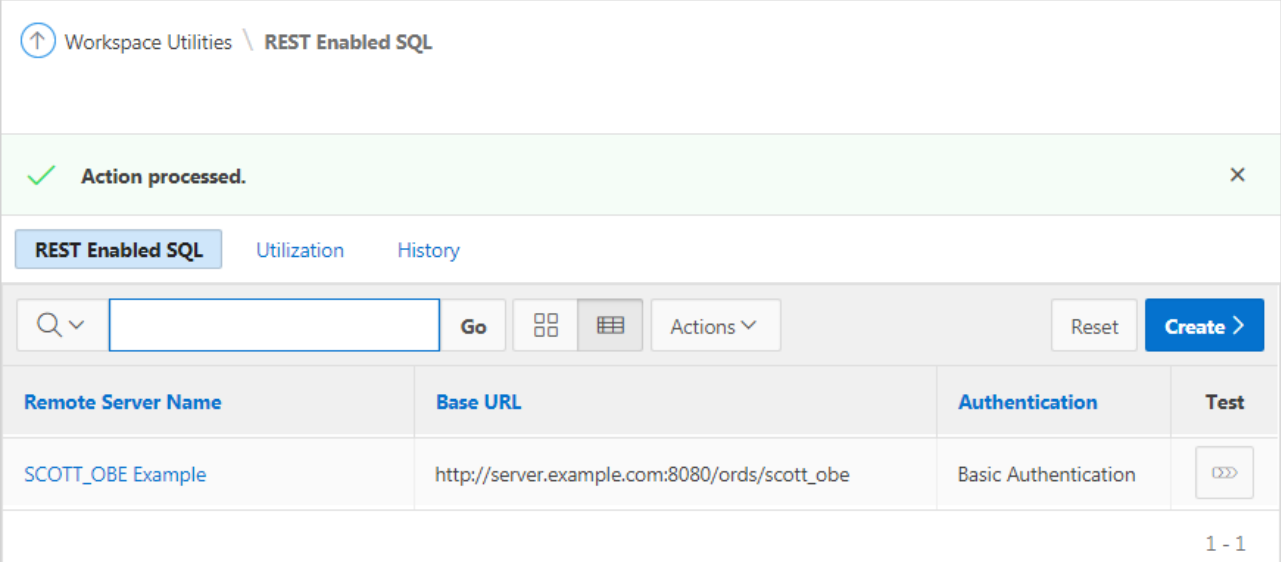
Edit existing REST Enabled SQL references on the Edit page.

To edit a REST Enabled SQL service:

1. Navigate to the REST Enabled SQL page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.

> **Tip:**
>
> You can also access the REST Enabled SQL page from Shared Components. On the App Builder home page, select the application and then click **Shared Components**. Find **Workspace Objects** and click **REST Enabled SQL Databases**.

The REST Enabled SQL page appears listing the Remote Server Name, Base URL, and Authentication.



2. Click the remote server name.

   The Edit page appears. Edit the attributes.

3. REST Enabled SQL attributes:
   a. Name - Enter a descriptive name for this REST Enabled SQL service.
   b. Endpoint URL - Enter the base URL for this REST Enabled SQL service.
   c. Prompt on Install - Choose whether prompts for this REST Enabled SQL service should display when the application is being imported on another APEX instance.
   d. Authentication Required - Choose whether this REST Enabled SQL aervice requires authentication.
   e. Credential - Select the credential to authenticate against.

   > **Tip:**
   >
   > To edit credentials, go to **Shared Components**, **Credentials**.

4. Session attributes:
   a. Initialization Code - Enter code to be executed immediately after connecting to the REST Enabled SQL service and before the component SQL is being executed.
   b. Cleanup Code - Enter code to be executed immediately after the component SQL is being executed.

5. Advanced attributes:
   a. Static ID - Use the Static ID to reference the Remote Server in API Calls. Static IDs are also used to identify an existing Remote Server when the application is being exported and imported to another workspace.
   b. Server Time Zone - Time Zone which the REST Enabled SQL service uses to decode DATE and TIMESTAMP values. This value is updated each time you click **Save and Test**.
6. Click **Apply Changes** or **Save and Test**.

> **See Also:**
>
> Managing Web Credentials

18.5.7 Deleting a REST Enabled SQL Service Reference

Delete REST Enabled SQL references on the Edit page by clicking **Delete**.

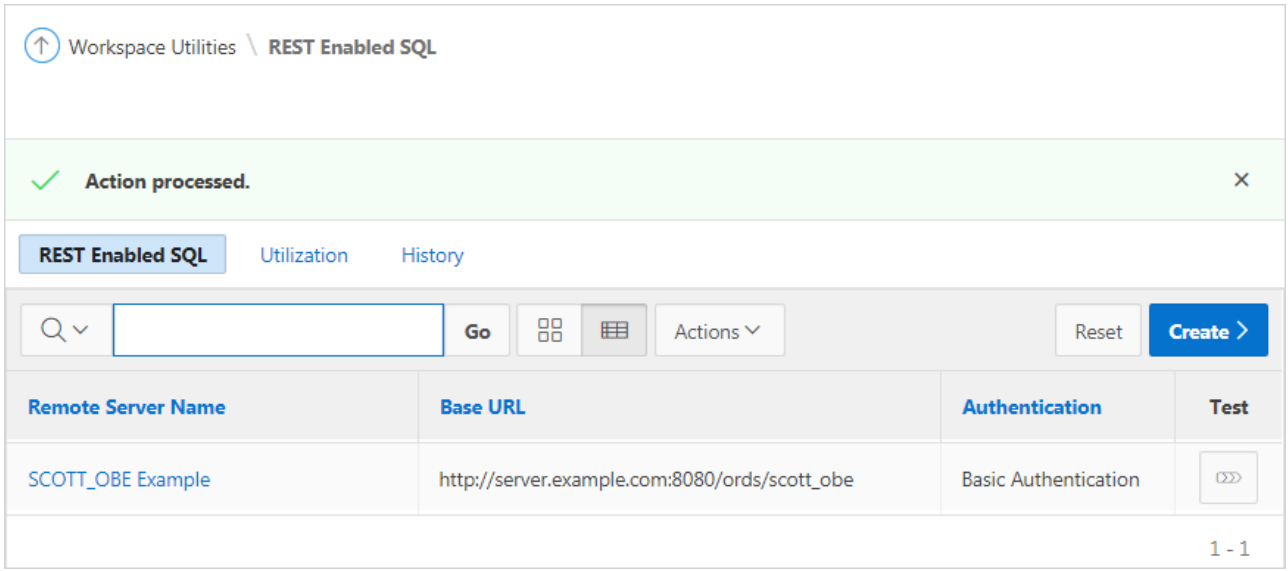
You cannot delete a REST Enabled SQL reference if it is being used by an APEX component.

To edit a REST Enabled SQL service:

1. Navigate to the REST Enabled SQL page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.

   > **Tip:**
   >
   > You can also access the REST Enabled SQL page from Shared Components. On the App Builder home page, select the application and then click **Shared Components**. Find **Workspace Objects** and click **REST Enabled SQL Databases**

   The REST Enabled SQL page appears listing the Remote Server Name, Base URL, and Authentication.



2. Click the remote server name.

   The Edit page appears.

3. Click **Delete**.

18.5.8 Viewing REST Enabled SQL Service Utilization

View where a REST Enabled SQL Service is being used within a workspace on the REST Enabled SQL Utilization report.

To view the REST Data Sources Utilization report:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.

   > **Tip:**
   >
   > You can also access the REST Enabled SQL page from Shared Components. On the App Builder home page, select the application and then click **Shared Components**. Find **Workspace Objects** and click **REST Enabled SQL Databases**

2. On the REST Enabled SQL page, click **Utilization**.

   The Utilization page displays where a REST Enabled SQL Service is being used within the workspace.

18.5.9 Viewing REST Enabled SQL Service History

View recent modifications made to REST Enabled SQL Services within a workspace on the REST Data Sources History report.

To view recent modifications to REST Enabled SQL Services:

1. Navigate to the REST Data Sources page:
   a. On the Workspace home page, click **App Builder**.
   b. Click **Workspace Utilities**.
   c. On the Workspace Utilities page, click **REST Enabled SQL Services**.

> **Tip:**
>
> You can also access the REST Enabled SQL page from Shared Components. On the App Builder home page, select the application and then click **Shared Components**. Find **Workspace Objects** and click **REST Enabled SQL Databases**

2. On the REST Enabled SQL page, click **History**.

   The History page displays recent modifications made to Enabled SQL Services in the current workspace.

18.5.10 Creating or Updating Components to Use a REST Enabled SQL

Create or update reports (classic reports, interactive reports, interactive grids, List View report, Toggle Column reports, and Reflow Reports), forms, CSS calendars, JET charts, and Tree regions to point to data on a remote database using a REST Enabled SQL reference.

- Creating a Report Using a REST Enabled SQL Reference
  Create a report on a remote database using a REST Enabled SQL reference.
- Creating a Chart Using a REST Enabled SQL Reference
  Create a chart on a remote database using a REST Enabled SQL reference.
- Creating a Calendar Using a REST Enabled SQL Reference
  Create a calendar on a remote database using a REST Enabled SQL reference.
- Specifying a REST Enabled SQL Reference in an Existing Component
  Configure existing components to point to a remote database using a REST Enabled SQL service reference.

18.5.10.1 Creating a Report Using a REST Enabled SQL Reference

Create a report on a remote database using a REST Enabled SQL reference.

The following example demonstrates how to create an interactive report based on a table using a REST Enabled SQL reference. The procedures for creating other report types is similar.

> **Tip:**
>
> Before creating a report using a using a REST Enabled SQL reference, you must complete the tasks described in Before You Begin: REST Enabled SQL Service Requirements and create the reference as described in Creating a REST Enabled SQL Service Reference.

To create an interactive report using a REST Enabled SQL reference:

1. Create a REST Enabled SQL reference.
2. Run the Create Page Wizard:

a. Navigate to the Workspace home page.

b. Click the **App Builder** icon.

c. Select an application.

d. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

3. Under Component, select **Interactive Report**.

4. Under Page Definition:

   a. Page Number - The page number is an integer value that identifies a page within an application.

   b. Name - Specify a text name for this page.

   > **Tip:**
   >
   > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.

   d. Include Form Page - Enable this flag to create a form page along with the report page.

      Form pages are always created as Modal Dialog pages. Enabling this flag displays additional attributes. To learn more about an attribute, see field-level Help.

5. Under Data Source:

   a. Data Source - Select **REST Enabled SQL**.

   b. REST Enabled SQL Service - Select a previously defined reference.

   c. Source Type - Specify the source of the new page. The UI changes based on your selection. Options include:

      - **Table**:
        - Table /View Owner - Select the owner of the table on which you are building the page.
        - Table/View Name - Select the table or view on which the page will be based.
      - **SQL Query**:
        - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

6. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

7. Click **Create Page**.

8. Click **Save and Run Page** to view the chart.

18.5.10.2 Creating a Chart Using a REST Enabled SQL Reference

Create a chart on a remote database using a REST Enabled SQL reference.

> **Tip:**
>
> Before creating a report using a using a REST Enabled SQL reference, you must complete the tasks described in Before You Begin: REST Enabled SQL Service Requirements and create the reference as described in Creating a REST Enabled SQL Service Reference.

To create a chart using a REST Enabled SQL reference:

1. Create a REST Enabled SQL reference.
2. Run the Create Page Wizard:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Create Page**.

      Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
3. Under Component, select **Chart**.
4. Select a chart type (for example, **Pie**).
5. Under Page Definition:
   a. Page Number - The page number is an integer value that identifies a page within an application.
   b. Name - Specify a text name for this page.

      > **Tip:**
      >
      > This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
6. Under Data Source:
   a. Data Source - Select **REST Enabled SQL**.
   b. REST Enabled SQL Service - Select a previously defined reference.
   c. Source Type - Specify the source of the new page. The UI changes based on your selection. Options include:
      - **Table**:
        – Table /View Owner - Select the owner of the table on which you are building the page.
        – Table/View Name - Select the table or view on which the page will be based.

- **SQL Query**:
  - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

7. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
8. Click **Next**.
9. Edit the chart attributes. The options that display vary depending upon the chart type selected. To learn more about an attribute, click the field-level Help.
10. Click **Create Page**.
11. Click **Save and Run Page** to view the chart.

> **See Also:**
>
> Creating Charts

18.5.10.3 Creating a Calendar Using a REST Enabled SQL Reference

Create a calendar on a remote database using a REST Enabled SQL reference.

> **Tip:**
>
> Before creating a report using a using a REST Enabled SQL reference, you must complete the tasks described in Before You Begin: REST Enabled SQL Service Requirements and create the reference as described in Creating a REST Enabled SQL Service Reference.

To create a calendar using a REST Enabled SQL reference:

1. Create a REST Enabled SQL reference.
2. Run the Create Page Wizard:
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Create Page**.

      Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
3. Under Component, select **Calendar**.
4. Under Page Definition:

a. Page Number - The page number is an integer value that identifies a page within an application.

b. Name - Specify a text name for this page.

> **Tip:**
>
> This text is also used for page **Title**. After page creation, you can modify the **Title** in Page Designer.

c. Page Mode - Identify the page mode. To learn more, see field-level Help.

5. Under Data Source:

   a. Data Source - Select **REST Enabled SQL**.

   b. REST Enabled SQL Service - Select a previously defined reference.

   c. Source Type - Specify the source of the new page. The UI changes based on your selection. Options include:

      - **Table**:
        - Table /View Owner - Select the owner of the table on which you are building the page.
        - Table/View Name - Select the table or view on which the page will be based.
      - **SQL Query**:
        - Enter a SQL SELECT statement - Enter the SQL query. To see an example, expand the **SQL Query Example** region.

6. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.

   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.

   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

7. Click **Next**.

8. Edit the calendar attributes:

   a. Display Column - Select the column which holds the text displayed for events on this calendar.

   b. Start Date Column - Select the column which holds the start date for events displayed on this calendar.

   c. End Date Column - Select the column which holds the end date for events displayed on this calendar. If **End Date** is specified, then the calendar displays duration based events.

   d. Show Time - Select **Yes** to show the time portion for events and also enable the Weekly and Daily calendar views. Select **No** to only show the dates for events.

9. Click **Create Page**.

10. Click **Save and Run Page** to view the chart.

> **See Also:**
>
> Creating Calendars

18.5.10.4 Specifying a REST Enabled SQL Reference in an Existing Component

Configure existing components to point to a remote database using a REST Enabled SQL service reference.

> **Note:**
>
> Before configuring a component to use a REST Enabled SQL service reference, , you must complete the tasks described in Before You Begin: REST Enabled SQL Service Requirements and create the reference as described in Creating a REST Enabled SQL Service Reference.

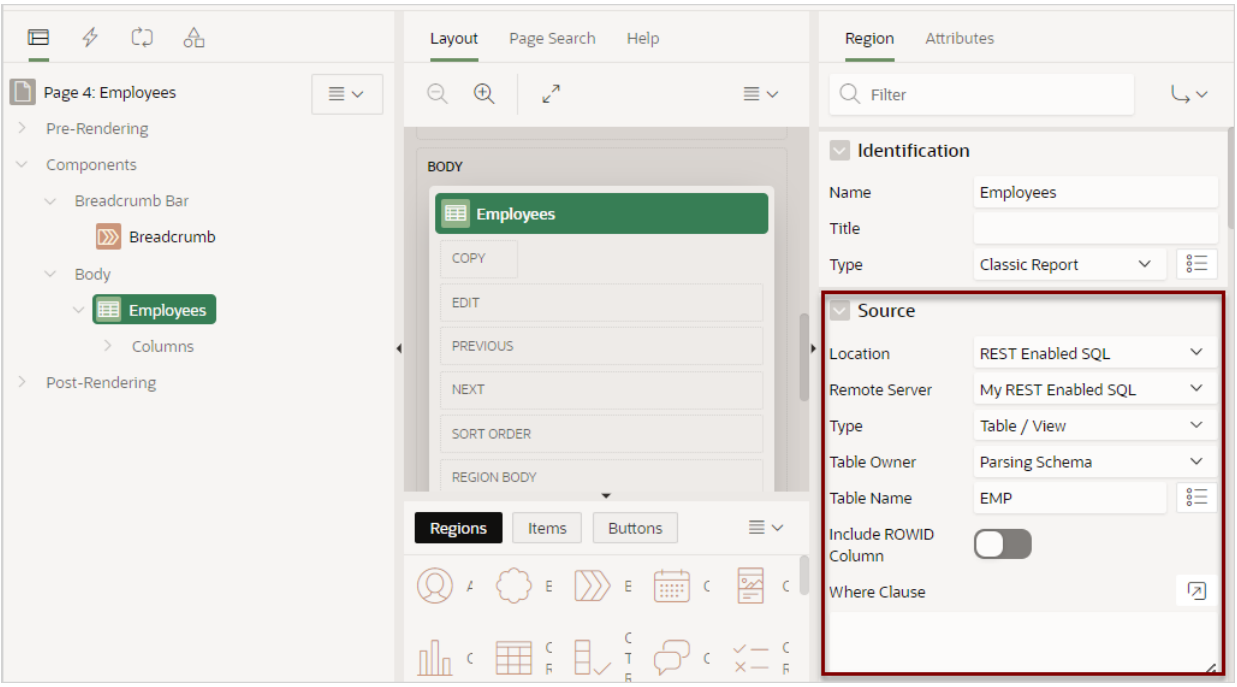
The following examples demonstrate how to configure the Source attributes for an existing Classic report and chart to point to a remote database using a REST Enabled SQL service reference named **My REST Enabled SQL**.

To configure existing components to point to a remote database using a REST Enabled SQL service reference:

1. View the page containing the components to be reconfigured in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

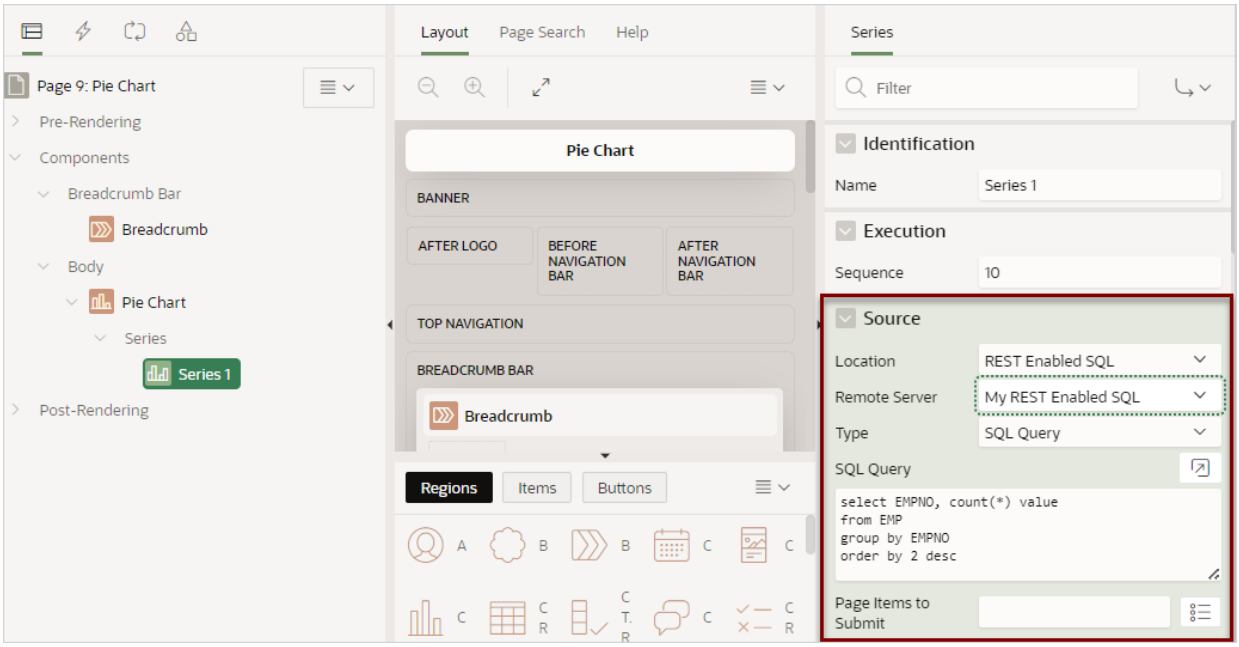
   Page Designer appears.
2. In Page Designer, select the region containing the component (if not already selected) and edit the Source attributes.
3. To specify a remote database for a report:
   a. In the Rendering tab (left pane), select the region (for example, **Classic Report**).
   b. In the Property Editor, edit the following:
      - Source, Location - Select **REST Enabled SQL**.
      - Source, Remote Server - Select the defined REST Enabled SQL Service reference (for example, **My REST Enabled SQL**.
      - Source, Type - Select how the data is queried. Select **Table/View** or **SQL Query**.
      - Configure the table or provide a SQL Query.
      - Configure the remaining attributes as needed.

      In the following example, **Classic Report** is selected and the Source points to a Remote Server named **My REST Enabled SQL**.



4. To specify a remote database for chart:
   a. In the Rendering tab (left pane), find and expand the region containing the chart.
   b. Expand **Series** and select the Series that defines the chart.
   c. In the Property Editor under Source, edit the following:
      - Location - Select **REST Enabled SQL**.
      - Remote Server - Select the defined REST Enabled SQL Service reference (for example, **My REST Enabled SQL**.
      - Type - Select how the data is queried. Select **Table/View** or **SQL Query**.
      - Configure the table or provide a SQL Query.
      - Configure the remaining attributes as needed.

      In the following example, **Series 1** is selected and the Source points to a Remote Server named **My REST Enabled SQL**.



5. Click **Save** or **Save and Run Page**.

18.6 Managing Remote Servers

Store REST Data Source server information as a Remote Server object.

- Understanding Remote Servers
  Create a Remote Server object to store REST Data Source server information.
- Creating a Remote Server
  Create a Remote Server from Shared Components.
- Editing or Deleting a Remote Server
  Edit or delete a Remote Server from Shared Components.
- Viewing Remote Server Utilization
  View the Remote Server Utilization report.
- Viewing Remote Server History
  View the Remote Server History report.

18.6.1 Understanding Remote Servers

Create a Remote Server object to store REST Data Source server information.

- About Remote Servers
  Create a Remote Server object to store REST Data Source server information.
- How APEX Stores REST Data Source Information
  Learn how Oracle APEX stores REST Data Source information.
- About Flexible Remote Servers
  A Flexible Remote Server enables you to populate the server portion of the Endpoint URL dynamically using a callback procedure you specify in the Configuration Procedure attribute.
- Exporting and Importing Remote Server Information
  Learn about exporting and importing Remote Server information.

> **See Also:**
>
> Managing REST Data Sources

18.6.1.1 About Remote Servers

Create a Remote Server object to store REST Data Source server information.

Oracle APEX stores REST Data Source server information (for example, REST APIs) as a Remote Server object. Remote Servers can be shared among multiple REST Data Sources. Changing Remote Server properties affects all REST Sources referencing that object. For example, you can change a Remote Server's Base URL to move all associated REST Sources from a test to a production system. Since Remote Servers are stored at the workspace-level, Remote Servers are visible in all applications in a workspace.

When you create a Remote Server you select one of the following Sever Types:

- **REST Data Source** - A Remote Server for consuming a remote REST API.
- **Authentication** - A Remote Server used for authentication.
- **Print Server** - A Remote Server for an external print server. You can create a Print Server in your workspace and use it in an application. By default, the print server configuration at instance level is used.
- **File Server** - A Remote Server for remote file storage.

18.6.1.2 How APEX Stores REST Data Source Information

Learn how Oracle APEX stores REST Data Source information.

Oracle APEX splits the endpoint URL of a REST Data Source into two parts. The first part is the server-specific part and is stored as a separate entity called the **Remote Server**. You can reuse a Remote Server with multiple REST Data Sources if each one uses the same server, port, and URL Path Prefix (context root). Remote Servers are stored at the workspace-level and therefore visible in all applications.

The second part of the endpoint URL and is specific to the REST Data Source. Multiple REST Data Sources can share one Remote Server, thus sharing information such as the Base URL and Authentication. If you change Remote Server attributes, the change impacts all REST Data Sources using the Remote Server. Remote Servers make it easy to move a collection of REST Data Source. For example, you can move from a test system to a production system by changing the URL within the Remote Server object.

18.6.1.3 About Flexible Remote Servers

A Flexible Remote Server enables you to populate the server portion of the Endpoint URL dynamically using a callback procedure you specify in the Configuration Procedure attribute.

Flexible Remote Server works as follows:

- When the REST Data Source is invoked, the APEX Engine evaluates the Remote Server, Configuration Procedure attribute.
- If the Configuration Procedure attribute has a value, either stored in the database or in an anonymous block in the **PL/SQL Code** attribute, the APEX Engine executes the procedure

to fetch a collection of substitution parameters that are then used to change the Endpoint URL.

Flexible Remote Server Use Cases

Common use cases for using a Flexible Remote Server include:

- **Environment dependencies** - A Flexible Remote Server is a good choice when the environment is dependent upon where the server or application is deployed (for example, in production, test or development instances). Instead of manually changing the Endpoint URL attribute on the Edit Remote Server page for each environment, you can use an API to change it dynamically.

  > **Tip:**
  >
  > You can also use `APPLICATION_ADMIN.SET_REMOTE_SERVER` procedure to change the Endpoint URL in a deployment script. See SET_REMOTE_SERVER Procedure in *Oracle APEX API Reference*

- **Application data or the application user dependencies** - A Flexible Remote Server enables you to use the Configuration Procedure attribute to change the Endpoint URL at runtime depending on dynamic parameters. These dynamic parameters are flexible and can be based on a configuration table, items in Session State, Application Items or Substitution Items.

  The results of the Configuration Procedure are cached and reused within the same request (page view or page processing).

> **Note:**
>
> Flexible Remote Servers are only supported for **REST Data Source** and **Authentication** Server Types or Remote Servers used in REST Enabled SQL.

Configuring a Flexible Remote Server

To configure a Flexible Remote Server, edit the Remote Server and enter a procedure name in the **Configuration Procedure** attribute. This procedure can be the name of a procedure stored in the database, stored in a database package procedure, or a procedure stored in the **PL/SQL Code** attribute.

This signature of the procedure must have two parameters of type:

- `p_info in apex_plugin.t_remote_server_info`
- `p_config in out apex_plugin.t_remote_server_config`

The parameter `p_config` has two optional attributes which can be changed in the configuration procedure:

- `p_config.base_url` - *`base_url`* changes the Endpoint URL.
- `p_config.substitutions` - *`substitutions`* assigns name/value pairs using `apex_t_varchar2`. Oracle APEX substitutes each `#NAME#` in the base url or Endpoint URL.

Together these two attributes determine the actual Endpoint URL the Remote Server uses.

> **Tip:**
>
> To view Configuration Procedure examples, see item Help for the **Configuration Procedure** attribute.

The following example changes the base URL if the application ID is 100:

```
procedure my_server_config(
  p_info    in  apex_plugin.t_remote_server_info,
  p_config  out apex_plugin.t_remote_server_config )
is
begin
  if v('APP_ID') = 100 then
    p_config.base_url := 'http://example100.com';
  else
    p_config.base_url := 'http://example.com';
  end if;
end;
```

You can also can use placeholders by enclosing the "*name*" with the number symbol (#) and use substitutions to replace the value in those placeholders. Consider the following example:

```
procedure my_server_config(
  p_info in apex_plugin.t_remote_server_info,
  p_config out apex_plugin.t_remote_server_config )
is
begin
  p_config.base_url := 'http://example#app_id#/#tenant#.com';
  p_config.substitutions := apex_t_varchar2();
  apex_string.plist_put( p_config.substitutions, 'app_id', v('APP_ID') );
  apex_string.plist_put( p_config.substitutions, 'tenant', v('TENANT') );
end;
```

> **See Also:**
>
> - Creating a Remote Server
> - Editing or Deleting a Remote Server

18.6.1.4 Exporting and Importing Remote Server Information

Learn about exporting and importing Remote Server information.

When you export an application, referenced Remote Servers are added to the export file. When you import the application into another workspace, APEX checks whether the target workspace already contains Remote Servers with the same **static ID**. If a Remote Server already exists, the application uses it. Otherwise the Remote Servers from the import file are created in the target workspace.

18.6.2 Creating a Remote Server

Create a Remote Server from Shared Components.

To create a Remote Server object:

1. Navigate to the Remove Server page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Remote Servers**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Remote Servers**.
2. On the Remote Servers page, click **Create**.
3. Under General:
   a. Name - Enter a name for the Remote Server object. Remote Server names must be unique within the application.
   b. Static Identifier - Enter a static ID for the Remote Server.
   c. Server Type - Select one of the following:
      - **REST Data Source** - A Remote Server for consuming a remote REST API.
      - **Authentication** - A Remote Server used for authentication.
      - **Print Server** - A Remote Server for an external print server. You can create a Print Server in your workspace and use it in an application. By default, the print server configuration at instance level is used.
      - **File Server** - A Remote Server for remote file storage.
   d. Endpoint URL - Enter the base URL of the Remote Server.
   e. Prompt on Install - Select **On** to have APEX prompt for the Endpoint URL after the application installs into another workspace.
4. Under Advanced:

   > **Tip:**
   >
   > Depending upon the Remote Server Type, some of the following attributes do not display.

   a. Static Identifier - Use the Static ID to reference the Remote Server in API calls.
   b. PL/SQL Code - Enter a procedure which configures the Endpoint URL at runtime. To view examples, see item Help.

c. Configuration Procedure - Enter the name of a procedure which configures the Endpoint URL at runtime. This procedure can be the name of a procedure stored in the database, stored in a database package procedure, or a procedure stored in the **PL/SQL Code** attribute. To view examples, see item Help.

d. Comments - Enter any comments or notes. These comments never display when running the application.

5. Click **Create**.

18.6.3 Editing or Deleting a Remote Server

Edit or delete a Remote Server from Shared Components.

To edit a Remote Server:

1. Navigate to the Credentials page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Remote Servers**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Remote Servers**.
2. On the Remote Server page, click the name of the Remote Server.

   The Edit Remote Server dialog appears.
3. To delete the current Remote Server, click **Delete**. Otherwise, edit the appropriate attributes.

   > **Tip:**
   >
   > Remote Servers which are referenced within an application cannot be deleted.

4. Under General:
   a. Name - Enter a name for the Remote Server object. Remote Server names must be unique within the application.
   b. Static Identifier - Enter a name for the remote server object here. Remote Server names must be unique within the application.
   c. Server Type - Options include:
      - **REST Data Source** - A Remote Server for consuming a remote REST API.
      - **Authentication** - A Remote Server used for authentication.
      - **Print Server** - A Remote Server for an external print server. You can create a Print Server in your workspace and use it in an application. By default, the print server configuration at instance level is used.

- **File Server** - A Remote Server for remote file storage.

d. Endpoint URL - Enter the base URL of the remote server.

e. Prompt on Install - If **On**, APEX prompts for the Endpoint URL after the application has been installed into another workspace.

5. Under Advanced:

> **Tip:**
>
> Depending upon the Remote Server **Type**, some of the following attributes do not display.

a. Static Identifier - Use the Static ID to reference the Remote Server in API calls.

b. PL/SQL Code - Enter a procedure which configures the Endpoint URL at runtime. To view examples, see item Help.

c. Configuration Procedure - Enter the name of a procedure which configures the Endpoint URL at runtime. This procedure can be the name of a procedure stored in the database, stored in a database package procedure, or a procedure stored in the **PL/SQL Code** attribute. To view examples, see item Help.

d. Comments - Enter any comments or notes. These comments only display within App Builder and never display when running the application.

6. To save your changes, click **Apply Changes**.

18.6.4 Viewing Remote Server Utilization

View the Remote Server Utilization report.

To view Remote Server utilization:

1. Navigate to the Remove Server page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Remote Servers**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Remote Servers**.
2. On the Remote Servers page, click **Utilization**.

   The Utilization page displays where Remote Servers are used within the current workspace.

18.6.5 Viewing Remote Server History

View the Remote Server History report.

To view recent modifications to Remote Servers:

1. Navigate to the Remove Server page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Remote Servers**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Remote Servers**.
2. On the Remote Servers page, click **History**.

   The History page displays recent modifications made to Remote Servers in the current workspace.

18.7 Managing Web Credentials

Store authentication credentials for external REST services or REST Enabled SQL services.

- Understanding Web Credentials
  Use Web credentials to connect to external REST services or REST Enabled SQL services.
- Creating Web Credentials
  Create Web credentials from either Workspace Utilities or Shared Components.
- Editing or Deleting Credentials
  Edit or delete Web credentials from either Workspace Utilities or Shared Components.
- Viewing Credential Utilization
  View the Web credential Utilization report.
- Viewing Credential History
  View the Web Credential History report.

18.7.1 Understanding Web Credentials

Use Web credentials to connect to external REST services or REST Enabled SQL services.

Creating Web Credentials securely stores and encrypts authentication credentials for use by Oracle APEX components and APIs. Credentials cannot be retrieved back in clear text. Credentials are stored at the workspace-level and therefore are visible in all applications.

- Protecting Web Credentials by Defining Valid URLs
  Protect Web credentials by adding valid URLs.

- About Supported Authentication Types in Web Credentials
  Learn about supported authentication types in Web credentials.
- Using a Database Credential with Web Credentials
  Learn about using a database credential with Web credentials.
- Exporting and Importing Web Credentials
  Learn about exporting and importing Web credentials.

> **See Also:**
>
> Managing REST Enabled SQL References

18.7.1.1 Protecting Web Credentials by Defining Valid URLs

Protect Web credentials by adding valid URLs.

You can optionally protect Web credentials by adding valid URLs to the **Valid for URLs** attribute. Adding URLs to the **Valid for URLs** attribute prevents APEX from accidentally sending sensitive credentials to a different server. Whenever a Web credential is used, APEX checks whether the URL matches what is defined in the **Valid for URLs** attribute. If the URL does not match, APEX raises the runtime error, `Credential is not allowed to be used for this URL endpoint`.

When adding URLs to this attribute, place each URL into a new line. The URL endpoint being used must start with one of the URLs provided here. See field-level Help for examples.

18.7.1.2 About Supported Authentication Types in Web Credentials

Learn about supported authentication types in Web credentials.

Web credentials support the following Authentication Types:

- **Basic Authentication** - Sends username and password in Base64-encoded form as the **Authorization** request header.
- **OAuth2 Client Credentials Flow** - APEX exchanges the client ID and client secret for an **Access Token** using a token server URL. The access token is then used to perform the actual request. If the access token is expired, APEX will transparently request a new one.
- **HTTP Header** - The credential is added to the REST Request as a HTTP Header. The name of the credential is the HTTP Header name, the **Secret** of the credential is the HTTP Header value. APEX does not add these secrets to the Debug log or any other logs.
- **URL Query String** - The credential is added to the URL of the REST Request as a Query String Parameter (for example: `?name=value`). This option can be useful with API keys. APEX does not add these secrets to the Debug log or any other logs. However, proxy or other intermediate servers will log these credentials in their log files.

18.7.1.3 Using a Database Credential with Web Credentials

Learn about using a database credential with Web credentials.

> **Note:**
>
> This feature is only available on Oracle Database release 23c or Autonomous Database release 19c or later.

For Oracle Database 23c or Autonomous Database 19c or later, Web credentials can reference a **database credential** to store the Client ID and Client Secret when the Web credential is of the Basic Authentication or OAuth2 Client Credentials type. Other Web credential types do not support database credentials.

Using a database credential changes the behavior of HTTP requests being done and how Web credentials work:

- If a database credential is used, all HTTP requests are done in the context of the application's parsing schema and not in the context of the APEX engine. Therefore, the PL/SQL network ACLs must be granted to the application's parsing schema. To learn more, DBMS_NETWORK_ACL_ADMIN in *Oracle Database PL/SQL Packages and Types Reference*.
- A database credential is completely maintained by the database, Oracle APEX does not have access to the stored secrets. Since APEX cannot enforce the **Valid for URLs** attribute, this attribute is disabled when a database credential is used.

> **Tip:**
>
> You can maintain a database credential using the DBMS_CREDENTIAL or DBMS_CLOUD packages.

About Instance and Schema Database Credentials

If a Web credential references a database credential, the database credential must be accessible to the application's parsing schema. If the `INSTANCE_DBMS_CREDENTIAL_ENABLED` instance parameter is enabled (set to the '`Y`' value), then a database credential, which is accessible to the APEX engine schema (`APEX_`*`NNNNNN`*) can be used in all workspaces.

> **See Also:**
>
> `INSTANCE_DBMS_CREDENTIAL_ENABLED` in Available Parameter Values, APEX_INSTANCE_ADMIN in *Oracle APEX API Reference*

18.7.1.4 Exporting and Importing Web Credentials

Learn about exporting and importing Web credentials.

When you export an application, used credentials are added to the export file. When you import the application into another workspace, APEX checks whether the target workspace

already contains credentials with the same static ID. If a credential already exists, the application uses it. Otherwise the credential from the import file is created in the target workspace.

18.7.2 Creating Web Credentials

Create Web credentials from either Workspace Utilities or Shared Components.

To create Web credentials:

1. Navigate to the Web Credentials page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Web Credentials**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Credentials**.
2. On the Web Credentials page, click **Create**.
3. Configure the Attributes on the Web Credentials page.
   a. **Name** - Enter a descriptive name.**Static ID**
   b. **Static ID** - Use the static identifier to reference the credential in API calls.
   c. **Authentication Type** - Select one of the following:
      - **Basic Authentication**
      - **OAuth2 Client Credentials**
      - **OCI Native Authentication**
      - **HTTP Header**
      - **URL Query String**
      - **Key Pair**

      To learn more about available Authentication Types, see field level Help.

      > **Note:**
      >
      > The the Authentication Type you select, determines what options that display next.

4. **Use Database Credential** - This switch only displays if the Authentication Type is **Basic Authentication** or **OAuth2 Client Credentials**.

   If the **Use Database Credential** switch is enabled, a section to configure the name of a database credential appears. Since a database credential is managed by the database

and not by APEX, the attributes for **Client ID**, **Client Secret**, and **Valid For URLs** disappear.

To learn more about **Use Database Credential**, see field-level Help.

5. If the Authentication Type, **Basic Authentication**:
   a. **Client ID or Username** - Enter the Username. APEX does not store this information encrypted.
   b. **Client Secret or Password** - Enter the password.
   c. **Verify Client Secret or Password** - Enter the password again to verify your input.
   d. **Valid for URLs** - APEX checks whether the URL a Web credentials uses matches the URLs in this attribute. When adding URLs to this attribute:
      - Place each URL on a new line.
      - The URL endpoint being used must start with one of the URLs provided here.

      See field-level Help for examples.
   e. **Prompt On Install** - Choose whether prompts for this credential display when the application is imported on another APEX instance.
   f. **Comments** - Enter any comments or notes here. These comments never display when running the application.
6. For Authentication Type, **OAuth2 Client Credentials**:
   a. **OAuth Scope** - Permissions represented by the Access Token in OAuth 2.0 terms are known as scopes. The scope parameter allows the application to express the desired scope of the access request.

      If your authentication server requires a scope to be specified for the access token request, provide it here. The OAuth2 access token will then be requested with the following request body:

      ```
      grant_type=client_credentials&scope={scope}
      ```
   b. **Client ID or Username** - Enter the Client ID. APEX does not store this information encrypted.
   c. **Client Secret or Password** - Enter the password.
   d. **Verify Client Secret or Password** - Enter the password again to verify your input.
   e. **Valid for URLs** - APEX checks whether the URL a Web Credential uses matches the URLs in this attribute. When adding URLs to this attribute:
      - Place each URL on a new line.
      - The URL endpoint being used must start with one of the URLs provided here.

      See field-level Help for examples.
   f. **Prompt On Install** - Choose whether prompts for this credential display when the application is imported on another APEX instance.
   g. **Comments** - Enter any comments or notes here. These comments never display when running the application.
7. For Authentication Type, **OCI Native Authentication**:
   a. **OCI User ID** - Enter the Oracle Cloud Infrastructure User OCID. APEX does not store this information encrypted.

b. **OCI Private Key** - Enter the private key in PEM format for OCI authentication. APEX stores this information encrypted and secure, so it cannot be retrieved back in clear text.

c. **OCI Tenancy ID** - Enter the Oracle Cloud Infrastructure Tenancy's OCID.

d. **OCI Public Key Fingerprint** - Enter the public RSA key fingerprint for OCI authentication.

e. **Valid for URLs** - APEX checks whether the URL a Web Credential uses matches the URLs in this attribute. When adding URLs to this attribute:

   - Place each URL on a new line.
   - The URL endpoint being used must start with one of the URLs provided here.

   See field-level Help for examples.

f. **Prompt On Install** - Choose whether prompts for this credential display when the application is imported on another APEX instance.

g. **Comments** - Enter any comments or notes here. These comments never display when running the application.

8. For Authentication Type, **HTTP Header**:

   a. **Credential Name** - Provide the name of the HTTP Header or URL Query String parameter to use for this credential.

   b. **Credential Secret** - Enter the value (or secret) of the credential.

   c. **Valid for URLs** - APEX checks whether the URL a Web Credential uses matches the URLs in this attribute. When adding URLs to this attribute:

      - Place each URL on a new line.
      - The URL endpoint being used must start with one of the URLs provided here.

      See field-level Help for examples.

   d. **Prompt On Install** - Choose whether prompts for this credential display when the application is imported on another APEX instance.

   e. **Comments** - Enter any comments or notes here. These comments never display when running the application.

9. For Authentication Type, **URL Query String**:

   a. **Credential Name** - Enter the name of the URL Query String parameter to use for this credential.

   b. **Credential Secret** - Enter the value (or secret) of the credential.

   c. **Valid for URLs** - APEX checks whether the URL a Web Credential uses matches the URLs in this attribute. When adding URLs to this attribute:

      - Place each URL on a new line.
      - The URL endpoint being used must start with one of the URLs provided here.

      See field-level Help for examples.

   d. **Prompt On Install** - Choose whether prompts for this credential display when the application is imported on another APEX instance.

   e. **Comments** - Enter any comments or notes here. These comments never display when running the application.

10. For Authentication Type, **Key Pair**:

a. **Public Key** - Enter the public key for for the key pair credentials. APEX does not store this information encrypted.

b. **Private Key** - Enter the private key for the key pair credentials. APEX stores this information encrypted, so it cannot be retrieved back in clear text.

c. **Verify Private Key** - Enter the private key again, to verify your input.

11. Click **Apply Changes**.

18.7.3 Editing or Deleting Credentials

Edit or delete Web credentials from either Workspace Utilities or Shared Components.

To Edit or delete Web credentials:

1. Navigate to the Web Credentials page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Web Credentials**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Credentials**.
2. On the Web Credentials page, select the credential name.

   The Create/Edit page appears.
3. To edit a credential:
   a. Edit the appropriate attributes. To learn more about an attribute, see field level Help.
   b. To save your changes, click **Apply Changes**.
4. To delete a credential:
   a. To delete a credential, click **Delete**.
   b. When prompted, click **OK**.

> **Tip:**
>
> A credential cannot be deleted when it is being referenced somewhere in the workspace.

18.7.4 Viewing Credential Utilization

View the Web credential Utilization report.

To view Web Credential Utilization report:

1. Navigate to the Web Credentials page:

- From Workspace Utilities:
  a. On the Workspace home page, click **App Builder**.
  b. Click **Workspace Utilities**.
  c. Click **Web Credentials**.
- From Shared Components:
  a. On the Workspace home page, click **App Builder**.
  b. Select an application.
  c. On the Application home page, click **Shared Components** in the center of the page.
  d. Under Workspace Objects, select **Credentials**.

2. On the Web Credentials page, click **Utilization**.

   The top of the Utilization page displays used credentials, the associated component type, the application id, and the component name. Unused credentials display at the bottom of the page.

3. To delete unused credentials, click **Delete Unused**.

18.7.5 Viewing Credential History

View the Web Credential History report.

To view Web Credential History report:

1. Navigate to the Web Credentials page:
   - From Workspace Utilities:
     a. On the Workspace home page, click **App Builder**.
     b. Click **Workspace Utilities**.
     c. Click **Web Credentials**.
   - From Shared Components:
     a. On the Workspace home page, click **App Builder**.
     b. Select an application.
     c. On the Application home page, click **Shared Components** in the center of the page.
     d. Under Workspace Objects, select **Credentials**.
2. On the Web Credentials page, click **History**.

   The History page displays recent modifications made to Credentials in the current workspace.

18.8 Managing Generative AI in APEX

Create Generative AI Services to include AI features in App Builder and enable support for developers to include Generative AI in their applications.

- Understanding Generative AI Support in APEX
  Learn about Generative Artificial Intelligence (AI) Service support in APEX.

- Managing Generative AI Services
  Create Generative AI Services on the Generative AI Services page in Workspace Utilities.
- Using APEX Assistant
  Use integrated AI Chat capability (called APEX Assistant) in Code Editors within App Builder.
- Including Generative AI in Applications
  Include Generative AI in your application by creating a Generative AI Service, configuring application attributes on the AI page, and creating a dynamic action to either display the AI Assistant or generate text with AI.

18.8.1 Understanding Generative AI Support in APEX

Learn about Generative Artificial Intelligence (AI) Service support in APEX.

About Generative AI Services

APEX stores Generative AI Service information as a Generative AI object (or Generative AI Service) on the Generative AI Services page in Workspace Utilities. Since Generative AI Services are stored at the workspace-level, they are visible and available to all applications in the workspace. Generative AI Services support many commercial Generative AI products including Open AI, OCI Generative AI Service, and Cohere.

When you create a Generative AI Service, APEX stores the AI Service key as a Web Credential. The Generative AI Service actually stores information about the AI Service being used.

If you export an application that uses Generative AI Service, the export file omits the actual Web Credential secret from the Web Credential definition. When importing the export file, you must set the Web Credential secret manually or set it using an API.

Administrator Controls

Instance administrators can manage support for AI Services at the workspace or instance-level. To learn more, see Editing Workspace Information and Managing Support for AI Services for an Instance in *Oracle APEX Administration Guide*.

About Enabling a Generative AI in App Builder

When you create a Generative AI Service, enabling the **Used by App Builder** setting makes the following features available:

- An integrated Generative AI option in the Create Application Wizard.
- Generative AI support (APEX Assistant) in all APEX Code Editors.

> **Note:**
>
> Only one Generative AI Service at a time can be configured with the **Used by App Builder** setting enabled.

See Creating an App Using Generative AI and Using APEX Assistant.

Including Generative AI in Runtime Apps

You can include AI in your applications by creating a dynamic action which opens the AI Assistant. The actual steps involved can be as simple as hooking a button to the dynamic action.

See Including Generative AI in Applications.

18.8.2 Managing Generative AI Services

Create Generative AI Services on the Generative AI Services page in Workspace Utilities.

> **Tip:**
>
> You can also navigate to the Generative AI Services page from Shared Components. In App Builder, select an application in and then click **Shared Components**. Find **Generative AI** and then click **AI Services**.

- Creating a Generative AI Service Object
  Create a Generative AI Service object on the Generative AI Services, Create/Edit page.
- Editing or Deleting a Generative AI Service
  Edit or delete a Generative AI Service on the Generative AI Services, Create/Edit page.
- Copying or Subscribing to an AI Configuration
  Copy an AI Configuration from the current application or from another application. When copying an AI Configuration from another application, you can also subscribe to it.
- Viewing Generative AI Service Utilization
  View a report that details where Generative AI Service objects are used within the current workspace.
- Viewing Generative AI Service History
  View a report of recent modifications made to Generative AI Services in this workspace.

18.8.2.1 Creating a Generative AI Service Object

Create a Generative AI Service object on the Generative AI Services, Create/Edit page.

> **Note:**
>
> Before creating a Generative AI Service, you need an API key or OCI Credentials from your AI Provider. To learn more, contact your AI Provider.

Each AI Service must have a unique Name and Static ID within the workspace. Developers can select from a list of supported AI Services.

> **Note:**
>
> Enable the **Used by App Builder** setting to add a Generative AI option in the Create Application Wizard and include Generative AI support (APEX Assistant) in all APEX Code Editors. Only one Generative AI Service at a time can be configured with the **Used by App Builder** setting enabled.

To create a generative AI Service:

1. Navigate to the Generative AI Services page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click the **Workspace Utilities** icon.

      > **Tip:**
      >
      > You can also access the Generative AI Services page from the menu bar. From the APEX menu bar, select **App Builder**, **Workspace Utilities**, and then **All Workspace Utilities**.

      The Workspace Utilities page appears.
   c. On the Workspace Utilities page, click **Generative AI**.

      The Generative AI Services page appears.
2. To create a Generative AI Service object, click **Create**.

   The Create/Edit page appears.
3. Under Identification:
   a. Identification, AI Provider - Select the AI Provider to use for this Generative AI Service. Options include:
      - **OCI Generative AI Service**
      - **Open AI**
      - **Cohere**
   b. Identification, Name - The name of the Generative AI Service. The name will display on the Generative AI Services page in Workspace Utilities.

      Example: `HCM Open AI Service`
   c. Identification, Static ID - The Static ID for the Generative AI Service. The static ID is used when using the service with the `APEX_AI` package (`APEX_AI.CHAT`).

      Example: `HCM_OPENAI`

   The UI changes depending upon the **AI Provider** you select. The steps that follow describe properties for the Open AI, OCI Generative AI Service, and Cohere.
4. If **AI Provider** is **OCI Generative AI Service**:
   a. OCI Generative AI, Compartment ID - The Oracle Cloud Infrastructure Compartment ID.
   b. OCI Generative AI, Region - The Oracle Cloud Infrastructure Region.

      The Default Region is `us-chicago-1`

c. OCI Generative AI, Model ID - The OCI Generative AI Model ID. The default Model ID is `cohere.command-r-16k`.

   To learn more about OCI Generative AI chat models, see OCI Generative AI chat models in *Oracle Cloud Infrastructure Documentation*

d. Settings, Used by App Builder - Indicates whether this Generative AI Service is used by App Builder.

e. Settings, Base URL - The base URL of the Generative AI Service. The base URL is typically the REST API endpoint for the specified Generative AI Provider. Make sure the URL in the selected Web Credential is reflected in the base URL of the Generative AI Service.

   For advanced use cases, you can specify Query Parameters as part of the Base URL by appending the following to the Base URL:

   ```
   ?param1=value1&param2=value2&param3=value3 ...
   ```

   Where `param<X>` is the name of the query parameter and `value<X>` the value of parameter `param<X>`.

f. Credentials, Credential - Select the Web Credential to use for this Generative AI Service. Customers must sign-up for or use existing credentials of their respective AI provider.

g. Credentials, OCI User ID - Enter the Oracle Cloud Infrastructure User OCID. Oracle APEX does not store this information encrypted.

h. Credentials, OCI Private Key - Enter the private key in PEM format for OCI authentication. Oracle APEX stores this information as encrypted and secure, so it cannot be retrieved back in clear text.

i. Credentials, OCI Tenancy ID - Enter the Oracle Cloud Infrastructure Tenancy's OCID.

j. Credentials, OCI Public Key Fingerprint - Enter the public RSA key fingerprint for OCI authentication.

   > **Tip:**
   >
   > To learn more, see Required Keys and OCIDs in *Oracle Cloud Infrastructure Documentation*.

k. To validate the information you enter prior to completing the setup, click **Test Connection**.

   If the connection is valid, the following message displays:

   ```
   Connection Succeeded!
   ```

   If the connection fails, resolve the errors that display.

l. Advanced, Additional Attributes - Click Additional Attributes to include additional attributes that will be added to the JSON request payload as part of performing a Generative AI Service operation.

m. Advanced, HTTP Headers - Additional HTTP headers used in the Generative AI Service (REST) request. HTTP headers are specified in the following format:

   ```
   name_1=value_1
   name_2=value_2
   ```

n. Advanced, Comments - Enter any developer comments or notes.

5. If **AI Provider** is **Open AI**:
   a. Settings, Used by App Builder - Indicates whether this Generative AI Service is used by App Builder.
   b. Settings, Base URL - The base URL of the Generative AI Service. For normal use, accept the default. The base URL is typically the REST API endpoint for the specified Generative AI Provider. Make sure the URL in the selected Web Credential is reflected in the base URL of the Generative AI Service.

      For advanced use cases, you can specify Query Parameters as part of the Base URL by appending the following to the Base URL:

      ```
      ?param1=value1&param2=value2&param3=value3 ...
      ```

      Where `param<X>` is the name of the query parameter and `value<X>` the value of parameter `param<X>`.
   c. Credentials, Credential - Select the Web Credential to use for this Generative AI Service. Customers must sign-up for or use existing credentials of their respective AI provider.

      The Web Credential can be created as part of creating the Generative AI service. Once created, the Web Credential will also appear on the Web Credentials page in Workspace Utilities.
   d. Credentials, API Key - If you are creating a new credential, enter the API key.

      The API key is generated from your Open AI account. For more information, see your Open AI documentation.
   e. To validate the information you enter prior to completing the setup, click **Test Connection**.

      If the connection is valid, the following message displays:

      ```
      Connection Succeeded!
      ```

      If the connection fails, resolve the errors that display.
   f. Advanced, Additional Attributes - Click Additional Attributes to include additional attributes that will be added to the JSON request payload as part of performing a Generative AI Service operation.

      For more information, see the REST API documentation of the selected AI provider.
   g. Advanced, AI Model - Enter an AI Model. For Open AI, the default model is `gpt-3.5-turbo`.
   h. Advanced, HTTP Headers - Additional HTTP headers used in the Generative AI Service (REST) request. HTTP headers are specified in the following format:

      ```
      name_1=value_1
      name_2=value_2
      ```
   i. Advanced, Comments - Enter any developer comments or notes.
6. If **AI Provider** is **Cohere**:
   a. Settings, Used by App Builder - Indicates whether this Generative AI Service is used by App Builder.
   b. Settings, Base URL - The base URL of the Generative AI Service. The base URL is typically the REST API endpoint for the specified Generative AI Provider. Make sure the URL in the selected Web Credential is reflected in the base URL of the Generative AI Service.

For advanced use cases, you can specify Query Parameters as part of the Base URL by appending the following to the Base URL:

```
?param1=value1&param2=value2&param3=value3 ...
```

Where `param<X>` is the name of the query parameter and `value<X>` the value of parameter `param<X>`.

c. Credentials, Credential - Select the Web Credential to use for this Generative AI Service. Customers must sign-up for or use existing credentials of their respective AI provider.

d. Credentials, Credential - Select the Web Credential to use for this Generative AI Service. Customers must sign-up for or use existing credentials of their respective AI provider.

   The Web Credential can be created as part of creating the Generative AI service. Once created, the Web Credential will also appear on the Web Credentials page in Workspace Utilities.

e. Credentials, API Key - If you are creating a new credential, enter the API key.

   The API key is generated from your Cohere account. For more information, see your Cohere documentation.

f. To validate the information you enter prior to completing the setup, click **Test Connection**.

   If the connection is valid, the following message displays:

   ```
   Connection Succeeded!
   ```

   If the connection fails, resolve the errors that display.

g. Advanced, Additional Attributes - Click Additional Attributes to include additional attributes that will be added to the JSON request payload as part of performing a Generative AI Service operation.

h. Advanced, AI Model - An optional model name or ID for the Generative AI Service. If no model information is given, the default model of the respective AI provider is used.

i. Advanced, HTTP Headers - Additional HTTP headers used in the Generative AI Service (REST) request. HTTP headers are specified in the following format:

   ```
   name_1=value_1
   name_2=value_2
   ```

j. Advanced, Comments - Enter any developer comments or notes.

7. Click **Create**.

18.8.2.2 Editing or Deleting a Generative AI Service

Edit or delete a Generative AI Service on the Generative AI Services, Create/Edit page.

To edit or delete a Generative AI Service:

1. Navigate to the Generative AI Services page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. On the App Builder home page, click the **Workspace Utilities** icon.

   c. On the Workspace Utilities page, click **Generative AI**.

      The Generative AI Services page appears.

2. To edit a Generative AI Service:
   a. Click the AI Service Name.

      Edit the attributes.
   b. Identification, AI Provider - Determines the AI Provider to use for this Generative AI Service. Options include:
      - **OCI Generative AI Service**
      - **Open AI**
      - **Cohere**
   c. Identification, Name - The name of the Generative AI Service. The name displays on the Generative AI Services page in Workspace Utilities.

      Example: `HCM Open AI Service`
   d. Identification, Static ID - The Static ID for the Generative AI Service. The static ID is used when using the service with the `APEX_AI` package (`APEX_AI.CHAT`).

      Example: `HCM_OPENAI`.

      The UI that displays next changes depending upon the selected **AI Provider**.
   e. Edit the appropriate attributes. To learn more about a specific attribute, see item Help.
   f. To validate the information you enter prior to completing the setup, click **Test Connection**.

      If the connection is valid, the following message displays:

      ```
      Connection Succeeded!
      ```

      If the connection fails, resolve the errors that display.
   g. Click **Apply Changes**.
3. To delete a Generative AI Service:

   > **Tip:**
   >
   > If a Generative AI Service is in use, you cannot delete it.

   a. Click the AI Service Name.
   b. Click **Delete**.
   c. Confirm your selection and click **Delete** again.

   > **Note:**
   >
   > Creating a Generative AI Service creates a Web Credential. When you delete a Generative AI Service, you must manually delete the associated Web Credential. See Editing or Deleting Credentials.

18.8.2.3 Copying or Subscribing to an AI Configuration

Copy an AI Configuration from the current application or from another application. When copying an AI Configuration from another application, you can also subscribe to it.

To copy an AI Configuration:

1. Navigate to the AI Configuration page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Applicatioin Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Services page appears.
2. On the AI Configuration, click **Copy**.
3. Copy From - Choose where to copy the AI Configuration from. Options include:
   - **This Application**
   - **Other Application**
4. To copy from the current application:
   a. Copy From - Select **This Application**.
   b. Copy Configuration - Select the configuration to copy.
   c. Name - Enter the name of the configuration you would like to create.
   d. Click **Copy**.
5. To copy from another application:
   a. Copy From - Select **Other Application**.
   b. Application - Select the application from which you want to copy.
   c. Copy Configuration - Select the configuration to copy.
   d. Subscribe - Enable **Subscribe** to subscribe to the configuration.
   e. Click **Copy**.

18.8.2.4 Viewing Generative AI Service Utilization

View a report that details where Generative AI Service objects are used within the current workspace.

To view the Generative AI Services Utilization report:

1. Navigate to the Generative AI Services page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **Generative AI**.

      The Generative AI Services page appears.
2. On the Generative AI Services page, click **Utilization**.

   The Generative AI Services Utilization report appears.

18.8.2.5 Viewing Generative AI Service History

View a report of recent modifications made to Generative AI Services in this workspace.

To view the Generative AI Services History report:

1. Navigate to the Generative AI Services page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click the **Workspace Utilities** icon.
   c. On the Workspace Utilities page, click **Generative AI**.

      The Generative AI Services page appears.
2. On the Generative AI Services page, click **History**.

   The Generative AI Services History report appears.

18.8.3 Using APEX Assistant

Use integrated AI Chat capability (called APEX Assistant) in Code Editors within App Builder.

> **Note:**
>
> Before you can use APEX Assistant, you must create an Generative AI Service and enable the **Used by App Builder** setting. See Managing Generative AI Services.

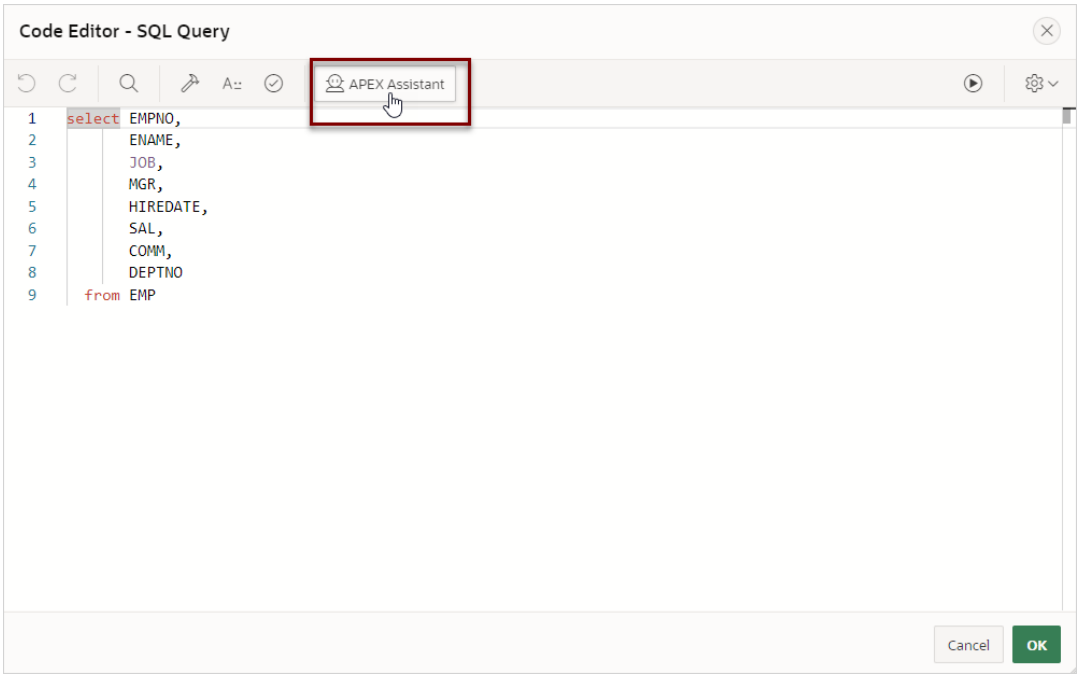
> **Tip:**
>
> APEX Assistant can also generate data models. For more information, see Creating a Data Model Using AI in *Oracle APEX SQL Workshop Guide*.

APEX Assistant adds AI chat capability to all APEX code editors to assist you with writing SQL queries, debugging code, and so on. To enable an iterative chat experience, the Chat Widget memorizes the context of the chat until the **Clear Chat** button is pressed. For example, given the SQL Query Builder use case below, a developer might ask 'add department information of the employee to the query'. Given the context in the chat, the Generative AI Service uses this information to enhance the existing query to include the desired information asked by the developer.

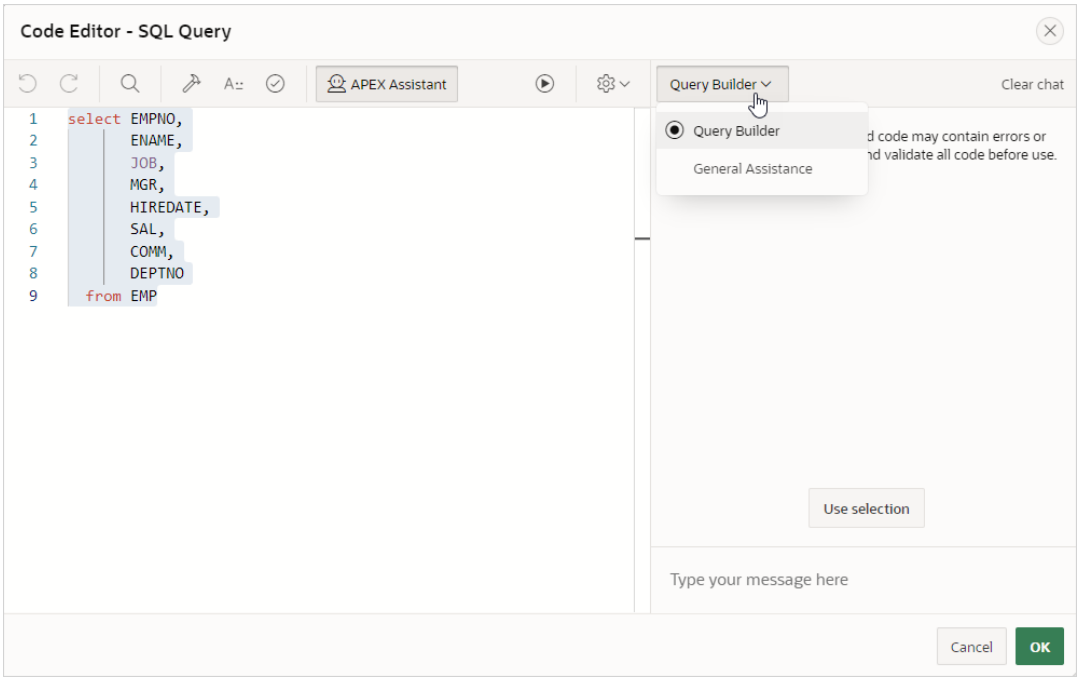
> **Caution:**
>
> AI-generated code may contain errors or security risks. Oracle recommends you always review and validate all code before use.

To use APEX Assistant:

1. Create an Generative AI Service and enable the **Used by App Builder** setting. See Managing Generative AI Services.
2. Access a Code Editor. The examples that follow show the Code Editor in Page Designer.
3. Click **APEX Assistant** on the menu bar.



The first time you access APEX Assistant a consent dialog appears. Review the dialog carefully and click **Accept** to continue. APEX Assistant appears in the right pane.



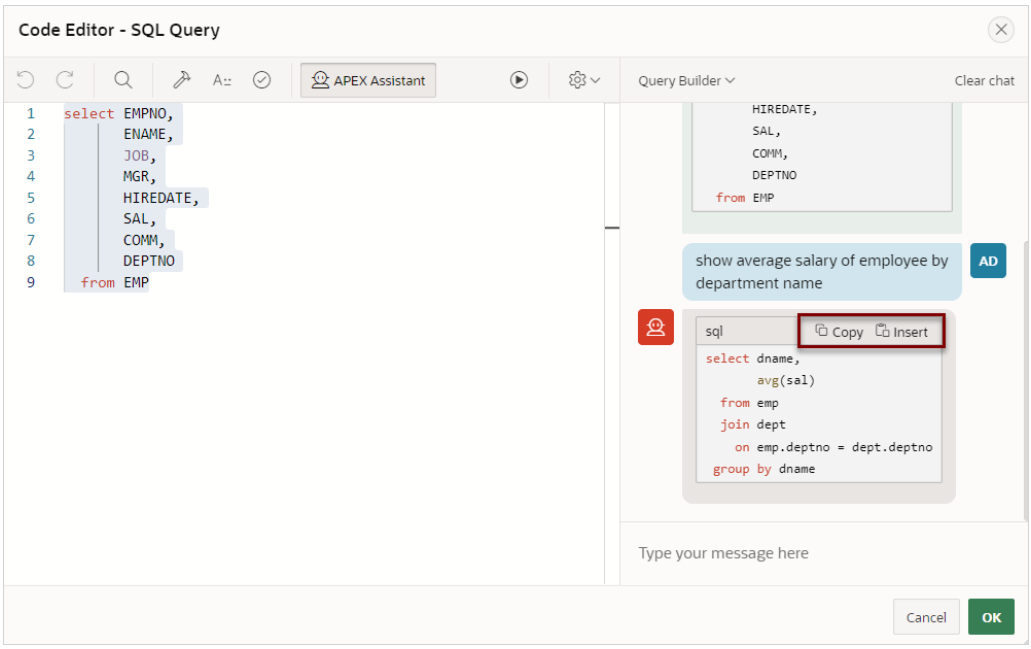
Note that the APEX Assistant menu contains two options:

- **Query Builder** - Use Query Builder to get back a simple query.

  In Query Builder mode, the AI assumes all questions are in the context of the customer schema in the database. For example, if your request that the AI create SQL for tables that do not exist in the database, APEX Assistant's responses will not be very helpful.

- **General Assistance** - Use General Assistance for general conversation, technical questions such as "Explain this" or "Improve this code." In General Assistance mode, APEX Assistant prompts you with default options such as **Use Selection**, **Improve Selection**, and **Explain Selection**.

4. **Query Builder Example**: Show average salary by department.
   a. From the menu, select **Query Builder** (if not already selected).
   b. In the Code Editor, select the query and click **Use Selection**.
   c. In **Type your message here**, enter `show average salary of employee by department name` and click **Send Message**.
   d. APEX Assistant responds with a query as shown in the following example.



   e. Click **Copy** or **Insert** to copy or insert the response into the Code Editor.
   f. To clear the chat window, click **Clear Chat**.

> **Tip:**
>
> If a query throws an error, APEX Assistant switches to General Assistance mode. Click **Help me fix this**.

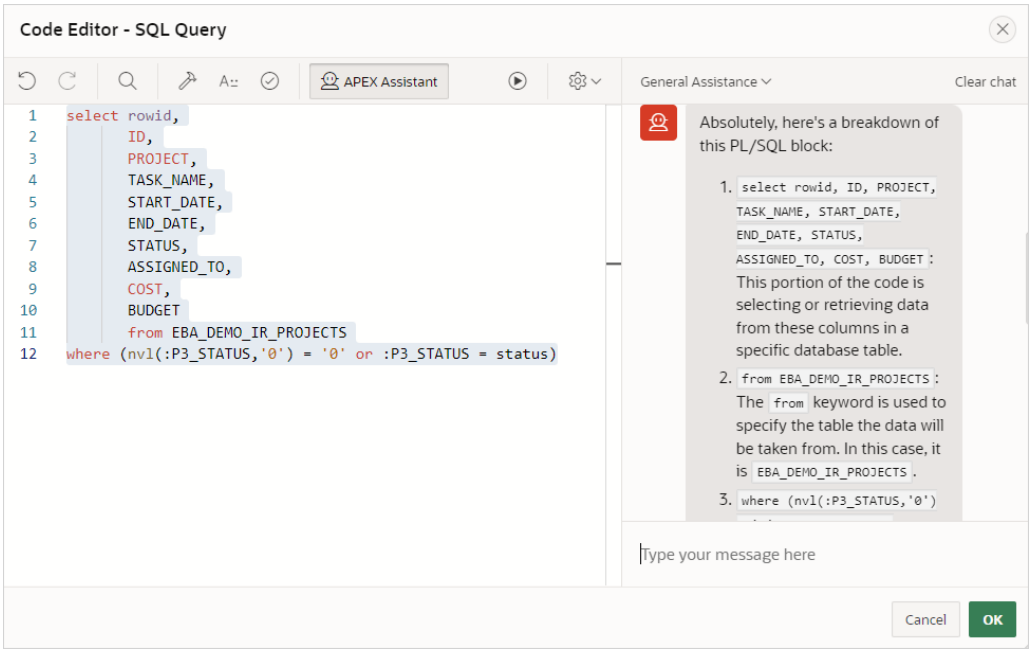
5. General Assistance example: Explain a selected query:
   a. From the menu, select **General Assistance**.

      APEX Assistant displays three options:

      - **Use selection**
      - **Improve selection**
      - **Explain selection**

   b. In the Code Editor, select the query and click **Explain selection**.

      APEX Assistant responds and describes the query.



c. To clear the chat window, click **Clear Chat**.

6. General Assistance example: Help with SQL query errors:

   If your query throws an error, APEX Assistant can help troubleshoot the problem.

   For example, suppose you have the following query and have inadvertently left off the "s" at the end the table name.

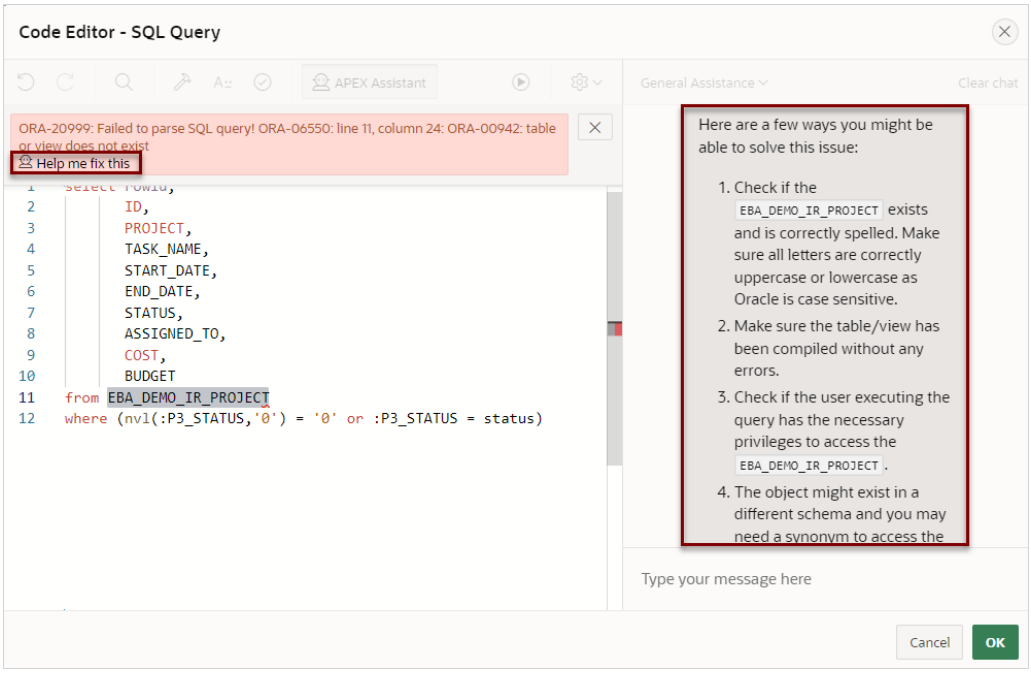
```
select rowid,
       ID,
       PROJECT,
       TASK_NAME,
       START_DATE,
       END_DATE,
       STATUS,
       ASSIGNED_TO,
       COST,
       BUDGET
from EBA_DEMO_IR_PROJECT
where (nvl(:P3_STATUS,'0') = '0' or :P3_STATUS = status)
```

   Since the table name should be `EBA_DEMO_IR_PROJECTS`, if you validate or try to run the query it throws the error:

```
ORA-20999: Failed to parse SQL query! ORA-06550: line 11, column 24:
ORA-00942: table or view does not exist.
```

   a. In the error, click **Help me fix this**.

      APEX Assistant reviews the code and error and suggests the table name may not be spelled correctly.



b. To clear the chat window, click **Clear Chat**.

18.8.4 Including Generative AI in Applications

Include Generative AI in your application by creating a Generative AI Service, configuring application attributes on the AI page, and creating a dynamic action to either display the AI Assistant or generate text with AI.

- About Including Generative AI in Applications
  Learn about including Generative AI (such as an AI Assistant) in your applications.
- Configuring AI Attributes for an Application
  Edit attributes on the AI page to select a Generative AI Service and define the Consent message for a application.
- Managing AI Configurations and RAG Sources
  Create AI Configurations and RAG Sources to enable richer interactions with Generative AI services.
- Creating a Dynamic Action to Display AI Assistant
  Create a dynamic action to display AI Assistant.
- Creating a Dynamic Action to Generate Text with AI
  Create a dynamic action to generate a text response from a Generative AI Service.
- Creating an AI Assistant to Suggest Salary
  Create an AI Assistant to suggest salary recommendations.

18.8.4.1 About Including Generative AI in Applications

Learn about including Generative AI (such as an AI Assistant) in your applications.

You can include AI in your applications by creating a dynamic action which displays the AI Assistant. The actual steps involved can be as simple as hooking a button to the dynamic action.

In general, the steps are as follows:

1. Create a Generative AI Service. See Creating a Generative AI Service Object.
2. Create an application. See Creating Applications.
3. Edit the application definition and configure the attributes on the AI tab. See Configuring AI Attributes for an Application.
4. Create a dynamic action with a True Action:
   - **Show AI Assistant** - Displays the AI Assistant, an AI-powered chat service, displayed either as a dialog or inline. .
   - **Generate Text With AI** - Invokes the configured Generative AI Service to generate a one-time response based on user content. This action is ideal for tasks like summarizing or translating text, extracting keywords, or drafting an email.

About Show AI Assistant True Action Behaviors

The **Show AI Assistant** True Action displays an AI-powered chat service as a dialog or inline. You can edit the Action attributes in Page Designer to define all kinds of behavior. Consider the following examples:

- **Generative AI, System Prompt** - Specify a system prompt to give the AI context and provide instruction on how to process the input. Supports application and page items and system variables.
- **Generative AI, Welcome Message** - Specify a welcome message the AI Assistant will display. Supports substitutions: Application and page items and system variables.
- **Appearance, Display As** - Select if AI Assistant should display as a dialog or inline.
- **Initial Prompt, Type** - Specify the initial prompt (or message) that displays to the user to make it appear it's coming from an actual person. The message can be stored as an item or a value from a JavaScript expression.
- **Use Response, Type** - Select how the AI Assistant should return responses. The term **response** refers to the message content of an individual chat message. You have the option to capture this response directly in a page item value or to process it based on more complex logic using JavaScript Code.
- **Quick Actions, Type** - Enter a message to serve as a chat-wide quick action. A quick action is a pre-defined phrase that, once clicked, will be sent as a user message. Supports template directives, application and page items. and system variables.

> **See Also:**
>
> To view an example, see Creating a Dynamic Action to Display AI Assistant.

About Generate Text with AI True Action Behaviors

The **Generate Text With AI** True Action calls the configured Generative AI Service and generates a one-time response based on user content. This True Action is ideal for tasks like summarizing or translating text, extracting keywords, or drafting an email. You can edit the Action attributes in Page Designer to define all kinds of behavior. Consider the following examples:

- **Generative AI, System Prompt** - Specify a system prompt to give the AI context and provide instruction on how to process the input. Supports application and page items and system variables.

- **Input Value, Type** - Select the type of user input to be delivered to the AI service. The term **input** refers to the request text to be sent to the AI service. You have the option to capture to provide this input directly in a page item value or returning it with more complex logic using a JavaScript expression.
- **Use Response, Type** - Select how the AI service should return the response. You have the option to capture this response directly in a page item value or to process it based on more complex logic using JavaScript Code. If an item is used, you can control if the response should trigger the change event.

> **Note:**
>
> To view an example, see Creating a Dynamic Action to Generate Text with AI.

Utilizing AI Infrastructure Programmatically

You also access the AI infrastructure programmatically using the `APEX_AI` package. Note that the `APEX_AI` APIs require an APEX session, must linked to an APEX app that has a Generative AI Service properly configured. Consider the following SQL Developer example:

```
set serveroutput on;

declare
    l_result clob;
begin
    apex_session.create_session (
        p_app_id   => 102,
        p_page_id  => 1,
        p_username => 'ADMIN' );

    l_result := apex_ai.generate( 'what''s 1+1?' );

    sys.dbms_output.put_line( l_result );

    apex_session.delete_session;
exception
    when others then
        apex_session.delete_session;
        raise;
end;
/
```

> **See Also:**
>
> APEX_AI in *Oracle APEX API Reference*

18.8.4.2 Configuring AI Attributes for an Application

Edit attributes on the AI page to select a Generative AI Service and define the Consent message for a application.

> **Tip:**
>
> Before you can configure attributes on the AI page, you must define a Generative AI Service. See Managing Generative AI Services.

To configure AI attributes for an application:

1. Define a Generative AI Service as described in Managing Generative AI Services.
2. Navigate to the AI page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Edit Application Definition** to the right of the application name.
   d. Click the **AI** tab.

   The AI page appears.
3. Configure the attributes on the AI page:
   a. Generative AI, Service - Select the default AI Service for this application.
   b. Generative AI, Consent Message - Enter a consent message for the use of the selected AI Service.

      The Consent Message you define displays to the user the first time they access the AI Service. The user must agree with the message before they interact with the AI. The user's choice (Accept or Deny) is stored as a preference so they are only prompted once.

      > **Note:**
      >
      > You can also use the `APEX_AI` package to programmatically set or clear user consent. See APEX_AI in *Oracle APEX API Reference*.

      Use the `APEX_AI` package to programmatically set or clear user consent.
4. Click **Apply Changes** to save your changes.

18.8.4.3 Managing AI Configurations and RAG Sources

Create AI Configurations and RAG Sources to enable richer interactions with Generative AI services.

- About AI Configurations and RAG Sources
  Learn about AI Configurations and RAG Sources.

- Creating an AI Configuration
  Create an AI Configuration in Shared Components on the Generative AI Configurations page.
- Editing or Deleting an AI Configuration
  Edit or delete an AI Configuration or RAG Source from the Generative AI Services, Create/Edit page.
- Copying or Subscribing to an AI Configuration
  Copy an AI Configuration from the current application or from another application. When copying an AI Configuration from another application, you can also subscribe to it.
- Viewing AI Configuration Utilization
  View a report that details where AI Configurations are used within the current workspace.
- Viewing AI Configuration History
  View a report of recent modifications made to AI Configurations in this workspace.

18.8.4.3.1 About AI Configurations and RAG Sources

Learn about AI Configurations and RAG Sources.

An AI Configuration is a shared component that centralizes key Generative AI settings, including the System Prompt, Welcome Message, and Retrieval-Augmented Generation (RAG) sources. Creating an AI Configuration enables the reusability of these settings across multiple AI-enabled components such as Show AI Assistant and Generate Text With AI dynamic actions. You can also use AI Configurations with PL/SQL APIs available in the `APEX_AI` package.

About AI Configurations

An AI Configuration contains the following:

- Name - Identifies the AI Configuration in the App Builder UI.
- Static ID - Use this ID to reference the AI Configuration in the APIs in the `APEX_AI` package.
- Service - Refers to the AI Service to be used by the AI Configuration. By default, the Application Default is used.
- System Prompt - Initial instructions or context for the AI Service.
- Welcome Message - The first message shown by the AI Assistant. Note that only components with a UI, such as Show AI Assistant, can utilize this setting.
- Temperature - Sets the level of randomness or predictability of the generated content.
- RAG Sources - Additional data to be sent to the AI Service.

> **See Also:**
>
> APEX_AI in *Oracle APEX API Reference*

About RAG Sources

Each AI Configuration can contain zero or more RAG Sources which help enrich the AI conversation with data specific to the application. A RAG Source can be a SQL Query, a Function Body Returning CLOB, or simply Static text. RAG Sources are dynamic and can individually be protected by a Server-side Condition. If the condition passes, the resulting

content is sent with the System Prompt to the AI Service with each request, in essence acting as additional System Prompts.

A RAG Source contains the following:

- Name - A name of the RAG Source in the App Builder UI.
- Description - A description of the included data to be sent to the AI Service.
- Source Type:
  - SQL Query - When chosen, the result set is included in CSV format. Make sure to only include the relavant columns, and limit the result set to the relevant rows.
  - Function Body Returning CLOB
  - Static text

RAG Sources and Server-side Conditions

Each RAG Source that passes a Server-side Condition is evaluated and included as an additional system message. Note that the RAG Sources are re-evaluated and sent with each call, which depending on the payload might have a large token usage.

To ensure only the relevant data to the conversation is included, you can use one of the following approaches for conditionally including RAG data:

- Using a Security, Authorization Scheme can be helpful for providing certain data sets for certain users only.
- Using a Server-side Condition offers even more flexibility. Besides the typical Condition types, RAG Sources support two Generative AI-specific Condition types: **Any User Prompt Contains** and **Last User Prompt Contains**. You can use these condition types to provide a comma-separated list of case-insensitive keywords. If any of the user prompts, or the last user prompts contains such a keyword, the RAG Source will be evaluated and the data will be included.
- For maximum flexibility, two new bind variables are available both when executing Server-side Condition code, as well as within the SQL Query or Function Body. `APEX$AI_ALL_USER_PROMPTS` and `APEX$AI_LAST_USER_PROMPT` are `CLOB` bind variables that you can use to perform custom RAG logic.

Available Dictionary Views

You can also use the following dictionary views to query an application's AI Configurations:

- `APEX_APPL_AI_CONFIGS`
- `APEX_APPL_AI_CONFIG_RAG_SRCS`

18.8.4.3.2 Creating an AI Configuration

Create an AI Configuration in Shared Components on the Generative AI Configurations page.

> **Tip:**
>
> Before you can create an AI configuration, you must define a Generative AI Service and create an application. See Managing Generative AI Services and Creating Applications.

To create an AI Configuration:

1. Navigate to the AI Configuration page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Application Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Services page appears.
2. Click **Create**.

   The Create / Edit page appears.
3. Configure the AI Configuration attributes:
   a. Identification, Name - Enter the name of the AI Configuration.
   b. Identification, Static ID - Enter a Static ID of the AI Configuration. Use this ID to reference the AI Configuration in the APIs in the `APEX_AI` package.
   c. Generative AI, Service - Select the Generative AI Service to be used by the AI Configuration. By default, the Application Default is used.
   d. Generative AI, System Prompt - Specify the initial instructions or context for the AI service. A well-crafted system prompt guides the AI's behavior and sets the tone for its responses. See item Help for more details.
   e. Generative AI, Welcome Message - Enter optional welcome message shown by the AI Assistant, for example:

      ```
      How may I help you today?
      ```
   f. Advanced, Temperature - Specify the creativity and randomness of the responses. A lower temperature makes the output more focused, deterministic, and conservative, favoring common or predictable answers. A higher temperature increases the models creativity, making it more likely to produce varied and less conventional responses. Adjust the temperature based on the desired balance between precision and creativity in the responses.

      Temperature may vary depending on the AI Provider. Oracle recommends the temperature values of the AI Service be evaluated when choosing the appropriate value.
   g. Advanced, Comments - Enter comments or notes.
   h. Click **Create**.

   Once you create an AI Configuration, you can continue to edit it or create a RAG Source. Retrieval-Augmented Generation (RAG) Sources are used by the Generative AI Service to improve response quality by providing relevant information
4. To create a RAG Source, click **Create RAG Source**.

   The RAG Source page appears.
5. Configure the RAG Source:
   a. Identification, Name - Enter the name of the RAG Source.
   b. Description, Description - Enter a brief description of the RAG Source that will be sent to the AI Service. This description should clearly explain the nature and content of the data being provided, helping the AI to better understand and process the information.
   c. Source, Type - Select the Source Type that defines the RAG Source for this Generative AI configuration. Options include:

- **SQL Query** - Data is sourced from a SQL Query from the local database.
- **Function Body** - Data is sourced from a Function Body that returns a CLOB.
- **Static** - User-defined static text, suitable for entering hard-coded text information that is not sourced from your database.

Depending upon your selection, enter the SQL Query, Function Body, or text in the Code Editor. See item Help adjacent to the Code Editor to learn more and view examples for each option.

d. Advanced, Maximum Tokens - Specify the maximum number of tokens that this RAG source can include. If a row exceeds the remaining token limit, it will be entirely skipped to ensure the response stays within the specified token count.

> **Note:**
>
> The token calculation is only an approximate estimate of the exact calculation used by the AI Service.

e. Server-Side Condition, Type - Select a condition type that must be met in order for this RAG Source to execute. In addition to the typical Condition types, RAG Sources support two Generative AI-specific Condition types: **Any User Prompt Contains** and **Last User Prompt Contains**. You can use these condition types to provide a comma-separated list of case-insensitive keywords. If any of the user prompts, or the last user prompts contains such a keyword, the RAG Source will be evaluated and the data will be included.

   You can use the following special bind variables to access and evaluate the user's prompt:

   - `APEX$AI_LAST_USER_PROMPT` - Retrieves the most recent user-entered prompt.
   - `APEX$AI_ALL_USER_PROMPTS` - Retrieves all user prompts, concatenated into a single string.

f. Configuration, Build Option - Optionally select an authorization scheme which must evaluate to TRUE in order for this RAG source to be executed.

g. Comments, Comments - Enter any comments or notes.

h. Click **Create**.

18.8.4.3.3 Editing or Deleting an AI Configuration

Edit or delete an AI Configuration or RAG Source from the Generative AI Services, Create/Edit page.

To edit or delete an AI Configuration or RAG Source:

1. Navigate to the AI Configuration page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Application Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Configurations page appears.

2. To edit an AI Configuration:
   a. On the Generative AI Configurations page, click the configuration Name.

      The Create / Edit page appears.
   b. Edit the appropriate attributes. To learn more about a specific attribute, see item Help.
   c. Click **Apply Changes**.
3. To delete an AI Configuration:

   > **Tip:**
   >
   > If an AI Configuration is in use, you cannot delete it.

   a. On the Generative AI Configurations page, click the configuration Name.
   b. Click **Delete**.
   c. Confirm your selection and click **Delete** again.
4. To edit a RAG Source:
   a. On the Generative AI Configurations page, click the RAG Source Name.

      The RAG Source page appears.
   b. Edit the appropriate attributes. To learn more about a specific attribute, see item Help.
   c. Click **Apply Changes**.
5. To delete a RAG Source:
   a. On the Generative AI Configurations page, click the RAG Source Name.

      The RAG Source page appears.
   b. Click **Delete**.
   c. Confirm your selection and click **Delete** again.

18.8.4.3.4 Copying or Subscribing to an AI Configuration

Copy an AI Configuration from the current application or from another application. When copying an AI Configuration from another application, you can also subscribe to it.

To copy an AI Configuration:

1. Navigate to the AI Configuration page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Applicatioin Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Services page appears.
2. On the AI Configuration, click **Copy**.
3. Copy From - Choose where to copy the AI Configuration from. Options include:
   - **This Application**

- **Other Application**

4. To copy from the current application:
   a. Copy From - Select **This Application**.
   b. Copy Configuration - Select the configuration to copy.
   c. Name - Enter the name of the configuration you would like to create.
   d. Click **Copy**.
5. To copy from another application:
   a. Copy From - Select **Other Application**.
   b. Application - Select the application from which you want to copy.
   c. Copy Configuration - Select the configuration to copy.
   d. Subscribe - Enable **Subscribe** to subscribe to the configuration.
   e. Click **Copy**.

18.8.4.3.5 Viewing AI Configuration Utilization

View a report that details where AI Configurations are used within the current workspace.

To view the AI Configurations Utilization report:

1. Navigate to the Generative AI Services page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Application Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Configurations page appears.
2. On Generative AI Configurations page, click **Utilization**.

   The Generative AI Configurations Utilization report appears.

18.8.4.3.6 Viewing AI Configuration History

View a report of recent modifications made to AI Configurations in this workspace.

To view the AI Configurations History report:

1. Navigate to the Generative AI Services page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. On the Application Home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Generative AI and click **AI Configurations**.

      The Generative AI Configurations page appears.
2. On the Generative AI Configurations page, click **History**.

   The Generative AI Configurations History report appears.

18.8.4.4 Creating a Dynamic Action to Display AI Assistant

Create a dynamic action to display AI Assistant.

> **Note:**
>
> Before you can add a dynamic action to display AI Assistant, you must complete the steps described in About Including Generative AI in Applications. Failure to create a Generative AI Service or edit the application, AI attributes results in an error.

The following simple example creates a dynamic action hooked to a button that displays the AI Assistant in a dialog. Note this example only describes the limited number of attributes. This example assumes you have already created an application and page.

To add a dynamic action to display AI Assistant:

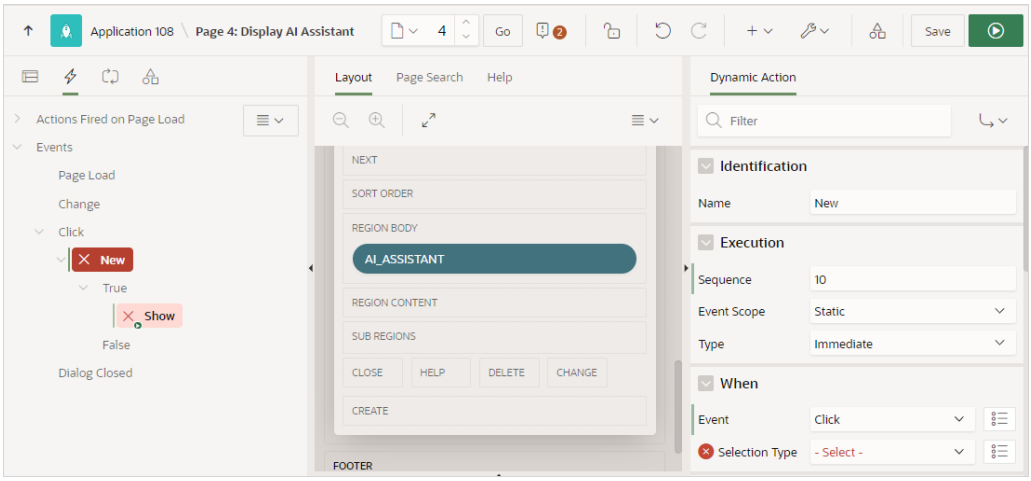
1. Navigate to the application page.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. Create a region to contain the button:
   a. Right-click Body and select **Create Region** from the context menu.

      In Page Designer, edit the Region attributes.
   b. Identification, Name - Enter a region name. The region name is used as the runtime region title if the title attribute is empty. For example: `Display AI Assistant Dialog`
3. Create a button:
   a. Right-click the region you just created and select **Create Button** from the context menu.

      In Page Designer, edit the Button attributes.
   b. Identification, Button Name - Enter the name used to reference the button being clicked. This name is used to reference the button. For example: `AI_ASSISTANT`
   c. Identification, Button Title - If the button is text, enter the text to be displayed on the button. If the button is an image, then this text becomes the ALT text for the image. For example, enter: `AI ASSISTANT`
4. Create a dynamic action that fires when the user clicks a button:
   a. Click the **Dynamic Actions** tab in the left pane.
   b. Under Events, right-click **Click** and select **Create Dynamic Action**.

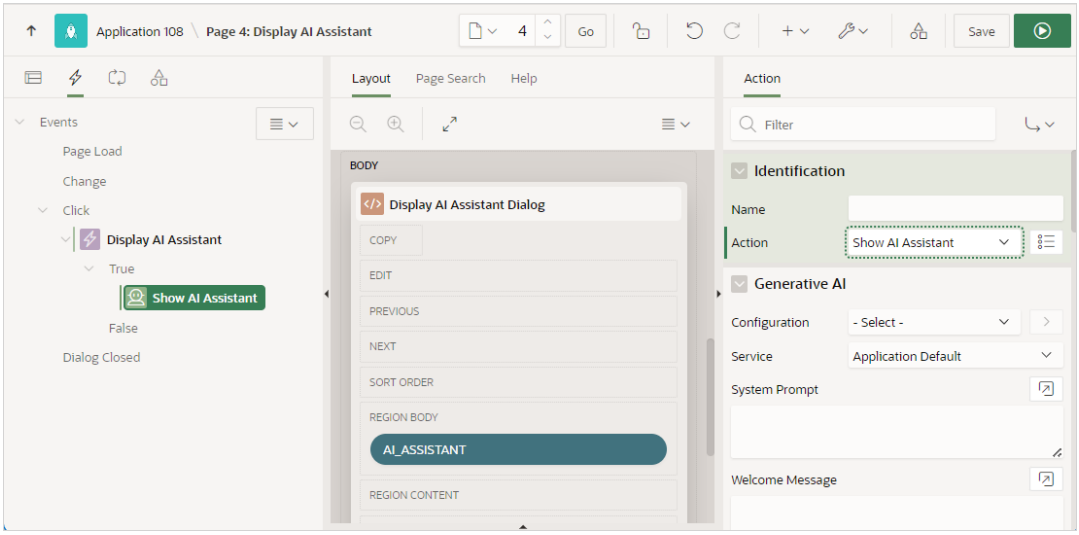
      A dynamic action named **New** appears in the left pane. The Property Editor displays Dynamic Action attributes.



5. In the Property Editor, edit the Dynamic Action attributes:
   a. Identification, Name - Enter the name of the dynamic action. For example, enter: `Display AI Assistant`
   b. When, Event - Specify the event that causes the dynamic action to fire. For this example, accept the default, **Click**
   c. When, Selection Type - Select the type of page element or construct to be used to trigger the event. For this example, select **Button**.
   d. When, Button - Select the type of page element or construct to trigger the event. For this example, select the button you previously created, **AI_ASSISTANT**.
6. Edit the True action, **Show**. In the Dynamic Action tab, click the Show action.

   Action attributes displays in the Property Editor.
7. Identification, Action - Select action you want to perform.

   For this example, select **Show AI Assistant**.



8. Generative AI attributes:

a. Generative AI, Service - Select the Generative AI Service to be used by the AI Assistant. **Application Default** uses the Generative AI Service specified on the AI page in the application attributes. See Configuring AI Attributes for an Application.

b. Generative AI, System Prompt - Specify the system prompt to be used by the AI Assistant. The System Prompt gives the AI a context. This message does not display on front end, but is sent to all AI service requests on the back end.

   You can enter text or use this attribute in conjunction with **Initial Prompt** to utilize substitutions using application items, page items, or system variables.

c. Generative AI, Welcome Message - Specify an optional welcome message shown by the AI Assistant. For example, enter: `How may I help you today?`

   Note this attribute also supports substitutions using application items, page items, or system variables

9. Appearance attributes:

   a. Appearance, Display As - Select how the AI Assistant should appear on your page. Options include: **Dialog** or **Inline**.

      For this example, select **Dialog**.

      Note that the UI changed depending upon your selection.

   b. Appearance, Title - Only displays if **Display As** is **Dialog**. Enter the title that displays on the dialog. For example, enter: `Customer Support`

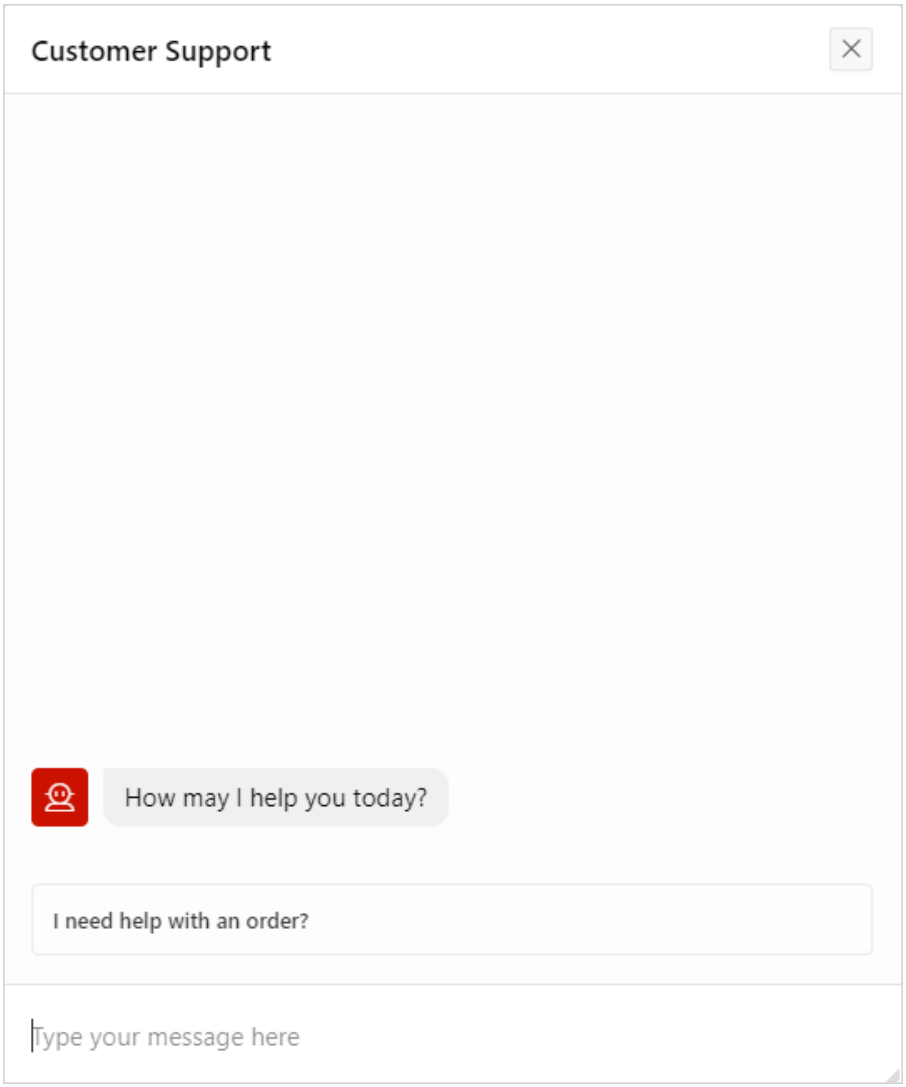
10. Quick Actions, Message - Enter a message to serve as a chat-wide quick action. A quick action is a predefined phrase that, once clicked, will be sent as a user message. For example, enter: `I need help with an order?`

11. View the page. Click **Save and Run Page.**.

    The page appears.

12. Click the **AI Assistant** button.

    The AI Assistant appears.



18.8.4.5 Creating a Dynamic Action to Generate Text with AI

Create a dynamic action to generate a text response from a Generative AI Service.

> **Note:**
>
> Before you can add a dynamic action to generate text with AI, you must complete the steps described in About Including Generative AI in Applications. Failure to create a Generative AI Service or edit the application, AI attributes results in an error.

The following simple example creates a dynamic action hooked to a button that invokes an AI and generates a one-time text response based on user content. Note this example only describes the limited number of attributes. This example assumes you have already created an application and page.

To add a dynamic action to generate one-time text response from an AI Service:

1. Navigate to the application page.

   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

2. Create a region named `AI` to contain the button and items:
   a. Right-click Body and select **Create Region** from the context menu.

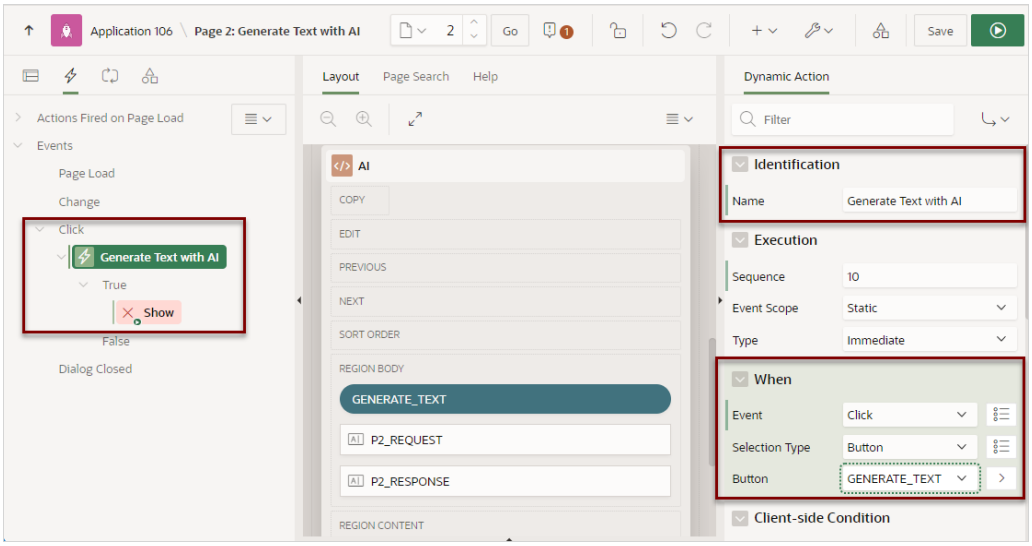
      In Page Designer, edit the Region attributes.
   b. Identification, Name - Enter a region name. The region name is used as the runtime region title if the title attribute is empty. For example, enter: `AI`
3. Create a button:
   a. Right-click the region you just created and select **Create Button** from the context menu.

      In Page Designer, edit the Button attributes.
   b. Identification, Button Name - Enter the name used to reference the button being clicked. This name is used to reference the button. For example, enter: `GENERATE_TEXT`
4. Create the Input Value item:
   a. Right-click the region you just created and select **Create Page Item** from the context menu.

      In Page Designer, edit the Item attributes.
   b. Identification, Item Name - Enter the name used to reference the button being clicked. This name is used to reference the button. For example, enter: `P2_REQUEST`
5. Create the Use Response item:
   a. Right-click the region you just created and select **Create Page Item** from the context menu.

      In Page Designer, edit the Item attributes.
   b. Identification, Item Name - Enter the name used to reference the button being clicked. This name is used to reference the button. For example, enter: `P2_RESPONSE`
6. Create a dynamic action that fires when the user clicks a button:
   a. Click the **Dynamic Actions** tab in the left pane.
   b. Under Events, right-click **Click** and select **Create Dynamic Action**.

      A dynamic action named **New** appears in the left pane. The Property Editor displays Dynamic Action attributes.
7. In the Property Editor, edit the Dynamic Action attributes:
   a. Identification, Name - Enter the name of the dynamic action. For example, enter: `Generate Text with AI`
   b. When, Event - Specify the event that causes the dynamic action to fire. For this example, accept the default, **Click**
   c. When, Selection Type - Select the type of page element or construct to be used to trigger the event. For this example, select **Button**.
   d. When, Button - Select the type of page element or construct to trigger the event. For this example, select the button you previously created, **GENERATE_TEXT**.



8. Edit the True action, **Show**. In the Dynamic Action tab, click the **Show** action.

   Action attributes displays in the Property Editor.

9. Identification, Action - Select action you want to perform. Select **Generate Text with AI**.

10. Generative AI attributes:

    a. Generative AI, Service - Select the Generative AI Service to be used by the AI Assistant. **Application Default** uses the Generative AI Service specified on the AI page in the application attributes. See Configuring AI Attributes for an Application.

    b. Generative AI, System Prompt - Specify the system prompt to be used by the AI Assistant. The System Prompt gives the AI a context.

       For this example, enter:

       ```
       Given a user input with a recipe name, reply by just providing a formatted shopping list
       ```

11. Input Value attributes:

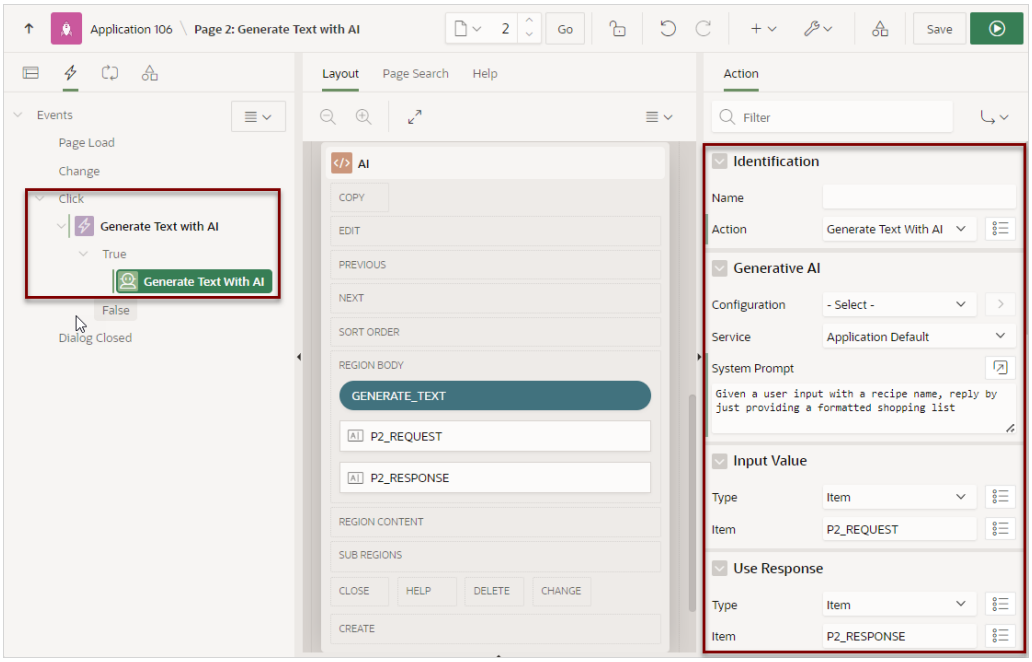
    a. Input Value, Type - Select how the input value to the AI will be provided. For this example, select **Item**.

    b. Input Value, Item - Select **P2_REQUEST**.

12. Use Response attributes:

    a. Use Response, Type - Select where the text generated by the AI should be returned to. For this example, select **Item**.

    b. Use Response, Item - Select **P2_RESPONSE**.



13. View the page. Click **Save and Run Page.**.

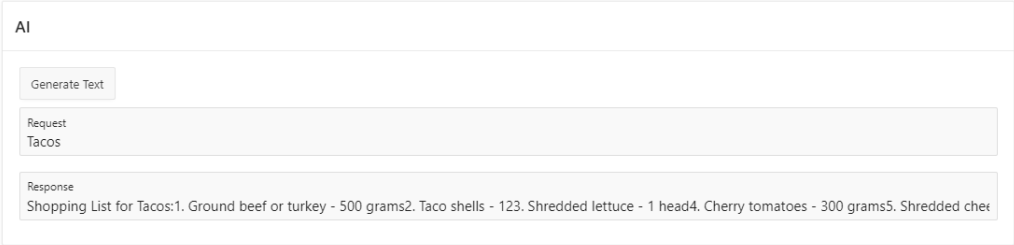
    The AI page appears.

14. Test the page:

    a. Request - Enter `Tacos`

    b. Click the **Generate Text** button.

    A shopping list for tacos appears on the Response field.



18.8.4.6 Creating an AI Assistant to Suggest Salary

Create an AI Assistant to suggest salary recommendations.

The example in this topic describes how to create an AI Assistant that provides salary recommendations for employees. This example assumes you have completed the following tasks:

- Have access the `EMP` table available in EMP / DEPT sample dataset. To learn more, see Using Sample Datasets in *Oracle APEX SQL Workshop Guide*
- Create an interactive report, named `Employees`, and form page, named `Edit Employee`, on the `EMP` table. See Managing Interactive Reports.

To create an AI Assistant to suggest salary recommendations:

1. View the form page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select the application.
   c. Select the form page, `Edit Employee`.

      The form page displays in Page Designer.
2. On the form page, add a button named SUGGEST_SALARY:
   a. Right-click the **Edit Employee** region and select **Create Button**.

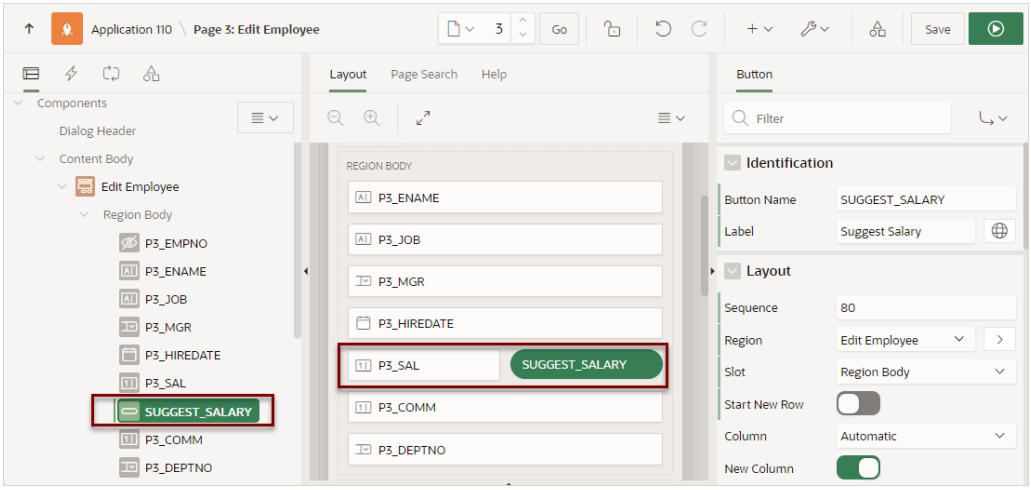
      In the Property Editor, edit the Button attributes.
   b. Identification, Button Name - Enter the name used to reference the button. For example, enter:

      `SUGGEST_SALARY`
   c. Change the button position so it displays to the right of Salary (`P3_SAL`).

      > **Tip:**
      >
      > You can edit the button attributes or drag and drop the button to the appropriate location in the Layout pane.

      - Layout, Sequence - Enter the display sequence for this item. Enter `60`
      - Layout, Start New Row - Disable **Start New Row**.

        Note **New Column** is now enabled.



   d. Appearance, Button Template - Select **Text with Icon**.
   e. Appearance, Hot - Enable **Hot**.
   f. Appearance, Template Options - Click **Use Template Defaults**. In Template Options dialog, configure the following:
      - Common, Size - Select **Large**.
      - Common, Width - Select **Stretch**.

- Advanced, Spacing Top - Select **Large**.
- Click **OK**.

g. Appearance, Icon - Select **fa-robot**.

h. Behavior, Action - Select **Defined by Dynamic Action**.

i. Behavior, Execute Validations - Disable **Execute Validations**.

j. Click **Save**.

3. Add a hidden item named P3_DEPARTMENT:

   a. Right-click the **Edit Employee** region and select **Create Page Item**

      In the Property Editor, edit the Page Item attributes.

   b. Identification, Name - Enter the item name. For example, enter:

      ```
      P3_DEPARTMENT
      ```

   c. Identification, Type - Select **Hidden**.

   d. Click **Save**.

4. Create a computation to retrieve the department name:

   a. Right-click **P3_DEPARTMENT** and select **Create Computation**.

      In the Property Editor, edit the Computation attributes.

   b. Computation, Type - Select **SQL Query (return single value)**.

   c. Computation, SQL Query - Enter the following:

      ```
      select dname
        from dept
       where deptno = :P3_DEPTNO
      ```

   d. Click **Save**.

5. Add a hidden item named P3_INIT_PROMPT:

   a. Right-click the **Edit Employee** region and select **Create Page Item**

      In the Property Editor, edit the Page Item attributes.

   b. Identification, Name - Enter the item name. For example, enter:

      ```
      P3_INIT_PROMPT
      ```

   c. Identification, Type - Select **Hidden**.

6. For P3_INIT_PROMPT, create a computation:

   a. Right-click **P3_INIT_PROMPT** and select **Create Computation**.

      In the Property Editor, edit the Computation attributes.

   b. Computation, Type - Select **Static Value**.

   c. Computation, Static Value - Enter the following:

      ```
      Department: &P3_DEPARTMENT. Current salary: &P3_SAL.
      ```

   d. Click **Save**.

7. Create a dynamic action that fires when the user clicks the SUGGEST_SALARY button:

   a. Click the **Dynamic Actions** tab in the left pane.

b. Right-click **Click** and select **Create Dynamic Action**.

   A dynamic action named **New** appears in the left pane. The Property Editor displays Dynamic Action attributes.

8. In the Property Editor, edit the Dynamic Action attributes:

   a. Identification, Name - Enter the name of the dynamic action. For example, enter:

      ```
      AI: Suggest Salary
      ```

   b. When, Event - Specifies the event that causes the dynamic action to fire. Accept the default, **Click**.

   c. When, Selection Type - Specifies the page element or construct that triggers the event. For example, select **Button**.

   d. When, Button - Specify the button that triggers the dynamic action. Select **SUGGEST_SALARY**.

9. Edit the True action, **Show**. In the Dynamic Action tab, click the **Show** action.

   Edit the Action attributes in the Property Editor.

10. Identification, Action - Select **Show AI Assistant**.

11. Configure Generative AI attributes:

    a. Generative AI, Service - Select the Generative AI Service to be used by the AI Assistant. **Application Default** uses the Generative AI Service specified on the AI page in the application attributes. See Configuring AI Attributes for an Application.

    b. Generative AI, System Prompt - The System Prompt gives the AI a context. This message does not display on front end, but is sent to all AI service requests on the back end.

       You will use this attribute in conjunction with **Initial Prompt**.

       Enter:

       ```
       You are an HR manager who provides suggestions for salary raises of
       employees.
       Based on the department the percentage of the increase differs.

       Here is the list of departments and the increase percentage values:
       - ACCOUNTING 5%
       - OPERATIONS 7%
       - RESEARCH 10%
       - SALES 3%

       You will get the department information and the current salary of an
       employee.
       Based on these information you will calculate a new salary.

       The calculated salary should be displayed without special characters
       and currency symbols. Also round decimal numbers to the next highest
       integer. Answer in complete but short sentences containing the
       calculated salary.

       Only answer questions to salary suggestions and nothing else.
       If needed parameters like the department name or current base salary
       are missing, reply that you need more information to proceed.
       ```

c. Generative AI, Welcome Message - Specify an optional welcome message shown by the AI Assistant. For example, enter:

```
Hi. I'm your personal HR assistant. I can assist you in calculating
salary raises.
```

12. Configure the Appearance attributes:

a. Appearance, Display As - Select how the AI Assistant should appear on your page. Select **Dialog**.

b. Appearance, Title - Enter the title that displays on the dialog.

Enter:

```
Personal HR Assistant
```

13. Initial Prompt attributes - The **Initial Prompt** is the first message/context from a user's perspective:

a. Initial Prompt, Type - Select **Item**.

This attribute determines how the AI Assistant should return responses. Select the type of initial prompt to be delivered to the AI Assistant. The initial prompt represents a message as if it were coming from the user. This can be either a specific item value or a value derived from a JavaScript expression.

b. Initial Prompt, Item - Enter a page item used to set the initial prompt.

Select **P3_INIT_PROMPT**.

c. Initial Prompt, Immediate Action Prompt - Enter a prompt that is used in conjunction with the **Initial Prompt** to display an immediate AI-generated response for the end user.

Enter:

```
Calculate the salary suggestion
```

14. Use Response - Returns the content of an answer (or AI message). For this example, you specify a page item.

a. Use Response, Type - Determines how the AI Assistant should return responses. Select **Item**.

b. Use Response, Item - Select **P3_SAL**.

c. Use Response, Button Label - Specify the label for the message action button, which is used to send the response to a designated target. Enter:

```
Apply suggestion
```

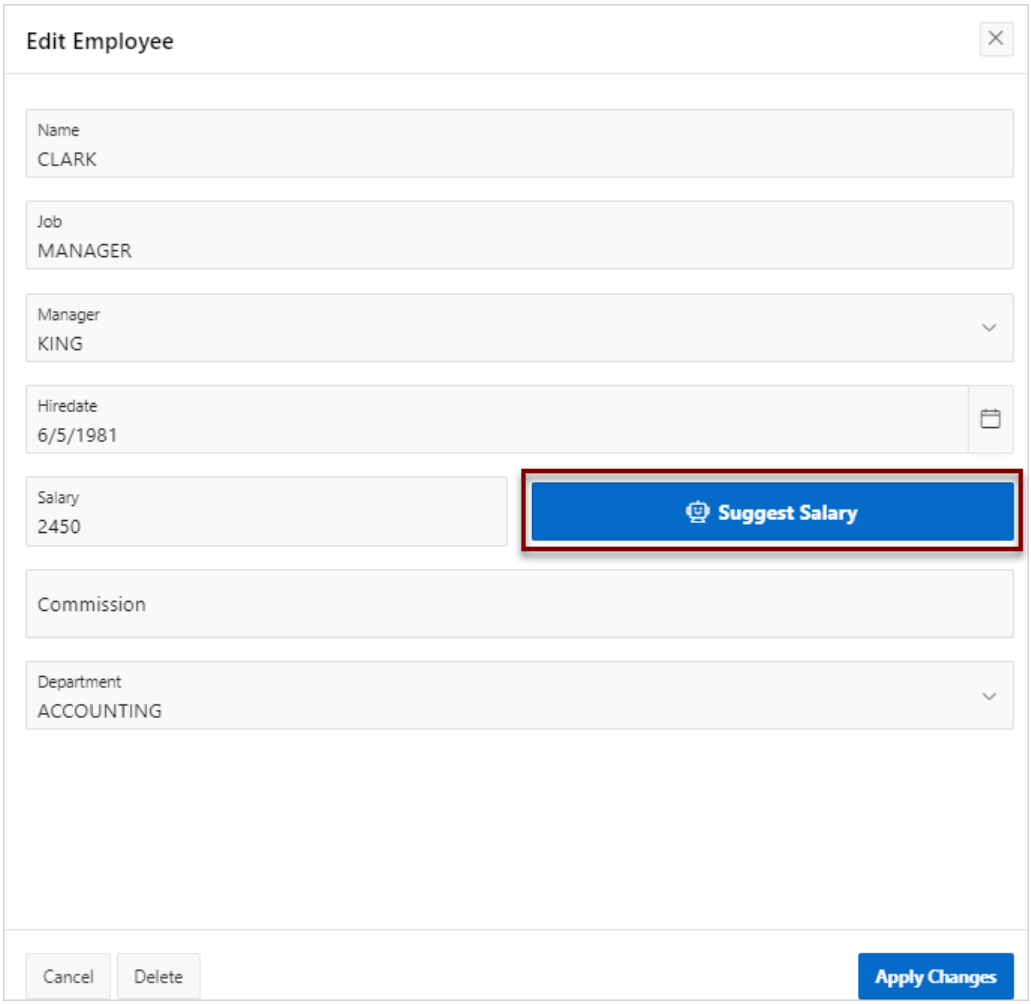
d. Use Response, JavaScript Detection Expression - Enter a JavaScript Expression to search for certain criteria within the message. This expression should return the text to be used in the response. Enter the code:

```
// search for numbers with a length of 3-5 and just return the number
if one is found
this.fullContent.match( /\b\d{3,5}\b/g ) ? this.fullContent.match( /
\b\d{3,5}\b/g )[0] : ""
```

15. Click **Save**.

16. Test your changes:
    a. Return the Employees page and run the page.
    b. Find an employee and click the Edit icon to view the Edit Employee form.

       The Edit Employee form appears.

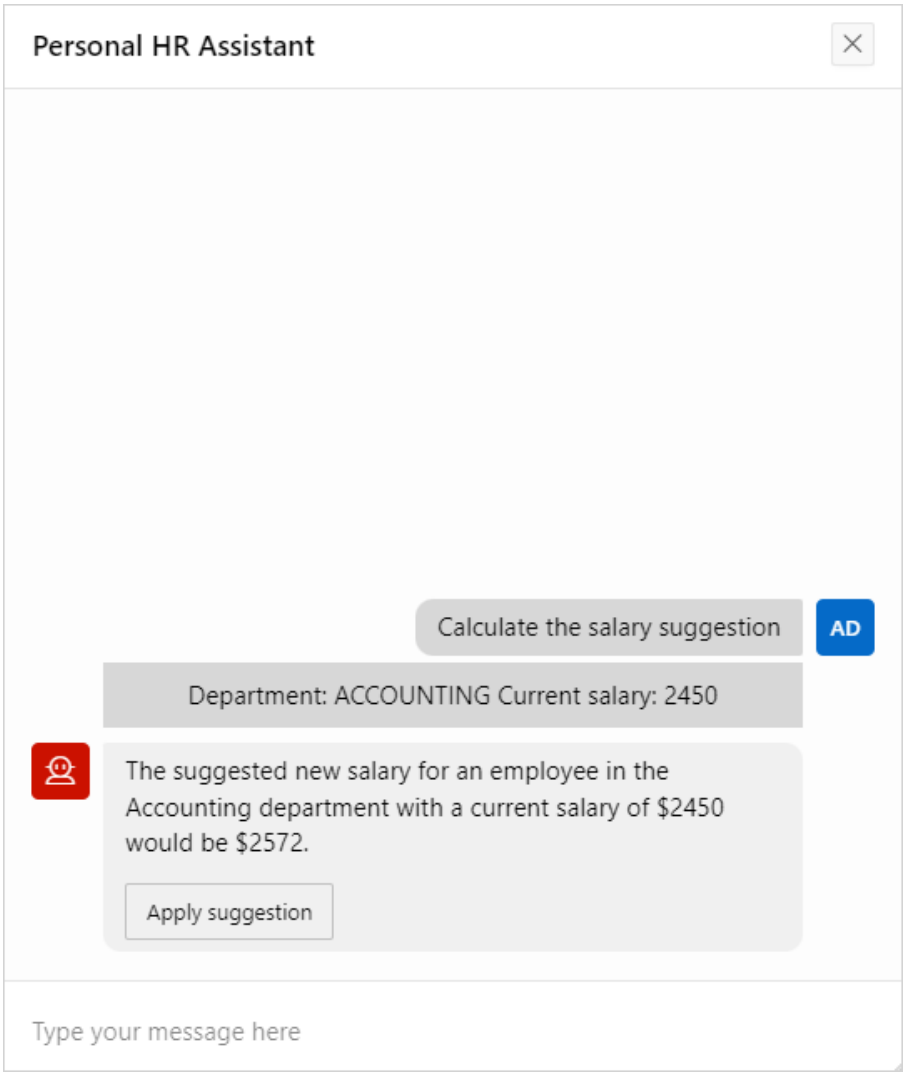


       Note that Clark's Salary is 2450.

       The Edit Employee form appears.

    c. Click **Suggest Salary**.

       The Personal HR Assistant appears.



d. Click **Apply Suggestion**.

   Note that the employee's Salary is no 2573.

e. Click **Apply Changes** to accept the change and exit the form.

18.9 Accessing Data with Database Links

Access data remotely by creating a database link.

- About Database Links
  A database link is a schema object in one database that enables you to access objects on another database.
- Creating a Database Link
  Create a database link in Object Browser.
- Viewing an Existing Database Link
  View existing database links in Object Browser.

See Also:

- Managing Database Objects with Object Browser in *Oracle APEX SQL Workshop Guide*
- Database Links in *Oracle Database Administrator's Guide*

18.9.1 About Database Links

A database link is a schema object in one database that enables you to access objects on another database.

Note:

By default, the CREATE DATABASE LINK system privilege is not granted to a provisioned workspace or database user. To use this feature, a DBA or administrator must grant this specific privilege to the database user in the user's workspace. See Creating Database Links in *Oracle Database Administrator's Guide*.

Because Oracle APEX runs in the Oracle database, you have access to all distributed Oracle database capabilities. Typically, you perform distributed database operations using database links.

Once you create a database link you can access the remote objects by appending `@dblink` to the table or view name where `dblink` is the Database Link Name you specify in the Create Database Object Wizard.

18.9.2 Creating a Database Link

Create a database link in Object Browser.

To create a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

   Object Browser home page appears.
2. Click **Create**.
3. Select **Database Link** and click **Next**.
4. Follow the on-screen instructions.

   Note that Database Link names must conform to Oracle naming conventions and cannot contain spaces, or start with a number or underscore.

18.9.3 Viewing an Existing Database Link

View existing database links in Object Browser.

To view an existing a database link:

1. On the Workspace home page, click **SQL Workshop** and then **Object Browser**.

Object Browser appears.

2. Select the object type **Database Links** at the top of the page.

19
Extending Application Capabilities

Learn about how to extend application capabilities including managing email, running background PL/SQL, utilizing plug-ins, and understanding jQuery support.

- Sending Email from an Application
  To configure and manage email in Oracle APEX, an Instance administrator must sign in to APEX Administration Services.
- Implementing Plug-ins
  Create plug-ins to declaratively extend, share, and reuse the built-in types available in APEX.
- Understanding jQuery Support
  Learn about jQuery support in Oracle APEX.
- Manually Refreshing Oracle APEX Components
  Use the `apex.region().refresh()` method to refresh the content of a region that supports being refreshed.
- Managing Automations
  Automations initiate a sequential set of actions based on a schedule.

> **See Also:**
>
> - Oracle APEX API Reference
> - Deploying an Application

19.1 Sending Email from an Application

To configure and manage email in Oracle APEX, an Instance administrator must sign in to APEX Administration Services.

- About Configuring Email
  To enable APEX to send email, an Instance administrator must configure email settings.
- About Email Provisioning
  Learn about email provisioning.
- About the Mail Queue
  Learn about the mail queue.
- About the APEX_MAIL Package
  Learn about the `APEX_MAIL` package.

19.1.1 About Configuring Email

To enable APEX to send email, an Instance administrator must configure email settings.

- About Enabling Network Services
  Learn about enabling network services.
- About Configuring Email in a Full Development Environment
  Learn about about configuring email in a full development environment.

> **See Also:**
>
> Configuring Email in *Oracle APEX Administration Guide*

19.1.1.1 About Enabling Network Services

Learn about enabling network services.

Enabling network services enables support for sending outbound mail in Oracle APEX, use of Web services in Oracle APEX, and PDF report printing.

By default, the ability to interact with network services is disabled in Oracle Database 11*g* Release 2 or later. Therefore, if you are running Oracle APEX with Oracle Database 11*g* Release 2 or later, you must use the new `DBMS_NETWORK_ACL_ADMIN` package to grant connect privileges to any host for the `APEX_240200` database user. Failing to grant these privileges results in issues with:

- Sending outbound mail in Oracle APEX.

  Users can call methods from the `APEX_MAIL` package, but issues arise when sending outbound email.
- Using Web services in Oracle APEX.
- PDF/report printing.

> **See Also:**
>
> - Enabling Network Services in Oracle Database in *Oracle APEX Installation Guide*
> - Configuring Email in *Oracle APEX Administration Guide*

19.1.1.2 About Configuring Email in a Full Development Environment

Learn about about configuring email in a full development environment.

To configure APEX to send mail in a full development environment your Instance administrator must sign in to APEX Administration Services, navigate to the Instance Settings page, and configure Email attributes.

> **See Also:**
>
> Configuring Email in *Oracle APEX Administration Guide*

19.1.2 About Email Provisioning

Learn about email provisioning.

To configure APEX to automatically email users their login credentials when a new workspace request has been approved. To accomplish this, you must complete the email configuration process and select the provisioning status, Request with Email Verification.

> **See Also:**
>
> - Configuring Email in *Oracle APEX Administration Guide*
> - About Specifying How Workspaces Are Created in *Oracle APEX Administration Guide*
> - Selecting a Provisioning Mode in *Oracle APEX Administration Guide*

19.1.3 About the Mail Queue

Learn about the mail queue.

Instance administrators can manage email sent from applications by monitoring email messages in the mail queue and mail log.

> **See Also:**
>
> Managing the Mail Queue in *Oracle APEX Administration Guide*

19.1.4 About the APEX_MAIL Package

Learn about the `APEX_MAIL` package.

You can send email from an APEX application using the `APEX_MAIL` package.

The `APEX_MAIL` package is built on top of the Oracle supplied `UTL_SMTP` package. Because of this dependence, the `UTL_SMTP` package must be installed and functioning to use APEX_MAIL.

See Also:

- APEX_MAIL in *Oracle APEX API Reference*
- Configuring Email in *Oracle APEX Administration Guide*

19.2 Implementing Plug-ins

Create plug-ins to declaratively extend, share, and reuse the built-in types available in APEX.

- Understanding Plug-ins
  Learn about plug-ins and how to find examples.
- Creating Plug-ins
  Create plug-ins by navigating to Shared Components and running the Create Plug-in Wizard.
- Editing Plug-ins
  Edit a plug-in to specify additional custom attributes, upload files such as image, CSS and JavaScript files, or add events.
- Copying or Subscribing to a Plug-in
  Copy a plug-in from the current application or from another applications. When copying a plug-in from another app, you can also subscribe to it.
- Deleting a Plug-in
  Delete plug-ins on the Edit page.
- Viewing the Plug-in Repository
  The Plug-in Repository provides a central location where developers can share and download plug-ins.
- Resetting the Plug-in Interactive Report
  Reset the plug-in interactive report to clear all current filters applied to the report.
- Viewing the Plug-in Utilization Page
  The Plug-in Utilization page displays which pages, components, and regions use each plug-in.
- Viewing the Plug-in History
  The Plug-in History page shows the actions taken on each plug-in, the developer that performed the action and the date of each action.

19.2.1 Understanding Plug-ins

Learn about plug-ins and how to find examples.

- About Plug-ins
  A plug-in is an extension to the built-in types available APEX.
- About Template Component Type Plug-ins
  Template Components are reusable UI components that work in Page Designer like any plug-in.
- Example Plug-ins
  View plug-in examples on apex.world and the Oracle APEX GitHub repository.

19.2.1.1 About Plug-ins

A plug-in is an extension to the built-in types available APEX.

Create plug-ins for declarative use of new item, region, process and dynamic action types in your application.

APEX supports a set group of authentication scheme, authorization scheme, item, region, dynamic action, and process types. Plug-ins offer a means of augmenting these built-in types by declaratively creating and using new types in your application. Because plug-ins are designed for reuse, developers can export and import them to other workspaces and also share them with the Oracle APEX Plug-in community by using the Plug-in Repository.

The process of implementing a plug-in involves the following steps:

1. Create a plug-in or import a plug-in into your application workspace.
2. Edit or create an authorization scheme, item, region, process, or dynamic action type to use the plug-in.
3. Run your application to test the plug-in.

19.2.1.2 About Template Component Type Plug-ins

Template Components are reusable UI components that work in Page Designer like any plug-in.

About Template Components

Template Components enable developers to create templates with placeholders, including action positions and action templates. APEX exposes the placeholders in the templates as custom plug-in attributes.

Use Template Components to add new region types to APEX which are not based on PL/SQL. Template Components support template directives, Actions and Menu buttons with row-level conditions, and both CSS and JS files.

About Creating Template Components

To create a Template Component, create a new plug-in and specify the Type as **Template Component**. Under **Templates**, enter markup for the template. You specify how the plug-in displays and where it can be used by configuring the **Available as** attribute:

- **Multiple (Report)** - Use the Template Component for a region to show multiple rows with pagination options.
- **Single (Partial)** - Use the Template Component for an interactive report column or for a region to show a single row.
- **Region Only** - When checked, the Template Component can be used as a region showing static values. The provided HTML template is displayed without any wrapping HTML elements around it. The region has a limited number of attributes in Page Designer (excludes Region Template, Header text, Footer text). Oracle recommends including the `APEX$DOM_ID` substitution as an ID attribute of an element in within the template, to ensure that existing APEX functionality works with this component.

Template Components can always be used in server-side Template Directives using `{with/}` and `{apply/}` syntax without checking any of the checkboxes. For example:

```
{with/}
PLACEHOLDER1:=Some value
PLACEHOLDER2:=Some value
{apply INTERNAL_NAME/}
```

> **Tip:**
>
> Template Component Type plug-in include support for template directives. To learn more, see Using Template Directives.

Row Selection

To enable selection support for a given Template Component plug-in, under Standard Attributes, select **Has Row Selection Support** on the plug-in Create or Edit page. You can also use the placeholder `APEX$SELECTOR` inside a row or partial html template. This placeholder will be rendered as a radio input per row (single selection) or a checkbox input per row (multiple selection).

Custom Attributes

To reference the Static IDs, edit the plug-in and add Custom Attributes with a Scope of **Component** and use the `#STATIC_ID#` placeholder syntax. You can quickly synchronize all placeholders as Custom Attributes using clicking the **Synchronize from Templates** button on the plug-in Edit page. These Custom Attributes are used by Page Designer to assign values to placeholders.

Action Positions and Action Templates

Editing a plug-in and adding Actions enables end users to interact with the Template Component. The **Action Template** defines if the template is a simple Button or displays a Menu button. The **Action Position** defines the position of the button within the template. You specify in Page Designer which action to perform on a click of the button.

Slots

> **Tip:**
>
> In earlier releases, **slots** were known as **Display Points** and later renamed to **Positions**.

Slots are placeholders in a template component markup where you can place page components, such as regions, items, and buttons. By configuring which page component is supported, you can use slots to restrict the placement of regions, items and buttons in a template component.

Template component slots:

- Suport nesting of multiple components which is essential for creating more advanced template components using layered UI designs.

- Give developers the ability to limit what type of components are allowed in a template component slot, thereby providing strong template guardrails for consistent appearance.

Impacts in Page Designer when using slots:

- Item Control - If a slot has item control, for example, it only allows Checkbox or Radio item types:
  - An item created in the slot will default to checkbox (first available).
  - In the Rendering tree and Layout tab, developers will be unable to drop unsupported item types.
  - In the Property Editor, unsupported item types ARE hidden under the Unsupported flag. Developers can view full list of all available item types by clicking the **Type** select list and selecting **Show Unsupported**.
  - Developers can move unsupported item types into a slot from the Property Editor (a soft restriction).
- Region Control - If a slot has region control, for example, it only allows Avatar or Static Content region types.
  - A region created in the slot will default to avatar (first available).
  - In the Rendering tree and Layout tab, developers will be unable to move unsupported regions.
  - In the Property Editor, unsupported regions will be hidden under the Unsupported flag. Developers can view full list of all available regions by clicking the **Type** select list and selecting **Show Unsupported**.
  - Developers can move unsupported regions in a slot from the Property Editor (a soft restriction).

Nested Templates

Oracle APEX enables you to embed another Template Component within a current Template Component. Nesting Template Components is a way to break templates into smaller reusable components. To nest a Template Component use the use the `{with/}` and `{apply/}` syntax of template directives.

Viewing Template Component Examples

The Universal Theme includes several Template Component Type plug-ins:

- **Comments** - Based on the classic report Comments template. Displays user comments and status updates declaratively using custom region attributes. Supports avatars.
- **Content Row** - Based on the classic report Content Row template. Displays region content and layout as a content row using custom region attributes. Supports avatars and badges.
- **Media List** - Based on the classic report Media List template. Displays region content and layout as a media list using custom region attributes. Supports avatars and badges.
- **Timeline** - Based on the classic report Timeline template. Displays a series of events by declaratively setting custom region attributes.

To view Template Component Type plug-ins available in the go to **Shared Components**, **Templates**, and then sort by **Type**, **Template Component**.

See Also:

- Viewing Templates on the Templates Page
- Creating a Plug-in from Scratch
- Adding Custom Attributes to a Plug-in
- Using Template Directives
- About Template Component Type Plug-ins

19.2.1.3 Example Plug-ins

View plug-in examples on apex.world and the Oracle APEX GitHub repository.

You can view plug-in implementation examples in the following locations:

- Explore APEX World Plug-ins.
- View Sample Plug-in in the Oracle APEX GitHub repository:
  - Go to https://github.com/oracle/apex.
  - Under **Branches**, select a release, and then **Plug-ins**.

19.2.2 Creating Plug-ins

Create plug-ins by navigating to Shared Components and running the Create Plug-in Wizard.

- Creating a Plug-in from Scratch
  Create a plug-in from scratch by running the Create Plug-in Wizard
- Standard Attributes for Item Type Plug-ins
  Review Standard Attributes for Item type plug-ins.
- Copying a Plug-in
  Create a plug-in by copying an existing plug-in from another application.

19.2.2.1 Creating a Plug-in from Scratch

Create a plug-in from scratch by running the Create Plug-in Wizard

Tip:

To learn more about an attributes described in this section see field-level Help.

To create a plug-in from scratch:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Other Components, click **Plug-ins**.
3. Click **Create**.

   The Create Plug-in wizard appears.
4. Create Plug-in - Select **From Scratch** and click **Next**.
5. Name:
   a. Name - Enter name of the plug-in.
   b. Internal Name - Enter the internal name of the plug-in. This name must be unique within the current application.

      To insure the internal name is a globally unique name worldwide, Oracle recommends that your organization domain name be used as a prefix to internal plug-in names. For example, a domain name of `example.com.com` prefixed to a plug-in named `Slider`, would result in an internal name of `COM.EXAMPLE.SLIDER`.

      The internal name of a Template Component type plug-in as part of a theme will always be prefixed with `THEME$`, for example, `THEME$COM.MYCOMPANY.BADGE`. Use this internal name to apply a Template Component manually where server-side template directives are supported.
   c. Type - Select the type of component that can use this plug-in. Options include:
      - **Authentication Scheme Type** - Used to establish each user's identity before accessing an application.
      - **Authorization Scheme Type** - Checks if a user can access a page, process, or component.
      - **Dynamic Actions** - Adds new actions and/or functionality that can by called dynamically on the client-side.
      - **Item** - Adds new item types to APEX.
      - **Process** - Adds an application and/or page processes.
      - **REST Data Source** - Used for certain REST Data Sources without further configuration or programming.
      - **Region** - Plug-in used to add new region types to APEX.
      - **Template Component** - Plug-ins based on HTML templates which support Template Directives. See Template Directives.

        Enables support for adding new region types to APEX which are not based on PL/SQL. These new region types can be used as regions, as reports, or as columns in interactive reports.

      Depending upon the plug-in type you select, the UI changes. For example, the options that display under Callbacks and Standard Attributes may differ.
   d. Category - Only displays if the selected type is Dynamic Action. Select the category the plug-in is displayed under on the user interface.
6. Source:
   a. PL/SQL Code - Enter a PL/SQL anonymous block of code that contains the procedures for rendering, validating, executing, and performing Ajax callbacks for this plug-in. For performance reasons you can also store this code in a PL/SQL package in the database.
   b. Do not validate PL/SQL code (parse PL/SQL code at runtime only) - Select this option to parse the PL/SQL code at runtime only. Otherwise, the code is parsed when the plug-in is created.

7. Callbacks - Configure that appropriate attributes. The attributes that display depend upon the plug-in type. To learn more about an attribute and view examples, see field-level Help.

> **Tip:**
>
> All Callback function names can reference a function of the anonymous PL/SQL code block, a function within a package or a standalone function in the database.

8. Standard Attributes - Select the attributes that apply to this plug-in. Standard Attributes do not display for some plug-ins.

   To learn more about Standard Attributes that apply only to Item type plug-ins, see Standard Attributes for Item Type Plug-ins .

9. Information:

   a. Version - Enter a string to identify the plug-in version.

   b. About URL - Enter a URL to the plug-in authors home page or to additional information about the plug-in.

10. For Help Text, enter help text used by the user to understand how the plug-in works.

11. For Comments, enter comments and notes that never display when the application is running.

    To learn more about each option, see field-level Help.

12. Click **Create Plug-in**.

    Now that the plug-in is created, you can edit it to specify additional custom attributes, upload files such as image, CSS, and JavaScript files to associate with your plug-in, and add events.

19.2.2.2 Standard Attributes for Item Type Plug-ins

Review Standard Attributes for Item type plug-ins.

Item Type Standard Attributes	Description
Is Visible Widget	The widget is visible.
Standard Form Element	The widget renders as a HTML form element (for example, input, select or textarea). If not checked, `APEX_PLUGIN_UTIL.GET_ELEMENT_ATTRIBUTES` will automatically add `ARIA-LABELLEDBY` to allow the non-standard form element to be labeled appropriately, so as to be accessible to screen readers. **Note**: This relies on the `#LABEL_ID#` substitution being defined in the item's corresponding label template. The **Is Visible Widget** attribute must also be checked
Session State Changeable	The value for an item can be changed in session state. The **Is Visible Widget** attribute must be checked.
Has Read Only Attribute	The user can specify a **Read Only** condition for the item. The **Is Visible Widget** attribute must be checked for this attribute to be enabled

Item Type Standard Attributes	Description
Has Escape Special Characters Attribute	The item has the **Escape special characters** field in the Security section when the item is edited. The **Is Visible Widget** attribute must be checked.
Has Quick Pick Attributes	The item has quick pick attributes. The **Is Visible Widget** attribute must be checked.
Has Source Attributes	The item has source related attributes such as **Source Used**, **Source Type**, and **Format Mask**.
Format Mask Date Only	For items that handle date information, the **Format Mask** field popup restricts the date selections by displaying date values to select from. The **Has Source Attributes** must be checked.
Format Mask Number Only	For items that handle numeric information, the **Format Mask** field popup restricts the numeric selections by displaying numeric values to select from.
Has Element Attributes	The item has element attributes such as **Horizontal/Vertical Alignment**, **HTML Form Element Attributes**, and **Pre/Post Element Text**.
Has Width Attributes	The item width can be controlled.
Has Height Attribute	The item height can be controlled.
Has Element Option Attribute	The item allows specification of additional option attributes when rendering multi-selection elements such as radio groups of check boxes.
Has Placeholder Attribute	The item supports the specification of placeholder text, which gets displayed in the element.
Has Icon Attribute	Enables users to select an icon in Page Designer for an item plug-in. That icon then displays on the item at runtime.
Has Encrypt Session State Attribute	When editing the item, the Store value encrypted in session state displays under the Security region.
Has List of Values	The item has an associated list of values. The **Is Visible Widget** attribute must be checked.
List of Values Required	For items that handle lists of values, a list of values must be defined. The **Has List of Values** attribute must be checked.
Has LOV Display Null Attributes	The item is associated with a list of values that is allowed to include null values. The **Has List of Values** attribute must be checked.
Has Cascading LOV Attributes	The item has cascading LOV related attributes. The **Has List of Values** attribute must be checked.
Join LOV for Column Display	The item can be a part of a Join statement for the LOV source. The Has List of Values attribute must be checked.
Optional Multiple Values	(Optional) Applicable to items where multiple values are at the developer's discretion.
Always Multiple Values	(Always) Designed for items like shuttles or checkboxes, which inherently support multiple values.

Item Type Standard Attributes	Description
Has Filter	Controls if the Page Designer shows the Column Filter - Enabled attribute of an interactive grid column.
Has Link	Controls if the Page Designer shows the Link - Target attribute of an interactive grid column.
Has "'Initialization JavaScript Code" Attribute	Component configuration can be controlled by a provided JavaScript function.
Session State Supports CLOB	Controls if the Page Designer will show Session State Group for the particular item.

19.2.2.3 Copying a Plug-in

Create a plug-in by copying an existing plug-in from another application.

> **Tip:**
>
> To learn more about plug-in attributes, see field-level Help.

To copy a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **Create**.

   The Create Plug-in wizard appears.
4. Create Plug-in - Select **As a Copy of an Existing Plug-in** and click **Next**.
5. Copy From:
   a. Copy From Application - Select an application from which to copy plug-ins.
   b. Theme - Identify the theme from which you wish to copy templates.
   c. Click **Next**.
6. Copy:
   - Identify the Plug-in to be copied and select the copy option. Options include:
     - **Yes** - Copy the Plug-in.
     - **No** - Do not copy this Plug-in.
     - **Copy and Subscribe** - Subscribingto a Plug-in puts a copy of it in your application and sets the plugin master reference.
7. Click **Copy Plug-ins**.

Now that the plug-in is created, you can edit it to specify additional custom attributes, upload files such as image, CSS, and JavaScript files to associate with your plug-in, and add events.

19.2.3 Editing Plug-ins

Edit a plug-in to specify additional custom attributes, upload files such as image, CSS and JavaScript files, or add events.

- Editing a Plug-in
  Edit a plug-in by navigating to Shared Components and selecting it on the Plug-ins page.
- Subscribing to a Plug-in
  Subscribe to a plug-in on the Plug-ins, Edit page.
- Adding Custom Attributes to a Plug-in
  Add Custom Attributes by editing the plug-in.
- Plug-in Attribute Types
  Review the attribute types supported for item plug-ins.
- Creating a File to Associate with a Plug-in
  Create a file to associate with a plug-in.
- Automatically Loading CSS and JavaScript Files
  Automatically load CSS and JavaScript files when a plug-in is used on a page.
- Creating Attribute Groups
  Define plugin attribute groups and assign them to custom plug-in attributes.
- Adding Events to a Plug-in
  Add events to an item, region, or dynamic action type plug-in, to enable them to be exposed to dynamic actions.

19.2.3.1 Editing a Plug-in

Edit a plug-in by navigating to Shared Components and selecting it on the Plug-ins page.

To edit a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to edit.

   The Edit page appears.
4. Make modifications.

   To learn more about each option, see field-level Help.
5. Click **Apply Changes**.

> **See Also:**
>
> Standard Attributes for Item Type Plug-ins

19.2.3.2 Subscribing to a Plug-in

Subscribe to a plug-in on the Plug-ins, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a list of values by copying it and then subscribing. See Copying or Subscribing to a Plug-in.

To subscribe to a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.

   The Plug-ins page appears. Note that the report view includes the following columns: Subscribed From, Subscription Status, and Subscribers.
3. Select a plug-in.

   The Plug-ins, Edit page appears.
4. To subscribe to a plug-in:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master plug-ins to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
5. To refresh a plug-in:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Plug-in - If the plugin is already subscribed, click **Refresh Plug-in**.
   c. Click **Apply Changes**.
6. To unsubscribe from a plug-in:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

19.2.3.3 Adding Custom Attributes to a Plug-in

Add Custom Attributes by editing the plug-in.

> **Tip:**
>
> The maximum number of custom attributes that can be added for an Item-type plug-in is 25.

Custom Attributes specified by the developer might contain items referenced with substitution syntax.

To add custom attributes to the plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to modify.

   The Edit page appears.
4. Custom Attributes - Enable or disable **Substitute Attribute Values**.

   Custom Attribute Values specified by the developer may contain items referenced with substitution syntax, for example `&P1_DNAME`. Options include:
   - **On** - APEX automatically replaces substitution syntax with their actual values.
   - **Off** - Substitution syntax is written unchanged into the `attribute_01` through `attribute_15` record type attributes of `p_plugin`, `p_item`, `p_region`, and so on. The plug-in developer is responsible for replacing those substitution syntax references with a call to `apex_plugin_util.replace_substitutions` or perform similar replacements.

     To learn more and view examples, see field-level Help.
5. To add an attribute click **Add Attribute**.

   The Edit Attribute page appears. Edit the appropriate attributes.
6. Name:
   - Scope - Options incude:
     – **Application** - The attribute can only be defined once for the application.
     – **Component** - The attribute can be defined each time the plug-in is referenced in a component.
     – **Report** - The attribute can only be defined if the region's display attribute is set to Multiple (Report).
     – **Report Group** - The attribute can only be defined if the region's display attribute is set to Multiple (Report).

- Attribute - Sequence which correlates with the `ATTRIBUTE_XX` columns (for example in `APEX_APPLICATION_PAGE_ITEMS`) and to the PL/SQL types defined in the package `APEX_PLUGIN`.
- Display Sequence - Specify the display sequence for this plug-in attribute in the App Builder.
- Static ID - Specify the Static ID for this attribute.

  Template component plug-ins should use uppercase IDs such as `MY_PLACEHOLDER`. The attribute value can then be referenced in the template by using the substitution syntax `#MY_PLACEHOLDER#`.

  Region and Item plug-ins should use lowercase IDs such as `my_static_id`. The attribute value can then be referenced in the code using `p_region.attributes.get_varchar2('my_static_id')` or `p_item.attributes.get_varchar2('my_static_id')`.
- Label - Specify the label which is displayed for that attribute in the App Builder.
- Attribute Group - Specify the group in Page Designer where this attribute displays.

7. Settings:
   - Type - Specify the type the attribute should have when it is displays in the App Builder. The type is also used to verify the entered data when the developer sets or modifies them and is used by the Advisor as well when checking the application.

     To learn more about a specific attribute Type, see Plug-in Attribute Types.
   - Common - Specify if this attribute should always be display in the Property Editor of Page Designer or only if advanced attributes are displayed.
   - Show in Wizard - Specify if this attribute should display in the Create Wizard if a new component is created based on the current plug-in.
   - Required - Specify if this attribute must contain a value.
   - Translatable - Specify whether or not this attribute is included in the XLIFF translation file.
   - Display Width - Specify the length (in characters) that is displayed for this attribute in the App Builder.
   - Maximum Length - Specify the maximum number of character users are allowed to type into this attribute.
   - Set Text Case - Specify if the entered plug-in attribute value should always be stored in upper or lowercase.
   - Unit - Enter the text to be displayed as the unit (for example, pixels) for the attribute's value in Page Designer.
8. Default Value:
   - Default Value - Specify a default value for this plug-in attribute. The default value is used when a new component is created that uses this plug-in. Use `Y and N` for attributes of Type **Yes/No**.
9. Condition:
   - Default Value - Specify a default value for this plug-in attribute. The default value is used when a new component is created that uses this plug-in. Use `Y and N` for attributes of Type **Yes/No**.
10. Help Text and Examples:

- Help Text - Specify help text for this plug-in attribute. The help text is displayed as context sensitive help for the attribute in App Builder.
- Examples -
  Specify examples for this plug-in attribute. The examples are displayed as part of the context sensitive help for the attribute in the App Builder.

11. Submit your changes:
    - Click **Create** to create the attribute and go back to the Edit page
    - Click **Create and Create Another** to create the attribute and then create another one.

> **Note:**
>
> If you click **Create** or **Create and Create Another** and the **Return To Page** checkbox on the right panel under Plug-ins is checked, this same Edit Attribute page displays.

19.2.3.4 Plug-in Attribute Types

Review the attribute types supported for item plug-ins.

The following table describes the attribute types supported for item plug-ins.

Attribute Types	Description
Checkboxes	Predefined list of checkboxes which allows the user to pick from. The values of all checked checkboxes are stored as colon separated string.
Color	For a HEX color code with Color Picker support in Page Designer (for example: `#FF0000` or `#000000`).
HTML	For HTML data with Code Editor support in Page Designer. The stored value can also be translated.
Icon	For Icon CSS classes with Icon Picker support in Page Designer.
Integer	Number without a decimal component (for example: `15` or `-12`).
JavaScript	For JavaScript code with Code Editor support in Page Designer.
Link to Target Page/URL	For a link with Link Builder support in Page Designer.
Number	Number with an optional decimal component (for example: `531.98`, `-363.55`, or `214`).

Attribute Types	Description

PL/SQL Code	An anonymous PL/SQL code block.
	For example:
	`begin`
	`  update emp set sal=sal*1.1;`
	`end;`
	
	For example:
	`case when :P7_CATEGORY='computer'`
	`then true else false end`

PL/SQL Expression Returning VARCHAR2	Simple PL/SQL code which returns a result string. Example:
	For example:
	`case when :P7_CATEGORY='computer'`
	`then 'Y' else 'N' end`

PL/SQL Function Body Returning BOOLEAN	Advanced PL/SQL code which returns TRUE or FALSE.
	For example:
	`declare`
	`  l_price number;`
	`begin`
	`  select product_price`
	`  into l_price`
	`  from demo_products`
	`  where id = :P7_ID;`
	
	`  return case when`
	`calculate_new_price(l_price) > 1000`
	`then true else false end;`
	`end;`

Attribute Types	Description

PL/SQL Function Body Returning VARCHAR2 :	Advanced PL/SQL code which returns a result string.
	For example:
	
	`declare`
	`  l_price number;`
	`begin`
	`  select product_price`
	`  into l_price`
	`  from demo_products`
	`  where id = :P7_ID;`
	
	`  return`
	`calculate_new_price(l_price);`
	`end;`

Page Item	Name of a page item.
Page Items	Comma delimited list of page items.
Page Number	Identifies a page in Oracle APEX.
Page Numbers	Comma delimited list of pages in Oracle APEX.
Region SQL Statement Column	List of columns used in the Region SQL Statement which allows the user to pick from. This attribute type is only available for Region Type plug-ins.
SQL Query	A SQL Statement with the specified number of columns in the statement.
Select List	Predefined list of values which allows the user to pick from.
Text	For alphanumeric data. The stored value can also be translated.
Textarea	For multi-line alphanumeric data. The stored value can also be translated.
XML	For XML data with Code Editor support in Page Designer. The stored value can also be translated.
Yes/No	Predefined select list with Yes/No for boolean decisions.

19.2.3.5 Creating a File to Associate with a Plug-in

Create a file to associate with a plug-in.

To create a file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Other Components, click **Plug-ins**.
3. Select the plug-in.

   The Edit page appears.
4. Locate the **Files** section.

   File Prefix determines the virtual path the Web server uses to point to the files of the plug-in. Do not specify anything to reference files which are stored with your plug-in definition in the database. For performance reasons you can also store your plug-in files on your Web Server. Use `#APEX_FILES#` or any valid URL to reference them.
5. To upload a file, click **Create File**.
6. On the Create page:
   a. Directory - Enter the name of the directory where the file should be stored. For example, `css` or `css/images`.

      If no directory is specified, the file is stored in the root directory.
   b. File Name - If creating a blank file (for example, `script.js`), enter the name of the file.
   c. Content - If uploading a file, do one of the following:
      - Drag and drop the file to the Content region.
      - Click the Content region and select the file.
   d. Click **Create** or **Create and Create Another**.

19.2.3.6 Automatically Loading CSS and JavaScript Files

Automatically load CSS and JavaScript files when a plug-in is used on a page.

You can have APEX automatically load CSS and JavaScript files when a plug-in is used on a page by configuring the File URLs to Load attributes. To specify which of the uploaded files should be loaded and in what order.

To automatically load a CSS or JavaScript file:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Select the plug-in.

   The Edit page appears.
4. Locate **File URLs to Load**:
   a. Cascading Style Sheet - Enter Cascading Style Sheet file URLs to be loaded with this plug-in. Each URL has to be written on a new line.

      If you provide a minified version of your file you can use the substitution string `#MIN#` to include .min, or `#MIN_DIRECTORY#` to include `minified/` in your file URL for a regular page view, and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#PLUGIN_FILES#` to substitute with the value of the plug-in's file prefix.

File URLs you enter here will be emitted within the `#APEX_CSS#` substitution string in the page template.

You do not need to include opening or closing link tags, just the file URL.

b. JavaScript - Enter JavaScript file URLs for code to be loaded with this plug-in.

Enter JavaScript file URLs for code to be loaded with this plug-in. Each URL has to be written on a new line. If you provide a minified version of your file you can use the substitution string `#MIN#` to include .min, or `#MIN_DIRECTORY#` to include minified/ in your file URL for a regular page view, and an empty string if the page is viewed in debug mode. You also have access to the substitution string `#PLUGIN_FILES#` to substitute the value of the plug-in's file prefix.

JavaScript file URLs you enter here will be emitted within the `#GENERATED_JAVASCRIPT#` substitution string in the page template.

You do not need to include opening or closing `script` tags, just the file URL.

To view examples, see field-level Help.

5. Click **Apply Changes**.

19.2.3.7 Creating Attribute Groups

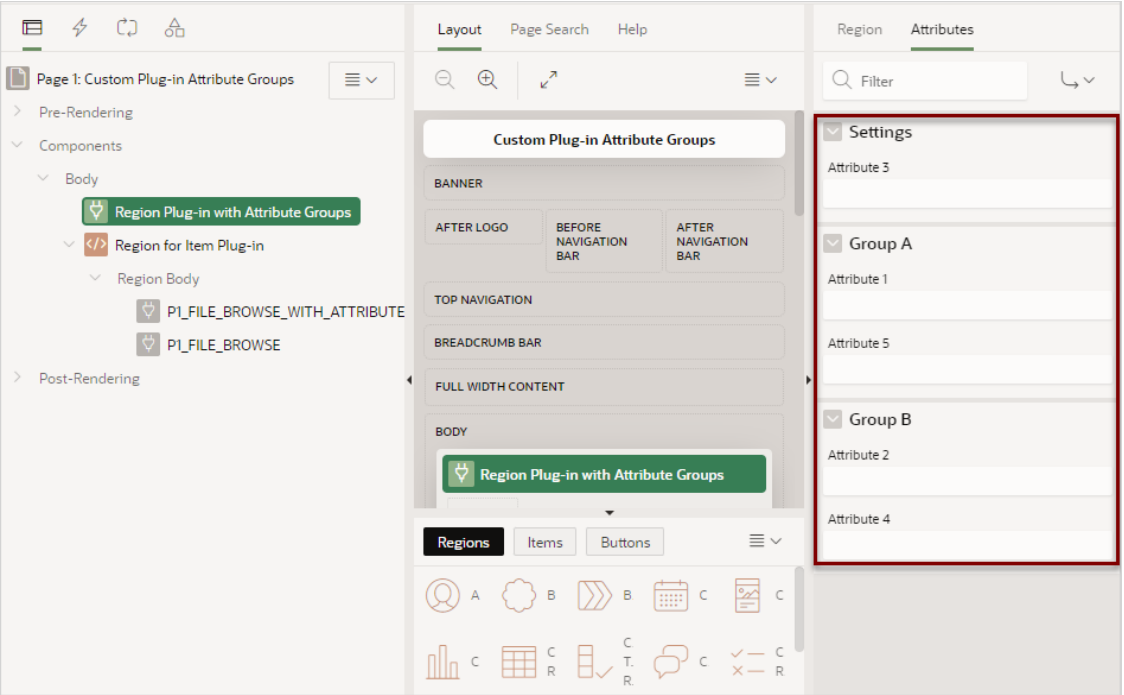
Define plugin attribute groups and assign them to custom plug-in attributes.

You can create attribute groups for the following plug-in types: Item, Region, Process and Dynamic Action. For each attribute group you specify a group Name and Sequence. Then, when you create custom attributes, you select the Attribute Group.

Other features of attribute groups include:

- Group names must be unique for each plug-in
- Attributes are ordered within an attribute group according to the value of their display sequence
- The attribute group Sequence defines the order in which groups are displayed in Page Designer
- Plug-in attributes that are not assigned to an attribute group display directly under Settings in the Attributes tab in Page Designer

The following example shows two attribute groups, Group A and Group B. Note that Attribute 3 is not assigned to an attribute group and displays directly under Settings.



To create an attribute group.

1. Create a plug-in as described in Creating Plug-ins.
2. On the Plug-ins page, click the plug-in you want to edit.

   The Edit page appears.
3. Add attribute groups:
   a. Under Attribute Groups, click **Add Group**.

      A new row displays under Attribute Groups.
   b. For each attribute group, edit the following:
      - Title - Enter a title for the attribute group.
      - Sequence - Enter the sequence.
   c. Click **Apply Changes**.
4. Add custom attributes as described in Adding Custom Attributes to a Plug-in. Under Name, Attribute Group, select a group. In Page Designer, the attribute group Sequence determines where the attribute displays.

19.2.3.8 Adding Events to a Plug-in

Add events to an item, region, or dynamic action type plug-in, to enable them to be exposed to dynamic actions.

For example, suppose you have a Slider plug-in that exposes events such as Start Slide, Sliding, and Stop Slide, and allows the creation of dynamic actions that can react when these events occur.

To add events to a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.

   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to edit.

   The Edit page appears.
4. Locate **Events** and Click **Add Event**.

   A new row displays under Events.
5. Under Events:
   a. Name - The display name under which the plug-in event appears in the dynamic action, for example: `Start Slide`.
   b. Internal Name - The name of the assigned JavaScript event that triggers the dynamic action, for example: `slidestart`.
6. Click **Add Event**.
7. Repeat steps 3 through 4 to add another event.
8. Click **Apply Changes**.

19.2.4 Copying or Subscribing to a Plug-in

Copy a plug-in from the current application or from another applications. When copying a plug-in from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.

   The Plug-ins page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
3. From the Tasks list, click **Copy from another app**.
4. Select plug-in to copy:
   a. Copy From Application - Select an application from which to copy.
   b. Theme - Identify the theme from which you wish to copy Template Component type plug-ins. Selecting a theme will only list those Template Components. Leave **Theme** empty if you wish to copy plug-ins which are not part of a theme.

   c. Click **Next**.

5. Copy:
   a. To Name - Optionally edit the plug-in name.
   b. Copy - Select an action:
      - **Yes** - Copy the plug-in.
      - **No** - Do not copy this plug-in. .
      - **Copy and Subscribe** - Copy and subscribe to this plug-in. Subscribing to a plug-in copies it to application and sets the plug-in master reference.
6. Click **Copy Plug-in**.

19.2.5 Deleting a Plug-in

Delete plug-ins on the Edit page.

You can delete a plug-in if it is not in use. If a plug-in is in use, the Delete button does not display.

To delete a plug-in:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to delete.

   The Edit page appears.
4. Click **Delete**.
5. To confirm, click **OK**.

19.2.6 Viewing the Plug-in Repository

The Plug-in Repository provides a central location where developers can share and download plug-ins.

The Plug-in Repository is located on the Oracle Technology Network.

To view the Plug-in repository:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **View Plug-in Repository**.

The APEX Plug-in Repository displays.

19.2.7 Resetting the Plug-in Interactive Report

Reset the plug-in interactive report to clear all current filters applied to the report.

To reset the interactive report:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. On the Plug-ins page, click **Reset**.

19.2.8 Viewing the Plug-in Utilization Page

The Plug-in Utilization page displays which pages, components, and regions use each plug-in.

To view the Plug-in Utilization page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **Utilization**.

   The Plug-in Utilization page appears.

19.2.9 Viewing the Plug-in History

The Plug-in History page shows the actions taken on each plug-in, the developer that performed the action and the date of each action.

To view the Plug-in History page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Other Components, click **Plug-ins**.
3. Click **History**.

   The Plug-in History page appears.

19.3 Understanding jQuery Support

Learn about jQuery support in Oracle APEX.

- About Upgrading to jQuery 3.6.4
  Learn about upgrading to jQuery 3.6.4.

19.3.1 About Upgrading to jQuery 3.6.4

Learn about upgrading to jQuery 3.6.4.

jQuery 3.x breaks compatibility with earlier 2.x versions. If you application relies on removed jQuery 2.x functionality, you can use the jQuery Migrate plug-in. To include the jQuery Migrate plug-in, go to the User Interface page and set **Include jQuery Migrate** to **On**.

> **Note:**
>
> If your application relies on removed 1.x jQuery APIs, be aware that this functionality no longer works as of Oracle APEX release 18.1. You must update the JavaScript to only use jQuery 3.5+ APIs. See the jQuery migration guides to learn more:
>
> - To migrate from 1.x to 2.x, see http://jquery.com/upgrade-guide/1.9/
> - To migrate from 2.x to 3.5.x, see https://jquery.com/upgrade-guide/3.5/

> **See Also:**
>
> Accessing the User Interface Page

19.4 Manually Refreshing Oracle APEX Components

Use the `apex.region().refresh()` method to refresh the content of a region that supports being refreshed.

Although the Dynamic Action framework provides refresh capability for supported APEX components, there may be cases where you wish to manually invoke a refresh from JavaScript code. To do this, use the `apex.region().refresh()` method as follows:

```
apex.region("myRegionStatidID").refresh();
```

19.5 Managing Automations

Automations initiate a sequential set of actions based on a schedule.

- About Automations
  Automations are a sequential set of actions, triggered by query results that monitor data and then perform the appropriate action.

- About Importing and Exporting Automations
  After importing an application, automations are disabled. Learn how to enable automations after importing.
- Creating an Automation
  Create an automation as a Shared Component on the Automations page.
- Editing an Existing Automation
  Edit an existing automation on the Automation page.
- Managing the Automation Execution Log
  View and clear the Automation Execution Log records on the Automations, Execution Log page.
- Viewing the Automation History Report
  View recent automation modifications on the Automations History report.

19.5.1 About Automations

Automations are a sequential set of actions, triggered by query results that monitor data and then perform the appropriate action.

Create automations to monitor data and then perform the appropriate action. Common automation use cases include approving specific requests and sending email alerts. An automation executes as a query or a PL/SQL function based on a defined schedule. Automations support Data Sources so the query can execute on a local database, a REST Enabled SQL service, or a REST Data Source.

Key features of automations include:

- **Define Automation Actions** - Define multiple actions and have them execute for each automation query row in sequential order. The typical automation action is PL/SQL code. Automation actions can be executed for each row of the query results when the query returns no result, or when the PL/SQL function returns TRUE. Conditions control whether or not an automation executes. Column values from the query result rows are available as bind variables. Alternatively, you can also use Process Type plug-ins as an automation action.
- **Specify Error Handling** - Define automation behavior for cases when an error occurs. Errors can be ignored, the automation can stopped, or set to "disabled" mode.
- **Trigger Automations Programmatically** - The `APEX_AUTOMATION` package provides a PL/SQL interface for triggering an automation programmatically.
- **Review Automation Logs** - View the Automation Execution Log and Log Messages view details about previous automation executions. The Automations History report lists a history of recent automation modifications.

A coordinator job runs automation executions in regular intervals. Each automation actually executes up to a few minutes after the Next Execution Due time. Thus, while the automation edit page shows when the next execution is due, the actual execution will happen a few minutes after the time shown.

> **Note:**
>
> The `CREATE JOB` privilege must be granted to the application's parsing schema in order to execute scheduled automations.

See Also:

- APEX_AUTOMATION package in *Oracle APEX API Reference*
- Configuring Background Jobs in *Oracle APEX Administration Guide*

19.5.2 About Importing and Exporting Automations

After importing an application, automations are disabled. Learn how to enable automations after importing.

Developers often import applications into a workspace multiple times, for versioning or other purposes. If automations were enabled during these imports, this would lead to multiple instances of the automation running simultaneously which is not the desired result. Therefore, Oracle APEX disables automations after you import an application.

You can activate automations directly in Shared Components, Automations, or in Runtime Only environment by using the following PL/SQL block as shown in the following examples:

Example 1: Enable automation and start execution schedule immediately.

```
declare
    l_app_id number := {new application ID};
begin
    apex_session.create_session(
        p_app_id         => l_app_id,
        p_page_id        => {one of the application pages},
        p_username       => '{workspace user}');

    apex_automation.enable(
        p_application_id => l_app_id,
        p_static_id      => '{automation static ID}');
end;
```

Example 2: Enable automation and set next execution timestamp explicitly

```
declare
    l_app_id number := {new application ID};
begin
    apex_session.create_session(
        p_app_id         => l_app_id,
        p_page_id        => {one of the application pages},
        p_username       => '{workspace user}');

    apex_automation.reschedule(
        p_application_id => l_app_id,
        p_static_id => '{automation static ID}',
        p_next_run_at => trunc( sysdate + 1 ) ); -- next midnight
end;
```

> **See Also:**
>
> Editing an Existing Automation

19.5.3 Creating an Automation

Create an automation as a Shared Component on the Automations page.

The following example demonstrates how to create an automation that runs every two hours. This automation increases the salary (`SAL`) of salesmen. For each salesman (`SALESMAN`) with a commission (`COMM`) greater than 500, the salary (`SAL`) increases by 1.5%. For all other salesman, the salary (`SAL`) increases by 1%.

> **Tip:**
>
> The examples in this section are built on the `EMP` table available in the sample dataset, EMP / DEPT. To install a sample dataset, see Using Sample Datasets in *Oracle APEX SQL Workshop Guide*.

To create an automation:

1. Create a new application.
2. Navigate to the Automations page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the new application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Workflows and Automations, select **Automations**.

   The Automations page appears.
3. To create an automation, click **Create**.

   The Create Automations Wizard appears.
4. Specify the automation name and define the execution schedule:
   a. Name - Enter a name for the new Automation. For example, `Increase SAL`.
   b. Type - Select an automation type. Select **Scheduled**.

      Scheduled executes automatically based on the selected frequency. **On Demand** executes when called explicitly using the `APEX_AUTOMATION` package.
   c. Actions initiated on - Select **Query**.

      This option determines whether the execution of Automation Actions should be initiated based on a **Query** (that is, the rows returned can then be used by the actions), or if the actions should **Always** be executed.
   d. Execution Schedule - Select an execution schedule for the new automation. Select **Custom**.

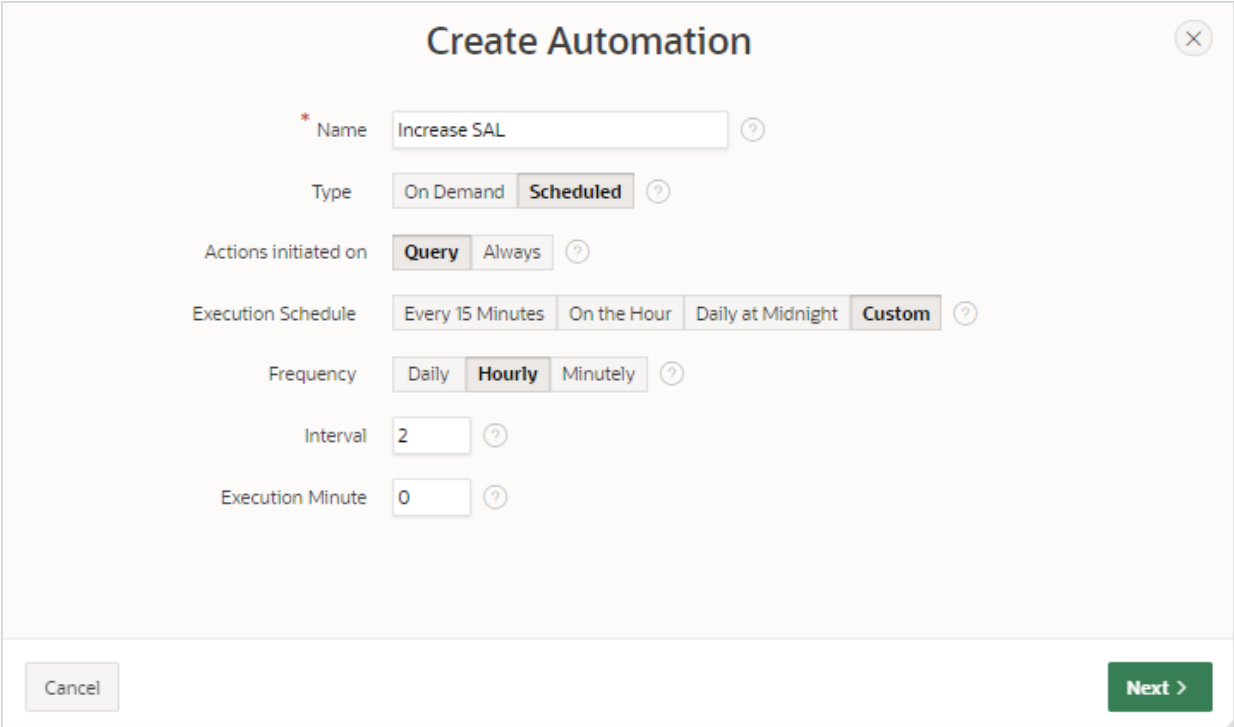
      The **Frequency**, **Interval**, **Execution Minute** attributes only display if Execution Schedule is set to **Custom**.

e. Frequency - Specify whether the execution schedule should be based on a daily, hourly or minutely basis. Select **Hourly**.

f. Interval - Enter `2` so that automation executes every 2 hours.

g. Execution Minute - Specify the execution minute, from 0 to 59. Leave this to the default, `0`.

The Create Automation Wizard should resemble the following illustration.



h. Click **Next**.

5. Define the source:

This example is based on the local table, `EMP`.

a. Data Source - Select the data source for this automation. Select **Local Database**.

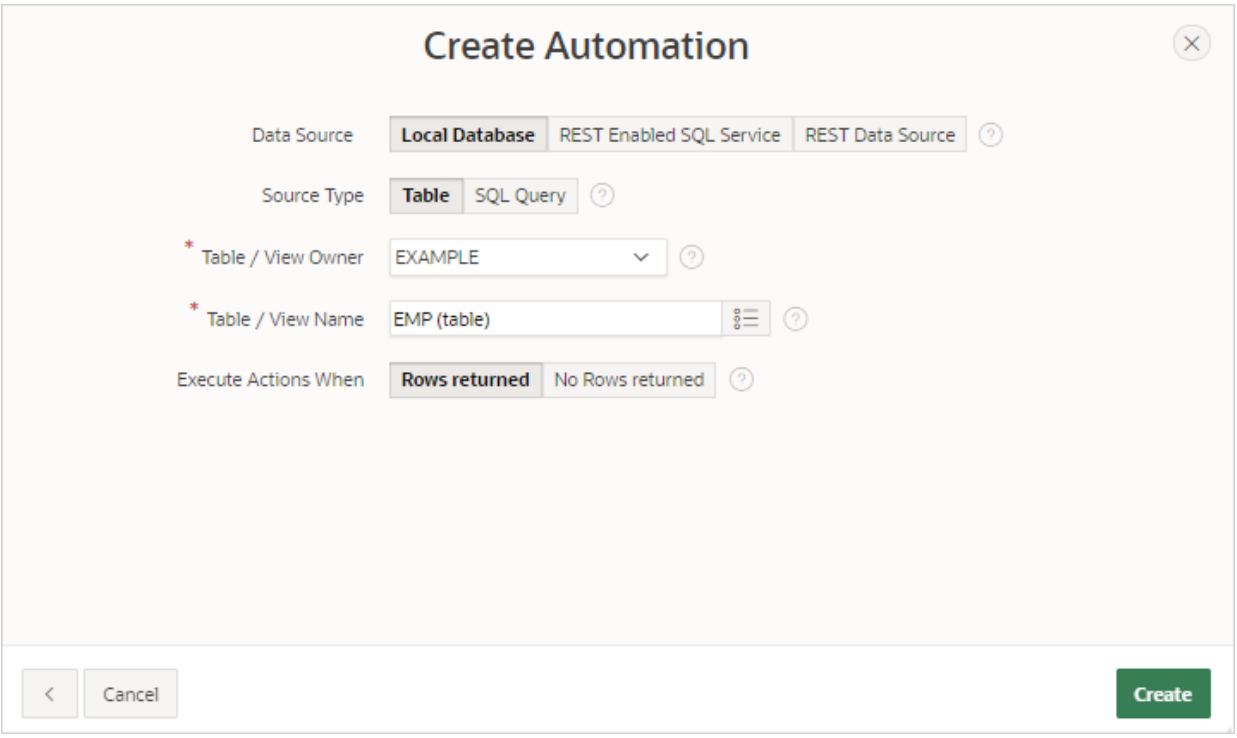
b. Source Type - Select **Table**.

c. Table/View Owner - Select the owner of the table on which you are basing this automation.

d. Table/View Name - Select the table or view on which the automation will be based. For this example, select **EMP (table)**.

e. Execute Actions When - This option determines if Automation Actions should execute when the query returns rows, or when the query does not return rows. Select **Rows returned**.

The Create Automation Wizard should resemble the following illustration.



f. Click **Create**.

The automation edit page appears. The following message displays at the top of the page:

```
Automation has been created and is in "Disabled" state. Now create Actions
to execute when the automation runs.
```

6. Edit the action for the first salary increase:

a. Scroll down to **Actions**.

Note that the wizard created an action, named **New Action**.

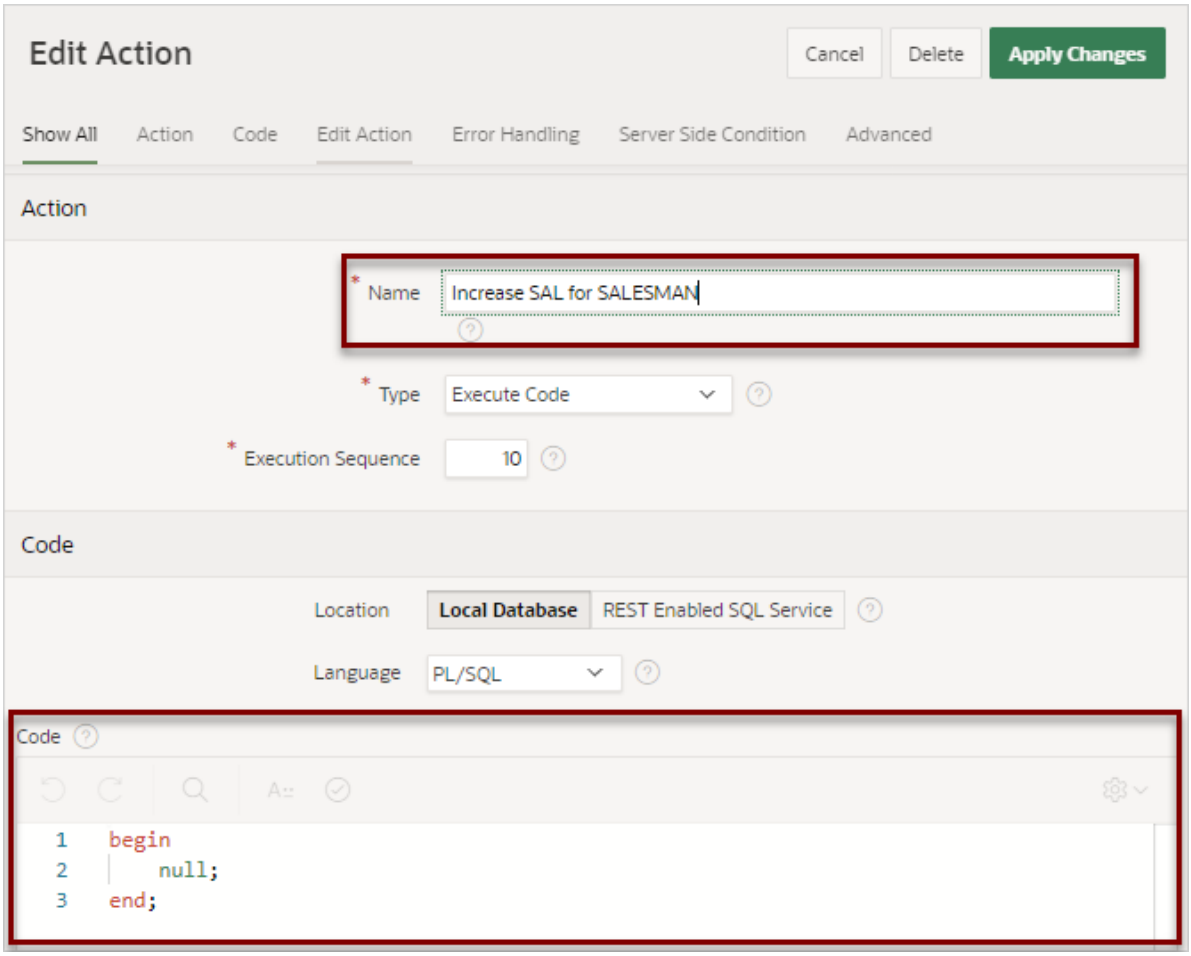
b. Edit the new action, **New Action**:

- Click the **Edit** icon adjacent to **New Action**.

  The Edit Action page appears.

- Name - Enter a meaningful name such as `Increase SAL for SALESMAN`.

  The Edit Action page should resemble the following illustration.



- Code - Add code for the first salary (SAL) increase.

  ```
  begin
  update emp set sal = round(sal * 1.015, 1) where empno = :EMPNO;
  apex_automation.log_info('SAL for ' || :ENAME || ' increased.');
  end;
  ```

  Note that the code will run on each row. In this example, each `SQL UPDATE` statement works on a single row based on the `EMPNO` primary key column. Also note the use of the `APEX_AUTOMATION.LOG_INFO` procedure . The `LOG_INFO` procedure writes a message with the "Information" level to the Execution Log for the automation. Available message levels as "Information" are "Warning" and "Error".

c. Server Side Condition - Create a server side condition for the salary increase.

- Server Side Condition, Condition Type - Select **Expression**.
- Language - Select **PL/SQL**.
- Expression 1- Enter an expression to only execute the action when the salesman (SALESMAN) has a commission (COMM) of more than 500. Enter:

  ```
  :JOB = 'SALESMAN' and :COM > 500
  ```

- Execute for Each Row - Enable this option to specify the condition execute for each row.

d. Click **Apply Changes**.

7. Create another action for the second salary increase:

   a. Scroll down to **Actions** again and click **Add Action**.

      The Edit Action page appears.

   b. Edit the new action:

      - Name - Enter a meaningful name such as `Increase SAL for all others`.
      - Code - Update the code to increase the salary (SAL) for all other salesman.

```
begin
update emp set sal = round(sal * 1.01, 1) where empno = :EMPNO;
apex_automation.log_info('SAL for ' || :ENAME || ' increased by
1%.');
end;
```

   c. Server Side Condition - Create a server side condition for the salary increase.

      - Condition Type - Select **Expression**.
      - Language - Select **PL/SQL**.
      - Expression - Enter an expression to only execute this action when the salesman (SALESMAN) has a commission (COMM) is less than or equal to 500 or null.

```
:JOB != 'SALESMAN' and :COM <= 500 or :COMM is null
```

      - Execute for Each Row - Enable this option to specify the condition execute for each row.

   d. Click **Create**.

8. Run the automation. On the Automation Edit page, click **Save and Run**.

   When you click **Save and Run**. the automations execute in the background. Note that the browser displays again immediately even if the actions are still running.

   The following message appears:

```
Changes applied. Automation Execution initiated.
```

9. Review the Automation Execution Log:

   a. Return to the Automations page by clicking the **Automations** breadcrumb.

   b. Click the **Execution Log** tab and verify the automation executed.

   c. Click the link under **Messages** to view individual messages for processed rows.

> **Tip:**
>
> You can also view the target table in SQL Workshop, Object Browser to validate that the automation is actually changing the data. See Browsing a Table in *Oracle APEX SQL Workshop Guide*.

   d. If viewing messages, click the Execution Log tab to return to the previous page.

10. Clear the Automation Execution Log:

   a. On the Execution Log page, click **Purge Log**.

The Clear Automation Log dialog appears.

b. In the Clear Automation Log dialog:
   - From the Automation list, select the Automation (for example, Increase Sal).
   - Click **Clear Log**.

11. Update the automation Settings to enable the schedule:
    a. Return to the Automations page by clicking **Automations** breadcrumb.
    b. Select the name of the automation to be edited. For example, select **Increase SAL**.
    c. **Settings**, **Schedule Expression** - Click **Schedule Builder** to right of the Schedule Expression field and update the following:
       - Frequency - Select **Weekly**.
       - Interval - Accept the default, `1`.
       - Execution Day - Select **Mon**, **Tue**, **Wed**, **Thu**, and **Fri**.
       - Execution Hour - Enter `14`.
       - Execution Minutes - Enter 30.
       - Click **Set Execution Interval**.

       In the previous example, the automation execution would be due every working day at 2:30pm. The automation scheduler runs in the time zone of the database server.
    d. Schedule Status - Select **Active**.
    e. To save your edits, click **Save Changes**.

> **See Also:**
>
> - Configuring Background Jobs in *Oracle APEX Administration Guide*
> - APEX_AUTOMATION package in *Oracle APEX API Reference*

19.5.4 Editing an Existing Automation

Edit an existing automation on the Automation page.

To edit an automation:

1. Navigate to the Automations page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Workflows and Automations, select **Automations**.
2. On the Automations page, click the automation Name.

   The Automation: *Name* edit page appears. Edit the appropriate attributes.
3. Name:

a. Name - Defines the name of the automation.

b. Static ID - Defines the Static ID for this automation. Use the Static ID when manually executing the automation with the `APEX_AUTOMATION` package (for example, `APEX_AUTOMATION.EXECUTE`).

4. Settings:

   a. Type - **Scheduled** executes automatically based on the selected frequency. **On Demand** executes when called explicitly using the `APEX_AUTOMATION` package.

   b. Schedule Expression - Use **Database Scheduler calendaring syntax** to define repeating execution schedules, such as "every Tuesday and Friday at 4:00 p.m." or "the second Wednesday of every month".

      The calendaring syntax can either be entered manually, or declaratively to build simple expressions by using the **Interval Builder**. Access Interval Builder by clicking the **Schedule Builder** icon to the right of the text area.

      `BACKGROUND_MIN_EXECUTION_CADENCE` controls the minimum amount of minutes which must be between two subsequent automation executions. Note that the automation scheduler works in the time zone of the database server.

   c. Schedule Status - Indicates the state of the Automation Schedule. Options include:

      - **Active** - Schedule is active and the automation will be executed based on the Schedule Expression.
      - **Disabled** - Schedule will not be executed. However, the automation can still be executed by calling the `APEX_AUTOMATION` package.
      - **Error** - Schedule is disabled because an error occurred during the last execution. However, the automation can still be executed by calling the `APEX_AUTOMATION` package.

   d. Actions Initiated On - Specify the data source type for this automation. Options include:

      - **Query** - A SQL Query will be executed. Automation actions can be executed when data is found (for each row), or when no data is found (once).
      - **Function Body Returning Boolean** - A PL/SQL function body is executed. If it returns TRUE, then the automation actions are executed once.
      - **Always** - Automation actions are always executed. This option is suitable to be used with the **Schedule** type to execute the actions based on a schedule regardless of the presence of certain data.

5. Source:

   - Data Source - Defines the Data Source for this automation. Options include:
     - **Local Database** - Data is sourced from the local database.
     - **REST Enabled SQL Service** - Data is sourced from a remote database, where the connection is defined using REST Enabled SQL. To create or maintain REST Enabled SQL references, navigate to Shared Components, REST Enabled SQL.
     - **REST Data Source** - Data is sourced from a RESTful web service defined using REST Data Sources. To create or maintain REST Data Sources, navigate to Shared Components, REST Data Sources.

     The option you select determines the subsequent UI options. To learn more about an attribute, see item Help.

6. Actions:

   a. Add an Action - Click **Add Action**.

The Edit Action page appears. Edit the attributes and click **Create**.

b. Edit an action - Click the **Edit** icon adjacent to the action Name.

The Edit Action page appears. Edit the attributes and click **Apply Changes**.

To learn more about an attribute on the Edit Action page, see item Help.

7. Action Execution:

   a. Execute Actions When - Specify whether the automation actions should execute when the query returns rows or when the query does not return rows.

   b. Primary Key Column - Optionally, specify the primary key column within the query result. The value of this column will be added to the log messages, in order to allow correlating log messages to source rows.

   c. Commit - Specify whether to commit after the actions for each row have been executed or to commit just once after all rows have been processed.

   d. Maximum Rows to Process - If defined, the automation execution stops when that maximum is reached and the automation is logged as `Incomplete`.

   e. Action Error Handling - Select what should happen when the automation encounters and error:

      - **Ignore** - Ignore error and continue processing automation
      - **Terminate Automation** - Terminate automation but leave it enabled
      - **Disable Automation** - Terminate automation and disable it

8. Additional Code Execution:

   a. Executable PL/SQL Code - Enter a PL/SQL anonymous block of code that contains procedures to be called for the initialization or cleanup of the automation, or before action processing for each row. For performance reasons, you can also store this code in a PL/SQL package in the database.

   b. Initialization Procedure Name - Provide the name of an Initialization Procedure to execute at the very beginning of an automation execution. Define the procedure as a PL/SQL object in the database schema, or above in **Executable PL/SQL Code**.

   c. Before Row Processing Procedure Name - Provide the name of a Before Row Processing Procedure to execute for each row, before executing the defined actions. Define the procedure as a PL/SQL object in the database schema, or above in **Executable PL/SQL Code**.

   d. Cleanup Procedure Name - Provide the name of a Cleanup Procedure to execute at the very end of the automation execution. Define the procedure as a PL/SQL object in the database schema, or above in **Executable PL/SQL Code**.

9. Server-Side Condition:

   a. Condition Type - Select a condition type from the list that must be met in order for this automation to be executed.

      The UI changes depending on the Condition Type you select. If you select **Expression**, the Language and Expression 1 attributes display.

   b. Language - Select a language that is used to evaluate the entered programming code.

   c. Expression 1 - Enter an expression to conditionally control the circumstances under which this automation will be executed. Enter values in this attribute based on the specific condition type selected. The automation will execute only if the specified condition is met.

10. Advanced:
    a. Build Option - Select a build option for this automation. Build options are predefined settings that determine whether or not components within an application are enabled.
    b. Comments - Enter any developer comments or notes here. These comments only display within App Builder and never display when running the application.
11. To save your edits, click **Save Changes**.
12. To execute the automation. click **Save and Run**.
13. To delete the automation, click **Delete**.

> **See Also:**
>
> - APEX_AUTOMATION package in *Oracle APEX API Reference*
> - Configuring Background Jobs in *Oracle APEX Administration Guide*
> - DBMS_SCHEDULER Operational Notes in *Oracle Database PL/SQL Packages and Types Reference*

19.5.5 Managing the Automation Execution Log

View and clear the Automation Execution Log records on the Automations, Execution Log page.

Oracle APEX purges log information by default every 14 days. Other retention periods can be configured in Administration Services.

- Viewing the Automation Execution Log
  View the Automation Execution Log for details about previous automation executions, such as start and end timestamps and details about processed rows (both successful and with errors).
- Purging the Automation Execution Log
  Purge the Automation Execution Log records by clicking Purge Log on Execution Log page.

> **See Also:**
>
> Configuring or Deleting the Automation Log in *Oracle APEX Administration Guide*

19.5.5.1 Viewing the Automation Execution Log

View the Automation Execution Log for details about previous automation executions, such as start and end timestamps and details about processed rows (both successful and with errors).

The Automation Execution Log contains start and end timestamps as well as details about processed rows (successful and with errors). Drill down into **Messages** to see individual messages for processed rows.

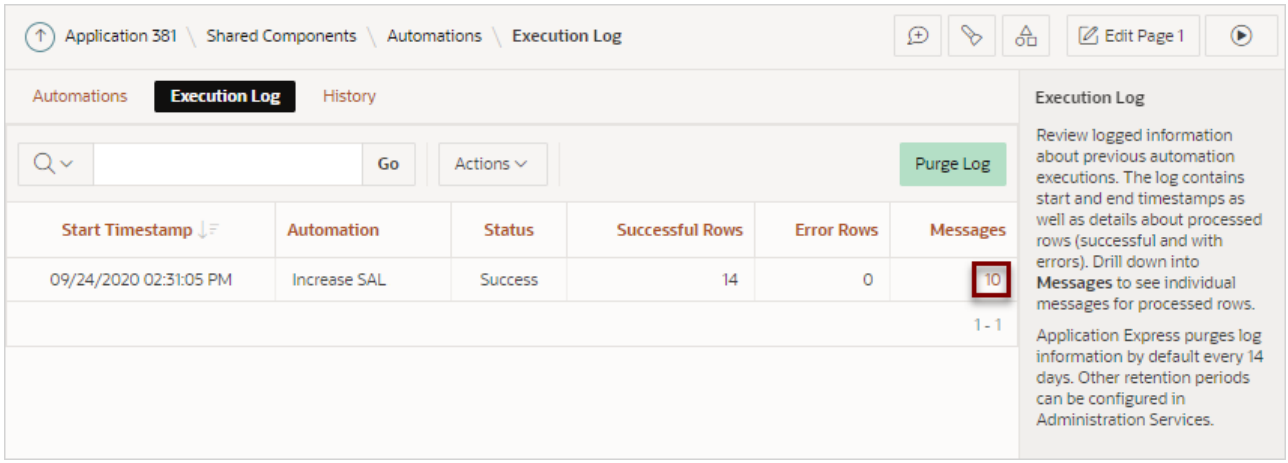
To access the Execution Log:

1. Navigate to the Automations page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the new application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Workflows and Automations, select **Automations**.

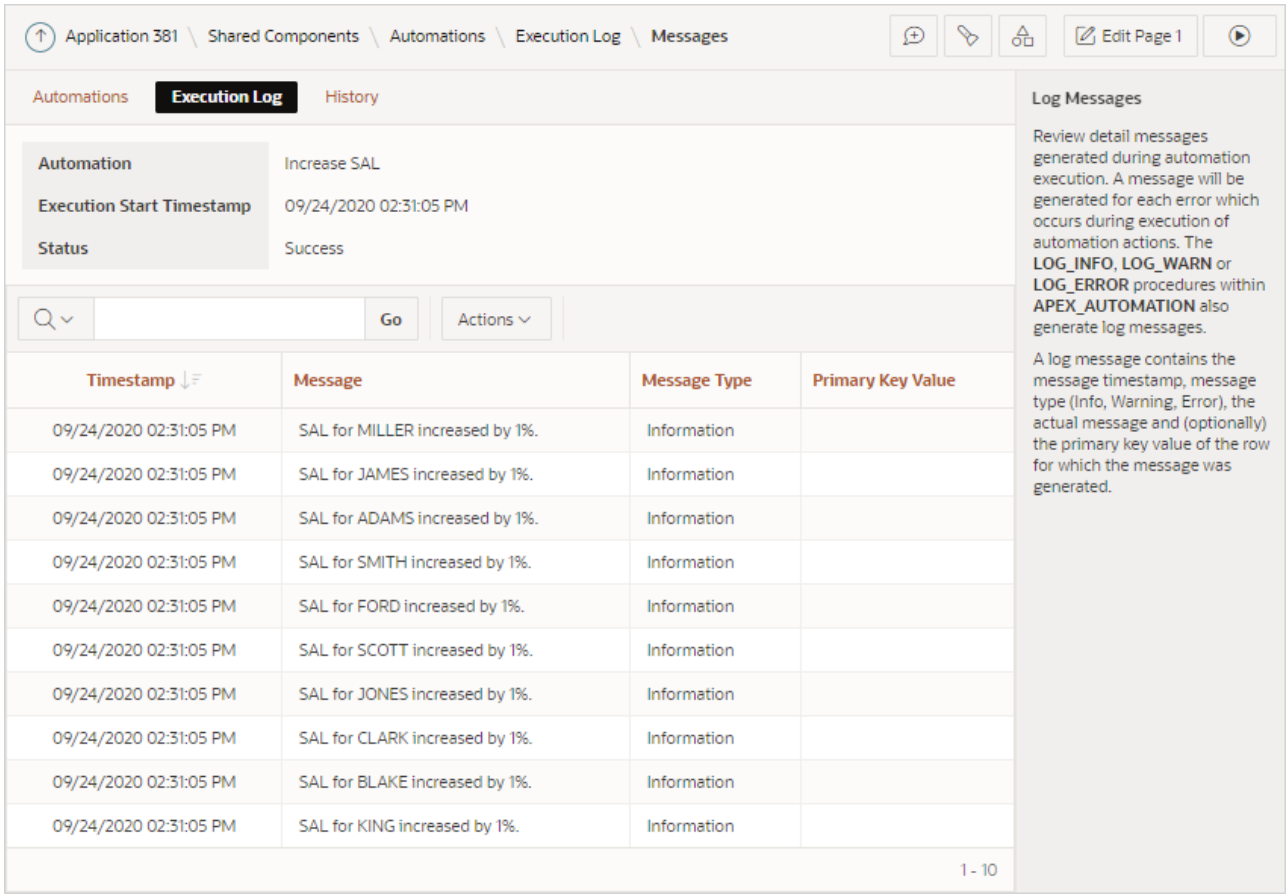
   The Automations page appears.
2. Click the **Execution Log** tab.

   The Execution Log appears displaying the following: Start Timestamp, Automation name, Status, Successful Rows, Error Rows, and Messages.



3. Click the link under Messages to view individual messages for processed rows.

   The Log Messages appear.



Each log message displays the following: Timestamp, the Message text, the Message Type (Information, Warning, or Error), and (optionally) the primary key value of the row for which the message was generated.

> **Tip:**
>
> You can view the target table in SQL Workshop, Object Browser to validate that the automation is actually changing the data. See Browsing a Table in *Oracle APEX SQL Workshop Guide*.

19.5.5.2 Purging the Automation Execution Log

Purge the Automation Execution Log records by clicking Purge Log on Execution Log page.

To purge the Automations Execution Log:

1. Navigate to the Automations page:
   a. On the Workspace home page, click **App Builder**.
   b. Select the new application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Workflows and Automations, select **Automations**.

The Automations page appears.

2. Click the **Execution Log** tab.

   The Execution Log page appears.

3. Click **Purge Log**.

4. In the Clear Automation Log dialog:

   a. From the Automation list, select the log to purge.

   b. Click **Clear Log**

19.5.6 Viewing the Automation History Report

View recent automation modifications on the Automations History report.

To view the Automations History report:

1. Navigate to the Automations page:

   a. On the Workspace home page, click **App Builder**.

   b. Select the new application.

   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

   d. Under Workflows and Automations, select **Automations**.

   The Automations page appears.

2. Click the **History** tab.

   The History report appears displaying the following: Audit Date, Action, Developer, Object Type, and Component Name.

20

Managing Workflows and Tasks

Learn about workflows (including designing workflows and using the workflow diagram builder) and tasks (including approval tasks and action tasks).

- About Workflows
  Oracle APEX Workflow provides basic workflow capabilities for APEX developers.
- Creating a Workflow
  Create a Workflow in Shared Components, Workflows and Automations.
- Copying a Workflow
  Copy a workflow from another application in Shared Components.
- Creating the Workflow Console, Workflow Details, and Workflow Dashboard Pages
  Create the workflow console, workflow details, and workflow dashboard pages with the Create Page Wizard.
- Managing Workflow Versions
  Each workflow can only have one Active workflow version.
- About Using Workflows in Page Processes
  Add the new Workflow Process Type plug-in to a page to expose workflow operations.
- Viewing the Workflow Instance ID in Debug Messages
  Display the workflow instance ID in Debug Messages when viewing workflow debug reports.
- About Workflow Reports
  The Workflows page shows a summary of all workflows in the application. Click on the workflow name to open Workflow Designer.
- Workflow Instance Retention
  Learn about workflow instance retention logic and workflow retention settings.
- Workflow Substitution Strings
  Lists available workflow substitution strings.
- Workflow Views
  Learn about workflow metadata views and workflow runtime views.
- About Workflow Keyboard Shortcuts
  Lists workflow keyboard shortcuts.
- Frequently Asked Questions
  Reference a list of frequently asked questions for workflows.
- About Tasks
  Use tasks to create items for users to approve or action.
- Managing Tasks
  Manage tasks by configuring task definitions, task details, and unified task lists. Use page processes to create approval tasks or action tasks.
- Creating Tasks from Task Definitions
  Create tasks in an Oracle APEX application from existing task definitions.
- About Creating Human Tasks Programmatically
  Learn about creating human tasks programmatically.

- About Creating a Task Definition Process Plug-in
  Learn about creating a task definition process plug-in.
- About Task Retention and Task Purging
  Learn about task retention, task purging, and the Archive of Purged Task Files report.
- Substitution Strings and Bind Variables for Tasks
  Oracle APEX uses substitution strings to pass information about a task to a page.
- Runtime Views for Tasks
  Learn about the runtime views for tasks.
- Troubleshooting Tasks
  Methods to troubleshoot issues with tasks and the approvals component.

20.1 About Workflows

Oracle APEX Workflow provides basic workflow capabilities for APEX developers.

A **workflow** is the automation of a business process where documents, information, or tasks move from one participant to another, following a set of procedural rules.

APEX workflows provide the following capabilities:

- **Workflow Versions** for managing both a development and an active version of a workflow
- **Workflow Definitions** for managing parameters, variables, activities, connections, and participants
- a visual **Workflow Designer** for creating and authoring sequential workflows
- a **Workflow Console** for monitoring and administering workflows
- a **Workflow Runtime Engine** for running workflow instances

You can manage workflows by creating workflow definitions in the Workflow Designer, activate the workflows and monitor them in the Workflow Console, and use page processes to initiate the Workflow Runtime Engine.

You can copy workflows from other applications. When you copy a workflow, the workflow versions, parameters, activities, participants, variables, and any associated shared components are copied from the selected application to the current application. If the copied workflow invokes one or more workflows, the invoked workflows are also copied to the current application if they are not already present.

- About Workflow Versions
  A workflow version can be in three different states: **In Development**, **Active**, or **Inactive**.
- About Workflow Definitions
  Workflow Definitions contain information about workflows including parameters, variables, activities, connections, and participants. Workflow definitions exist on the application level in the Shared Components section.
- About Workflow Designer
  Use Workflow Designer to create or edit workflow definitions.
- About the Workflow Console, Workflow Details, and Workflow Dashboard
  The **Workflow Console** displays all workflows for a given report context. The **Workflow Details** shows details for a specific workflow instance. The **Workflow Dashboard** shows analytics for all workflows in a given report context.

- About the Workflow Runtime Engine
  The **Workflow Runtime Engine** executes workflow instances according to the execution sequence.
- States and Transitions
  Learn about states and transitions for workflows and for workflow activities.
- Example: Workflows
  View the workflows functionality in the **Sample Workflows, Approvals, and Tasks** app.

20.1.1 About Workflow Versions

A workflow version can be in three different states: **In Development**, **Active**, or **Inactive**.

A **Workflow Version** refers to a specific version of a workflow definition. There are three types of workflow versions:

- **In Development** workflows are editable, but you can only run them in the developer session. Only one version of a workflow can be in development at a time.
- **Active** workflows are partially editable. Only one version of a workflow can be active at a time. You cannot move an active workflow back to in development.
- **Inactive** workflows are no longer active, and you cannot use an inactive version of a workflow to start a new workflow instance. Multiple versions of a workflow can be inactive at a time. You can delete an inactive workflow. Deleting an inactive workflow definition also deletes any workflow instances associated with it.

> **Note:**
>
> A workflow instance that is currently running continues to run until complete, even if the workflow definition the instance is based on becomes inactive while the instance is still running.

When you create a workflow version, it is **In Development**. Once you are satisfied with the workflow model, you can move the workflow version to **Active**. If you move a development version to active, APEX moves the currently active workflow to inactive. You can delete or duplicate inactive workflows.

For applications running in a developer session, APEX uses the development version of the workflow to start a new workflow instance. Otherwise, APEX uses the active version of the workflow.

20.1.2 About Workflow Definitions

Workflow Definitions contain information about workflows including parameters, variables, activities, connections, and participants. Workflow definitions exist on the application level in the Shared Components section.

- About Workflow Parameters
  Workflow Parameters pass information into a workflow when a new workflow instance starts.
- About Workflow Variables
  Workflow Variables are specific to a workflow version and can change during the workflow runtime.

- About Workflow Activities
  Workflow Activities specify what happens during each step of a workflow.
- About Workflow Connections
  Workflow Connections link workflow activities and determine the path through the workflow at runtime.
- About Workflow Participants
  Workflow Participants have operational privileges on workflows.

20.1.2.1 About Workflow Parameters

Workflow Parameters pass information into a workflow when a new workflow instance starts.

Workflow Parameters are inputs for the workflow that APEX passes to the workflow when a new workflow instance starts. For example, a workflow that approves an employee's request for a new laptop could include parameters like **Employee ID** and **Laptop Type**.

Workflow parameters support data types and format masks, and can be based on SQL queries or page items. Workflow parameters can have directions: **In**, **Out**, and **In/Out**. **Out** and **In/Out** parameters can only be based on workflow items (variables and parameters). **Out** parameters can only be set when the workflow is invoked from another workflow. **Out** parameters can not be set when the workflow is used in a page process.

When the direction is set to **In**, workflow parameters are read-only. When the direction is set to **In/Out** or **Out**, the workflow parameter can be updated while the workflow instance is running.

20.1.2.2 About Workflow Variables

Workflow Variables are specific to a workflow version and can change during the workflow runtime.

Workflow Variables are inputs for the workflow that are specific to the workflow version. Workflow variables support data types and format masks. They can be based on SQL queries, expressions, function bodies, or workflow parameters. Workflow variables can initiate as a `NULL` value.

The value of a workflow variable can change during the workflow runtime. Workflow variables update due to:

- the result of a completed workflow activity
- the workflow administrator explicitly calling the `apex_workflow.update_variables()` API
- the workflow administrator updating the variable from the workflow details page

> **Note:**
>
> If the workflow variable is a `CLOB` datatype and the value exceeds the `VARCHAR2` limit, use the `apex_workflow.update_variables()` API instead of the details page.

20.1.2.3 About Workflow Activities

Workflow Activities specify what happens during each step of a workflow.

A **Workflow Activity** is a logical step in a workflow. Each workflow activity requires human or machine resources to support workflow execution. For example, a workflow activity could

programmatically evaluate data from a table in order to conditionally branch an activity, or an activity could request input from a workflow participant.

In APEX, each workflow activity corresponds to a process type plugin. All workflows have:

- exactly one Start Activity at the beginning
- at least one End Activity at the end
- support for existing process type plug-ins, including Execute Code, Send E-Mail, Human Task - Create, Send Push Notification, Invoke API, and Server Side Geocoding (supported only on Oracle Autonomous Database)
- support for workflow-specific process type plug-ins, including Workflow Start, Workflow End, Wait, Workflow Switch, and Invoke Workflow

Each workflow activity, regardless of type, has the following common properties:

- Due On - Specifies activity timeouts. Wait activities have their own timeout specifications
- Additional Workflow Data - Specifies a SQL Query. The workflow can use the columns from the query as bind variables or substitution strings in the activity
- Sequence - The sequence in which the workflow appears in the Designer Tree. The sequence and the placement of the activity in the workflow are not the same, and changing the sequence of an activity does not move the activity earlier in the workflow
- Label - A display name that appears in the workflow diagram

In addition to the activity properties, a workflow activity may also have one or more **Activity Variables**. Activity variables also have their own properties, and can capture information about the workflow activity that can be used later on in the workflow.

You can also configure activities for error handling. Error handling is not managed on the activity itself—instead, configure **Error Connections** from the activity that dictate how specific errors are resolved. For more on configuring error connections, see Adding a Workflow Connection.

20.1.2.4 About Workflow Connections

Workflow Connections link workflow activities and determine the path through the workflow at runtime.

A **Workflow Connection** is a link between two workflow activities. Workflow connections determine the path through a particular workflow at runtime.

There are two types of connections: **branches** and **transitions**.

Branches

Branches are connections from switch activities. Switch activities split the workflow path from one activity to multiple other activities depending on the result of the condition specified in the switch activity. Each of these splits is called a branch. For example, a **True False Check** switch has one branch if the condition specified in the activity evaluates as true, and one branch if the condition specified in the activity evaluates as false.

Transitions

Transitions are connections that go directly from one activity to another. For example, a **Start** activity can transition directly to an **Execute Code** activity. There are three types of transitions that connect activities in a workflow.

Normal
The default connection between two workflow activities. Every workflow activity must have at least one incoming and one outgoing normal connection with the exception of start activities, which cannot have an incoming connection, and end activities, which cannot have an outgoing connection.

Timeout
Every activity can have a timeout connection specified if there is already a timeout value in the deadline section of the activity properties. The only exception is the wait activity, since it already has its own timeout specification and cannot have another timeout connection stemming from it.

Error
Every activity can have one or more error connections specified. Each error connection specifies the SQL errorcode. When an error occurs during runtime, the workflow executes the target activity of the error connection instead of going into a faulted state.

20.1.2.5 About Workflow Participants

Workflow Participants have operational privileges on workflows.

Workflow Participants are APEX users with operational privileges over a workflow. Workflow participants are associated with a specific workflow version. They are not represented in the workflow diagram, but they are visible in the **Workflow Designer Tree**.

You can start an **In Development** workflow without adding participants. However, in order to start an **Active** workflow, you must specify participants.

You can specify **Workflow Owners** and **Workflow Administrators** in the tree.

Workflow Owner
Can start and terminate an instance of the workflow. Can retry a faulted workflow.

Workflow Administrator
Can start, terminate, suspend, resume, and retry a workflow instance. Can update variable values of a workflow instance.

An APEX user does not need to be a workflow participant to start a workflow. The user who starts a workflow is known as the **Workflow Initiator**. Initiators can terminate workflow instances, but only for workflows they initiated.

20.1.3 About Workflow Designer

Use Workflow Designer to create or edit workflow definitions.

Workflow Designer is a graphical user interface that models Oracle APEX workflows. From Workflow Designer, you can:

- create a new workflow
- edit an existing workflow
- view all workflows for your application in tree form
- expand existing workflows to show additional information, including the workflow parameters, activities, transitions, variables, and participants
- use the **Workflow Diagram Builder** to model workflows
- switch from one workflow definition to another

- view an invoked workflow from the Invoke Workflow activity of the parent workflow

> **Tip:**
>
> For information on navigating Workflow Designer with a keyboard, see About Workflow Keyboard Shortcuts.

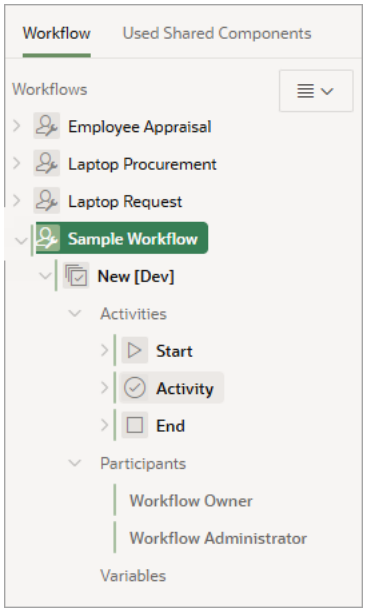
- The Workflow Designer Tree
  The Workflow Designer Tree shows all workflow definitions for your application in tree form.
- The Workflow Diagram Toolbar
  The diagram toolbar contains tools for interacting with the Diagram Builder.
- The Activities Palette
  The Activities Palette is a panel that displays all available activity types, including any custom process plugins you have configured.
- The Workflow Diagram Builder
  The **Workflow Diagram Builder** is the central pane of the Workflow Designer.

20.1.3.1 The Workflow Designer Tree

The Workflow Designer Tree shows all workflow definitions for your application in tree form.

The **Workflow Designer Tree** is the left pane in the Workflow Designer. When you expand a workflow definition, APEX shows the parameters for that workflow definition, and each version of the workflow. Expanding a workflow shows the following:

- workflow activities, including activity variables
- outgoing and incoming connections for each activity
- workflow participants
- workflow variables



The **Rendering Menu** in the designer tree provides context-sensitive menu options. For example, if you right-click on a specific activity, the rendering menu may display:

- Create Activity Above
- Create Activity Below
- Create Activity Variable
- Create Connection
- Duplicate
- Expand All Below
- Collapse All Below
- Delete

If you select a workflow version, the rendering menu may display:

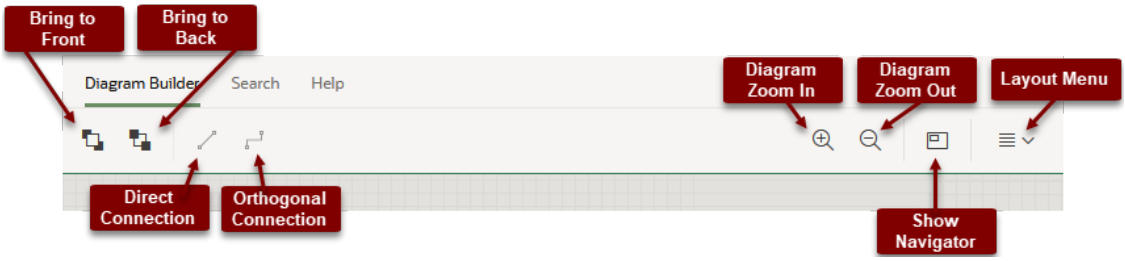
- Deactivate
- Create Activity
- Create Variable
- Create Participant
- Expand All Below
- Collapse All Below
- Delete

> **Caution:**
>
> The workflow designer tree displays all items according to the sequence value specified in the properties. This is *not* the same as the workflow execution order, which is determined by the workflow connections.

20.1.3.2 The Workflow Diagram Toolbar

The diagram toolbar contains tools for interacting with the Diagram Builder.

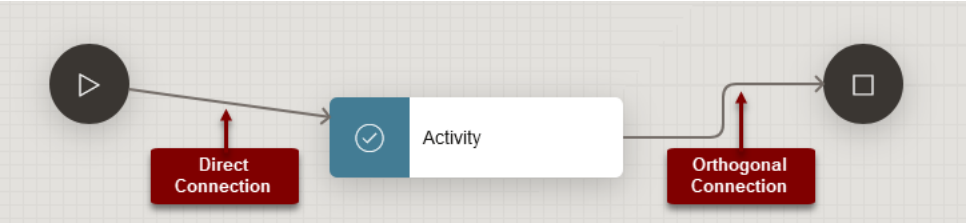
The diagram toolbar at the top of the Diagram Builder contains context-sensitive tools for changing how the workflow displays on the **Drawing Area** in the center of the page.



Use **Bring to Front** and **Bring to Back** to move activities on the drawing area forward or backward. This can help in places where activities on the drawing area overlap.

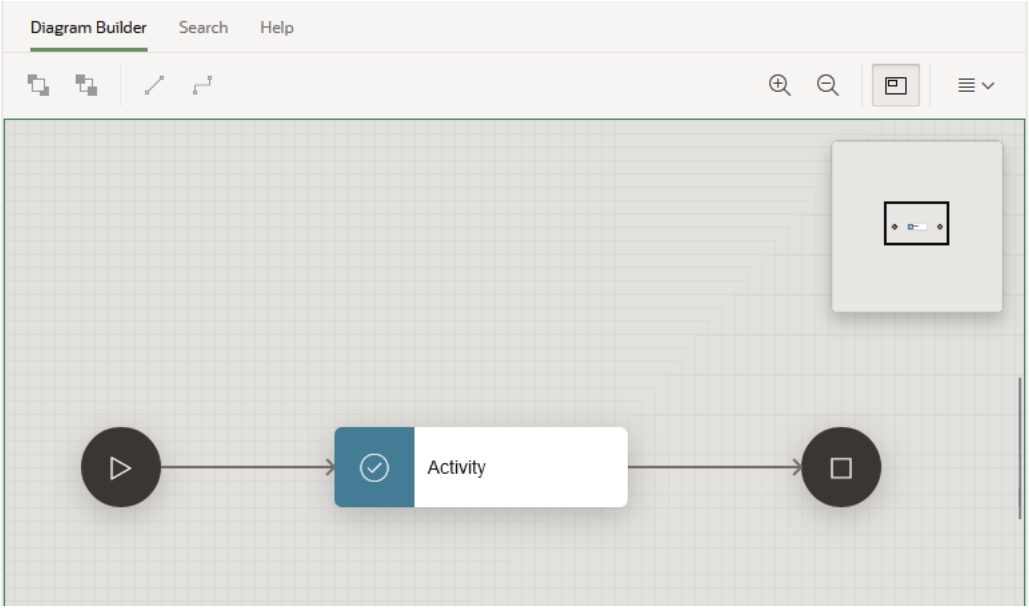
Use **Direct Connection** and **Orthogonal Connection** to change the appearance of specific connections. Direct connections are straight lines between two activities. Orthogonal connection use a combination of vertical lines, horizontal lines, and ninety-degree angles to connect two activities.

> **Tip:**
>
> You can use the **Default Router Setting** option on the **Layout Menu** to set the default connection style for the workflow. Changing the default router setting does not update any connectors you have already individually configured.

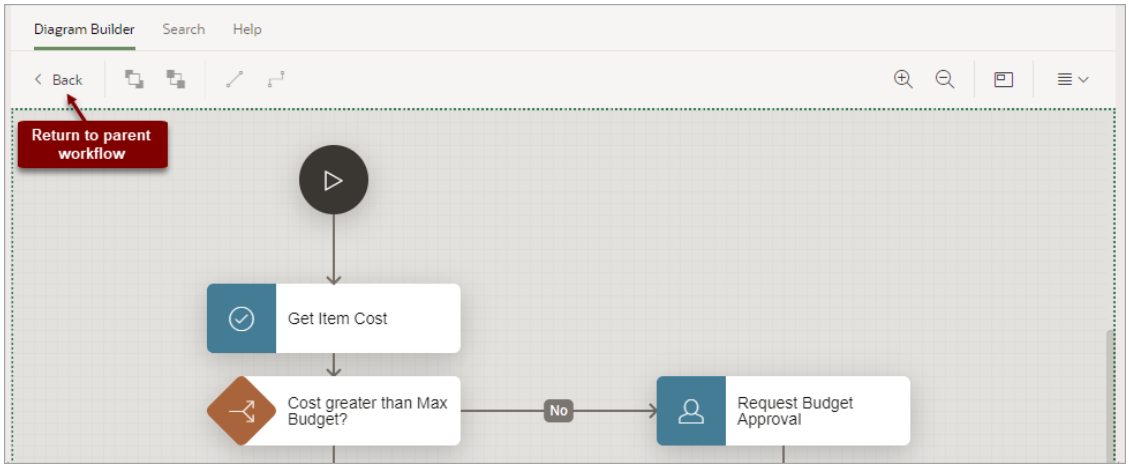


Use **Diagram Zoom In** and **Diagram Zoom Out** to zoom the drawing area in and out.

Use **Show Navigator View** to toggle **Navigator View**, which is an overview that reflects the current state of the drawing area. To pan the drawing area in navigator view, click and drag the visible view rectangle.



If you are viewing a workflow that is invoked from another workflow, a **Back** option appears in the toolbar to easily navigate from the invoked workflow back to the parent workflow.

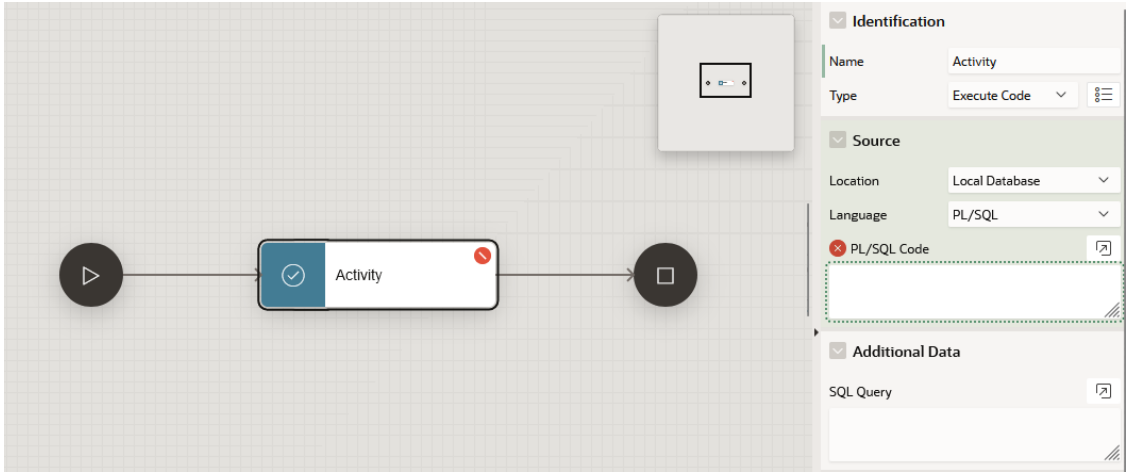


20.1.3.3 The Activities Palette

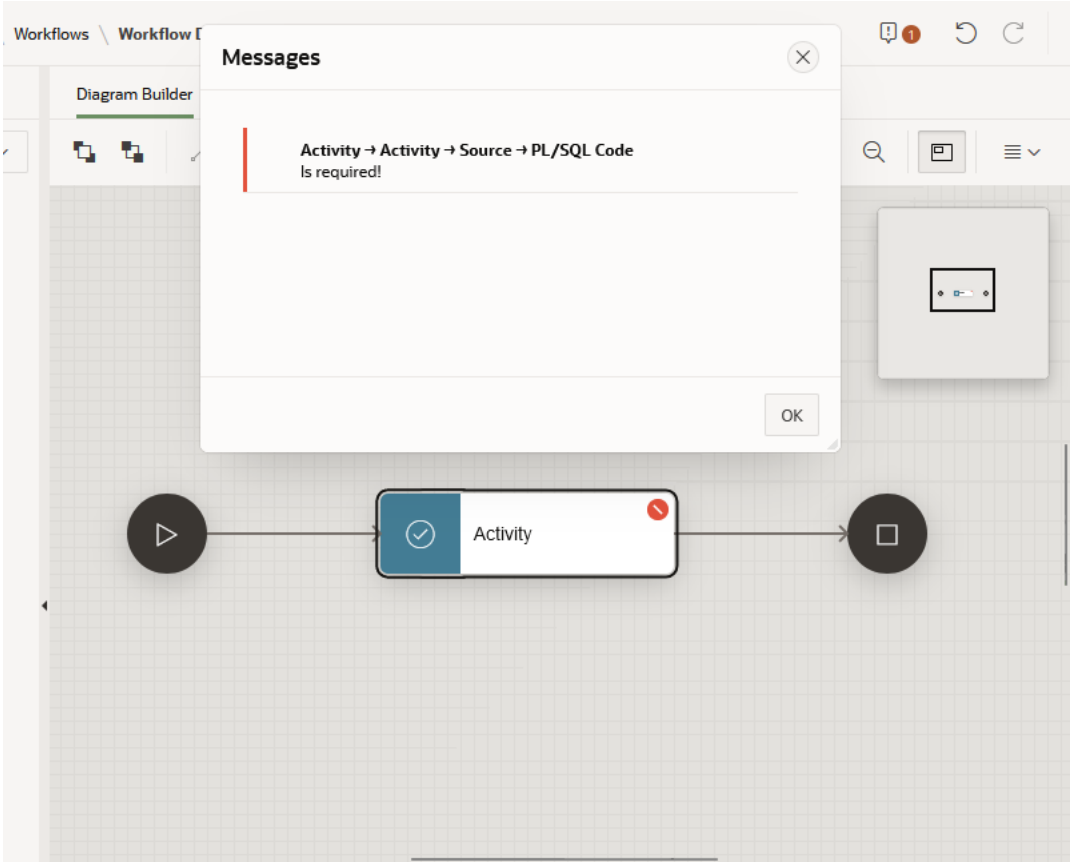
The Activities Palette is a panel that displays all available activity types, including any custom process plugins you have configured.

The **Activities Palette** is located at the bottom of the drawing area. You can add activities to your workflow by dragging them from the activity palette onto the drawing area.

Once you drag an activity onto the drawing area, a context-sensitive icon appears in the top right corner. The icon is red when there is an error or missing information in the activity. Selecting the activity brings up the activity properties with the error location(s) marked.



The **Show Messages** icon displays the number of errors in the workflow. You can select the icon to display a list of the error messages. You can select a specific error message to bring up the activity properties.



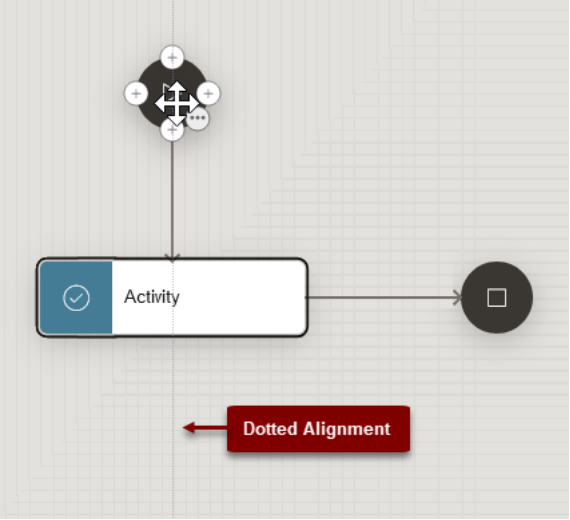
20.1.3.4 The Workflow Diagram Builder

The **Workflow Diagram Builder** is the central pane of the Workflow Designer.

The Diagram Builder models the information in the workflow tree in graphical form, with a visual representation of all activities and connections in a particular version of a workflow. In the Diagram Builder, you can create activities, move activities around, delete activities, and connect activities with transitions.

> **Tip:**
>
> The Diagram Builder does not show workflow variables, parameters, activity variables, or participants. You can view those in the workflow tree.

The **Drawing Area** has a 10 pixel grid in the background. You can use the drawing area to reposition and move activities and connections around. When you reposition activities and links, they move in 10 pixel steps, staying aligned with the grid. If you reposition an activity, the connectors attached to the activity move with it. As you drag an activity, a dotted line appears when the activity aligns horizontally or vertically with other activities on the drawing area.



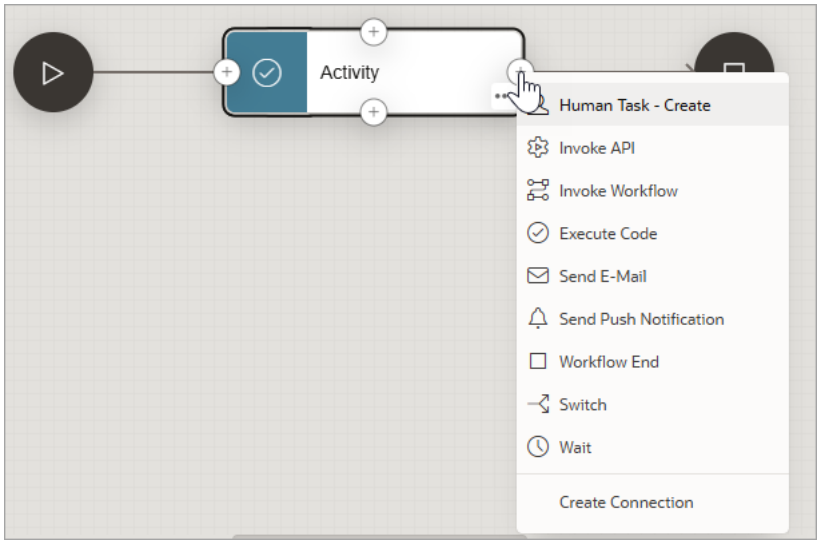
The drawing area resizes automatically. If you move an activity outside of the drawing area, the drawing area automatically expands to accommodate the activity. If you delete activities, the drawing area shrinks to contain the remaining activities.

You can scroll the drawing area with the vertical and horizontal scrollbars, and zoom in and out with the buttons in the toolbar.

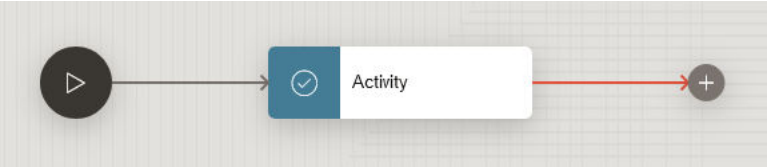
Activity Tools

Activities show two kinds of tools: the **Connect** tool and the **Menu** tool. These tools are context-sensitive, and are not visible if they are not available. For example, an end activity does not show the **Connect** tool, because end activities cannot have outbound connections.

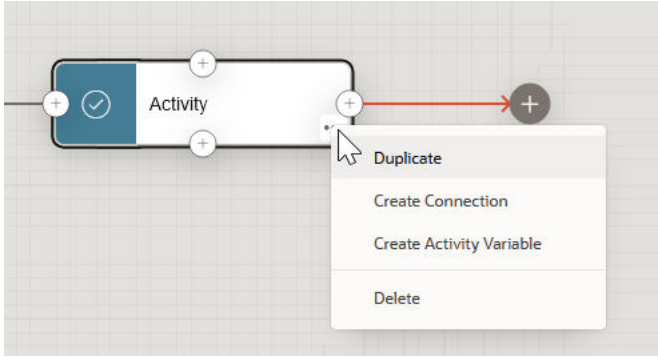
Use the **Connect** tool when connecting two activities. The connect tool is available on multiple sides (top, right, bottom, or left) of each activity. Clicking the tool opens a menu where you can select the type of activity. APEX creates the new activity on the same side you clicked the connect tool on.



You can also use the **Connect** tool to create a new connection. Drag the end of the connection over an existing activity to connect it, or drag out from the connection tool on the existing activity to draw a new connection.



Use the **Menu** button to open a menu with the available actions for the activity.



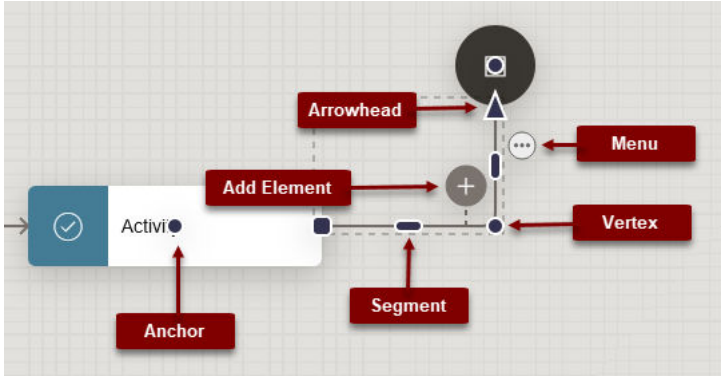
Tip:

For **Invoke Workflow** activities, use the **Open** option on the menu to open the invoked workflow in the Workflow Diagram Builder. To return to the original workflow, select the **Back** link on the toolbar.

Connection Tools

Connections may show different types of tools depending on which conditions they meet and whether they use normal or orthogonal routers:

- **Arrowhead** - Appears at the end of the visible connection at the connection point with the activity. At the source activity, it is a rounded rectangle. At the target, it is a triangle. Use arrowheads to reposition the ends of the connection. If you drag and release the arrowhead over an activity, the point where you release it is the **anchor**. You can drag an arrowhead to a new activity.
- **Menu** - Opens a menu with available actions for the connection.
- **Add Element** - Appears near the center of the connection or at the end of a connection not attached to an activity. Clicking the tool opens a menu with activities to place at the mid-point of the connection, effectively splitting it in two, or at the end of the connection.
- **Vertex** - Appears as point(s) on a connection. Connections always route through their vertices. Drag the center of a link to create a new vertex. Double-click an existing vertex to remove it. Drag an existing vertex to reposition it.
- **Anchor** - Appears over the source or target activity when a link is connected to it. Reposition anchors by dragging the circle or releasing the arrowhead over an activity. Double-clicking the anchor sets the anchor position to the center of the activity.
- **Segment** - Appears between two horizontally- or vertically-aligned connection points. Drag a segment to move that segment horizontally or vertically. This may create new vertices.



Positioning

If the diagram or the activity is read-only, you cannot change the position of the cells.

Otherwise, you can reposition elements on the drawing area by dragging them, or by using the keyboard shortcuts. For the keyboard shortcuts to work, you must place focus in the diagram.

For more on keyboard shortcuts, see About Workflow Keyboard Shortcuts.

Connecting Activities in the Workflow Diagram Builder

You can connect (link) activities by:

- clicking the **Connect** tool and selecting a menu entry
- dragging a link from the **Connect** tool and connecting it to an existing element
- dragging an activity from the palette and dropping it over an element when the element is highlighted

- dragging an activity from the palette and dropping it over a transition or link when the transition or link is highlighted
- dragging an activity from the palette or drag-repositioning an existing one, and dropping it over a placeholder at the end of the link

20.1.4 About the Workflow Console, Workflow Details, and Workflow Dashboard

The **Workflow Console** displays all workflows for a given report context. The **Workflow Details** shows details for a specific workflow instance. The **Workflow Dashboard** shows analytics for all workflows in a given report context.

When you create the Workflow Console using the Create Page Wizard, the wizard creates two pages in the application: an overview page with all the workflows for the given report context, and a details panel that opens when a user selects a workflow from the overview page. If you select the **Include Dashboard Page** option, the wizard also creates a dashboard page with charts that compare workflow performance and workflow states in the given report context.

You can further customize all three pages with Page Designer.

The Workflow Console Page

The **Workflow Console** allows workspace users to view and manage their workflow instances. When you configure the workflow console, you have different **Report Contexts** to choose from:

- **My Workflows**, for workflow owners
- **Admin Workflows**, for workflow administrators
- **Initiated by Me**, for workflow initiators

Clicking on a workflow in the Workflow Console brings you to the workflow details page.

The Workflow Details Page

The workflow details page shows:

- activity instances and their states
- when an activity was started, completed, retried, faulted, or terminated
- workflow instance variables, which can be edited by workflow administrators

> **Tip:**
>
> The **Workflow Details** page shows `VARCHAR2` representations of the variables. For `CLOB` variables, the workflow details page displays the section of the `CLOB` that is accommodated within the `VARCHAR2` limit.

- workflow instance parameters
- an audit history of the workflow instance including, if applicable, the parent workflow and parent activity instance
- (optional) a read-only visual representation of the workflow
- (invoked workflows only) a button to navigate to the details of the parent workflow

Depending on the workflow context, users can perform various actions from the workflow details page, including suspending, resuming, terminating, and retrying a workflow instance. To learn more about these actions, see Working with Workflows in *Oracle APEX End User's Guide*.

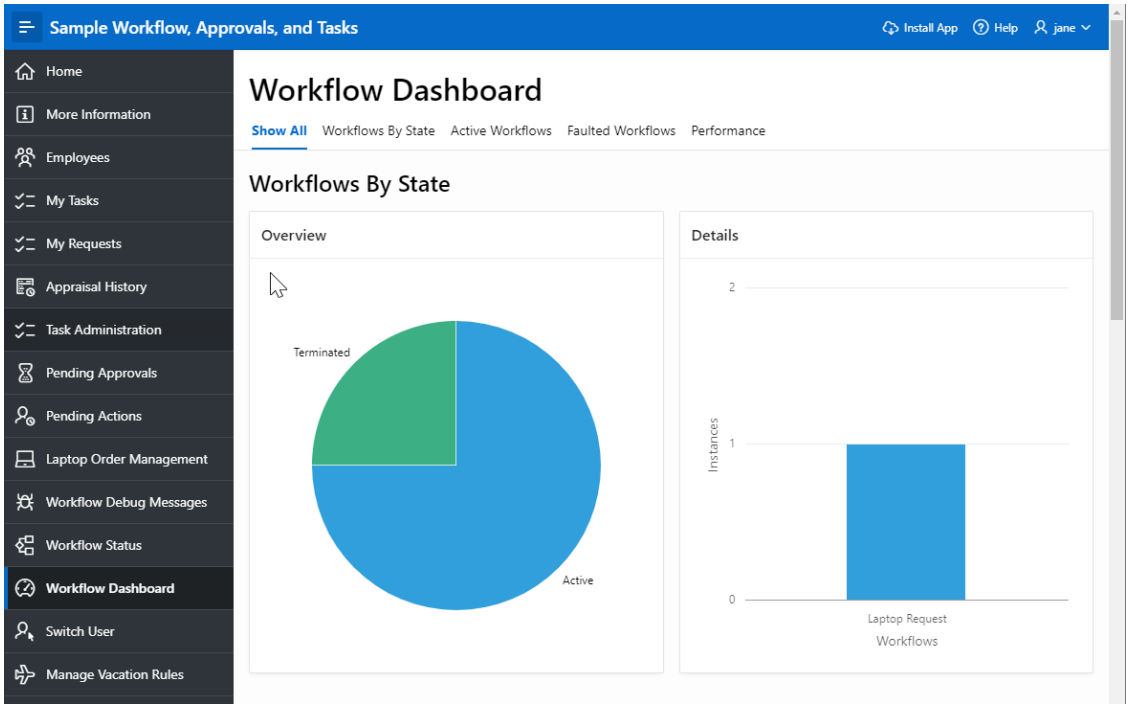
The workflow details page includes a **Workflow Diagram** region. The workflow diagram region is a read-only visual representation of the workflow instance. It shows the progress of the workflow instance and the activities already completed.

The Workflow Dashboard Page

The workflow dashboard page shows different tabs of information for a given workflow:

- **Workflows by State** - Shows a pie chart with the distribution of workflows by state. Clicking on a particular state shows the number of workflow instances currently in that state.
- **Active Workflows** - Shows the number of workflows in an **active** or **suspended** state. Clicking on a particular workflow shows the number of workflow instances waiting at a particular activity.
- **Faulted Workflows** - Shows the number of workflows in a **faulted** state. Clicking on a particular workflow shows the number of workflow instances that are faulted at a given activity.
- **Performance** - Shows the average completion time for workflows. Use the **Time Unit** dropdown list to change the chart's time unit.

The workflow dashboard has the same report context as the workflow console.



20.1.5 About the Workflow Runtime Engine

The **Workflow Runtime Engine** executes workflow instances according to the execution sequence.

When the workflow runtime engine starts a new workflow instance, the instance is based on the active version of the workflow definition unless you currently have a developer session open. If a developer session is open, the instance is based on the development version of the workflow.

> **Tip:**
>
> If you want to run the active version of the workflow while you have a developer session open, run your application in an incognito browser window.

When workflows execute, they begin at the start activity. The execution order of a workflow is determined by the connections and activities in the workflow, not the sequence of the items in the workflow designer tree. In cases where there are multiple options for connections (such as a case switch activity or multiple error connections), the workflow runtime engine evaluates the options based on the execution sequence specified in the property editor.

20.1.6 States and Transitions

Learn about states and transitions for workflows and for workflow activities.

- Workflow States and Transitions
  Learn about available workflow states and potential transitions.
- Activity States and Transitions
  Learn about available activity states and potential transitions.

20.1.6.1 Workflow States and Transitions

Learn about available workflow states and potential transitions.

Workflow States

A **workflow state** is the current state of a workflow instance. There are five separate workflow states, and a workflow may only be in one state at a time.

Active
A workflow that is currently running.

Suspended
A workflow that is currently suspended at a particular activity. A suspended workflow can be resumed by the **Workflow Administrator**.

Completed
A workflow that has reached an end activity and completed successfully.

Terminated
A workflow that has been stopped by the **Workflow Owner** or **Workflow Administrator** or a workflow that has reached an end activity with the state attribute set to terminated.

Faulted
A workflow that produced an error at one of its activities. The **Workflow Administrator** can retry or stop a faulted workflow.

Workflow State Transitions

A **workflow transition** describes a workflow moving from one state to another. Workflow state transitions happen either by the workflow engine itself, by appropriate APIs, or as a result of a user action. For example, a workflow may transition from **Active** to **Completed** after a user approves a task.

Initial State	Category	Possible Transitions
Active	Normal	• **Completed**, when the workflow reaches an end activity. • **Faulted**, when the creation of the workflow instance fails due to parameters that cannot be handled correctly, if there is more than one start activity, or if there is no start activity. • **Suspended**, per API or user request. • **Terminated**, per API or user request, or because the workflow reaches an end activity that moves the workflow to a terminated state.
Completed	Normal	None.

Initial State	Category	Possible Transitions
Faulted	Exception **Tip:** Workflows transition to faulted if an activity cannot run successfully or if there is an error in the workflow itself due to a failure to retrieve workflow variables or a system error. Faulted	• **Active**, when the workflow is retried by API or user request and retrying successfully brings the workflow back into a normal state of operation. • **Suspended**, per API or user request. • **Terminated**, per API or user request.

Initial State	Category	Possible Transitions
	workflows can be retried after modifying variables, or terminated if the user wants to end the workflow instance and no longer run the workflow.	

Initial State	Category	Possible Transitions
Suspended	Normal **Tip:** Typically, a workflow is suspended due to a Wait activity. **Tip:** By default, workflows resume at the current activity. Use `APEX_WORKFLOW.RESUME` to resume the workflo	• **Active**, when the workflow is resumed by API or user request. • **Faulted**, when an outstanding (asynchronous) activity returns with an error and the activity and workflow instance transition to faulted. • **Terminated**, by API or user request.

Initial State	Category	Possible Transitions
		w at any acti vity.
Terminated	Exception	None.
		Tip: A ter min ated wor kflo w inst anc e can not be retri ed.

20.1.6.2 Activity States and Transitions

Learn about available activity states and potential transitions.

Activity States

An **activity state** is the current state of an activity. An activity can only be in one state at a time.

Active
An activity that is currently running.

Waiting
An activity that is currently waiting for a result. For example, an activity might be waiting on approval from a task, or waiting for an invoked workflow to run.

Completed
An activity that completed successfully.

Terminated
An activity that was in active state when the workflow was stopped.

Faulted
An activity that fails to run.

Activity State Transitions

An **activity transition** describes an activity moving from one state to another.

Initial State	Category	Possible Transitions
Active	Normal	• **Completed**, when the activity completes successfully and the workflow continues. • **Faulted**, when the activity fails to run, either because the activity itself faulted, or because something leading into or out of the activity faulted. • **Terminated**, when the workflow instance is terminated. • **Waiting**, when the activity is waiting for a result to come back. For example, asynchronous activities or approval activities may move into a waiting state.
Completed	Normal	None.
Faulted	Exception	• **Active**, when the workflow instance is retried by API or user request. Any faulted activities move to an **active** state and the workflow runtime engine attempts to execute the activity again.
Terminated	Exception	None.
Waiting	Normal	• **Active**, when the activity operation is complete and signals to the workflow that it is okay to continue. The duration of this transition may be very short, so the user may never experience an activity in **waiting** state, even when there is a pause in the workflow. • **Faulted**, when the activity processing fails. • **Terminated**, when the workflow instance is terminated.

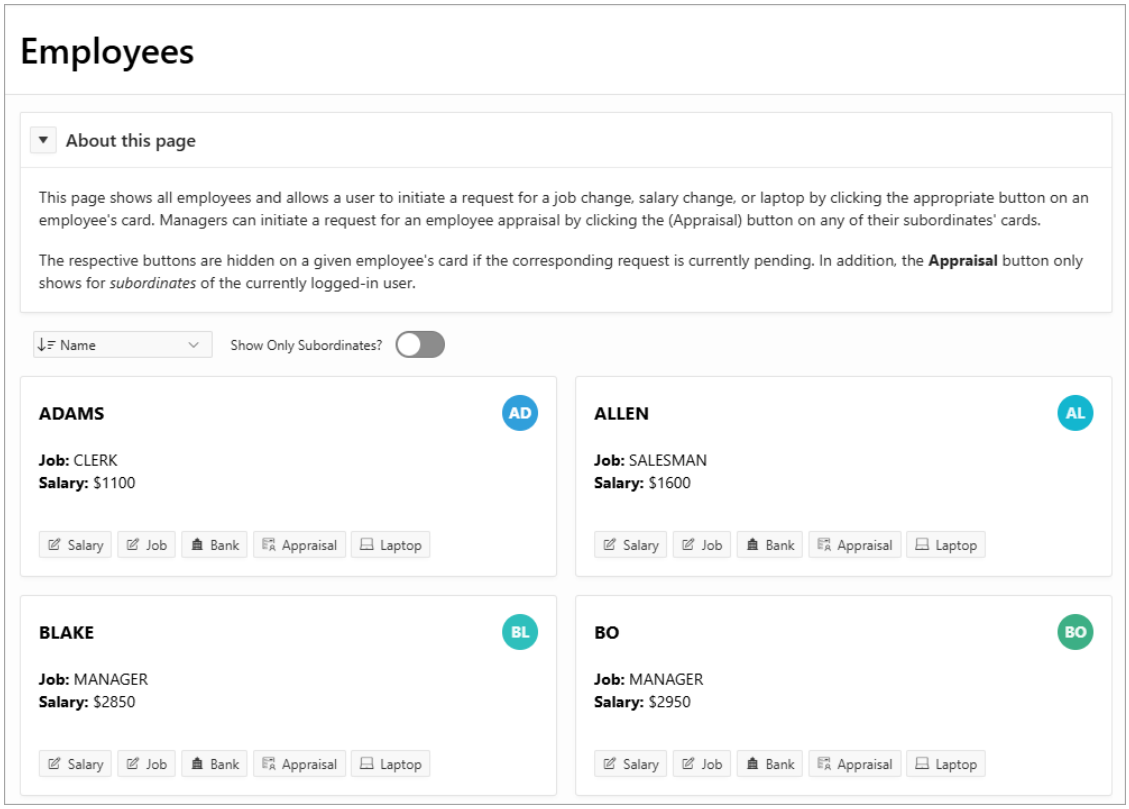
20.1.7 Example: Workflows

View the workflows functionality in the **Sample Workflows, Approvals, and Tasks** app.

The Sample Workflows, Approvals, and Tasks Application

Sample apps highlight specific APEX functionality and serve as a developer guide for how to use particular features. For workflows, you can install the **Sample Workflows, Approvals, and Tasks** app from the Gallery.

> **Tip:**
>
> To view this example, install the sample app, **Sample Workflows, Approvals, and Tasks**. To learn more, see Installing Apps from the Gallery.
>
> Once you install **Sample Workflows, Approvals, and Tasks**, you can reference the **More Information** page of the sample app for an overview of workflow features, approval and action tasks features, and a discussion of how to use the sample app.



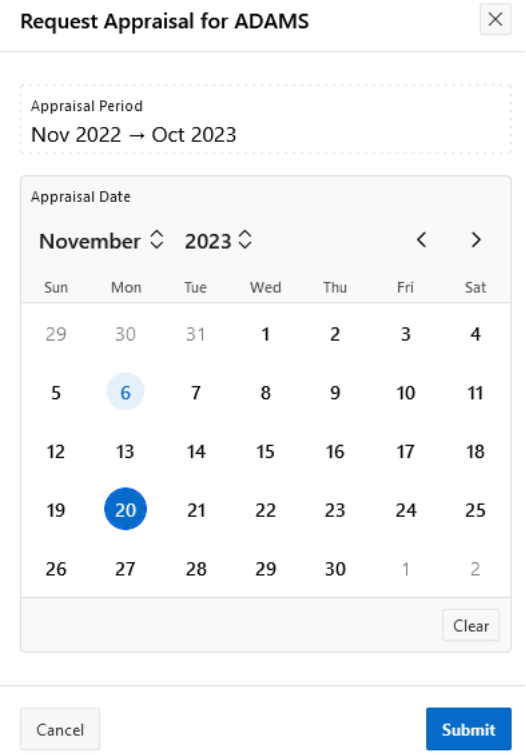
The **Employees** screen shows all employees. Users can request the following changes for an employee:

- **Salary** - Requests a change to the employee's salary. This functionality uses an **approval task** to review and approve the change and a customized task details page.
- **Job** - Requests a change to the employee's job. This functionality uses an **approval task** to review and approve the change and a standard task details page.

- **Bank** - Requests a change to the employee's bank account. This functionality uses an **approval task** that the initiating employee can approve.
- **Appraisal** - Begins the process of an employee's appraisal. This functionality uses a **workflow** with **action tasks** for the employee, the employee's manager, and, if applicable, a VP.
- **Laptop** - Requests the purchase of a new laptop for the employee. This functionality uses a **workflow** with an **approval task** and an **action task**. The workflow is programmatically completed when the laptop is delivered.

Employee Appraisal Workflow

The **Request Appraisal for ADAMS** screen allows users to request an appraisal for an employee and set a completion date for the appraisal.

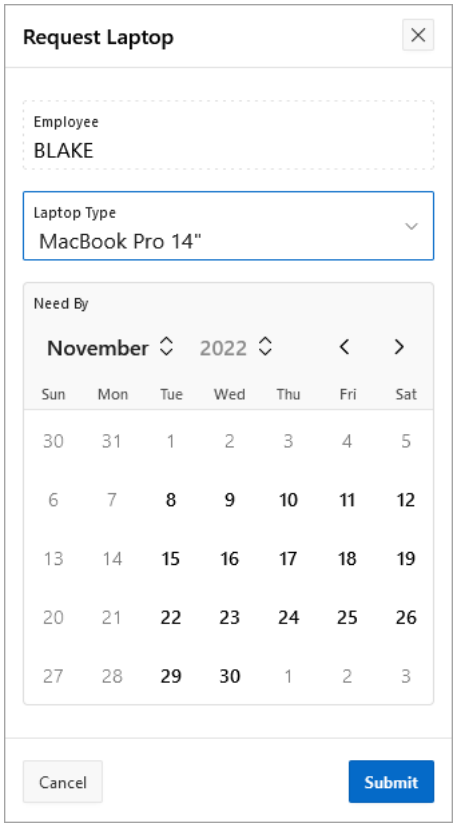


Once a user requests an employee appraisal, the **Appraisal** button on the Employees screen is no longer visible.

The workflow for the employee appraisals includes a combination of activities including human tasks, Invoke APIs, and a switch.

Laptop Request Workflow

The **Request Laptop** screen allows users to request a specific laptop by a specific date for an employee.

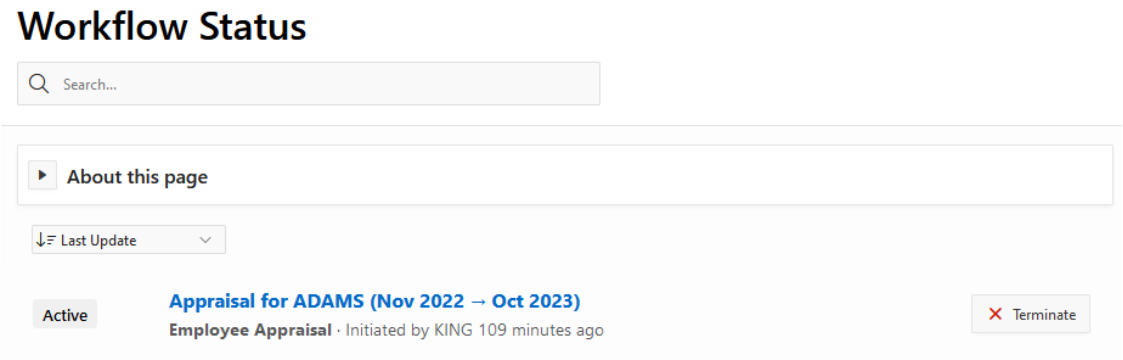


Once a user requests a laptop, the **Laptop** button on the Employees screen is no longer visible until a member of the Operations team marks the laptop as **delivered**.

The workflow for laptops includes a combination of activities including human tasks, Invoke APIs, switches, emails, and a wait activity.

Workflow Administration

The Workflow Status page, based on the **My Workflows** context, gives workflow participants a way to easily view workflow status. If the participant is also a workflow administrator, they can administer the workflows from this page as well.



Clicking on the title of each workflow instance gives the workflow participant (in the OPERATIONS department) access to perform further actions with the workflow, including checking workflow activities, variables, parameters, and history.

20.2 Creating a Workflow

Create a Workflow in Shared Components, Workflows and Automations.

This process is a high-level summary of the steps needed to create and activate a workflow.

To create a new workflow:

1. Select an application.
2. Navigate to the Workflows page:
   a. On the application home page, select **Shared Components**.
   b. Under Workflows and Automations, select **Workflows**.
3. Click **Create**.

   The workflow definition opens.
4. In the workflow tree, select the **Workflow**.
5. Under Identification, specify the workflow **Name**.

   The workflow name displays in the workflow tree.
6. Under Identification, specify the user-friendly workflow **Title**.
7. Configure the workflow:
   a. Add **Workflow Parameters** by right-clicking the workflow and choosing **Create Parameter**.

      For more on creating parameters, see Creating a Workflow Parameter.
   b. Add **Workflow Activities** by right-clicking the workflow version and choosing **Create Activity**.

      For more on creating activities, including different methods of creating activities, see Creating a Workflow Activity.
   c. Add **Workflow Variables** by right-clicking the workflow version and choosing **Create Variable**.

      For more on creating workflow variables, see Creating a Workflow Variable.
   d. Add **Workflow Participants** by right-clicking the workflow version and selecting **Create Participant.**

      For more on adding participants, see Adding a Workflow Participant.
   e. Add **Workflow Connections** between activities by right-clicking the activity and selecting **Create Connection**.

      For more on creating connections, including different methods of creating workflow connections, see Adding a Workflow Connection.
8. Click **Save** to save the workflow definition.
9. Add the workflow to the application as a page process.

   For more on using workflows in page processes, see About Using Workflows in Page Processes.
10. Activate the workflow by right-clicking the workflow version and selecting **Activate**.

    For more on workflow versions, see Saving and Activating a Workflow Version.

- Creating a Workflow Parameter
  Create a workflow parameter by right-clicking on the workflow and selecting Create Parameter.
- Creating a Workflow Variable
  Create a workflow variable by right-clicking on the workflow version and selecting Create Variable.
- Creating a Workflow Activity
  Create a workflow activity by right-clicking the workflow version and selecting Create Activity.
- Adding a Workflow Connection
  Add a workflow connection by right-clicking on an activity and selecting Create Connection.
- Adding a Workflow Participant
  Add a workflow participant by right-clicking the workflow version and selecting Create Participant.

20.2.1 Creating a Workflow Parameter

Create a workflow parameter by right-clicking on the workflow and selecting Create Parameter.

A workflow parameter is an input for the workflow. APEX passes parameters to the workflow instance when the workflow starts. A parameter applies to all versions of a workflow, and the value of a parameter does not change during the workflow runtime.

To create a workflow parameter:

1. Right-click on the workflow and choose **Create Parameter**.

   Create Version and Create Parameter are options on the **Workflow** context menu. Create Activity, Create Variable, and Create Participant are options on the **Workflow Version** context menu.

2. Define the parameter attributes:

   a. Static ID - The unique name of the parameter.

   b. Label - The user-friendly label for the parameter.

   c. Data Type - Allowable data types are `VARCHAR2`, `TIMESTAMP`, `TIMESTAMP WITH LOCAL TIME ZONE`, `TIMESTAMP WITH TIME ZONE`, `NUMBER`, `BOOLEAN`, `CLOB`.

   d. Direction - The direction of the parameter as defined in the workflow. Select **In**, **Out**, or **In/Out**.

      Note that **Out** parameters do not appear in the **Page Process Component Parameters** section when the workflow is used in a page process. Out parameters only appear when the workflow is used in the **Invoke Workflow** activity of another workflow.

   e. Value - Specify whether the parameter is required and must always be set.

   f. Session State Format Mask - Only applicable for some data types. For more information, see item-level help.

   g. Application Format Masks - Only applicable for some data types. For more information, see item-level help.

3. Under Value:

   a. If the parameter is required, turn on the **Required** toggle.

   b. Set the Default Value.

20.2.2 Creating a Workflow Variable

Create a workflow variable by right-clicking on the workflow version and selecting Create Variable.

A workflow variable is an input for the workflow that is specific to the workflow version. The value of a workflow variable may change during the workflow runtime.

You can only create a workflow variable if the workflow is in development. You cannot create a variable if the workflow is active.

To create a workflow variable:

1. Right-click on the workflow version and choose **Create Variable**.

   You can also right-click on the **Variables** region, and select **Create Variable**.

2. Fill in the properties for the variable:

   a. Static ID - The unique name of the variable.

   b. Label - The user-friendly label for the variable.

   c. Data Type - The type of data expected for the variable. Select from `VARCHAR2`, `TIMESTAMP`, `TIMESTAMP WITH LOCAL TIME ZONE`, `TIMESTAMP WITH TIME ZONE`, `NUMBER`, `BOOLEAN`, and `CLOB`.

   d. Value - The initial value for the variable. Select from Static Value, Item, SQL Query (return single value), Expression, Function Body, or Null.

20.2.3 Creating a Workflow Activity

Create a workflow activity by right-clicking the workflow version and selecting Create Activity.

A workflow activity specifies what kind of work APEX performs when the workflow arrives at that activity.

You can only create a workflow activity if the workflow is in development. You cannot create a workflow activity if the workflow is active.

To create a workflow activity:

1. Choose one of the following options:

   - Right-click on the workflow version and choose **Create Activity**.
   - Drag and drop a specific activity from the Activity Palette into the Diagram Builder.

     > **Tip:**
     >
     > You can drop an activity on an existing activity to automatically place the new activity next in the workflow execution order. You can also drop an activity on a connection to place the new activity between the original source and the target of the connection.

   - Right-click an existing activity in the workflow tree and select **Create Activity Below**.

> **Tip:**
>
> The new activity is not connected to any existing activities. You must explicitly create connections to other activities.

- Right-click an existing activity in the workflow tree and select **Create Activity After**.

> **Tip:**
>
> The new activity is connected to the existing activity. If the existing activity was connected to another activity, the new activity replaces that connection.

By default, the new activity type is **Execute Code**.

2. Enter the activity **Name**.

   The name displays in the workflow tree and in the Diagram Builder. The activity name must be unique for the workflow.

3. Select the activity **Type**:
   - Custom Process Type Plug-in - An activity that uses an existing process type plug-in. See Adding a Custom Process Type Plug-in as an Activity.
   - Execute Code - An activity that executes the entered code.
   - Human Task - Create - An activity that links to an existing task definition. See Adding a Workflow Human Task Activity.
   - Invoke API - An activity that invokes an API.
   - Invoke Workflow - An activity that invokes another workflow in the current application. See Adding an Invoke Workflow Activity.
   - Send Email - An activity that sends an email.
   - Send Push Notification - An activity that sends a push notification.
   - Switch - An activity that defines branches in the workflow. See Adding a Workflow Switch Activity.
   - Wait - An activity that represents a pause in the workflow. See Adding a Workflow Wait Activity.
   - Workflow End - An activity that ends the workflow or a particular branch of a workflow. See Adding a Workflow End Activity.
   - Workflow Start - The single activity that starts the workflow. See Adding a Workflow Start Activity.
4. Fill in the remaining settings for the activity type.
5. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
6. (Optional) Under Deadline, configure **Due On Type**.
7. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of activities in the workflow execution chain.

8. Continue configuring the activity:

- **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
- Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
- Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
- Create another activity connected to this activity by clicking the **Connect** tool on any side of the existing activity and selecting the type for the new activity.

9. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

10. Click **Save** to save your changes.

- Adding a Workflow Start Activity
  A **Start** activity begins a workflow.
- Adding a Workflow End Activity
  An end activity ends a workflow or a workflow branch.
- Adding a Workflow Switch Activity
  A switch activity defines conditional branches in your workflow.
- Adding a Workflow Wait Activity
  A **Wait** activity pauses workflow execution.
- Adding a Workflow Human Task Activity
  A **Human Task - Create** activity creates a human task using an existing task definition.
- Adding a Custom Process Type Plug-in as an Activity
  A custom process type plugin uses a custom plugin as an activity.
- Adding an Invoke Workflow Activity
  An invoke workflow activity calls another workflow in the same application.
- Defining an Activity Variable
  Add an activity variable by right-clicking the activity in the workflow tree and selecting **Create Activity Variable**.

20.2.3.1 Adding a Workflow Start Activity

A **Start** activity begins a workflow.

There can only be one start activity in a workflow.

To create a workflow start activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the Workflow Start activity from the Activities Palette into the Diagram Builder.

2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.

3. Set the Type to **Workflow Start**.

4. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
5. (Optional) Under Deadline, configure **Due On Type**.
6. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.
7. Continue configuring the activity:
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
   - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
   - Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.
8. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
9. Click **Save** to save your changes.

A workflow can only have one Start Activity. The Start Activity cannot have any incoming connections.

20.2.3.2 Adding a Workflow End Activity

An end activity ends a workflow or a workflow branch.

To create a workflow end activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the Workflow End activity from the Activities Palette into the Diagram Builder.
2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.
3. Set the type to **Workflow End**.
4. Select the **End State**:
   - Completed - When the workflow reaches this activity, the workflow ends with the **Completed** state.
   - Terminated - When the workflow reaches this activity, the workflow ends with the **Terminated** state. This state indicates that something unexpected happened in the workflow.
5. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
6. (Optional) Under Deadline, configure **Due On Type**.
7. Under Layout, set the sequence.

The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

8. Continue configuring the activity:
   - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
   - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
9. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
10. Click **Save** to save your changes.

A workflow can have one end activity or multiple end activities. An end activity does not have any outgoing connections.

20.2.3.3 Adding a Workflow Switch Activity

A switch activity defines conditional branches in your workflow.

To create a workflow switch activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the Switch activity from the Activities Palette into the Diagram Builder.
2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.
3. Set the Type to **Switch**.
4. Select the switch type:
   - **True False** - For branches based on a true/false outcome of the switch condition. See Configuring a True False Switch.
   - **Check Workflow Variable** - For branches based on the value of a workflow variable. See Configuring a Check Workflow Variable Switch.
   - **Case** - For branches based on the value of a case expression. See Configuring a Case Switch.
   - **If Elsif Else** - For branches based on the evaluation of more complex conditional expressions. See Configuring an If Elsif Else Switch.
5. Fill in the remaining settings for the activity type.
6. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
7. (Optional) Under Deadline, configure **Due On Type**.
8. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

9. Continue configuring the activity:
   - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
   - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
   - Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.
10. Click **Show Messages** to view and fix errors.

    The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
11. Click **Save** to save your changes.

- Configuring a True False Switch
  A **True False** switch is based on the result of a single condition that returns a boolean.
- Configuring a Check Workflow Variable Switch
  A **Check Workflow Variable** switch is based on the value of a workflow variable.
- Configuring a Case Switch
  A **Case** switch is based on the result of a computation.
- Configuring an If Elsif Else Switch
  An **If Elsif Else** switch is based on `AND/OR` comparisons between the conditions.

20.2.3.3.1 Configuring a True False Switch

A **True False** switch is based on the result of a single condition that returns a boolean.

To configure a **True False** Switch:

1. Set the switch type to **True False Check**.
2. Set the Condition Type:
   - Rows Returned - Defined by a SQL Query.
   - Rows Not Returned - Defined by a SQL Query.
   - Expression - Defined by an PL/SQL Expression.
   - Function Body - Defined by a PL/SQL Function Body.
   - Workflow Variable = Value - Select the workflow parameters, version variables, or activity variables.
   - Workflow Variable != Value - Select the workflow parameters, version variables, or activity variables.
   - Workflow Variable is NULL.
   - Workflow Variable is NOT NULL.
3. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
4. (Optional) Under Deadline, configure **Due On Type**.
5. Under Layout, set the sequence.

The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

Add the switch branches. To create a new branch:

6. Right-click on the Switch activity in the tree and select **Create Connection**.

   You can also click the context menu at the bottom right corner of the Switch activity in the Diagram Builder and select **Create Connection**.

7. In the properties of the Connection, specify the following attributes:

   a. When - Specify whether this connection is for a True, False, or Null value of the condition.

   b. To - Select the target activity that the condition leads to.

8. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

9. Click **Save** to save your changes.

20.2.3.3.2 Configuring a Check Workflow Variable Switch

A **Check Workflow Variable** switch is based on the value of a workflow variable.

To configure a **Check Workflow Variable** Switch:

1. Set the switch type to **Check Workflow Variable**.
2. For Compare Variable, select the workflow parameter, version variable, or activity variable for the comparison.
3. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
4. (Optional) Under Deadline, configure **Due On Type**.
5. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

Add the switch branches. To create a new branch:

6. Right-click on the Switch activity in the tree and select **Create Connection**.

   You can also click on the menu at the bottom right corner of the Switch activity in the Diagram Builder and select **Create Connection**.

7. In the properties of the Connection, specify the following attributes:

   a. Operator - From the dropdown, select the appropriate operator.

   b. To - Select the target activity that the condition leads to.

8. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

9. Click **Save** to save your changes.

20.2.3.3.3 Configuring a Case Switch

A **Case** switch is based on the result of a computation.

To configure a **Case** Switch:

1. Set the switch type to **Case**.
2. Specify the **Compare Data Type** by selecting the appropriate data type for the result of the case statement or expression.

   Case supports the following data types: `VARCHAR2`, `NUMBER`, `BOOLEAN`, `TIMESTAMP`, `TIMESTAMP WITH TIME ZONE`, `TIMESTAMP WITH LOCAL TIME ZONE`.
3. Specify the Compare Type by selecting either SQL Query, Expression, or Function Body.
4. In the Code Editor, enter the Compare Query.
5. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
6. (Optional) Under Deadline, configure **Due On Type**.
7. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

Add the switch branches. To create a new branch:

8. Right-click on the switch activity in the tree and select **Create Connection**.

   You can also click on the menu at the bottom right corner of the Switch activity in the Diagram Builder and select **Create Connection**.
9. In the connection properties, specify the following attributes:
   a. Sequence - Specify the sequence. The sequence determines the order that APEX evaluates case branches at runtime.
   b. Operator - Select the branch condition expression.
   c. Value - Specify the value to compare with.
   d. To - Select the target activity the branch leads to.
10. Click **Show Messages** to view and fix errors.

    The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
11. Click **Save** to save your changes.

20.2.3.3.4 Configuring an If Elsif Else Switch

An **If Elsif Else** switch is based on `AND/OR` comparisons between the conditions.

To configure an **If Elsif Else** switch:

1. Set the switch type to **If Elsif Else**.
2. (Optional) Under Deadline, configure **Due On Type**.
3. Under Layout, set the sequence.

The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

Add the switch branches. To create a new branch:

4. Right-click on the switch activity in the tree and select **Create Connection**.

   You can also click on the menu at the bottom right corner of the Switch activity in the Diagram Builder and select **Create Connection**.

5. In the properties of the connection, specify the following attributes:

   a. Sequence - Specify the sequence. This determines the order that APEX evaluates the branches in at runtime.

   b. Type - From the dropdown, select the appropriate condition that must be satisfied for this branch.

   c. To - Select the target activity the branch leads to.

6. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

7. Click **Save** to save your changes.

20.2.3.4 Adding a Workflow Wait Activity

A **Wait** activity pauses workflow execution.

A **Wait** activity pauses workflow execution. The wait state updates in two ways: a **Time Out** after the specified duration elapses, or a call from `apex_workflow.continue_activity`.

To create a workflow wait activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the workflow Wait activity from the Activities Palette into the Diagram Builder.

2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.

3. Set the type to **Wait**.

4. For Timeout Type, specify Static, SQL Query, Expression, or Function Body.

   For Static, use ISO Duration format values.

5. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.

6. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

7. Continue configuring the activity:

   - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.

- Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
- Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.

8. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

9. Click **Save** to save your changes.

20.2.3.5 Adding a Workflow Human Task Activity

A **Human Task - Create** activity creates a human task using an existing task definition.

In order to use a task as part of a workflow, you must already have the task definition configured in the application. For more information on setting up task definitions, see About Task Definitions.

To create a workflow human task activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the Human Task - Create activity from the Activities Palette into the Diagram Builder.

2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.

3. Set the Type to **Human Task - Create**.
4. Under Settings, set the Human Task properties:
   a. Definition - Select the Task Definition from the dropdown.
   b. Subject - Set the subject if you want to override the subject from the task definition.
   c. Priority - Set the priority if you want to override the priority from the task definition.
   d. (Approval tasks only) Initiator Can Complete - Set whether the initiator can also complete the task if you want to override this value from the task definition.
   e. Initiator Item - The ID of the user initiating the task. When not specified, this value defaults to the current logged-in user.
   f. Task ID Item - The page item where the ID of the newly created task instance can be stored.
   g. Details Primary Key Item - The workflow item (variable or parameter) pointing to the Primary Key for the system of records that the task definition uses.
   h. (Optional) Outcome - A workflow variable to store the task outcome.
   i. (Optional) Owner - A workflow variable to store the person who approved, rejected, or completed the task.

> **Tip:**
>
> When you create a Human Task activity, APEX creates two workflow variables: `TASK_OUTCOME` and `APPROVER`. In the activity settings, you can set the Outcome and Approver attributes to these variables. At runtime, after the task completes, the workflow variable `TASK_OUTCOME` is set to `APPROVED` or `REJECTED` based on the outcome of the task. Subsequent activities in the workflow can use these workflow variables. You can choose to ignore the outcome and approver details from the activity, delete the auto-generated workflow variables, and leave the corresponding task attributes empty.
> For example, a Switch activity that follows a Human Task activity can use the **Check Workflow Variable** type to look at `TASK_OUTCOME`, and branch the workflow based on `TASK_OUTCOME = APPROVED` and `TASK_OUTCOME = REJECTED`.
>
> Note that if there are subsequent Human Task activities in the workflow, these workflow variables are not automatically created again. You can choose to re-use the same workflow variables to store the task outcome and actual owner, or you can manually create new workflow variables if you want each Human Task activity outcome stored in a different workflow variable.

5. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
6. (Optional) Under Deadline, configure **Due On Type**.
7. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.
8. Continue configuring the activity:
   - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
   - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
   - Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.
9. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
10. Click **Save** to save your changes.

20.2.3.6 Adding a Custom Process Type Plug-in as an Activity

A custom process type plugin uses a custom plugin as an activity.

To configure the process type plug-in for workflow use:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.

   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.

2. Under Other Components, click **Plug-ins**.
3. Click the plug-in you want to edit.

   The Edit page appears.

4. In Supported For, enable **Workflow Activities**.

To configure the workflow to wait on this process and only progress after the process completes:

5. Under Standard Attributes, select **Wait for Completion**.
6. (Optional) In Callbacks, specify the Completion function for the process.

   When the process completes and calls back into the workflow using `apex_workflow.continue_activity`, the workflow engine executes the Completion function before continuing with the rest of the workflow activities. Refer to APEX_WORKFLOW in *Oracle APEX API Reference* to learn more about the `continue_activity()` API.

7. (Optional) In Callbacks, specify the Termination function for the process.

   Termination functions are useful if you want to free up resources held by your process if the workflow is terminated. You can only add a termination function if you have already selected **Wait For Completion**.

To add the custom process type plug-in to the workflow:

8. Right-click on the workflow version and select **Create Activity**.
9. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.

10. Set the Type to the name of your plug-in. Available custom process plug-ins display under the **Application Components** heading.

    You can also drag the custom process type plug-in activity from the Activities Palette into the Diagram Builder.

11. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
12. (Optional) Under Deadline, configure **Due On Type**.
13. Under Layout, set the sequence.

    The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.

14. Continue configuring the activity:
    - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
    - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
    - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.

- Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.

15. Click **Show Messages** to view and fix errors.

    The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

16. Click **Save** to save your changes.

20.2.3.7 Adding an Invoke Workflow Activity

An invoke workflow activity calls another workflow in the same application.

The invoked workflow must already be configured for the application.

To add an invoke workflow activity:

1. Right-click on the workflow version and select **Create Activity**.

   You can also drag the Invoke Workflow activity from the Activities Palette into the Diagram Builder.

2. Enter the activity **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The activity name must be unique for the workflow.

3. Set the Type to **Invoke Workflow**.

4. Under Settings, set the Invoke Workflow properties:

   a. Definition - Select the Workflow Definition from the dropdown.

   > **Note:**
   >
   > A workflow cannot invoke itself or any of its caller workflows. This avoids cyclic dependency.

   b. Initiator Item - Provide an optional workflow variable or parameter that specifies the initiator of the invoked workflow.

      If Initiator Item is set, the initiator of the invoked workflow is the value of this item. Otherwise, the initiator of the invoked workflow is the initiator of the parent workflow.

   c. Details Primary Key Item - The workflow item (variable or parameter) pointing to the Primary Key for the system of records that the invoked workflow uses.

   d. Workflow ID Item - The workflow item where the ID of the newly invoked workflow instance is stored.

   e. Wait for Completion - Turn this switch **On** if the workflow waits for the invoked workflow instance to complete before progressing to the next activity. Turn this switch **Off** if the workflow progresses to the next activity without waiting for the invoked workflow instance to complete.

      This switch is turned on by default.

   f. Retry Point - Defines the point where an invoked workflow is retried if it faults. **Resume** retries the invoked workflow from the faulted activity. **Start Over** terminates the faulted instance of the invoked workflow and starts a new instance.

Start Over is only applicable when the Retry operation is performed on the faulted parent workflow instance.

> **Note:**
>
> Retry Point is only applicable for invoked workflows with **Wait for Completion** set to `On`.

5. Set the values for the parameters of the invoked workflow:
   a. Under the Invoke Workflow activity, expand the Parameters region to display the parameters of the invoked workflow.
   b. Ignore Output - Specify whether the parameters for Out and In/Out should be ignored. By default, Ignore Output is turned on. If Ignore Output is turned off, then the value can only be set to an item (parameter, variable, or activity variable) of the current workflow.

> **Note:**
>
> If a workflow parameter is set to `Required` and its Value Type is set to `null`, using this workflow in an Invoke Workflow activity throws an exception when the Invoke Activity is executed at runtime.

6. (Optional) Under Additional Data, specify a SQL Query to read additional data. For more information, see item-level help.
7. (Optional) Under Deadline, configure **Due On Type**.
8. Under Layout, set the sequence.

   The sequence determines where the activity displays in the workflow tree. The sequence does not impact the order of the activities in the workflow execution chain.
9. Continue configuring the activity:
   - **Duplicate** the activity by clicking on the context menu and selecting **Duplicate**.
   - Configure **Workflow Connections** by clicking the context menu and selecting **Create Connection**. For more on workflow connections, see Adding a Workflow Connection.
   - Create an **Activity Variable** by clicking the context menu and selecting **Create Activity Variable**. For more on activity variables, see Defining an Activity Variable.
   - Create another activity connected to this activity by clicking **Connect** on any side of the existing activity and selecting the type for the new activity.
10. Click **Show Messages** to view and fix errors.

    The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
11. Click **Save** to save your changes.

20.2.3.8 Defining an Activity Variable

Add an activity variable by right-clicking the activity in the workflow tree and selecting **Create Activity Variable**.

To add a workflow activity variable:

1. Right-click on the activity in the workflow tree and select **Create Activity Variable**.
2. Fill in the properties for the variable:
   a. Static ID - The unique name of the variable.
   b. Label - The user-friendly label for the variable.
   c. Data Type - Select from `VARCHAR2`, `TIMESTAMP`, `TIMESTAMP WITH LOCAL TIME ZONE`, `TIMESTAMP WITH TIME ZONE`, `NUMBER`, `BOOLEAN`, and `CLOB`.
   d. Value Type - The type of value for the variable.
   e. Value Language - Language, if applicable.
   f. (Item only) Item - The workflow parameter, version variable, or activity variable.
   g. (Boolean only) Format Mask - If applicable, fill in the variable or parameter value used to determine the values.

20.2.4 Adding a Workflow Connection

Add a workflow connection by right-clicking on an activity and selecting Create Connection.

At least two activities must be present on the workflow to add a workflow connection. You can only add a connection if the workflow is in development. You cannot add a connection if the workflow is active. For more information, see About Workflow Versions.

To add a new connection to a workflow activity:

1. Right-click on the activity and select **Create Connection**.

   You can also click the menu at the right bottom corner of the activity in the Diagram Builder and select **Create Connection**.
2. Enter the connection **Name**.

   The name displays in the workflow tree and in the workflow diagram builder. The connection name does not need to be unique for the workflow.
3. In the Connection properties, specify the connection type:
   - **Normal** - The default connection type.
   - **Error** - Indicates a target activity for a specific error code. If an error occurs and there is no Error connection, the workflow goes into a Faulted state.
   - **Timeout** - Only available if there is a Due On specified for the activity the connection comes from.
4. Fill in the connection parameters:
   a. (Error only) Sequence - Specify the sequence. The sequence determines the order that APEX evaluates error connections in at runtime.
   b. From - Select the target activity the connection leads from.
   c. To - Select the target activity the connection leads to.
   d. (Error only) Operator - Select the condition expression.
   e. (Error only) Value - The SQL error code. For example, `-01043` for `NO_DATA_FOUND`.
5. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can

click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

6. Click **Save** to save your changes.

20.2.5 Adding a Workflow Participant

Add a workflow participant by right-clicking the workflow version and selecting Create Participant.

To create a new workflow participant:

1. Right-click on the workflow version and select **Create Participant**.
2. Enter the participant **Name**.

   The name can represent the participant's role in the workflow. For example, **Medical Appointments Agent** or **Requisition Manager**.
3. Specify **Type**:
   - **Workflow Owners** can start and terminate instances of the workflow, or retry a faulted workflow.
   - **Workflow Administrators** can start, terminate, suspend, resume, and retry instances of the workflow. Workflow administrators can also update variable values of a workflow instance.
4. Under Value:
   a. Type - Select how the participant is populated.
   b. Code Editor - Specify the value or query that produces a valid participant or list of participants.
5. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.
6. Click **Save** to save your changes.

20.3 Copying a Workflow

Copy a workflow from another application in Shared Components.

To copy a workflow:

1. Select an application.
2. Navigate to the Workflows page:
   a. On the application home page, select **Shared Components**.
   b. Under Workflows and Automations, select **Workflows**.
3. Click **Copy from Other App**.

   The Copy Workflow wizard opens.
4. Select the application to copy the workflow from.
5. Select the workflow to be copied.
6. Specify the workflow **Name**.

The name must be unique for the application.

7. Specify the workflow **Static ID**.

   The static ID must be unique for the application.

8. Click **Copy Workflow**.

The workflow, along with all its versions, parameters, activities, participants, variables, and shared components is copied from the selected application to the current application. If the copied workflow invokes one or more workflows, the invoked workflows are also copied to the current application if they are not already present.

20.4 Creating the Workflow Console, Workflow Details, and Workflow Dashboard Pages

Create the workflow console, workflow details, and workflow dashboard pages with the Create Page Wizard.

To create the workflow console, details, and dashboard pages:

1. Open the Create Page Wizard.
   a. On the Workspace home page, click App Builder.
   b. Select the application that needs a Workflow Console and Details page.
   c. On the Application home page, select Create Page.

      The Create Page Wizard appears
2. Under Component, select **Workflow Console**.
3. For Page Definition, set the following:
   a. Page Number - The integer value that identifies the workflow console page in the application.
   b. Name - The page title for the workflow console page.

      After page creation, you can modify the name in Page Designer.
   c. Report Context - Select My Workflows, Admin Workflows, or Initiated by Me.

      My Workflows is for workflow owners, Admin Workflows is for workflow administrators, and Initiated by Me is for users who initiate workflows.
   d. Include Dashboard Page - Specify whether to create a **Workflow Dashboard** page that allows monitoring of running workflows and comparison of cycle times for activities and workflows.
   e. Form Page Number - The integer value that identifies the workflow details page in the application.
   f. Form Page Name - Specify the page title for the workflow details page.
   g. Form Page Mode - Select Normal, Modal Dialog, or Drawer. See item-level Help for more details.
4. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.

5. Select **Create Page**.

The Workflow Console opens in Page Designer. Navigate back to Page Designer to edit the Workflow Details page or the Workflow Dashboard page.

20.5 Managing Workflow Versions

Each workflow can only have one Active workflow version.

- Creating a Workflow Version
  Create a workflow version by right-clicking on a workflow and selecting Create.
- Saving and Activating a Workflow Version
  Activate a workflow version by right-clicking the workflow version and selecting Activate.
- Editing a Workflow Version
  Edit a workflow version by clicking on it in the workflow tree.
- Copying a Workflow Version
  Copy a workflow version by right-clicking the active workflow and selecting Duplicate.
- Deactivating a Workflow Version
  Deactivate a workflow by right-clicking the active workflow and selecting Deactivate.
- Importing a New Application Version Containing Workflows
  When replacing an application, any running workflow instances are automatically suspended, and must be restarted by a workflow administrator.

20.5.1 Creating a Workflow Version

Create a workflow version by right-clicking on a workflow and selecting Create.

To create a workflow version:

1. Right-click on the workflow and select **Create Version**.

   If the workflow already has an existing active version, you can either duplicate the existing version or create a new version.

2. (Optional) In the Property Editor, set the workflow version.

   The Workflow Version can be a name (`New`) or a number (`1.0`).

3. (Optional) Select a source for **Additional Data**.

   You can specify a SQL Query or table the workflow uses for bind variables during workflow execution. If the source is a table, you must identify the primary key column.

4. (Optional) Set the Debug Level:

   - Info - The default debug level for the workflow version.
   - Warning - Debug level for logging less critical errors.
   - Error - Debug level for logging critical errors.
   - Trace - Debug level for tracing messages with procedures and functions.

   For more information on debug messages, see Utilizing Debug Mode and Viewing the Workflow Instance ID in Debug Messages.

5. Click **Show Messages** to view and fix errors.

   The **Show Messages** icon indicates whether there are any workflow modeling or workflow configuration errors. When you click the icon, APEX displays the list of errors. You can

click on an individual error to focus on that activity in the workflow. Note that you cannot save the workflow until you address any errors.

6. Click **Save** to save your changes.

20.5.2 Saving and Activating a Workflow Version

Activate a workflow version by right-clicking the workflow version and selecting Activate.

In order to activate an **In Development** workflow, you must have participants defined for the workflow.

If your workflow has one or more Invoke Workflow activities, the invoked workflow(s) must have an active version. A workflow cannot be activated if any of the workflows it invokes are in development.

To resolve any errors in the workflow:

1. Click **Show Messages**.

   A list of errors displays. Each error lists the location of the error and the error message.

2. Click on a specific error to go to the error location in Workflow Designer.
3. Make any changes necessary to resolve each error.

   Show Messages disappears when all errors are resolved.

To save changes to a workflow version:

4. Click **Save**.

To activate an **In Development** workflow version:

5. Right-click on the workflow version and select **Activate**.

   Alternatively, in the Property Editor, click **Activate** next to the State property.

   If a workflow already has an existing active version, the existing active version moves to **Inactive** and the current version becomes **Active**.

   If the current workflow has an Invoke Workflow activity and the invoked workflow does not have an Active version, the current workflow cannot be activated.

6. Click **OK** to confirm.
7. Click **Save**.

The workflow version is now **Active**.

20.5.3 Editing a Workflow Version

Edit a workflow version by clicking on it in the workflow tree.

To edit an **In Development** workflow:

1. In the Designer Tree, click the **In Development** version of the workflow.
2. Edit the workflow by:
   - Adding and deleting activities and connections
   - Adding new variables
   - Deleting existing variables
   - Deleting the workflow version

Once you delete a workflow version, APEX deletes any workflow instances based on that version. You cannot delete the **In Development** version of a workflow if that workflow is used in the Invoke Workflow activity of another workflow.

3. Click **Save**.

To edit an **Active** workflow:

4. In the Designer Tree, click the **Active** version of the workflow.
5. Edit the workflow by:
   - Adding activity variables local to an activity
   - Modifying branch conditions on switch activities
   - Adding new workflow participants

   You cannot delete existing participants on an active workflow. You also cannot delete the workflow version.
6. Click **Save**.

20.5.4 Copying a Workflow Version

Copy a workflow version by right-clicking the active workflow and selecting Duplicate.

You must have an **Active** version of the workflow in order to create a copy.

To copy a workflow version:

1. Locate the **Active** version of the workflow in the Designer Tree.
2. Right-click the workflow version and select **Duplicate**.

APEX duplicates the active workflow. The duplicated workflow is now **In Development**.

> **Note:**
>
> Only one version is allowed to be **In Development** at a time. If duplicating the active workflow creates two **In Development** workflows, you must delete one in order to continue.

20.5.5 Deactivating a Workflow Version

Deactivate a workflow by right-clicking the active workflow and selecting Deactivate.

If a workflow is used in the Invoke Workflow activity of another workflow, you cannot deactivate the Active version of that workflow unless there is an In Development version available.

To deactivate a workflow version:

1. Locate the **Active** version of the workflow.
2. Right-click the workflow version and select **Deactivate**.

   Alternatively, you can select the workflow version, and click **Deactivate** in the property editor.
3. Confirm you want to deactivate the workflow by clicking **OK**.

The workflow is now **Inactive**. Any running workflow instances based on this workflow definition continue to run, but no new workflow instances can start.

20.5.6 Importing a New Application Version Containing Workflows

When replacing an application, any running workflow instances are automatically suspended, and must be restarted by a workflow administrator.

When replacing an existing version of an application with a new version of the same application, any running workflow instances are automatically **suspended**. The workflow administrator can then decide which existing workflow instances should be resumed and which should be terminated.

APEX does not automatically resume suspended workflow instances since the workflow definition in the imported application could have drastically changed. If the workflow definition is changed, automatically resuming workflow instances could lead to irregular results.

20.6 About Using Workflows in Page Processes

Add the new Workflow Process Type plug-in to a page to expose workflow operations.

The **Workflow Process Type** plug-in is a new Process Type available in the processing section of application pages. This plug-in exposes specific operations for workflows.

- Starting a Workflow Instance
  Start a workflow instance with the Workflow Start page process.
- Terminating a Workflow Instance
  Terminate a running workflow with the Workflow Terminate page process.
- Suspending a Workflow Instance
  Suspend a running workflow with the Workflow Suspend page process.
- Resuming a Workflow Instance
  Resume a suspended workflow with the Workflow Resume page process.
- Retrying a Workflow Instance
  Retry a faulted workflow with the Workflow Retry page process.

20.6.1 Starting a Workflow Instance

Start a workflow instance with the Workflow Start page process.

For APEX to create tasks, you must define a **Workflow Process Type** plugin, and add it to a page process. When the page process runs, APEX creates a new workflow instance based on the workflow definition. In order to use this process, at least one workflow definition must exist as a shared component in the application.

To start a workflow instance:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process must occur. Right-click and select **Create Process**.
3. Under Identification:
   a. Name - Specify the name of the process.

   b. Type - Select **Workflow**.

4. Under Settings:
   a. Type - Select **Start**.
   b. Definition - Select the workflow definition.
5. Bind the Workflow Parameters to values.

   These values can be Static Content, Page Item (with optional format mask), SQL Query, Expression, Function Body, Default, and Null. You only need to assign values to required parameters.
6. (Optional) Under Settings, specify the **Details Primary Key** by selecting the page item that contains the value of the Primary Key of the Source Table specified in the corresponding workflow version.
7. (Optional) Under Settings, specify the **Workflow ID Item** - If the ID of the Workflow Instance that gets created must be returned into a page item, specify the page item here.
8. Configure the rest of the Process.

   For more information, see Creating Page Processes.
9. Set the Page Process Parameters.

   If the Workflow Definition has parameters specified, these appear in the Process Tree on the page. Parameters can be set to static values or page items. If a parameter is not required, it can be left as `Null`. If a parameter is marked `Required` in the Workflow Definition, it must be set in the process. If a required parameter is not set, there is an error when saving the page.
10. Click **Save**.

20.6.2 Terminating a Workflow Instance

Terminate a running workflow with the Workflow Terminate page process.

Terminating a workflow that was invoked from another workflow also terminates the parent workflow if the parent workflow is configured to wait for the current workflow to complete.

To terminate a workflow instance:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Under Identification:
   a. Name - Specify the name of the process.
   b. Type - Select **Workflow**.
4. Under Settings:
   a. Type - Select **Terminate**.
   b. Workflow ID Item - Specify the page item that contains the workflow instance ID.
5. Configure the rest of the Process.

   For more information, see Creating Page Processes.

6. Click **Save**.

20.6.3 Suspending a Workflow Instance

Suspend a running workflow with the Workflow Suspend page process.

To suspend a workflow instance:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Under Identification:
   a. Name - Specify the name of the process.
   b. Type - Select **Workflow** .
4. Under Settings:
   a. Type - Select **Suspend**.
   b. Workflow ID Item - Specify the page item that contains the workflow instance ID.
5. Configure the rest of the Process.

   For more information, see Creating Page Processes.
6. Click **Save**.

20.6.4 Resuming a Workflow Instance

Resume a suspended workflow with the Workflow Resume page process.

To resume a workflow instance:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Under Identification:
   a. Name - Specify the name of the process.
   b. Type - Select **Workflow**.
4. Under Settings:
   a. Type - Select **Resume**.
   b. Workflow ID Item - Specify the page item that contains the workflow instance ID.
   c. (Optional) Activity Static ID Item - Specify the static ID of the activity where the workflow resumes.

5. Configure the rest of the Process.

   For more information, see Creating Page Processes.

6. Click **Save**.

- Resuming Suspended Workflows at Any Activity
  Learn about resuming suspended workflows at any activity.

20.6.4.1 Resuming Suspended Workflows at Any Activity

Learn about resuming suspended workflows at any activity.

A suspended workflow can be resumed at any activity in the workflow by using `APEX_WORKFLOW.RESUME` and specifying the workflow instance ID and the static ID of the activity the workflow should resume with. The default is the current activity, which is the activity where the workflow was suspended.

If the workflow is resumed at a different activity than the current activity, the current activity is terminated and the workflow resumes at the specified activity.

Example 20-1 Exposing the Workflow Activity ID in the Workflow Admin Console

To expose the workflow activity ID in the Workflow Admin Console, add a new page item `P<X>_ACTIVITY_STATIC_ID` of type **Select List** in the Dialog Footer of the page with the following SQL query:

```
select wa.name      as name,
       wa.static_id as value
  from apex_appl_workflow_activities wa,
       apex_appl_workflow_versions   wv,
       apex_workflows                wf$
 where wf$.workflow_id          = :P<X>_WORKFLOW_ID
   and wf$.workflow_version_id  = wv.version_id
   and wv.version_id            = wa.version_id
   and wv.workflow_static_id    = wa.workflow_static_id
```

and then modify the **Resume** processing element to take the page item value of `P<X>_ACTIVITY_STATIC_ID` for item **Activity Static ID**.

With this, a business administrator can select the static ID of the activity from a dropdown list and click **Resume** to resume the workflow at the specified activity.

20.6.5 Retrying a Workflow Instance

Retry a faulted workflow with the Workflow Retry page process.

To retry a workflow instance:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Under Identification:

   a. Name - Specify the name of the process.
   b. Type - Select **Workflow**.
4. Under Settings:
   a. Type - Select **Retry**.
   b. Workflow ID Item - Specify the page item that contains the workflow instance ID.
5. Configure the rest of the Process.

   For more information, see Creating Page Processes.
6. Click **Save**.

20.7 Viewing the Workflow Instance ID in Debug Messages

Display the workflow instance ID in Debug Messages when viewing workflow debug reports.

To view workflow debug messages, the workflow must have a **Debug Level** set. You can specify the debug level in the workflow version. For more information on setting the debug level, see Creating a Workflow Version.

To view debug messages for a specific workflow instance:

1. On the Workspace home page, click **App Builder**.
2. Select the desired application.
3. Click **Utilities**.
4. On the Utilities page, click **Debug Messages**.
5. Click **Actions**, **Columns**.

   The Select Columns window appears.
6. In the Do Not Display section, select **Workflow Instance**.
7. Click **Move** to move Workflow Instance to Display in Report.
8. Click **Apply**.

The workflow instance ID now displays in the report. For more information on working with debug messages, see Utilizing Debug Mode.

20.8 About Workflow Reports

The Workflows page shows a summary of all workflows in the application. Click on the workflow name to open Workflow Designer.

- Viewing the Workflow Utilization Report
  The Workflow Utilization Report is available in Shared Components, Workflows.
- Viewing the Workflow History Report
  The workflow history report is available in Shared Components, Workflows.

20.8.1 Viewing the Workflow Utilization Report

The Workflow Utilization Report is available in Shared Components, Workflows.

To view the workflow utilization report:

1. Select the application.

2. Navigate to the Workflow page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Workflows**.
3. Select **Utilization**.

Utilization shows all pages and page components where workflows are in use.

20.8.2 Viewing the Workflow History Report

The workflow history report is available in Shared Components, Workflows.

To view the workflow history report:

1. Select the application.
2. Navigate to the Workflow page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Workflows**.
3. Select **History**.

History shows recent modifications made to workflows.

20.9 Workflow Instance Retention

Learn about workflow instance retention logic and workflow retention settings.

An application can create complex workflows with several activities. This means workflow instance data can grow exponentially, causing space and performance overloads. Periodically purging workflow data prevents overloading the system.

Workflow Instance Retention Logic

A `DBMS_SCHEDULER` job runs on a daily basis to check all workflow instances. When `DBMS_SCHEDULER` runs, it:

- keeps workflow instances in Active, Suspended, and Faulted states
- purges completed workflow instances after 30 days, unless the retention period is set to keep workflow instances for a longer period of time
- purges canceled workflows
- purges terminated workflows

> **Note:**
>
> Terminated workflows do not have a retention period, and only remain in the system until the daily `DBMS_SCHEDULER` job runs. This means terminated workflows exist in the workflow instance table for less than twenty-four hours.

Workflow Retention Settings

The workflow retention policy determines how long to keep completed workflows. The default workflow retention period is thirty days and the maximum retention period is one hundred days.

An instance administrator can configure the workflow retention period in Instance Settings. For more information, see Configuring Workflow Settings in *Oracle APEX Administration Guide*.

20.10 Workflow Substitution Strings

Lists available workflow substitution strings.

Substitution Strings are used to pass information about a workflow to an Oracle APEX page.

You can use workflow substitution strings to pass information about a workflow instance to an APEX page. For example, you might want to show different regions in the workflow details page depending on the workflow instance state.

Substitution String Syntax	Description
`APEX$WORKFLOW_ACTIVITY_ID`	The ID of the workflow activity instance currently being executed by the workflow
`APEX$WORKFLOW_CREATED_ON`	The timestamp for workflow creation in `YYYY-MM-DD"T"HH24:MI:SS.FF3TZH:TZM` format.
`APEX$WORKFLOW_DETAIL_PK`	The primary key value of the system of records that are the source for this workflow. The source is defined in the workflow versions. `APEX$WORKFLOW_DETAIL_PK` can be used as a bind variable in any PL/SQL query used by workflow activities, variables, and participants. When used as a bind variable, the syntax is `:APEX$WORKFLOW_DETAIL_PK`.
`APEX$WORKFLOW_ID`	The workflow instance ID.
`APEX$WORKFLOW_INITIATOR`	The user who initiated the workflow. This defaults to the logged-in user who runs the application that starts a new workflow.
`APEX$WORKFLOW_STATE`	The workflow instance's current state.

20.11 Workflow Views

Learn about workflow metadata views and workflow runtime views.

Workflow **metadata views** provide information about the workflow definition at design time. The metadata views represent the workflow model in workflow designer, including the activities, parameters, variables, and participants of a workflow definition.

Workflow **runtime views** contain information about specific workflow instances. You can use workflow runtime views to query the current state of an active workflow instance, or to query the history of a completed workflow instance.

- Workflow Metadata Views
  Learn about workflow metadata views.
- Workflow Runtime Views
  Learn about workflow runtime views.

20.11.1 Workflow Metadata Views

Learn about workflow metadata views.

View Name	Description
APEX_APPL_WORKFLOWS	Workflow definitions, including static ID of workflow.
APEX_APPL_WORKFLOW_VERSIONS	Workflow definition versions.
APEX_APPL_WORKFLOW_ACTIVITIES	Workflow activity definitions.
APEX_APPL_WORKFLOW_TRANSITIONS	Workflow transitions between activities.
APEX_APPL_WORKFLOW_BRANCHES	Workflow branches going out of a switch activity.
APEX_APPL_WORKFLOW_VARIABLES	Workflow variable definitions.
APEX_APPL_WORKFLOW_ACT_VARS	Workflow activity variable definitions.
APEX_APPL_WORKFLOW_PARAMS	Workflow parameter definitions.
APEX_APPL_WORKFLOW_PARTICIPANT	Workflow participant definitions.
APEX_APPL_WORKFLOW_COMP_PARAMS	Workflow component parameters for Page Designer.

20.11.2 Workflow Runtime Views

Learn about workflow runtime views.

View Name	Description
APEX_PURGEABLE_WORKFLOWS	Purgeable workflow instances with references to the workflow definition.
APEX_PURGEABLE_WF_ACTIVITIES	Purgeable workflow activity instances with references to the workflow activity definition.
APEX_PURGEABLE_WF_VARIABLES	Purgeable workflow variable values with information about the variable value static ID, data type, and format mask.
APEX_PURGEABLE_WF_PARTICIPANT	Purgeable workflow participant values.
APEX_PURGEABLE_WF_AUDIT	Purgeable workflow audit log for workflow instances.
APEX_WORKFLOWS	Workflow instances with references to the workflow definition.
APEX_WORKFLOW_ACTIVITIES	Workflow activity instances with references to the workflow activity definition.
APEX_WORKFLOW_PARAMETERS	Workflow parameter values with information about the parameter value static ID, data type, and format mask.
APEX_WORKFLOW_VARIABLES	Workflow variable values with information about the variable value static ID, data type, and format mask.
APEX_WORKLFOW_ACTIVITY_VARS	Workflow activity variable values with information about the activity variable value static ID, data type, and format mask.
APEX_WORKFLOW_PARTICIPANTS	Workflow participant values.
APEX_WORKFLOW_AUDIT	Workflow audit log for workflow instances.

20.12 About Workflow Keyboard Shortcuts

Lists workflow keyboard shortcuts.

Workflow Designer Keyboard Shortcuts

To view keyboard shortcuts in Workflow Designer, open Workflow Designer and press `Alt+Shift+F1` or do a spotlight search for `keyboard shortcuts`.

Action	Keys
Change selection mode in diagram: activity, connection, or both	Shift+C
Deselect activity or connection in diagram	Esc
Go to Diagram Builder	Control+/, D
Go to Gallery	Control+/, G, A
Go to Help	Windows: Alt+F1 Mac: Option+F1
Go to Messages	Ctrl+/, M
Go to Property Editor	Ctrl+/, E
Go to Used Shared Components	Ctrl+/, S
Go to Workflow Tree	Ctrl+/, W
Keyboard Shortcuts	Windows: Alt+Shift+F1 Mac: Option+Shift+F1
Move diagram's activity or connection down	Down arrow
Move diagram's activity or connection up	Up arrow
Move diagram's activity or connection left	Left arrow
Move diagram's activity or connection right	Right arrow
Open bottom menu for selected activity in diagram	K
Open left menu for selected activity in diagram	J
Open menu for target at end of connection	L
Open menu to insert activities into the connection	I or K
Open right menu for selected activity in diagram	L
Open top menu for selected activity in diagram	I
Page Search	Ctrl+/, F
Redo	Ctrl+Y
Remove selected activity or connection from diagram	Delete or Backspace
Reset zoom for diagram	Ctrl+0
Restore/Expand	Windows: Alt+F11 Mac: Option+F11
Save	Windows: Alt+F7 Mac: Option+F7
Scroll diagram area up	W
Scroll diagram area left	A
Scroll diagram area down	S
Scroll diagram area right	D
Search	Ctrl+Quote (')

Action	Keys
Select next activity or connection in diagram	Shift+Up arrow Shift+Right arrow
Select previous activity or connection in diagram	Shift+Down arrow Shift+Left arrow
Show activity or connection menu in diagram	Shift+F10
Undo	Ctrl+Z
Zoom in diagram	Ctrl+Plus sign (+)
Zoom out diagram	Ctrl+Minus sign (-)

Workflow Diagram Keyboard Shortcuts

Action	Keys
Scroll diagram area up	Up arrow
Scroll diagram area left	Left arrow
Scroll diagram area down	Down arrow
Scroll diagram area right	Right arrow
Reset zoom	Ctrl+0
Zoom in diagram	Ctrl+Plus sign (+)
Zoom out diagram	Ctrl+Minus sign (-)

20.13 Frequently Asked Questions

Reference a list of frequently asked questions for workflows.

Workflow Designer

Topic	Question	Answer
Switch Activities	For more complex conditions, should I use a true/false type switch and a PL/SQL expression or function body that returns a boolean result?	You can create more complex conditions by using an if-elsif-else switch, where there is flexibility to specify different types of conditions on each of the switch branches. The true/false type switch is the simplest condition because the only available results are true, false, or null.
Wait Activities	Why can't I see the deadline section for wait activities in the property editor?	Wait activities have their own timeout attributes. The generic deadline section is not visible.

Topic	Question	Answer
Error Connections	How can I effectively use error connections in my workflow model?	Consider an order management workflow with an Invoke API activity that fetches the order number from the `ORDERS` table before proceeding to the next activity in the workflow. If the query returns no records, the Invoice API activity throws a `no_data_found` exception, and the workflow status transitions to faulted. If you configure an error connection, you can avoid this message. In the settings for the error connection, select = as the condition, and the SQL Error Code for `no_data_found` (`01403`) as the value. To handle this error, you can set up an Invoke API activity to create the missing order and add it to the `ORDERS` table. This method efficiently handles the error, and prevents the workflow from going into a faulted state.
Workflow and Activity Variables	How do I decide when to use workflow variables and when to use activity variables?	Workflow variables are alive through the lifetime of a workflow instance. Their values can be accessed and altered by all the workflow activities. Activity variables only live during the execution of a particular activity. Typically, activity variables fetch additional data that may be needed by the activity during its execution.

Workflow Runtime

Topic	Question	Answer
Application Reimport/Migration	Why are active workflow instances suspended after reimporting or moving the application from a development environment to a production environment?	During application reimport, running instances of workflows are suspended to ensure that no workflow instance inadvertently moves to a faulted state due to temporary unavailability of the workflow definition. A workflow administrator must resume these suspended instances either through the Workflow Console or by using the `APEX_WORKFLOW.resume()` API.

Topic	Question	Answer
Tasks and Workflows - Deadlines and Expiration	When a workflow contains a task activity with a deadline specified in the workflow and a deadline specified in the task definition, which deadline is honored?	The workflow activity deadline gets precedence over the task deadline. If there is no workflow activity deadline, the workflow uses the deadline in the task definition.
Tasks and Workflows - Deadlines and Expiration	Tasks can expire and be auto-renewed. Does this also apply to tasks created using workflow?	At runtime, if the task fails to complete by the due date, the expiration policy in the task definition activates, and the task is automatically renewed or expired accordingly. If the expiration policy in the task definition is `NONE`, the task still expires because it is a workflow-generated task, and it cannot remain active while governed by workflow timeout. Once the task expires, control goes back to the workflow engine, and the engine follows the timeout connection in the workflow definition. If there is no timeout connection, the workflow transitions to a faulted state.
Tasks and Workflow - Deadlines and Expiration	If a task is created using workflow, can the business administrator of the task manually renew or update the due date?	No. Tasks created by a workflow can only be auto-renewed based on their expiration policy settings. The business administrator cannot manually renew or update the due date of a task because the due date is governed by the workflow timeout.
Tasks and Workflow - Errored or Canceled Tasks	What happens to the workflow if its task becomes errored or canceled?	When the initiator or business administrator cancels a task, the workflow runtime engine receives no task outcome, and continues executing the next activity in the workflow. If the next activity in the workflow is a switch based on the outcome of the task, the switch activity transitions to faulted because the task-based workflow activity has no outcome. When a task transitions to **errored**, control returns to the workflow, which transitions to **faulted**. The workflow administrator can retry a faulted workflow after fixing the underlying cause of the error.
Tasks and Workflow - Errored or Canceled Tasks	What happens when a workflow is terminated while its tasks are still active and not yet completed?	The tasks move to a Canceled state if the associated workflow is terminated.

Topic	Question	Answer
Tasks and Workflow - Using Approval Task Outcomes in Workflow	How can the outcome of an approval task defined as a workflow activity be used by other activities in the workflow?	For every approval activity in a workflow, two associated workflow variables exist: `TASK_OUTCOME` and `APPROVER`. In the activity settings, you can set the task outcome and approver attributes to these variables. After the actual owner completes the task, the workflow runtime engine completes the activity and sets the workflow variable `TASK_OUTCOME` to approved or rejected. The activity also sets the `APPROVER` variable to the actual owner. Subsequent workflow activities can use these workflow variables. For example, set a switch activity that follows the human task activity to **Check Workflow Variable**, specify the variable as `TASK_OUTCOME` with a condition of `= APPROVED` or `= REJECTED`, and use that variable to branch the workflow. **Note:** If there is more than one approval task activity in the workflow, the subsequent activities do not

Topic	Question	Answer
		have workflow variables automatically created for them. You can choose to re-use the same workflow variables to store the task outcome and actual owner or you can manually create new workflow

Topic	Question	Answer
		variables if each human task activity outcome must be stored in a different workflow variable.
Tasks and Workflow - Using Approval Task Outcomes in Workflow	I accidentally rejected an approval task instead of accepting it. Can the workflow administrator fix this in the associated workflow?	Yes, the workflow administrator can fix this if the workflow has a wait activity right after the approval task activity. After an approval task is rejected, the `TASK_OUTCOME` workflow variable value is `Rejected`. During the wait activity, the workflow administrator can suspend the workflow from the Admin Console Details page. The administrator can then expand the variables section on that page, click **Edit** next to the `TASK_OUTCOME` variable, and manually update the value to `Approved`. When this is done, the workflow administrator can resume the workflow. Note: Only workflow administrators can update workflow variables, and only when the workflow is suspended or faulted.
Workflow Subprocess	Can a workflow be used as an activity in another workflow?	Yes. Use the **Invoke Workflow** activity type. See Adding an Invoke Workflow Activity.

Topic	Question	Answer
Workflow - Retry	Can auto-retry be configured for workflow activities?	No. Workflow activities that transition to **faulted** must be manually retried by either: • the workflow administrator clicking **Retry** in the Workflow Console Details Page • calling `apex_workflow.retry()` and passing the workflow instance ID Retrying a workflow always re-executes the activity where the workflow faulted, and then continues the rest of the workflow execution chain. There is no limit to the number of times a workflow can be manually retried.
Workflow - Retry and Resume	What is the difference between **Retry** and **Resume** for a workflow?	**Retry** is only applicable to faulted workflows, and can be performed by both the workflow administrator and the workflow owner. **Resume** is only applicable to suspended workflows, and can be performed by the workflow administrator.
Workflow - Resume	Can I resume workflows at any activity?	Yes, workflows can be resumed at any activity, but this requires additional developer configuration. For more information, see Resuming Suspended Workflows at Any Activity.
Workflow Versions	My application has a workflow with two versions: 1.0 is the active version, and 1.1 is the development version. When I run the application from App Builder, can I specify in the workflow page process which version of the workflow APEX uses?	The workflow page process only takes the workflow definition as input, not the version. Running an application from App Builder implies the application is running in a developer session, and APEX uses the development (1.1) version of the workflow. If you run the application without any active developer (App Builder) sessions, then APEX uses the active (1.0) version of the workflow.

Topic	Question	Answer
Workflow Versions	I am running the application from App Builder. My application has a workflow with only one version, 1.0, which is in development. When I run the application from App Builder, **Suspend** and **Edit** link for workflow variables are available for both the workflow owner and the workflow administrator. However, the documentation mentions that only the workflow administrator should have access to these operations. Why can the owner see them?	When running the development version of a workflow, APEX does not perform authorization checks. APEX only performs authorization checks for the active version of a workflow.
Workflow Versions	While running an application containing an active workflow version, I started a few workflow instances. Since then, I have added two new workflow owners and one new workflow administrator to the workflow. How can I reflect the new participants in the currently running workflow instances?	Call `apex_workflow.refresh_participants` from a valid APEX session. This API re-evaluates the participants in the workflow definition and updates the currently running workflow instances.
Workflow Versions	I have moved my development workflow version to active. I want to delete all the workflow instances I created while the workflow was still in development. Is this possible?	Yes. Call `apex_workflow.clear_test_data`, passing the application ID and the static ID of the workflow definition. This deletes all workflow instances created while running the development version of the workflow.
Workflow Versions	I have moved my development workflow version to active. I had a previously active version, which is inactive now. What happens to the workflow instances created based on the previously active version? Are they terminated or deleted?	No. The workflow instances created from the previously active workflow version all continue to run until complete. However, you cannot start any new instances based on the previously active version of the workflow. All new workflow instances are based on the current active workflow definition.

Topic	Question	Answer
Workflow Debugging	My workflow instance faulted at an activity. I did not set any debug level for the workflow version. Can I still enable debugging and retry the activity?	Yes. Call `apex_workflow.set_log_level()`, passing the workflow instance ID and the log level you want (`apex_debug.c_log_level_app_trace`). This enables logging for the specific workflow instance. You can check the log entries by querying: `select message from apex_debug_messages where workflow_instance_id =workflow instance ID order by message_timestamp asc;`

20.14 About Tasks

Use tasks to create items for users to approve or action.

Human Tasks in APEX are items that need to be actioned by people, rather than activities performed by a script or automation. There are two types of tasks available:

- **Approval Tasks**, for tasks that can be approved or rejected. The user who owns the task evaluates the information in the task, and then either **Approves** or **Rejects** the task. A task that asks a manager to approve or deny a laptop request for an employee is an approval task.
- **Action Tasks**, for tasks that need to be actioned. The user who owns the task performs the action requested in the task, and then **Completes** the task. A task that asks an employee to fill in a self-evaluation is an action task.

Because the task component contains a data model, when you configure tasks, you focus on the details and customization for a specific type of task. You can set up task owners and administrators at design time or determine them dynamically at runtime based on data related to the task.

APEX provides the following building blocks for tasks:

- **Task Definition** - A shared component used to configure task parameters, participants, actions, and due dates.
- **Unified Task List** - A page type in the Create Page Wizard, used to create a summary of tasks that functions like an inbox.
- **Task Details Page** - A page that shows details for a specific task, which can include metadata, history, comments, and actions.
- **Human Task - Create** and **Human Task - Manage** - Page processes that create and act on task instances in your pages.

- the **APEX_HUMAN_TASK** API package - Used for creating, retrieving, and managing tasks programmatically.

By customizing these building blocks, you can create custom tasks specific to your application or shared across many applications. Once you create the task definition, unified task list, and task details page, you add the task definition to a page as a **Human Task - Create** page process. When the process activates, the task engine creates the task from the task definition.

Users can view and take action on their tasks in the unified task list. Based on the user action, the **task state** changes. For more on task states, see About Task States and Transitions.

> **Tip:**
>
> By default, approval tasks do not allow the user who initiated the task to complete the task. However, you can change this behavior in the task definition settings.

- About Task States and Transitions
  Learn about the task lifecycle in the Oracle APEX approval component.
- Example: Approval Tasks and Action Tasks
  View the approvals component and action tasks functionality in the **Sample Workflows, Approvals, and Tasks** app.

20.14.1 About Task States and Transitions

Learn about the task lifecycle in the Oracle APEX approval component.

The task **lifecycle** is the path a task takes from the time the task is created to the time that the task is complete. There are three lifecycle stages: **Initial**, when a task is created, **Intermediate**, as the task moves through various states, and **Final**, when a task is complete.

A **state** is the current location of the task in the task lifecycle. A **transition** refers to the movement of a task between states. Transitions can occur as part of the normal task lifecycle, or they can occur due to an exception or a user interruption.

An **errored** state occurs when there is a non-recoverable error during the execution of the task, including:

- an underlying database error
- a code bug
- an invalid Task Definition
- an error during execution of the action specified for the Task Definition

A **canceled** state occurs when the initiator or the business administrator of the task explicitly cancels the task.

An **expired** state occurs when the due date arrives, the task is not complete, and either the task does not have a renewal policy or the maximum number of renewals has already been reached.

Task states and possible transitions are available in the following diagram, or in the Task States and Transitions table.

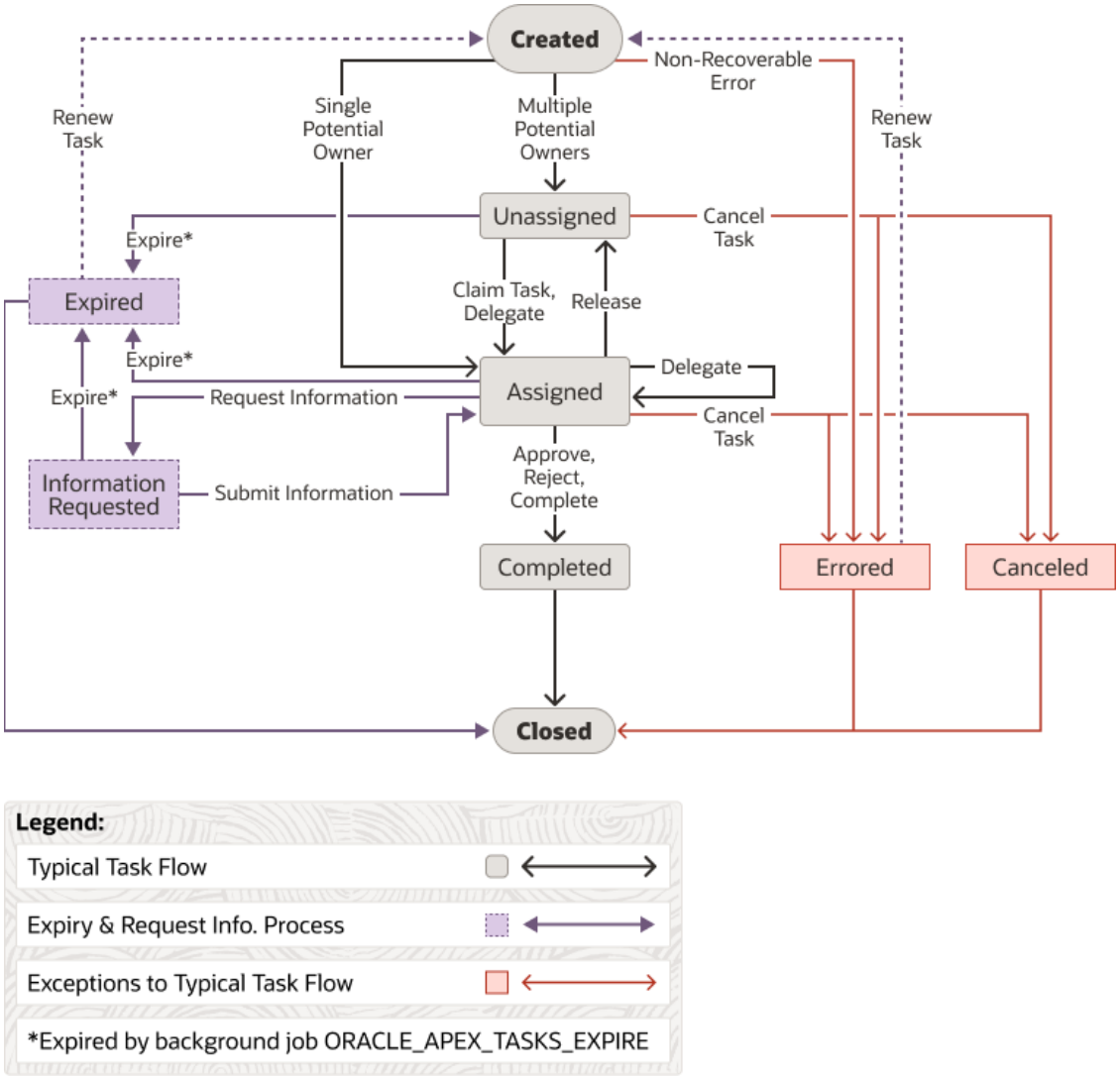


Table 20-1 Task States and Transitions

Task State	Lifecycle Stage	Category	Possible Transitions
Created	Initial	Normal	• **Unassigned**, when there are multiple potential owners • **Assigned**, when there is one potential owner • **Errored**, when there is a non-recoverable error during execution

Table 20-1 (Cont.) Task States and Transitions

Task State	Lifecycle Stage	Category	Possible Transitions
Unassigned	Intermediate	Normal	• **Assigned**, when a user claims the task or delegates the task • **Completed**, when a potential owner approves or rejects the task from the unified task list • **Errored**, when there is a non-recoverable error during execution • **Canceled**, when the initiator cancels the task • **Expired**, when the task is not completed by the due date the expiration policy for the task definition is either: – Expire (APEX does not create a new task) – Renew (APEX does create a new task as long as the maximum number of renewals is not yet reached)

Table 20-1 (Cont.) Task States and Transitions

Task State	Lifecycle Stage	Category	Possible Transitions
Assigned	Intermediate	Normal	• **Completed**, when the owner approves, rejects, or completes the task • **Unassigned**, when the owner releases the task • **Info Requested**, when the owner requests information from the initiator • **Assigned**, when a user delegates the task to another potential owner • **Errored**, when there is a non-recoverable error during execution • **Canceled**, when the initiator cancels the task • **Expired**, when the task is not completed by the due date the expiration policy for the task definition is either: – Expire (APEX does not create a new task) – Renew (APEX does create a new task as long as the maximum number of renewals is not yet reached)

Table 20-1 (Cont.) Task States and Transitions

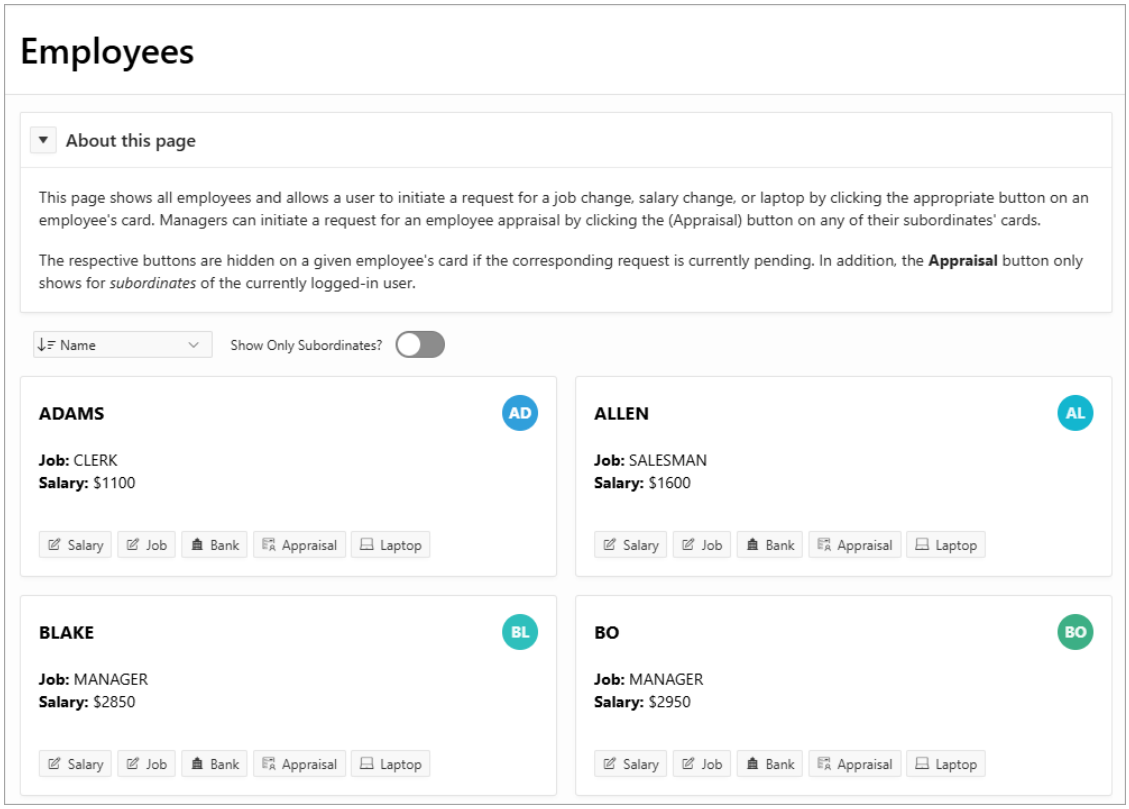
Task State	Lifecycle Stage	Category	Possible Transitions
Info Requested	Intermediate	Normal	• **Assigned**, when the initiator submits the requested information • **Expired**, when the task is not completed by the due date the expiration policy for the task definition is either: – Expire (APEX does not create a new task) – Renew (APEX does create a new task as long as the maximum number of renewals is not yet reached)
Completed	Final	Normal	None
Errored	Final	Exception	• **Renewed**, when the Business Administrator manually renews the task. APEX creates a new task based on the underlying task definition.
Canceled	Final	User Interrupted	None
Expired	Final	Normal	None • If the task's Expiration Policy is set to Renew and the maximum number of renewals is not yet reached,APEX creates a new task. • If the task's Expiration Policy is set to Expire, or if it is set to Renew but the maximum number of renewals is already reached, APEX does not create a new task. The business administrator can manually renew the task.

20.14.2 Example: Approval Tasks and Action Tasks

View the approvals component and action tasks functionality in the **Sample Workflows, Approvals, and Tasks** app.

The Sample Workflows, Approvals, and Tasks Application

Sample apps highlight specific APEX functionality and serve as a developer guide for how to use particular features. For approvals and action tasks, you can install the **Sample Workflows, Approvals, and Tasks** app from the Gallery.

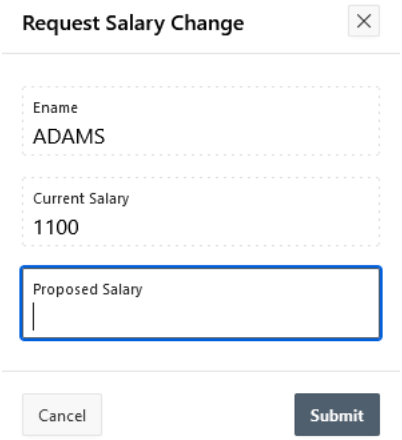


The **Employees** screen shows all employees. Users can request the following changes for an employee:

- **Salary** - Requests a change to the employee's salary. This functionality uses an **approval task** to review and approve the change and a customized task details page.
- **Job** - Requests a change to the employee's job. This functionality uses an **approval task** to review and approve the change and a standard task details page.
- **Bank** - Requests a change to the employee's bank account. This functionality uses an **approval task** that the initiating employee can approve.
- **Appraisal** - Begins the process of an employee's appraisal. This functionality uses a **workflow** with **action tasks** for the employee, the employee's manager, and, if applicable, a VP.
- **Laptop** - Requests the purchase of a new laptop for the employee. This functionality uses a **workflow** with an **approval task** and an **action task** that is programmatically completed when the laptop is delivered.

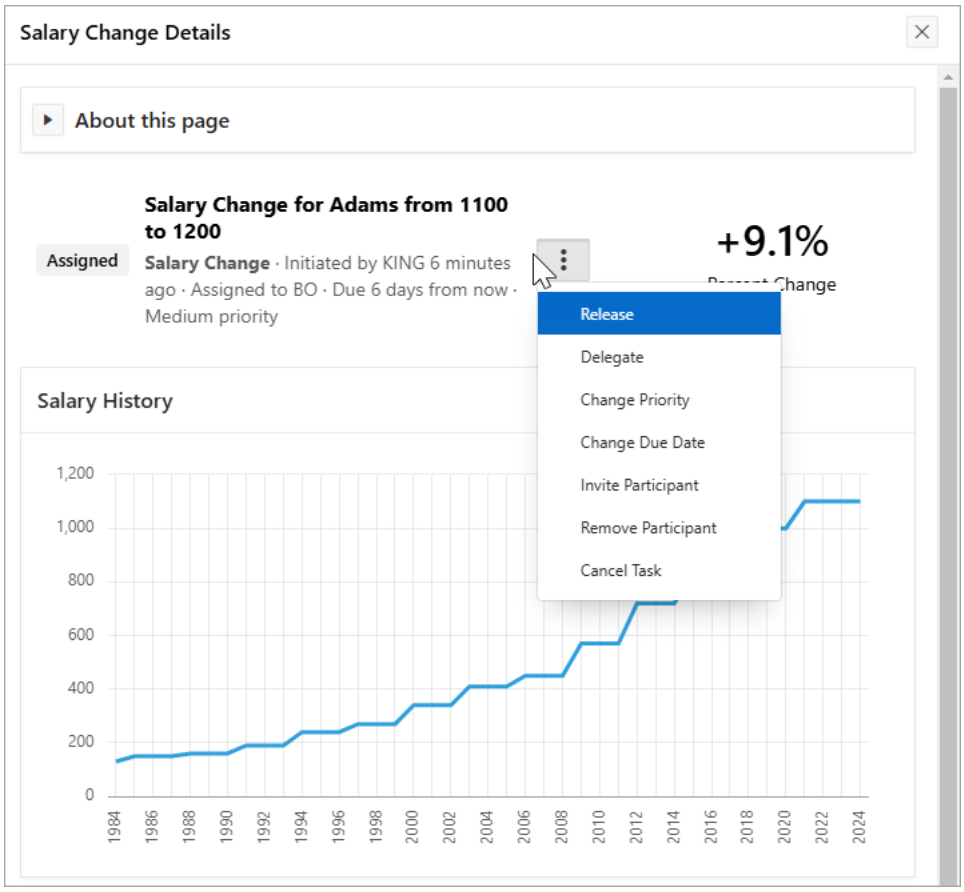
Request Salary Change

The **Request Salary Change** screen allows users to view the current salary for an employee and enter a proposed salary.



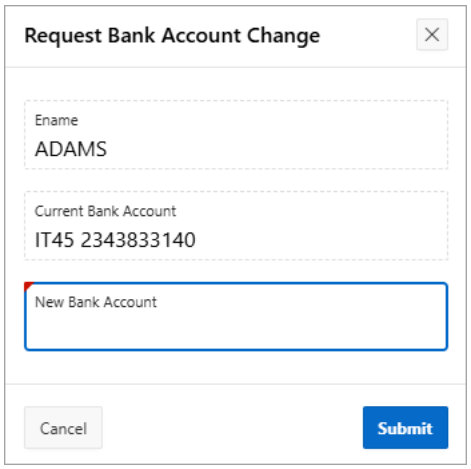
Once a user requests a salary change, the **Salary** button on the Employees screen is no longer visible until the salary change request is approved or rejected.

The **Salary Change Details** screen is a customized task details page that displays additional information about the specific task instance, including a graph of the employee's salary history, and the percent change that the requested salary change represents.



Request Bank Account Change

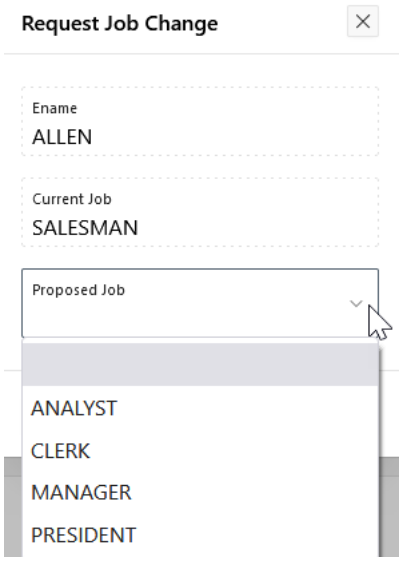
The **Request Bank Account Change** screen allows users to view the current bank account information for an employee and enter a new bank account number.



The approval task definition associated with the bank account change has the **Initiator Can Complete** switch set to `ON`, which means that the initiator can approve their own task.

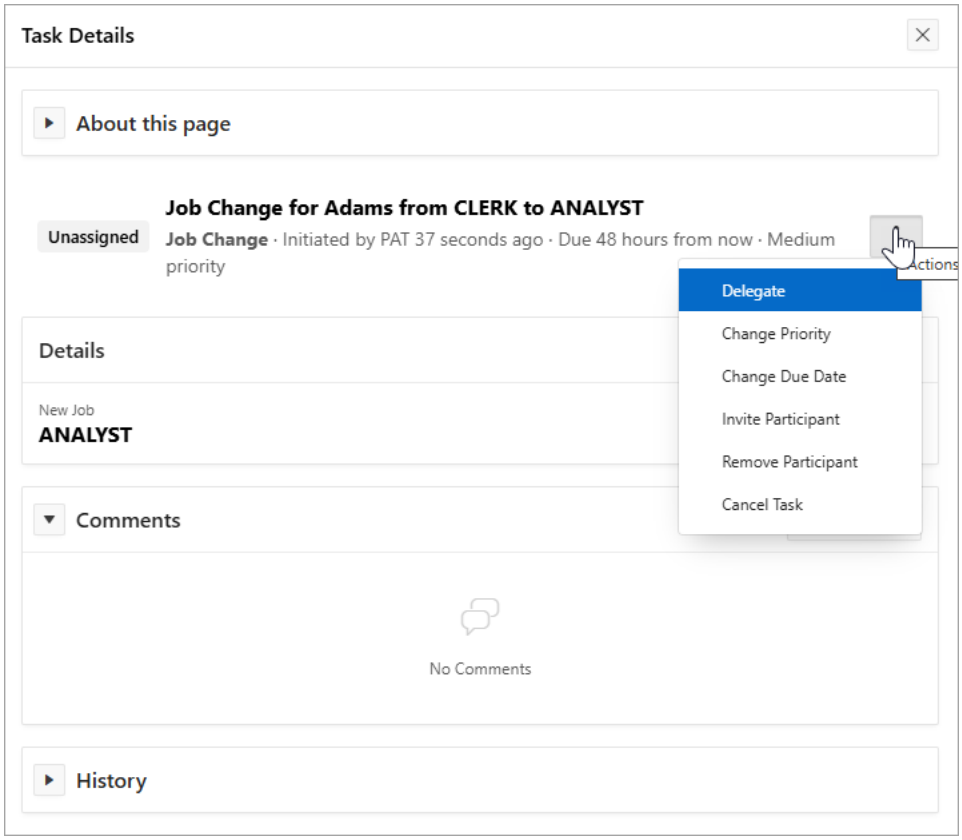
Request Job Change

The **Request Job Change** screen allows users to view the current job for an employee and select a proposed job.



Once a user requests a job change, the **Job** button on the Employees screen is no longer visible until the job change request is approved.

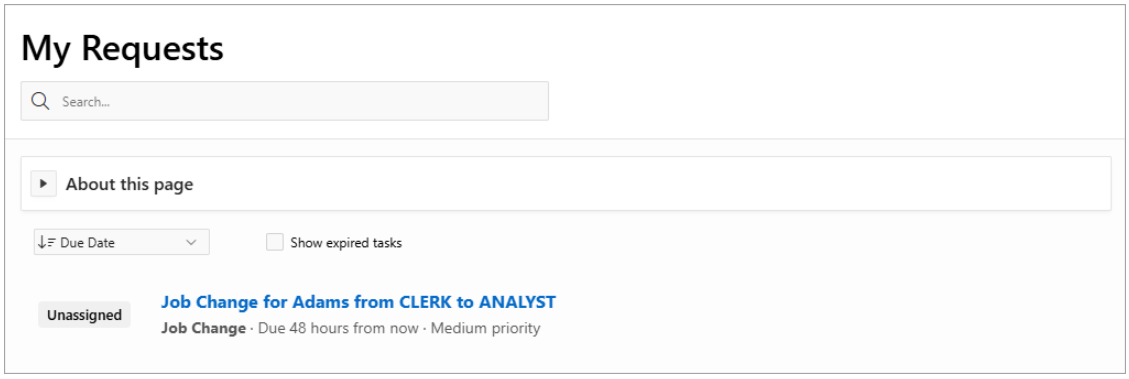
In contrast to the **Salary Change Details** screen, the **Task Details** screen for the job change task is not customized.



Unified Task Lists

There are three unified task lists included as part of the sample app:

1. My Tasks, based on the **My Tasks** context
2. My Requests, based on the **Initiated by Me** context
3. Task Administration, based on the **Admin Tasks** context



Each unified task list shows tasks relevant to the user currently logged in, allowing them to manage, track, or administer both approval tasks and action tasks.

> **Tip:**
>
> To view this example, install the sample app, **Sample Workflows, Approvals, and Tasks**. To learn more, see Installing Apps from the Gallery.
>
> Once you install **Sample Workflows, Approvals, and Tasks**, you can reference the **More Information** page of the sample app for an overview of workflow features, approval and action tasks features, and a discussion of how to use the sample app.

20.15 Managing Tasks

Manage tasks by configuring task definitions, task details, and unified task lists. Use page processes to create approval tasks or action tasks.

- Managing Task Definitions
  Task Definitions set up the structure for a particular set of tasks, including participants, parameters, and actions.
- Managing Task Details
  Task Details display when you open the task instance.
- Managing Unified Task Lists
  The Unified Task List allows workspace users to view and manage their tasks.

20.15.1 Managing Task Definitions

Task Definitions set up the structure for a particular set of tasks, including participants, parameters, and actions.

- About Task Definitions
  Task definitions include task type, parameters, participants, parameters, and actions.
- Creating a Task Definition
  Create a Task Definition from the Shared Components page, under the Workflows and Automations heading.
- Copying a Task Definition
  Copy an existing Task Definition from the Shared Components page, under the Workflows and Automations heading.
- Editing a Task Definition
  Edit Task Definition details from the Shared Components page, under the Workflows and Automations heading.
- Viewing the Task Definition Utilization Report
  The Task Definition Utilization Report shows all pages and page components where Task Definitions are used in a specific application.
- Viewing the Task Definition History Report
  The Task Definition History Report shows recent modifications made to Task Definitions.

20.15.1.1 About Task Definitions

Task definitions include task type, parameters, participants, parameters, and actions.

Task Definitions

Task Definitions contain information about tasks including deadlines, expiry settings, and notification settings. Task definitions exist on the application level in the Shared Components section, and are made up of task settings, participants, parameters, and actions:

- **Task Type** - Defines whether the task is an approval task or an action task.
- **Settings** - The general configuration options for the task, including the default priority for tasks, the task detail page URL, and the task deadline settings.
- **Task Definition Participants** - Users that have permission to act on individual tasks. Participants can be:
  - Task Initiators, who can start new tasks or provide more information about existing tasks.
  - Potential Owners, who can claim unassigned tasks. Tasks can have multiple potential owners.
  - Actual Owners, who can make changes to their assigned tasks, including requesting more information about the task, approving, rejecting, or completing the task, and updating the parameters of the task.
  - Business Administrators, who can renew expired tasks, reassign tasks to new owners, remove existing potential owners, cancel tasks, or update the task priority.

> **Note:**
>
> You can define a **vacation rule procedure** that determines how to handle vacation rules for participants.

- **Task Definition Parameters** - Attributes for a task that contain information about the task. These parameters can be visible on the task details page, and you can configure whether or not a parameter is editable after the task is initiated.
- **Task Definition Actions** - Possible events that can happen when specific criteria are met. Available actions are:
  - Claim
  - Complete
  - Delegate
  - Update Comment
  - Update Priority
  - Update Parameter
  - Release
  - Cancel
  - Create
  - Request Information

- Submit Information
- Before Expire
- Expire

You can update a Task Definition any time. Updates do not change existing tasks, but are applicable to new tasks.

Vacation Rules

A **vacation rule** is a PL/SQL procedure that returns additional alternate participants. You can define a vacation rule at the task definition level or the application definition level. If you define a vacation rule at the application level, it applies to all task definitions in the application.

> **Tip:**
>
> A vacation rule defined at the task definition level overrides a vacation rule defined at the application level.

APEX evaluates vacation rules at runtime:

- **On task instance creation:** APEX executes the vacation rule to see if alternate participants are returned by the procedure for any participants of the task. If so, APEX adds the alternate participants to the list of existing task participants.
- **On task instance delegation:** APEX checks whether a vacation rule exists for the task. If so, APEX delegates the task to the first alternate participant returned by the procedure.

20.15.1.2 Creating a Task Definition

Create a Task Definition from the Shared Components page, under the Workflows and Automations heading.

To create a task definition:

1. Select an application.
2. Navigate to the Task Definitions page:
   a. On the application home page, select **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Click **Create**.

   The Create Task Definition wizard appears.
4. Specify the task definition name and define the metadata.
   a. Name - Enter a name for the new Task Definition.
   b. Type - Select `Approval Task` for tasks with an outcome, or `Action Task` for tasks with no outcome.
   c. Subject - Provide a subject for the task definition.

      For example, `Purchase Requisition for Order &ORDER_ID.`, where `ORDER_ID` is a defined parameter of the task definition. `ORDER_ID` is substituted with the parameter value on task creation.

      For more on substitution strings, see Substitution Strings and Bind Variables for Tasks.

d. Static ID - Provide a static ID for this task definition. The static ID is used when creating the task with the `APEX_HUMAN_TASK` package (`APEX_HUMAN_TASK.CREATE_TASK`).

e. Priority - Set the priority for the task. Choose a number between 1 and 5, where 1 is Urgent and 5 is Lowest. The default priority is 3 - Medium.

f. Potential Owner - The user who owns the task.

   To determine potential owners dynamically, edit the participants in the Task Definition Details. For approval tasks, the user who initiates the task cannot own that task, even if they are listed as a Potential Owner. For action tasks, the user who initiates the task can also complete the task.

g. Business Admin - The user who has administrative rights on all tasks for this task definition.

   To determine business administrators dynamically, edit the participants in the Task Definition Details.

5. Click **Create**.

   The Task Definition details page opens.

6. (Approval tasks only) Set the **Initiator Can Complete** toggle:
   - **On** - The task initiator can also approve or reject the task.
   - **Off** - (Default.) The task initiator is not allowed to approve or reject the task.

7. Set the Task Details Page Number. If a Task Details Page already exists for this application, enter the page number here. Otherwise, select **Create Task Details Page** to create a new page.

   The Task Details Page allows you to view the details of the task instance when the task is selected on the Unified Task List Report. See Editing the Task Details Page.

8. Set the Actions Source - Specify the source of the data that is evaluated when the task actions execute.

   These actions are based on tables or SQL queries. An Actions SQL Query can reference the bind variable `:APEX$TASK_PK` to access the value of the primary key in the task detail row.

9. Set **Due on Type**:
   - **Interval** - The ISO or SQL format duration that specifies when a task is due. Only positive interval values are allowed.
   - **SQL Query** - The SQL query that returns a valid due date for the task.
   - **Function Body** - The function body that returns a valid due date for the task.
   - **Expression** - The PL/SQL expression that returns a valid due date for the task.
   - **Scheduler Expression** - The Database Scheduler calendaring syntax that returns a due date. For example, `first Friday of the month at 4:00 p.m.`.

10. After setting the Due on Type, set the **Expiration Policy**:
    - **None** - After the due date, the task stays in its current state, and owners can still approve or reject the task. This is the default setting.
    - **Expire** - After the due date, the task expires. Expired tasks cannot be approved or rejected.
    - **Renew** - After the due date, the task expires, and a new task is created from the same task definition. If you select Renew, set the **Maximum Renewal Count** to specify the

maximum number of times a task is automatically renewed. A task can be renewed up to 99 times.

11. Specify the **Vacation Rule Procedure**, if applicable, in the format `[schemaName.[packageName]].procedureName`.

    The vacation rule procedure returns additional alternate participants for this task definition. It implements the following interface:

    ```
    procedure my_vacation_rule (
        p_param    in apex_human_task.t_vacation_rule_input,
        p_result  out apex_human_task.t_vacation_rule_output );
    ```

    For an example of a vacation rule procedure, see item-level help or Example: Adding a Vacation Rule Procedure.

    > **Tip:**
    >
    > A vacation rule defined at the task definition level overrides a vacation rule defined at the application level.

12. Create Participants to assign additional people to the Task Definition.
    a. Select **Add Row**.
    b. Edit the Participant Type to select **Potential Owner** or **Business Administrator**.

       For approval tasks, the user who initiates a task cannot approve their own task, even if they are listed as a Potential Owner. For action tasks, the user who initiates a task can own and complete their own task.
    c. Set the Value Type.

       SQL Query, Function Body, and Expression all allow users to be dynamically selected, and can return a single user name or a comma-separated list of multiple user names.
    d. Set the Value.
13. Set Parameters.
    a. Select **Add Row** and fill in the parameter fields.
    b. Static ID - Identifier used to reference the parameter.
    c. Label - A user-friendly name for the task parameter.
    d. Data Type - Choose `String`.
    e. Required - Specify whether the parameter is mandatory.
    f. Visible - Specify whether the parameter is visible in the task details page.
    g. Updatable - Specify whether the parameter can be updated in the Task Details page after the task is started.
    h. Comment (Optional) - Describe the parameter.

    > **Note:**
    >
    > Parameters set in the Task Definition also need to be set in the page process. See Creating Tasks from Task Definitions.

14. Select **Apply Changes** to save all existing changes, and then click on the task definition name to continue editing.
15. Add Actions.
    a. Click **Add Action**.

       The Edit Action page opens.
    b. Name - Enter a name for the task definition action.
    c. Type - Select **Execute Code**, **Send Email**, or **Send Push Notification**.

       For Send Email, you can use predefined email templates with substitution strings. For example, you can configure a notification email to the initiator when a task is approved or rejected. For more information, see Managing Email Templates.

       If you have a plug-in set up for task definitions, you can use that plug-in here. For more information, see About Creating a Task Definition Process Plug-in.
    d. Execution Sequence - Specify the execution sequence. Actions are executed in numerical order from low to high.
    e. On Event - Select the event that triggers the action.
    f. Success Message - Enter a success message that indicates the action is successfully processed. The success message is included in the task history if the Log Messages When attribute in the Error Handling section is set to `All` or `Success`.
16. Configure the Action Type.
    - Set attributes for Execute Code:
      - Location - Specify whether your PL/SQL code is executed in the **Local Database** or on a **REST Enabled SQL Service**.
      - Language - Select the language used to evaluate the code.
      - Code - Provide the PL/SQL code to be executed.
        * Column values from the task definition action query are available as bind variables by column name, such as `:EMPNO` and `:ENAME`. For more on bind variables, see Substitution Strings and Bind Variables for Tasks. You can expand the **Examples** section to see code examples.
    - Set attributes for Send Email:
      - From - The email address that displays as the sender of the email.
      - To - The email recipients.
      - Cc - The carbon copy email recipients.
      - Bcc - The blind carbon copy email recipients.
      - Reply To - The email address replies are sent to. If left blank, the From email is used.
      - Email Template - Choose an email template, and set the placeholder values. For more information, see Managing Email Templates.
      - Subject - The subject of the email.
      - Body Plain Text - The text of the email. Plain text is used if the recipient's email client does not support rich text. See the in-app help text for more details.
      - Body HTML - The HTML text of the email. See the in-app help text for more details.

  - Attachment SQL - Enter a SQL statement that returns one or more BLOBs that are attached to the email as files.
- Set attributes for Send Push Notification:
  - To - User name of the application user who receives the push notification. To receive the notification, the user must have subscribed to the application first. User name is case sensitive.
  - Title - Title of the push notification.
  - Body - Body of the push notification.
  - Link Target - URL of the page that opens when the user clicks on the push notification. This defaults to the home page of the application. You can use the `&APP_ID.`, `&SESSION.`, and `&DEBUG.` substitution strings with `&COLUMN_NAME.` syntax to set the row ID or primary key on the target page, or to set additional attribute values. For more information, on URL syntax, see Understanding f?p URL Syntax.

17. Set up Error Handling:
    a. Error Message - Enter an error message that can be added to the task log if the action errors while executing. The error message is included in the task history if the Log Messages When attribute in the Error Handling section is set to `All` or `Failure`. Use `#SQLERRM#` as a placeholder for the actual ORA error message.
    b. Stop Execution on Error - Enable if processing for the current task stops after an error. Subsequent actions are not processed, and the task goes into the `ERRORED` state.
    c. Log Messages When - Specify the actions that are logged in the task history. You can choose between None, Success, Failure, and All. When actions are logged in the history, they include the Success Message or Error Message text.
18. Finish configuring the Action.
    a. Condition Type - Select a condition type that must be met for this action to be executed.
    b. Build Option - Select a build option for this action. Build options determine whether or not components in an application are enabled.
    c. Comments - Enter developer notes here. These comments only display in App Builder and are not visible in the application.
19. Click **Create** to go back to the Task Definition.
20. Add any comments for this task definition.
21. Click **Apply Changes**.

APEX creates the task definition.

20.15.1.3 Copying a Task Definition

Copy an existing Task Definition from the Shared Components page, under the Workflows and Automations heading.

To copy a task definition:

1. Select an application.
2. Navigate to the Task Definitions page:
   a. On the application home page, select **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.

3. Click **Copy**.

   The Copy Task Definition wizard appears.

4. Specify the task definition you want to copy:

   a. Copy Source - Select This Application or Other Application.

      If you select Other Application, specify the application to copy the task definition from.

   b. Copy Task Definition - Choose the task definition to copy.

   c. New Name - Provide a name for the copied task definition

   d. New Static ID - Provide a static ID

5. Click **Copy Task Definition**.

6. APEX copies the task definition:

   Copy Task Definition copies the settings, deadline, participants, actions, and parameters from the original task, but does not copy the task details page. If the task definition uses any shared components (like email templates or Rest Data Sources), Copy Task Definition copies those shared components to the target application if they are not already there. Copy Task Definition does not copy subscriptions.

7. Click on the task definition name to edit it.

8. Set the Task Details Page Number. If a Task Details Page already exists for this application, enter the URL here. Otherwise, select **Create Task Details Page** to create a new page.

9. Click **Apply Changes** to confirm any changes in the copied task definition.

20.15.1.4 Editing a Task Definition

Edit Task Definition details from the Shared Components page, under the Workflows and Automations heading.

- Example: Editing Task Definition Participants
  Edit Task Definition Participants by accessing the Task Definition in the Shared Components page.
- Example: Adding a Vacation Rule Procedure
  Add a vacation rule by editing the Task Definition in the Shared Components page.
- Example: Editing Task Definition Parameters
  Edit Task Definition Parameters by accessing the Task Definition in the Shared Components page.
- Example: Adding and Editing Task Definition Actions
  Add and edit Task Definition Actions by accessing the task definition in the Shared Components page.

20.15.1.4.1 Example: Editing Task Definition Participants

Edit Task Definition Participants by accessing the Task Definition in the Shared Components page.

This example uses the **Sample Workflows, Approvals, and Tasks** application, which can be installed from the Gallery. In this example, suppose that Bo, one of the potential owners in the Job Change task definition, has become a business administrator for that definition.

To edit Task Definition Participants:

1. Select the Sample Workflows, Approvals, and Tasks app.
2. Navigate to the Task Definitions page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Click **Job Change**.
4. Under Participants, select the row where Participant Type is `Potential Owner` and Value is `Bo`.
5. Click **Edit**.
6. Change the Participant Type from `Potential Owner` to `Business Administrator`.
7. Click **Apply Changes**.

Bo is now a business administrator on all new Job Change tasks. Job Change tasks that already exist do not change.

20.15.1.4.2 Example: Adding a Vacation Rule Procedure

Add a vacation rule by editing the Task Definition in the Shared Components page.

> **Note:**
>
> This section describes adding a vacation rule procedure to a task definition. You can also add a vacation rule procedure to the application definition. If you add a vacation rule procedure to the application definition, the vacation rule applies to all tasks in the task definition.

> **Tip:**
>
> A vacation rule defined at the task definition level overrides a vacation rule defined at the application level.

This example uses the **Sample Workflows, Approvals, and Tasks** application, which can be installed from the Gallery. In this example, suppose that Jane, a task participant, is on leave. When Jane is on leave, Scott takes responsibility for Jane's approvals.

Before adding the vacation rule procedure, you must add the following procedure to the APEX workspace:

```
procedure alternate_job_approvers(
                p_param    in apex_human_task.t_vacation_rule_input,
                p_result  out apex_human_task.t_vacation_rule_result)
    is
        l_result            apex_human_task.t_vacation_rule_result;
        l_changes           apex_human_task.t_task_participant_changes;
        l_participants      apex_human_task.t_task_participants;
        l_new_participant   apex_human_task.t_task_participant;
        l_old_participant   apex_human_task.t_task_participant;
        l_has_changed       boolean                             := false;
        l_index             pls_integer                         := 1;
    begin
```

```
        l_participants := p_param.original_participants;
        for i in 1..l_participants.count loop
            if l_participants(i).value = 'JANE' then
                l_old_participant               :=  l_participants(i);
                l_new_participant.value         := 'SCOTT';
                l_new_participant.type          := 'POTENTIAL_OWNER';
                l_new_participant.identity      := 'USER';
                l_has_changed                   := true;
            end if;
        end loop;
        if l_has_changed then
            l_participants(l_participants.count + 1) := l_new_participant;

            l_changes(l_index).old_participant   := l_old_participant;
            l_changes(l_index).new_participant   := l_new_participant;
            l_changes(l_index).change_reason     := 'Jane is out on maternity
leave';
        end if;
        l_result.participant_changes            := l_changes;
        l_result.has_participant_changes        := l_has_changed;
        p_result                                := l_result;
    end;
```

For more information on adding a procedure, see Creating a Procedure in *Oracle APEX SQL Workshop Guide*.

To add a vacation rule procedure:

1. Select the Sample Workflows, Approvals, and Tasks app.
2. Navigate to the Task Definitions page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Click **Job Change**.
4. Under Participants, enter the vacation rule procedure name in **Vacation Rule Procedure**.

   The procedure name should be in the format `[schemaName.[packageName]].procedureName`.
5. Click **Apply Changes**.

Scott is now an alternate participant for Jane.

20.15.1.4.3 Example: Editing Task Definition Parameters

Edit Task Definition Parameters by accessing the Task Definition in the Shared Components page.

This example uses the **Sample Workflows, Approvals, and Tasks** application, which can be installed from the Gallery. In this example, suppose that for Job Change requests, you want to display the old job title as well as the new job title on 2 - Task Details.

To edit Task Definition Parameters:

1. Select the Sample Workflows, Approvals, and Tasks app.
2. Navigate to the Task Definitions page:
   a. On the Application home page, click **Shared Components**.

   b. Under Workflows and Automations, select **Task Definitions**.
3. Click **Job Change**.
4. Under Parameters, click **Add Row**.
   a. Static Id - `JOB`
   b. Label - `Previous Job`
   c. Data Type - `String`
   d. Required - `Yes`
   e. Visible - `Yes`
   f. Updatable - `Yes`
5. Click **Apply Changes**.
6. Go back to the application home page, and open **7 - Request Job Change** in Page Designer.
7. Open the Processing tab, and expand the Submit Task process.
8. Under Parameters, select **Previous Job**.
9. Under Value, set the Type to `Item`.
10. Set the item to `P7_JOB`.
11. Click **Save**.

    Now, when 7 - Request Job Change is saved, the Previous Job value will be stored.
12. Open **2 - Task Details** in Page Designer, and confirm that the **Details** section contains the columns `PARAM_LABEL` and `PARAM_VALUE`.

    All visible parameters, including the new Previous Job parameter, are visible in the Details section of the Task Details page. Since the parameter is marked as **updatable**, the task details section has an Edit button to edit the task parameter when the logged-in user is the actual owner of the task.

To test this change, go to the Employees page, and request a new job change. After submitting the Request Job Change page, access the task details page through My Requests or Approvals Administration. Under the Details section, you can see both the new job and the previous job.

20.15.1.4.4 Example: Adding and Editing Task Definition Actions

Add and edit Task Definition Actions by accessing the task definition in the Shared Components page.

This example uses the **Sample Workflows, Approvals, and Tasks** application, which can be installed from the Gallery. In this example, suppose that you want to add an action that logs an event every time a task is complete.

This example requires a metrics table to be added to the Sample Workflows, Approvals, and Tasks app through the SQL Commands page of the SQL Workshop.

```
CREATE TABLE EBA_DEMO_APPR_METRICS(
  ID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY,
  TASK_DEF_STATIC_ID VARCHAR2(255),
  OUTCOME_CODE VARCHAR2(32),
  APPROVER VARCHAR2(255),
  APPROVAL_DATE DATE,
```

```
  DAYS_ELAPSED NUMBER
);
```

To add and edit Task Definition Actions:

1. Select the Sample Workflows, Approvals, and Tasks app.
2. Navigate to the Task Definitions page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Click **Salary Change**.
4. In the Actions section, click **Add Action**.
   a. Name - `Salary Change Metrics`
   b. Type - `Execute Code`
   c. Execution Sequence - Leave the default value.
   d. On Event - `Complete`
   e. Outcome - Leave blank.
   f. Success Message - Leave blank.
   g. Location - `Local Database`
   h. Language - `PL/SQL`
   i. Code - Enter the sample insert statement.

```
insert into
eba_demo_appr_metrics(task_def_static_id,outcome_code,approver,days_elap
sed,approval_date)
select task_def_static_id, outcome_code, actual_owner, sysdate -
cast(created_on as date), sysdate
from apex_tasks
where task_id = :APEX$TASK_ID;
```

5. The **Error Handling**, **Server Side Condition**, and **Advanced** sections can be left as-is.
6. Click **Create**.
7. Click **Apply Changes**.

The new action adds a row into `EBA_DEMO_APPR_METRICS` every time a Salary Change task is completed. You could set up an interactive report based on this table to allow users to view metrics on approval tasks.

20.15.1.5 Viewing the Task Definition Utilization Report

The Task Definition Utilization Report shows all pages and page components where Task Definitions are used in a specific application.

To view the Task Definition Utilization Report:

1. Select the application.
2. Navigate to the Task Definition page:
   a. On the Application home page, click **Shared Components**.

   b. Under Workflows and Automations, select **Task Definitions**.

3. Select **Utilization**.

   The Task Definition Utilization Report appears.

20.15.1.6 Viewing the Task Definition History Report

The Task Definition History Report shows recent modifications made to Task Definitions.

To view the Task Definition History Report:

1. Select the application.
2. Navigate to the Task Definition page:
   a. On the Application home page, click **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Select **History**.

   The Task Definition History Report appears.

The History Report displays recent modifications made to Task Definitions in the current application.

20.15.2 Managing Task Details

Task Details display when you open the task instance.

- About the Task Details Page
  The task details page shows information about a particular task instance.
- Editing the Task Details Page
  Edit the Task Details Page using Page Designer.
- Configuring the Task History
  Configure the Task History from the Task Definition.

20.15.2.1 About the Task Details Page

The task details page shows information about a particular task instance.

The **Task Details** page contains information about a specific task instance. Depending on the status of the task, and the role of the person logged in, the task details page may show the following buttons:

- Release - Releases the task, setting it from Assigned to Unassigned so that it can be claimed by other potential owners.
- Delegate - Assigns a task exclusively to another potential owner, who then becomes the actual owner.
- Change Priority - Updates the priority of a task.
- Change Due Date - Updates the task's due date.
- Request Information - Requests more information from the task initiator.
- Submit Information - Sends more information back to the task owner.
- Invite Participant - Adds a new potential owner.
- Remove Participant - Removes an existing potential owner.

- Cancel Task - Cancels a task.
- Renew Task - Renews an expired task.
- (Approval tasks only) Approve - Approves the task, setting the state to Completed.
- (Approval tasks only) Reject - Rejects the task, setting the state to Completed.
- (Action tasks only) Save - Saves the task without completing it.
- (Action tasks only) Complete - Completes a task, setting the state to Completed.
- Claim - Claims a task, setting it from Unassigned to Assigned and changing the current user to the actual owner.
- Add Comment - Adds a comment to the task.

The task details page also shows comments users make and a log of changes for the task.

Typically, you set up one task details page for each task definition. This ensures you can customize the task details page to include information relevant for the specific task definition.

20.15.2.2 Editing the Task Details Page

Edit the Task Details Page using Page Designer.

The Task Details Page is a generated page that you can customize according to your requirements. It displays when the task is selected on the Unified Task List Report.

To edit a Task Details page:

1. Select the application.
2. In Page Designer, select the Task Details page.

   You can also find the Task Details Page URL in the Task Definition.
3. Edit the Task Details in Page Designer.

> **See Also:**
>
> Editing Pages in Page Designer

20.15.2.3 Configuring the Task History

Configure the Task History from the Task Definition.

The task history logs events for an individual task. You can view the task history on the task details page.

To configure the task history:

1. Select an application.
2. Navigate to the Task Definitions page:
   a. On the application home page, select **Shared Components**.
   b. Under Workflows and Automations, select **Task Definitions**.
3. Click on the task definition.
4. Navigate to the **Actions** section.

5. Click the pencil icon next to an action to edit the action.
6. Under **Error Handling**, navigate to **Log Messages When**.
7. Choose from the available options:
   - None (Default) - No action is logged in the task history.
   - Success - Only successful actions are logged in the task history.
   - Failure - Only failed actions are logged in the task history.
   - All - All actions are logged in the task history.
8. Repeat this step for each action.

20.15.3 Managing Unified Task Lists

The Unified Task List allows workspace users to view and manage their tasks.

- About Unified Task Lists
  Learn about the types of Unified Task Lists available, including My Tasks for potential task owners, Admin Tasks for business administrators, and Initiated by Me for task initiators.
- Creating Unified Task Lists
  Create a Unified Task List using the Create Page Wizard.
- Editing Unified Task Lists
  Edit a Unified Task List using Page Designer.

20.15.3.1 About Unified Task Lists

Learn about the types of Unified Task Lists available, including My Tasks for potential task owners, Admin Tasks for business administrators, and Initiated by Me for task initiators.

The Unified Task List allows users to view, action, or edit task instances. When you configure the Unified Task List, you have different **Report Contexts** to choose from:

- **My Tasks**, for users who are potential or actual owners of tasks.
  - Shows all tasks that are either **Assigned** to the user or **Unassigned** with the user as a potential owner. Tasks that are approved, rejected, or completed do not show in the list.
  - Sorts by descending creation date, with newly created tasks at the top of the list.
- **Admin Tasks**, for users who are business administrators of tasks.
  - Shows all tasks that list the user as a business administrator, including **Errored**, **Completed**, and **Canceled** tasks. **Expired** tasks only show if the user selects **Show expired tasks**.
- **Initiated by Me**, for users who have initiated tasks.
  - Shows all tasks that the user has initiated.

The Unified Task List can be in the same application as one of the task definitions or a completely separate application in the same workspace. If the Unified Task List is in a separate application, you must configure session sharing so users can still open the task details.

20.15.3.2 Creating Unified Task Lists

Create a Unified Task List using the Create Page Wizard.

To create a Unified Task List:

1. Open the Create Page Wizard.
   a. On the Workspace home page, click **App Builder**.
   b. Select the application that needs a Unified Task List.
   c. On the Application home page, select **Create Page**.

      The Create Page Wizard appears.
2. Under Components, select **Unified Task List**.
3. For Page Definition, set the following:
   a. Page Number - An integer value that identifies a page within an application.
   b. Name - Specify a text name for this page. The same text is also used for the page title.

      After page creation, the title can be modified using Page Designer.
   c. Report Context - Select **My Tasks**. **Admin Tasks**, or **Initiated by Me**.

      My Tasks is for the potential and actual task owners. **Admin Tasks** is for the business administrators. **Initiated by Me** is for users who initiate tasks.
4. Navigation - Expand the region and enable or disable navigation. If either flag is enabled, additional attributes appear. To learn more about an attribute, see field-level Help.
   a. Use Breadcrumb - Enable to create a breadcrumb entry for this page. By default, the page name is used as breadcrumb entry name.
   b. Use Navigation - Enable to create a navigation menu entry for this page. By default, the page name is used as navigation menu name.
5. Select **Create Page**.

The Unified Task List opens in Page Designer.

20.15.3.3 Editing Unified Task Lists

Edit a Unified Task List using Page Designer.

To edit a Unified Task List:

1. Open the Unified Task List in Page Designer.
   a. On the Workspace home page, click **App Builder**.
   b. Select the application that contains the Unified Task List.
   c. On the Application home page, select the Unified Task List page you want to edit.
2. Edit the Unified Task List.
3. Select **Save** or **Save and Run Page** to save changes.

> **See Also:**
>
> Using Page Designer

20.16 Creating Tasks from Task Definitions

Create tasks in an Oracle APEX application from existing task definitions.

For APEX to create tasks, you must define a **Human Task - Create** page process. When the page process runs, APEX creates a new task based on the task definition. In order to be able to use this process, at least one Task Definition must exist as a shared component in the application.

To add a Human Task - Create page process to a page:

1. View the page in Page Designer.
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.
2. In either the Rendering or Processing tab, locate the node where the process will occur. Right-click and select **Create Process**.
3. Under Identification:
   a. Name - Specify the name of the process.
   b. Type - Select **Human Task - Create**.
4. Under Settings:
   a. Definition - Select the Task Definition.
   b. Subject - If specified here, this will override the subject in the task definition.
   c. Priority - If specified here, this will override the priority in the task definition.
   d. (Approval tasks only) Initiator Can Complete - If specified here, this will override the value in the task definition.
   e. Initiator Item - The page item whose value can be used to specify who initiates the task. By default, the logged-in user running the page will be the initiator. This value can be specified to override the logged-in user, allowing a user to initiate a task on behalf of another user.
   f. Task ID Item - When specified, this saves the internal id of the task created when the page is run.
   g. Details Primary Key Item - When specified, this allows Oracle APEX to uniquely identify a record from the table or query that is specified as the Actions Source in the task definition.

      For example, in the **Sample Workflows, Approvals, and Tasks** app, entering `P4_EMPNO` associates the task with the corresponding record in the `EBA_DEMO_APPR_EMP` table. When the actual owner approves the Job Change Request, APEX updates the table with the new information from the task.
   h. Due Date Item - An optional page item representing the task's due date. When specified, this value overrides the due date calculated from the associated task definition.
5. Configure the rest of the Process.

   For more information, see Creating Page Processes.
6. Set the Page Process Parameters.

If the Task Definition has parameters specified, these appear in the Process Tree on the page. Parameters can be set to static values or page items. If a parameter is not required, it can be left as `Null`. If a parameter is marked `Required` in the Task Definition, it must be set in the process or there will be an error when saving the page.

7. Click **Save**.

- Example: Creating a Human Task from a Form Submission
  You can configure both approval and action tasks, collectively known as **human tasks**, as page processes that can run upon submission of a form.

20.16.1 Example: Creating a Human Task from a Form Submission

You can configure both approval and action tasks, collectively known as **human tasks**, as page processes that can run upon submission of a form.

This example uses the **Sample Workflows, Approvals, and Tasks** application, which you can install from the Gallery. This example demonstrates how to set up the process that initiates a job change approval task when a user submits the Request Job Change form.

To create approval tasks when the Request Job Change form is submitted:

1. Open **7 - Request Job Change** in Page Designer.
2. Open the Processing tab.
3. Right-click **Processing** and select **Create Process**.
4. Under Identification:
   a. Name - `Submit Task`
   b. Type - `Human Task - Create`
5. Under Settings:
   a. Definition - `Job Change`
   b. Details Primary Key Item - `P7_EMPNO`
6. Under Execution Options:
   a. Sequence - `20`
   b. Point - `Processing`
   c. Run Process - `Once Per Page Visit (Default)`
7. Under Server-side Condition:
   a. When Button Pressed - `SUBMIT_FOR_APPROVAL`
   b. Type - Leave as – `Select –`
8. Set the Page Process Parameters.
   a. On the Processing tab, under Submit Task, expand **Parameters**.
   b. Select **New Job**.
   c. Under Value, select `Item`.
   d. Set Item to `P7_PROPOSED_JOB`.
9. Click **Save**.

APEX creates an approval task when a user submits the Request Job Change form.

> **Note:**
>
> You can use the same steps to create an action task from a form submission.

20.17 About Creating Human Tasks Programmatically

Learn about creating human tasks programmatically.

Use the `APEX_HUMAN_TASK` API package to manage tasks programmatically. The package includes functionality to create new tasks and handle lifecycle management for existing tasks. The package is part of Oracle APEX workflow functionality.

> **See Also:**
>
> APEX_HUMAN_TASK in *Oracle APEX API Reference*.

20.18 About Creating a Task Definition Process Plug-in

Learn about creating a task definition process plug-in.

You can extend a task definition with a process plug-in as part of the plug-in functionality available with Oracle APEX. By setting the plug-in **Type** to `Process` and **Supported for** to `Task Definition Actions`, the plug-in becomes available in task definition actions.

For example, if you want to update an external system when an approval task completes, you can make those updates with standard task functionality by adding custom PL/SQL as a task action. However, if this code is repeated across many task definitions in your application, you can use a process plug-in to hide the details of the task processing, and instead have a simple declarative definition using the plug-in within each task action.

For more on plug-ins, see Implementing Plug-ins and Understanding Plug-ins.

20.19 About Task Retention and Task Purging

Learn about task retention, task purging, and the Archive of Purged Task Files report.

APEX can create tasks with complex payloads. This means task data can grow exponentially, causing space and performance overloads. Periodically purging task data prevents overloading the system.

Task Retention Logic

A `DBMS_SCHEDULER` job runs on a daily basis to check all records in the task instance table. When `DBMS_SCHEDULER` runs, it:

- Keeps active, unassigned, and assigned tasks
- Purges completed tasks with a retention period that is less than or equal to the current date
- Purges all terminated, canceled, and errored tasks

> **Note:**
>
> Terminated and canceled tasks do not have a retention period, and only remain in the system until the daily `DBMS_SCHEDULER` job runs. This means terminated or canceled tasks exist in the task instance table for less than twenty-four hours.

The Archive of Purged Tasks Report saves a `JSON` document of tasks that are eligible for purge. After the report is generated, the tasks eligible for purge are cleaned up.

Task Retention Settings

The task retention policy determines how long to keep completed tasks. The default task retention period is seven days and the maximum task retention period is thirty days. An instance administrator can configure the task retention period in Instance Settings. For more information, see Configuring Workflow Settings in *Oracle APEX Administration Guide*.

Archive of Purged Task Files Report

The report of tasks that have been purged by the Task Purge job is located in Workspace Administration. For more information, see Accessing the Monitor Activity Page in Administration Services in *Oracle APEX Administration Guide*.

Files in the Purged Task Files report area are cleaned up every 30 days.

- Retrieving Tasks Before Purge
  Example code for retrieving tasks before purge.

20.19.1 Retrieving Tasks Before Purge

Example code for retrieving tasks before purge.

This example uses an API to fetch purgeable tasks, retrieve the data, and handle those tasks in their own schema.

```
merge
    into user_task_table    u
using apex_purgeable_tasks  t
    on (u.task_id = t.task_id)
when not matched then insert
    (u.task_id, u.task_type, u.subject, u.priority)
values(t.task_id, t.task_type, t.subject, t.priority);
```

20.20 Substitution Strings and Bind Variables for Tasks

Oracle APEX uses substitution strings to pass information about a task to a page.

You can use substitution strings to pass information about a task instance to an APEX page. For example, you might want to show different regions in the task details page depending on the task's priority or the date a task was created.

You can configure substitution strings in the task definition.

Substitution String Syntax	Description
APEX$TASK_CREATED_ON	The timestamp for task creation in YYYY-MM-DD"T"HH24:MI:SS.FF3TZH:TZM format
APEX$TASK_DUE_ON	The timestamp for the task due date in YYYY-MM-DD"T"HH24:MI:SS.FF3TZH:TZM format
APEX$TASK_ID	The task id
APEX$TASK_INITIATOR	The user who initiated the task • Defaults to the logged in user unless a different value is explicitly specified in the Human Task - Create Page Process
APEX$TASK_MAX_RENEWAL_COUNT	The maximum number of times a task can be renewed according to the expiration policy in the task definition
APEX$TASK_OUTCOME	The task outcome • Only populated for approval tasks in the Completed state
APEX$TASK_OWNER	The user who currently owns the task
APEX$TASK_PK	The primary key value • APEX$TASK_PK can also be used as a bind variable in any PL/SQL query used by task definition actions and task definition participants. When used as a bind variable, the syntax is :APEX$TASK_PK
APEX$TASK_PREVIOUS_ID	The previous id for a task • Links an expired task to the renewed task • If task T1 is expired and T2 is the renewed task, then APEX$TASK_PREVIOUS_ID of T2 is the ID of task T1 (T1.id)
APEX$TASK_RENEWAL_COUNT	The number of times a task has been renewed
APEX$TASK_STATE	The task's current state
APEX$TASK_SUBJECT	The task subject
APEX$TASK_TEXT	The text message sent as part of the **Add Comment**, **Request Information**, and **Submit Information** task actions, which you can use as a bind variable to access the text in **Execute Code** or **Send Email** actions defined for these task operations

20.21 Runtime Views for Tasks

Learn about the runtime views for tasks.

View Name	Description
APEX_PURGEABLE_TASKS	The properties of each purgeable task instance.
APEX_PURGEABLE_TASK_HISTORY	The chronological series of operations performed during the lifecycle of a purgeable task instance, including the user associated with the update and the date the operation was performed.
APEX_PURGEABLE_TASK_COMMENTS	All comments entered on each purgeable task instance.

View Name	Description
APEX_PURGEABLE_TASK_PARAMETERS	The parameters defined for a purgeable task instance along with their computer runtime values in String format.
APEX_TASKS	The properties of each task instance.
APEX_TASK_COMMENTS	All comments entered on each task instance.
APEX_TASK_HISTORY	The chronological series of operations performed during the lifecycle of a task instance, including the user associated with the update and the date the operation was performed.
APEX_TASK_PARAMETERS	The parameters defined for a task instance along with their computed runtime values in String format.
APEX_TASK_PARTICIPANTS	The list of participants on each task instance. Note that this list can differ from the list in the task definition due to business administrators inviting new participants or removing existing participants.

For more information on purgeable tasks, see About Task Retention and Task Purging.

20.22 Troubleshooting Tasks

Methods to troubleshoot issues with tasks and the approvals component.

Troubleshooting Tasks

Problem	Solution
Encountering Error **ORA-20987: APEX: Not authorized for Task operations like complete_task, release, delegate, cancel_task, etc**	Confirm the user name case in APEX_TASKS_PARTICIPANT and the user name case for the log in name are identical. APEX users are saved in uppercase (USER). Potential owners specific in the task definition should be specified in uppercase. Many of the authorization exceptions are false alarms due to incorrect case.

Problem	Solution

Approving a task through DB Scheduler Job is throwing a **Task Not Found** error.	When you use a DBMS Scheduler job to run `apex_human_task` APIs, the tasks are not in the user session, and cannot be found.
	To resolve this error, create a valid APEX session in the job using `apex_session.create_session()` that passes the name of the approver as the user name and then calls the `approve_task` API.
	
	`PROCEDURE CREATE_SESSION (`
	`p_app_id IN NUMBER,`
	`p_page_id IN NUMBER,`
	`p_username IN VARCHAR2,`
	`p_call_post_authentication IN`
	`BOOLEAN DEFAULT FALSE );`
	
	After running `create_session`, you can call the `apex_human_task` API to approve or reject the task.

Task not expiring after setting the due date using `apex_human_task.set_task_due`	APEX has a scheduled job that checks once an hour for tasks past the due date. Expired tasks do not show as expired until the scheduled job runs. You can trigger the scheduled job manually by running `APEX_HUMAN_TASK.HANDLE_TASK_DEADLINES`. Run this API to confirm your task goes to the expired state as specified in the expiration policy of the task definition. This is convenient for testing task expiration policies without waiting for the background job to run.
`APEX$TASK_ID` substitution string does not work when used in the Actions SQL Query in the Edit Task Definitions page	You cannot use `APEX$TASK_ID` in the action source because the ID is generated only after the task is created. This is a known bug that is planned to be fixed in an upcoming release. As a workaround, use the `APEX$TASK_PK` substitution string wherever possible.

21
Managing Application Security

Administrators are primarily responsible for ensuring the security of the Oracle APEX installation and developers are responsible for building secure applications. Learn about how to implement best practices for APEX application security.

> **Note:**
>
> To ensure the security and performance of your development environment, some administration functionality is not available in Oracle APEX instances running in Oracle Cloud.

- Understanding Administrator Security Best Practices
  Learn about security best practices for Oracle APEX administrators.
- Understanding Developer Security Best Practices
  Learn about security best practices for Oracle APEX developers.
- Controlling Access to Applications, Pages, and Page Components
  Control access to an application, individual pages, or page components by creating an access control list.
- Establishing User Identity Through Authentication
  Use authentication to establish a user's identity to control access to an application. Authentication may require a user identify a user name and password or could involve the use of digital certificates or a secure key.
- Providing Security Through Authorization
  Extend the security of your application by creating an authorization scheme.

> **See Also:**
>
> Running Advisor to Check Application Integrity

21.1 Understanding Administrator Security Best Practices

Learn about security best practices for Oracle APEX administrators.

> **Note:**
>
> To ensure the security and performance of your development environment, some administration functionality is not available in Oracle APEX instances running in Oracle Cloud.

This section references many settings in the APEX Administration Services application.

- About Oracle APEX Administrator Roles
  Oracle APEX includes two different types of administrators: Workspace administrators and Instance administrators.
- Web Server Security Considerations
  Review Oracle REST Data Services security considerations.
- Managing Instance Security
  Learn about security best practices when managing an Oracle APEX instance.
- Configuring Instance Settings
  Security best practices when configuring an Oracle APEX instance include determining whether to automate the creation workspaces., enabling and configuring email, configuring storage options, and creating wallets.
- About Configuring Workspace Purge Settings
  Save storage space and improve system performance by purging inactive workspaces.
- Understanding Workspace Management
  Instance administrators can define associations between workspaces and schemas. Administrators can manage developers and users at the workspace or instance-level and control user access to key components such App Builder, Team Development, and SQL Workshop, and PL/SQL editing.
- About Integrating with Oracle BI Publisher
  Oracle recommends that you employ Secure Sockets Layer (SSL) when integrating Oracle APEX with Oracle BI Publisher. Once SSL (HTTPS protocol) is configured for your Oracle BI Publisher server, you must create a wallet and specify the HTTPS protocol for the Report Server in the internal administration Environment Settings.
- About the Advantages of the Oracle APEX Runtime Environment
  An Oracle APEX runtime environment enables you to run applications, but does not provide a web interface for administration or direct development of these applications.
- Enabling Network Services in Oracle Database
  Database administrators must enable network services in Oracle Database to send outbound mail, invoke web services, or use template-based PDF report printing with external print servers in Oracle APEX.

> **See Also:**
>
> Understanding Developer Security Best Practices

21.1.1 About Oracle APEX Administrator Roles

Oracle APEX includes two different types of administrators: Workspace administrators and Instance administrators.

To access the APEX development environment, users sign in to a shared work area called a workspace. **Workspace administrators** are users who perform administrator tasks specific to a workspace such as managing user accounts, monitoring workspace activity, and viewing log files. **Instance administrators** are superusers that manage an entire hosted instance using the APEX Administration Services application.

> **Tip:**
>
> The role of Instance administrator is not available in Oracle Database Cloud Service (Database Schema).

> **See Also:**
>
> Workspace and Application Administration and Oracle APEX Administration Services in *Oracle APEX Administration Guide*

21.1.2 Web Server Security Considerations

Review Oracle REST Data Services security considerations.

- About Configuring Oracle REST Data Services with Oracle APEX
  Oracle APEX requires access to the web server, Oracle REST Data Services

21.1.2.1 About Configuring Oracle REST Data Services with Oracle APEX

Oracle APEX requires access to the web server, Oracle REST Data Services

Oracle REST Data Services (formerly known as Oracle APEX Listener) is a J2EE application which communicates with the Oracle Database by mapping browser requests to the APEX engine database over a SQL*Net connection. In a production environment, you deploy Oracle REST Data Services web archive files to a supported Java EE application server, like Oracle Web Logic Server. Each deployment can be configured individually and serves the same purpose as a `mod_plsql` Database Access Descriptor, which is to communicate with an Oracle database.

An Oracle REST Data Services deployment configuration contains several security related parameters. In a configuration for APEX, Oracle recommends to set the parameter `security.requestValidationFunction` to `wwv_flow_epg_include_modules.authorize`. This activates the white list of callable procedures which ships with APEX and prohibits calls to other procedures. This can be extended using the validation functions shipped with APEX.

21.1.3 Managing Instance Security

Learn about security best practices when managing an Oracle APEX instance.

> **Note:**
>
> To ensure the security and performance of your development environment, some administration functionality is not available in Oracle APEX instances running in Oracle Cloud.

- About Creating Login Controls
  Administrators can configure login controls for an entire instance or for individual workspaces.
- About Enabling Public File Upload
  Administrators can use the Allow Public File Upload attribute to control whether unauthenticated users can upload files in applications that provide file upload capability
- About Restricting User Access by IP Address
  Administrators can restrict user access to an APEX instance by entering a comma-delimited list of allowable IP addresses in the Restrict Access by IP Address attribute on the Manage Instance, Security page.
- About Specifying an Instance Proxy
  Administrators can use the Instance Proxy attribute to configure an entire APEX instance to use a proxy for all outbound HTTP traffic.
- About Utilizing Secure Sockets Layer
  Secure Sockets Layer (SSL) is a protocol for managing the security of data transmitted on the Internet. For web applications, SSL is implemented by using the HTTPS protocol. Oracle recommends that you run APEX applications using SSL (HTTPS protocol) to prevent any sensitive data from being sent over an unencrypted (cleartext) communication channel.
- About Enabling RESTful Access
  Administrators can use the Allow RESTful Access attribute to control whether developers can expose report regions as RESTful services.
- About Rejoin Sessions
  Use the Rejoin Sessions attribute to control if APEX supports application URLs that do not contain session IDs.
- About Isolating Workspaces
  Isolating workspaces is an effective approach to preventing browser attacks.
- About Utilizing Session Timeout
  Configure Session Timeout attributes to reduce your application's exposure.
- Restricting Password Reuse
  Site administrators can restrict password reuse for all administrator, developer, and end-user accounts based on a history of passwords previously used for the accounts.
- About Enabling RESTful Services
  Administrators can control the ability to create and access RESTful Services at either the workspace or instance-level.

> **See Also:**
>
> Configuring Security in *Oracle APEX Administration Guide*

21.1.3.1 About Creating Login Controls

Administrators can configure login controls for an entire instance or for individual workspaces.

Workspace administrators and developers can create user accounts for the purpose of logging in to the APEX development environment and for end-user authentication to applications developed within their workspaces.

Administrators can configure login controls at the instance or workspace level. For example, if an Instance administrator configures account login controls in APEX Administration Services, that configuration applies to all APEX accounts in all workspaces across an entire development instance.

If the Instance administrator does *not* enable login controls across an entire instance, then each Workspace administrator can enable the following controls on a workspace-by-workspace basis:

- End-user account expiration and locking.
- A maximum number of failed login attempts for end-user accounts.
- The password lifetime (or number of days an end-user account password can be used before it expires for end-user accounts).

> **See Also:**
>
> Managing Workspace Account Login Controls and in Configuring Authentication Controls for an Instance in *Oracle APEX Administration Guide*

21.1.3.2 About Enabling Public File Upload

Administrators can use the Allow Public File Upload attribute to control whether unauthenticated users can upload files in applications that provide file upload capability

> **See Also:**
>
> Controlling Public File Upload in *Oracle APEX Administration Guide*

21.1.3.3 About Restricting User Access by IP Address

Administrators can restrict user access to an APEX instance by entering a comma-delimited list of allowable IP addresses in the Restrict Access by IP Address attribute on the Manage Instance, Security page.

> **See Also:**
>
> Restricting User Access by IP Address in *Oracle APEX Administration Guide*

21.1.3.4 About Specifying an Instance Proxy

Administrators can use the Instance Proxy attribute to configure an entire APEX instance to use a proxy for all outbound HTTP traffic.

Setting a proxy at the instance-level supersedes any proxies defined at the application-level or in web service references. If a proxy is specified, regions of type URL, Web services, and report printing will use the proxy.

> **See Also:**
>
> Configuring a Proxy Server for an Instance in *Oracle APEX Administration Guide*

21.1.3.5 About Utilizing Secure Sockets Layer

Secure Sockets Layer (SSL) is a protocol for managing the security of data transmitted on the Internet. For web applications, SSL is implemented by using the HTTPS protocol. Oracle recommends that you run APEX applications using SSL (HTTPS protocol) to prevent any sensitive data from being sent over an unencrypted (cleartext) communication channel.

Instance administrators can configure both their APEX instance and all related applications to require HTTPS by configuring the **Require HTTPS** and **Require Outbound HTTPS** attributes. Configuring these attributes forces authenticated pages within the App Builder to require HTTPS which encrypts network communications. Changing the Require HTTPS option does not affect the web server in any way. You will still need to enable the web server for HTTPS for these options to have any meaning. To learn more about enabling HTTPS, see the documentation for your specific web server.

Require HTTPS options include:

- **Always** - Enforces HTTPS for all applications (including the APEX development and administration applications) to require HTTPS.
- **Development and Administration** - Forces all internal applications within APEX (that is, App Builder, SQL Workshop, Instance Administration and so on) to require HTTPS.
- **Application specific** - Makes HTTPS dependent on application-level settings.

> **See Also:**
>
> Configuring HTTP Protocol Attributes in *Oracle APEX Administration Guide*

21.1.3.6 About Enabling RESTful Access

Administrators can use the Allow RESTful Access attribute to control whether developers can expose report regions as RESTful services.

> **See Also:**
>
> Enabling RESTful Services for an Instance in *Oracle APEX Administration Guide*

21.1.3.7 About Rejoin Sessions

Use the Rejoin Sessions attribute to control if APEX supports application URLs that do not contain session IDs.

Tip:

For security reasons, Oracles recommends that administrators disable Rejoin Sessions unless they implement workspace isolation by configuring the Allow Hostname attribute at the workspace or instance-level. See About Isolating Workspaces.

When rejoin sessions is enabled, APEX attempts to use the session cookie to join an existing session, when a URL does not contain a session ID. To use Rejoin Sessions, administrators must enable Rejoin Sessions at the application or page-level. Note that a more restrictive value of Rejoin Sessions at the instance-level overrides application and page settings.

While supporting session joins is convenient, it does present some serious security risks:

- **Triggering malicious session state changes or other modifications**.

  If an attacker tricks the user into clicking an application link, this can trigger malicious session state changes or other modifications. Because of this risk, APEX requires a checksum that the attacker will not be able to guess when processing requests that alters data.

- **Triggering unintended changes between applications**.

  If a user has two applications open in separate tabs on the same APEX instance, one application could trigger unintended changes in the other one. The attacking application could be written by a hacker or it could have an XSS vulnerability that enables an attacker to inject code. This application could use Ajax calls that simulate user interactions with the other application.

See Also:

- About Enabling Support for Bookmarks
- Session Management
- Configuring Rejoin Sessions for a Page
- Configuring Rejoin Sessions for an Instance in *Oracle APEX Administration Guide*

21.1.3.8 About Isolating Workspaces

Isolating workspaces is an effective approach to preventing browser attacks.

The only way to truly isolate a workspace is to enforce different domains in the URL by configuring the Allow Hostnames attribute in APEX Administration Services. When the URLs of the attacker and the victim have different domains and hostnames, the browser's same-origin policy prevents attacks.

Workspace isolation by configuring Allow Hostnames is a counter measure against client side attacks that attempt to cross workspace boundaries. This security measure is not necessary if you trust all applications that are accessible using the instance's host which includes applications that are written in other frameworks and languages such as Oracle Application Development Framework (ADF) and Java.

> **See Also:**
>
> - Isolating All Workspaces in an Instance in *Oracle APEX Administration Guide*
> - Isolating a Workspace to Prevent Browser Attacks in *Oracle APEX Administration Guide*

21.1.3.9 About Utilizing Session Timeout

Configure Session Timeout attributes to reduce your application's exposure.

Users often leave their computers unattended for extended periods and do not close applications before departing. Therefore, an unauthorized person can easily assume the user's identity within the application. By setting the session and idle timeout, users are automatically logged out of their application after the specified timeout.

Session Timeout attributes include:

- Maximum Session Length in Seconds
- Maximum Session Idle Time in Seconds
- Session Timeout Warning in Seconds
- Session Timeout URL (application-level setting)
- Idle Timeout URL (application-level setting)

Developers can configure session timeout for a specific application, or administrators can configure sessions timeout for a workspace or an entire instance.

> **See Also:**
>
> - Session Management to learn how to configure these attributes at an application-level.
> - Configuring Session Timeout for a Workspace and Configuring Session Timeout for an Instance in *Oracle APEX Administration Guide*

21.1.3.10 Restricting Password Reuse

Site administrators can restrict password reuse for all administrator, developer, and end-user accounts based on a history of passwords previously used for the accounts.

To restrict password reuse:

1. Start SQLcl and connect to the database where Oracle APEX is installed as `SYS`, for example:
   - On Windows:

```
SYSTEM_DRIVE:\ sql /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: SYS_password
```

- On UNIX and Linux:

```
$ sql /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: SYS_password
```

> **Tip:**
>
> The `APEX_ADMINISTRATOR_ROLE` must be granted directly to a user and not through another database role.

2. Execute the following statement:

```
ALTER SESSION SET CURRENT_SCHEMA = APEX_240200;
```

3. Execute the following statement:

```
BEGIN
APEX_INSTANCE_ADMIN.SET_PARAMETER('PASSWORD_HISTORY_DAYS',365);
COMMIT;
END;
/
```

This restricts new or updated passwords to those that have not been used for the account for the previous 365 days. Recording of previously used passwords (actually, hashed representations of passwords) begins upon installation of Oracle APEX.

4. To disable this feature, run the block above, providing 0 for the parameter value. For example:

```
BEGIN
APEX_INSTANCE_ADMIN.SET_PARAMETER('PASSWORD_HISTORY_DAYS',0);
COMMIT;
END;
/
```

> **See Also:**
>
> - STRONG_PASSWORD_CHECK Procedure in *Oracle APEX API Reference*
> - STRONG_PASSWORD_VALIDATION Function in *Oracle APEX API Reference*

21.1.3.11 About Enabling RESTful Services

Administrators can control the ability to create and access RESTful Services at either the workspace or instance-level.

Disabling the **Enable RESTful Services** attribute prevents developers from creating and editing RESTful Web Services mapped to SQL and PL/SQL in conjunction with Oracle REST Data Services 2.0 or later.

To configure Enable RESTful Services:

- In Workspace Administration — Sign in to APEX and click the **Administration** menu. Select **Manage Service**, **Set Workspace Preferences**, **SQL Workshop** and find the **Enable RESTful Services** attribute.
- In Instance Administration — Sign in to APEX Administration Services. Click **Manage Instance**. Under Instance Settings, click **Feature Configuration** and click **Feature Configuration**. Find the **REST** tab and then **Enable RESTful Services** attribute.

> **See Also:**
>
> Enabling RESTful Services and Defining a Path Prefix for a Workspace and Enabling RESTful Services for an Instance in *Oracle APEX Administration Guide*

21.1.4 Configuring Instance Settings

Security best practices when configuring an Oracle APEX instance include determining whether to automate the creation workspaces., enabling and configuring email, configuring storage options, and creating wallets.

> **Note:**
>
> To ensure the security and performance of your development environment, some administration functionality is not available in Oracle APEX instances running in Oracle Cloud.

- About Enabling Automation When Creating Workspaces
  Learn about enabling automation when creating workspaces.
- About Enabling and Configuring Email
  Learn about enabling and configuring email.
- About Configuring Storage Options
  Learn about configuring storage options.
- About Creating Wallets
  Learn about creating wallets.

> **See Also:**
>
> Configuring Instance Settings in *Oracle APEX Administration Guide*

21.1.4.1 About Enabling Automation When Creating Workspaces

Learn about enabling automation when creating workspaces.

Administrators determine the amount of automation when provisioning (or creating) new workspaces.

To determine how provisioning works, an Instance Administrator selects one of the following options on the Instance Settings page:

- **Manual** - An Instance administrator creates new workspaces and notifies the Workspace administrator regarding the Sign In credentials.
- **Request** - Users request a workspace. Once an administrator approves the request, the user receives an email containing an email verification link. After the user clicks the email verification link, the workspace is created.
- **Automatic** - Works similar to **Request** except requests are automatically approved with no administrator review required.

> **Note:**
>
> To enable users to request a workspace using a link on the Sign In page, the Instance administrator must choose either the **Request** or **Automatic** Provisioning Method and configure email. If the Provisioning Method is set to **Manual**, no link appears on the Sign In page.

> **See Also:**
>
> About the Workspace Request Approval Process and Selecting a Provisioning Mode in *Oracle APEX Administration Guide*

21.1.4.2 About Enabling and Configuring Email

Learn about enabling and configuring email.

In order to send email from an APEX application, an Instance administrator must define the email SMTP settings and set the Use SSL/TLS attribute to use a secure connection for Oracle Database 11*g* R2 or later.

> **See Also:**
>
> - Configuring Email in *Oracle APEX Administration Guide*
> - Enabling Network Services in Oracle Database

21.1.4.3 About Configuring Storage Options

Learn about configuring storage options.

Administrators can configure the following storage options: require a new schema when requesting a workspace, auto extend tablespaces, or delete uploaded files after a specified number of days.

> **See Also:**
>
> Configuring Storage in *Oracle APEX Administration Guide*

21.1.4.4 About Creating Wallets

Learn about creating wallets.

A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL. You must create a wallet if you:

- Call a SSL-enabled URL (for example, by invoking a Web service).
- Create a region of type URL that is SSL-enabled.
- Configure secure SMTP by setting the Use SSL/TLS attribute to **Yes**.
- Have applications with LDAP authentication schemes that are configured to use SSL with Authentication.

> **See Also:**
>
> Configuring Wallet Information in *Oracle APEX Administration Guide*

21.1.5 About Configuring Workspace Purge Settings

Save storage space and improve system performance by purging inactive workspaces.

Inactive workspaces consume valuable storage space and degrade system performance. By enabling Workspace Purge Settings, administrators can configure Oracle APEX to purge inactive workspaces.

> **See Also:**
>
> Purging Inactive Workspaces in *Oracle APEX Administration Guide*

21.1.6 Understanding Workspace Management

Instance administrators can define associations between workspaces and schemas. Administrators can manage developers and users at the workspace or instance-level and control user access to key components such App Builder, Team Development, and SQL Workshop, and PL/SQL editing.

- About Managing Workspace to Schema Assignments
  Learn about managing workspace to schema assignments.

- About Managing Developers and Users
  Learn about managing developers and users.
- About Managing Component Availability
  Learn how workspace administrators can control user access to key components.

> **See Also:**
>
> Managing Existing Workspaces in *Oracle APEX Administration Guide*

21.1.6.1 About Managing Workspace to Schema Assignments

Learn about managing workspace to schema assignments.

When users sign in to Oracle APEX, they sign in to a shared work area called a workspace. Each workspace can have multiple associated (or mapped) schemas. Instance administrators can define associations between workspaces and schemas as appropriate. By associating a workspace with a schema, developers in that workspace can create new database objects in that schema and build applications that interact with the database objects in that schema.

> **See Also:**
>
> Managing Workspace to Schema Assignments in *Oracle APEX Administration Guide*

21.1.6.2 About Managing Developers and Users

Learn about managing developers and users.

Administrators can define what schemas specific users have access to and also their user role, that is if they are a workspace administrator, a developer, or an end user. Administrators can also limit developer access to specific APEX components and lock accounts. APEX includes two separate interfaces for managing developers and users. Workspace administrators manage user accounts within their workspace and Instance administrators manage all user accounts across an APEX development instance.

> **See Also:**
>
> Managing Users in a Workspace and Managing Users Across an APEX Instance in *Oracle APEX Administration Guide*

21.1.6.3 About Managing Component Availability

Learn how workspace administrators can control user access to key components.

Workspace administrators can control user access to key components such as App Builder, Team Development, SQL Workshop, and PL/SQL editing, which determines whether developers have the ability to edit and compile PL/SQL program units from Object Browser. For example, suppose an administrator wants users to be able to build database components,

run SQL statements, but not create applications, he or she could define a workspace with rights to a specific schema and then configure the users as developers using the options on the Set Workspace Preferences page.

> **See Also:**
>
> Configuring Workspace Preferences in *Oracle APEX Administration Guide*

21.1.7 About Integrating with Oracle BI Publisher

Oracle recommends that you employ Secure Sockets Layer (SSL) when integrating Oracle APEX with Oracle BI Publisher. Once SSL (HTTPS protocol) is configured for your Oracle BI Publisher server, you must create a wallet and specify the HTTPS protocol for the Report Server in the internal administration Environment Settings.

> **See Also:**
>
> Configuring Wallet Information and Configuring Report Printing for an Instance in *Oracle APEX Administration Guide*

21.1.8 About the Advantages of the Oracle APEX Runtime Environment

An Oracle APEX runtime environment enables you to run applications, but does not provide a web interface for administration or direct development of these applications.

> **Note:**
>
> To ensure the security and performance of your development environment, this functionality is not available in Oracle APEX instances running in Oracle Cloud.

Oracle recommends that you run any sensitive production APEX applications with a runtime installation of APEX. A runtime installation does not expose the web-based application development environment, thus preventing the use of App Builder, SQL Workshop, and related utilities on a production installation. Additionally, a runtime environment only includes the APEX database objects and privileges necessary to run applications, making it a more hardened environment.

> **See Also:**
>
> Installing Exported Applications in a Runtime Environment in *Oracle APEX Administration Guide*

21.1.9 Enabling Network Services in Oracle Database

Database administrators must enable network services in Oracle Database to send outbound mail, invoke web services, or use template-based PDF report printing with external print servers in Oracle APEX.

To learn more, see Enabling Network Services in Oracle Database in *Oracle APEX Installation Guide*.

> **Note:**
>
> Enabling network services does not apply to APEX instances running on Oracle Autonomous Database. APEX can communicate with external endpoints over the internet without additional configuration.

21.2 Understanding Developer Security Best Practices

Learn about security best practices for Oracle APEX developers.

- About Items of Type Password
  Password items do not emit the text entered to the web browser screen. When creating password items, Oracle recommends using password attributes that do not save session state to prevent the password from being saved in the database in the session state tables.
- Identifying At Risk Password Items
  Identify at risk password items by viewing the Security Profiles report and Password Items report.
- Understanding Cross-Site Scripting Protection
  Protect your application from a cross site-scripting security breach.
- About Session State and Security
  Learn about managing session state and security.
- Preventing URL Tampering
  Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.
- About Securing File Uploads
  Learn about developer best practices for securing file uploads.

> **See Also:**
>
> - Understanding Administrator Security Best Practices
> - About Oracle APEX Administrator Roles

21.2.1 About Items of Type Password

Password items do not emit the text entered to the web browser screen. When creating password items, Oracle recommends using password attributes that do not save session state to prevent the password from being saved in the database in the session state tables.

Configurable password item type attributes include:

- **Settings**, **Submit when Enter pressed** - If set to **On**, the page is submitted when the user presses `ENTER`.
- **Appearance**, **Template Options**, **Common**, **Hide Password Visibility** - By default, the password field has a **Show Password** button that users can click to show their password. If you select **Hide Password Visibility**, the **Show Password** button is disabled.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. If set to **Off**, a NULL value is accepted.
- **Session State**, **Storage** - If set to **Per Request (Memory Only)**, the value is not written to the database and is only available for the current request. For highly sensitive values, this is the preferred setting.

  > **Caution:**
  >
  > If you set **Storage** to **Per Session (Persistent)**, be sure to set **Store value encrypted in session state** to **Yes**.

- **Security**, **Authorization Scheme** - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
- **Security**, **Session State Protection** - Select the level of session state protection:
  - **Unrestricted** - The item can be set by passing the item in a URL or in a form. No checksum is required in the URL.
  - **Checksum Required - Application Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace and application.
  - **Checksum Required - User Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace, application, and user.
  - **Checksum Required - Session Level** - The item can be set by passing the item in a URL that includes a checksum specific to the session.
  - **Restricted - May not be set from browser** - The item can not be altered using the URL or POSTDATA. Select this option to restrict what can set the item value to internal processes, computations, and so on. This attribute only applies to items that are not used as data entry items and is always observed, even if Session State Protection is disabled.
- **Security**, **Store value encrypted in session state** - To encrypt an item when stored in session state, set this attribute to **On**. To learn more, see About Session State and Security.

> **See Also:**
>
> - About Item Types
> - Editing Page Items

21.2.2 Identifying At Risk Password Items

Identify at risk password items by viewing the Security Profiles report and Password Items report.

At risk password items are those that either do not use a password item type that does not save session state, or store an unencrypted value in session state.

- Viewing the Security Profiles Report
  View the Security Profiles Report by navigating to Workspace Utilities and selecting it from Cross Application Reports.
- Viewing the Password Items Report
  View the Password Items Report by navigating to Workspace Utilities and selecting it from Cross Application Reports.

21.2.2.1 Viewing the Security Profiles Report

View the Security Profiles Report by navigating to Workspace Utilities and selecting it from Cross Application Reports.

To view the Security Profiles Report:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.

   The App Builder home page appears.
3. Click the **Workspace Utilities** icon.
4. Locate **Cross Application Reports** on the right side of the window.
5. Under Cross Application Reports, click **Security Profiles** report.

   This report list the following information about all applications in the current workspace:

   - Application
   - Name
   - Parsing Schema
   - Application Level Authorization Scheme
   - Authentication
   - Authorization Schemes
   - Authorization Schemes
   - Pages
   - Encrypted Items
   - At Risk Password Items

21.2.2.2 Viewing the Password Items Report

View the Password Items Report by navigating to Workspace Utilities and selecting it from Cross Application Reports.

To identify at risk password items:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.

   The App Builder home page appears.
3. Click the **Workspace Utilities** icon.
4. Locate **Cross Application Reports** on the right side of the window.
5. Under Cross Application Reports, click **Password Items**.

   The Password Items report shows all of the password items within the application and indicates if they use encryption and whether they save state. Password items that do neither are highlighted as `At Risk`.

> **Tip:**
>
> For pages that contain password items, set the page attribute **Form Auto Complete** to **Off**. Setting that attribute to **Off** prevents the web browser from attempting to auto complete items on the page.

21.2.3 Understanding Cross-Site Scripting Protection

Protect your application from a cross site-scripting security breach.

- About Cross Site-scripting Security Breaches
  Learn about cross site-scripting security breaches.
- Protecting HTML Regions and Other Static Areas
  Learn how to protect HTML regions and other static display areas.
- About Protecting Dynamic Output
  Learn about protecting dynamic output.
- About Protecting Report Regions
  Learn about protecting report regions.
- About Protecting Form Items
  Learn about protecting form items.
- About Restricting Characters Entered on an Item
  Learn about restricting characters entered on an item.

21.2.3.1 About Cross Site-scripting Security Breaches

Learn about cross site-scripting security breaches.

Cross-site scripting (also referred to as XSS) is a security breach that takes advantage of dynamically generated web pages. In an XSS attack, a web application is sent a script that activates when it is read by a user's browser. Once activated, these scripts can steal data, even session credentials, and return the information to the attacker.

If malicious code were introduced into an Oracle APEX application, it could be rendered into HTML regions and other places within the application during normal page rendering. To prevent the introduction of malicious code into session state, the APEX engine escapes characters in certain cases.

21.2.3.2 Protecting HTML Regions and Other Static Areas

Learn how to protect HTML regions and other static display areas.

- About Protecting HTML Regions and Other Static Areas
  Learn how to protect HTML regions and other static display areas.
- About Safe Item Display Types
  Learn about safe item display types.
- About the Rules Used to Determine Whether to Escape Values
  Review rules used to determine whether to escape values.
- About Using Safe Item Types to Hold and Emit HTML Markup
  Learn about using safe item types to hold and emit HTML markup.

21.2.3.2.1 About Protecting HTML Regions and Other Static Areas

Learn how to protect HTML regions and other static display areas.

In HTML regions and other static display areas, you can reference session state using the substitution strings. Special substitution strings available within a template are denoted by the number symbol (#), for example, `#ABC#`. To reference page or application items use the `&ITEM.` notation.

Examples of static display areas include HTML regions, page headers and footers, region headers and footers, region titles, button labels, help text, form item labels and post-element text, templates, radiogroup (before and after field text), event success messages, event error messages, navigation bar attributes, application static substitution string values, chart labels and legends, breadcrumbs and list framing text, and calendar text, labels, or legends.

Developers can also append an exclamation mark (`!`) followed by a predefined filter name to a page or application item name or to a report column reference to escape special characters in the substitution value.

> **See Also:**
>
> - About Using Substitution Strings
> - Controlling Output Escaping in Substitution Strings

21.2.3.2.2 About Safe Item Display Types

Learn about safe item display types.

When session state is referenced in this way, the value emitted to the page will not have special characters (`<`, `>`, `&`, `"`) escaped if the referenced item is **Display Only** and the **Format** attribute is set to **HTML**, or if the item type exposes the **Escape Special Characters** attribute which is set to **No**.

If the referenced item has a display type other than **Display Only** with the attribute **Save Session State** set to **No**, the value emitted to the page will have special characters escaped. Although application-level items are also considered to have a safe display type, they do not actually have display properties like form items do.

> **See Also:**
>
> Understanding Page Items

21.2.3.2.3 About the Rules Used to Determine Whether to Escape Values

Review rules used to determine whether to escape values.

The APEX engine uses predefined smart escaping rules to determine if and when to escape values fetched from session state.

The reason for these rules is that items that use the display types listed previously are often for text containing HTML that is intended to be emitted to the browser without being filtered (that is, escaped). The only way this can be made safe is by the enforcement of the rule that these types of items are always escaped on input to the application. For example, if a user passes some text into a safe item using an APEX `f?p` URL syntax, the APEX engine escapes special characters when saving the value into session state. This has two intended results:

1. If the value contains no special characters, the value passed in is saved into session state exactly as it was provided.
2. If the value contains special characters, those characters are escaped when the value is saved into session state.

In either situation, the item can now safely be referenced using an `&ITEM.` notation in any HTML region or other static area mentioned previously.

21.2.3.2.4 About Using Safe Item Types to Hold and Emit HTML Markup

Learn about using safe item types to hold and emit HTML markup.

You can use the safe item types listed previously to hold and emit HTML markup to the browser. For example, suppose you have a requirement to render some text in bold face by referencing a safe page item named `P1_XXX` (using `&P1_XXX.`) The item `P1_XXX` is presumed to contain the following HTML:

```
<b>ABABABAB</b>
```

You can achieve this by using application controls (computations, processes, item source expressions, item default values, and so on) to store values into these safe items. When values are introduced in this way, you ensure the safety of the content. When you use these methods, the APEX engine does not escape any special characters when saving the values into session state.

Finally, the safety of safe items is ensured by a rule that prevents those items from being posted on a page and submitted to the APEX engine as part of a page submission.

21.2.3.3 About Protecting Dynamic Output

Learn about protecting dynamic output.

Items fetched from session state and rendered using `htp.p` or other methods should be explicitly escaped by the code where it is appropriate to do so. For example, suppose a PL/SQL dynamic content region on a page uses the following:

```
htp.p(v('SOME_ITEM'));
```

If the value of the item fetched from session state could contain unintended tags or scripts, you might want to use the following in the region:

```
htp.p(apex_escape.html(:SOME_ITEM));
```

However, if you are confident that the fetched value is safe for rendering, you do not need to escape the value. As a developer, you must determine when it is appropriate to not escape output.

As a best practice, follow this rule:

- Never emit an item fetched from session state without escaping it unless the item is a safe type.

The reason for this is that as a developer, there is no way you can prevent a hacker from posting a malicious value into a non-safe item. Even if your application does not present these items visibly to ordinary users, be aware that a hacker can mount a XSS attack using your application if you do not follow this rule.

> **See Also:**
>
> About Safe Item Display Types

21.2.3.4 About Protecting Report Regions

Learn about protecting report regions.

The APEX engine escapes data rendered in the body of a report. References to session state in report headings and messages are fetched from session state using the smart escaping rules so that the values of safe item types are not escaped and the values of other item types are escaped.

APEX automatically escapes HTML special characters of a report column when the column's **Escape special characters** attribute is set to **On**. If you need to render HTML fragments instead of plain column values (for example, for highlighting), instead of concatenating the HTML fragment in the query itself (which prevents you from using Escape special characters), you should use the report column's **HTML Expression** attribute. In the **HTML Expression** attribute, you can enter static HTML and embed escaped column values with the `#COLUMN#` notation. The extended column notation gives you control regarding how APEX should escape a column value:

- `#COLUMN!HTML#` - Escapes reserved HTML characters.
- `#COLUMN!ATTR#` - Escapes reserved characters in a HTML attribute context.
- `#COLUMN!JS#` - Escapes reserved characters in a JavaScript context.
- `#COLUMN!RAW#` - Preserves the original item value and does not escape characters.

- `#COLUMN!STRIPHTML#` - Removes HTML tags from the output and escapes reserved HTML characters.

For example, suppose you have a report based on this query:

```
SELECT
    empno,
    ename,
    NULL DELETE_LINK
FROM emp
```

In this example, all columns are escaped. You could define a HTML Expression on `DELETE_LINK` as follows:

```
<a href="javascript:if (confirm('Do you really want to delete #ENAME!JS#?'))
doSubmit('DELETE-#EMPNO#');">Delete</a>
```

This example renders a link that asks if you would like to delete an employee and submits a request to delete the row if the user is confirmed. If you had not used `#ENAME!JS#` but `#ENAME#`, a name like O'Neill would cause a syntax error and an attacker could exploit the improper escaping for cross-site scripting.

> **See Also:**
>
> - Controlling Output Escaping in Substitution Strings
> - Editing Interactive Report Column Attributes

21.2.3.5 About Protecting Form Items

Learn about protecting form items.

When form items, including hidden items, obtain their values during the generation of the form page to be sent to the browser, the resulting text is escaped before rendering. Some of the safe item types are exceptions to this rule to support the intended behavior of each display type.

Some item types have the Security attribute **Escape special characters**. Use the **Escape special characters** attribute to specify whether or not the value should be escaped. To prevent Cross-Site Scripting (XSS) attacks, always set this attribute to **On**. If you need to render HTML tags stored in the page item or in the entries of a list of values, you can set this flag to **Off**. In such cases, you should take additional precautions to ensure any user input to such fields are properly escaped when entered and before saving.

Developers can also append an exclamation mark (`!`) followed by a predefined filter name to a page or application item name or to a report column reference, to escape special characters in the substitution value.

> **See Also:**
>
> Controlling Output Escaping in Substitution Strings

21.2.3.6 About Restricting Characters Entered on an Item

Learn about restricting characters entered on an item.

Limit cross-site scripting (XSS) and other injection attacks by restricting the characters users can save in session state. To accomplish this, edit the page item and configure the **Security**, **Restricted Characters** attribute. Restricted Characters can be saved in session state. Available options include:

- **All characters can be saved.**

  No restriction applies.

- **Allowlist for a-Z, 0-9 and space**

  Only allow characters a-z, A-Z, 0-9, and space.

- **Blocklist HTML command characters (<>")**.

  Reserved HTML characters can NOT be saved.

- **Blocklist &<>"/;,*|=% and --**

  The following characters: &, <, >, ", /, ;, ",", *, |, =, % and "--" (PL/SQL comment) can NOT be saved.

- **Blocklist &<>"/;,*|=% or -- and new line**

  The following characters: &, <, >, ", /, ;, ",", *, |, =, %, "--", and new line characters can NOT be saved.

If you select a restriction, APEX displays an error message if a user tries to save data which does not conform to the selected character restriction.

> **See Also:**
>
> Configuring Page Item Security

21.2.4 About Session State and Security

Learn about managing session state and security.

Oracle APEX persists session state in database tables. Session state is preserved in database tables because it is more secure to store the session state on the server side than on the client. Because APEX applications use the stateless HTTP protocol, an application's session state across multiple page views is preserved in database tables. Not maintaining a synchronous database connection for each APEX application user significantly lessens memory and database resource requirements.

Developers can query the session state stored by APEX applications using the App Builder and built-in monitoring pages. Developers and administrators can access session state for any application in the workspace to which they are authenticated.

Developer best practices for managing session state include:

- As a standard part of implementing program control flow, clear the session state of unneeded values using clear-cache page processes or clear-cache directives in URLs used to navigate to pages.

- Use password page item types that do not save state. This prevents the entered passwords from being saved in session state tables in the database.
- When sensitive data must persist in a session, it should be saved in APEX session state tables in encrypted form. To use this feature, set the page item's **Store value encrypted in session state** attribute to **On**. Encrypted stored values are automatically decrypted when read.

> **Note:**
>
> The objective of encrypted session state is to persist session state such that the unencrypted values cannot be read by other APEX developers or database administrators who might have access to the APEX session state tables, debug output, or database data files. Oracle recommends that database backups be performed using facilities that encrypt data in the backup files.

> **Tip:**
>
> You can encrypt item values up to 4000 bytes in length. Attempts to encrypt values longer than 4000 bytes produces an error message.

21.2.5 Preventing URL Tampering

Session State Protection is a built-in functionality that prevents hackers from tampering with the URLs within your application. URL tampering can adversely affect program logic, session state contents, and information privacy.

- How Session State Protection Works
  Learn session state protection works.
- Enabling Session State Protection
  Enable Session State Protection in Shared Components.
- Configuring Session State Protection
  Learn how to configure Session State Protection.

21.2.5.1 How Session State Protection Works

Learn session state protection works.

Enabling Session State Protection is a two-step process. First, you enable the Session State Protection feature in Shared Components. Second, you set page and item security attributes. You can perform these steps using a wizard, or you can set security attributes for pages and items manually on the Session State Protection page.

When enabled, Session State Protection uses the **Page Access Protection** and the item **Session State Protection** attributes with checksums positioned in `f?p=` URLs to prevent URL tampering and unauthorized access to and alteration of session state. When Session State Protection is disabled, the page and item attributes related to session state protection are ignored and checksums are not included checksums in generated `f?p=` URLs.

21.2.5.2 Enabling Session State Protection

Enable Session State Protection in Shared Components.

> **Tip:**
>
> To disable Session State Protection, perform the steps described in this topic, but select **Disable** instead of **Enable**. Disabling Session State Protection does not change existing security attribute settings, but those attributes are ignored at runtime.

To enable Session State Protection:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Security, select **Session State Protection**.

   The Session State Protection page appears. Note that the page displays the application name and current Session State Protection status (that is, **Enabled** or **Disabled**).
2. Click **Set Protection**.

   The Session State Protection page appears.
3. In the Session State Protection wizard:
   a. Select Action - Select **Enable**.

      When enabled, Session State Protection, uses the Page Access Protection page attributes and the Session State Protection item attributes in conjunction with checksums positioned in `f?p=` URLs to prevent URL tampering and unauthorized access to and alteration of session state.

      > **Tip:**
      >
      > You can also adjust page and item security attributes manually. Select **Enable**, return to the Session State Protection page, and click the appropriate icon.

   b. Click **Next**.
   c. Click **Enable**.

      Next, determine whether to set security attributes for pages and items.

21.2.5.3 Configuring Session State Protection

Learn how to configure Session State Protection.

> **Tip:**
>
> Before you can configure security attributes, you must first enable Session State Protection. See Enabling Session State Protection.

- About Configuring Session State Protection
  Learn about configuring Session State Protection.
- Reviewing Existing Session State Protection Settings
  Review a summary of Session State Protection settings for pages, items, and application items on the first page of the Session State Protection wizard.
- Configuring Session State Protection Using a Wizard
  Configure Session State Protection using a wizard.
- Configuring Session State Protection for a Page
  Configure Session State Protection for a page in Page Designer.
- Configuring Session State Protection for Page Items
  Configuring Session State Protection for page items in Page Designer.
- Configuring Session State Protection for Application Items
  Configuring Session State Protection for application items in Shared Components.

21.2.5.3.1 About Configuring Session State Protection

Learn about configuring Session State Protection.

Once you have enabled Session State Protection, the next step is to configure security attributes. You can configure security attributes in two ways:

- Use a wizard and select a value for specific attribute categories. Those selections are then applied to all pages and items within the application.
- Configure values for individual pages, items, or application items.

21.2.5.3.2 Reviewing Existing Session State Protection Settings

Review a summary of Session State Protection settings for pages, items, and application items on the first page of the Session State Protection wizard.

To view summaries of existing Session State Protection settings:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Security, select **Session State Protection**.

      The Session State Protection page appears.
2. Click **Set Protection**.
3. Expand and review the following regions at the bottom of the page:
   - **Page Level Session State Protection Summary**
   - **Page Item Session State Protection Summary**

- **Application Item Session State Protection**

21.2.5.3.3 Configuring Session State Protection Using a Wizard

Configure Session State Protection using a wizard.

When you configure Session State Protection using a wizard, you set a value for specific attribute categories. Those selections are then applied to all pages and items within the application.

To configure Session State Protection using a wizard:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the App Builder icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Security, select **Session State Protection**.

      The Session State Protection page appears.
2. Click **Set Protection**.

   The Session State Protection wizard appears.
3. Select Action - Select **Configure** and click **Next**.

   Select security attributes for pages and items. You may accept the default settings displayed here, or make new selections. Note that the value you choose for an attribute category will be applied to all pages and items throughout the application.
4. Page Access Protection:
   - **Unrestricted** - The page may be requested using a URL with or without session state arguments (Request, Clear Cache, Name/Values).
   - **Arguments Must Have Checksum** - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The checksum type must be compatible with the most stringent Session State Protection attribute of all the items passed as arguments.
   - **No Arguments Allowed** - A URL may be used to request the page but no Request, Clear Cache, or Name/Value arguments are allowed.
   - **No URL Access** - The page may not be accessed using a URL; however, the page may be the target of a Branch to Page branch type, which does not do a URL redirect.
5. Page Data Entry Item Protection:
   - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
   - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace and application. A user-level checksum or a session-level checksum will also suffice (see the next option). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
   - **Checksum Required: User Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace, application, and user. A session-level checksum will also suffice (see the next option). Use this option when you want to allow the item to be set only by URLs

having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.

- **Checksum Required: Session Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the current session. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

6. Page Display-Only Item Protection:
   - **Unrestricted** - The item may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
   - **Restricted: May not be set from browser** - The item may not be altered via the URL or POSTDATA. Use this when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is always observed, even if Session State Protection is disabled. This attribute may be used with any of these **Display As** types:
     - Display Only (Save State=No)
     - Display as Text (does not save state)
     - Display as Text (based on LOV, does not save state)
     - Display as Text (based on PLSQL, does not save state)
     - Text Field (Disabled, does not save state)
     - Stop and Start Grid Layout (Displays label only)
   - **Checksum Required: Application Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace and application. A user-level checksum or a session-level checksum will also suffice (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
   - **Checksum Required: User Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace, application, and user. A session-level checksum will also suffice (see next bullet). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.
   - **Checksum Required: Session Level** - The item's session state may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the current session. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.
7. Application Item Protection:
   - **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is only applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled. This attribute may be used for application items or for page items with any of these **Display As** types:
     - Display Only (Save State=No)
     - Text Field (Disabled, does not save state)
     - Stop and Start Grid Layout (Displays label only)

- **Checksum Required: Application Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace and application. A user-level checksum or a session-level checksum will also suffice (see next bullets). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
- **Checksum Required: User Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace, application, and user. A session-level checksum will also suffice (see next bullet). Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.
- **Checksum Required: Session Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the current session. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.

8. Click **Next**.
9. Click **Finish**.

21.2.5.3.4 Configuring Session State Protection for a Page

Configure Session State Protection for a page in Page Designer.

To configure Session State Protection for a page:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab, select the page name.

   The Property Editor displays the page attributes in the right pane. Attributes are organized in groups.
3. Find the **Security** group and edit the **Page Access Protection** attribute. Options include:
   - **Unrestricted** - The page may be requested using a URL, with or without session state arguments, and without having to have a checksum.
   - **Arguments Must Have Checksum** - If Request, Clear Cache, or Name/Value arguments appear in the URL, a checksum must also be provided. The checksum type must be compatible with the most stringent **Session State Protection** attribute of all the items passed as arguments.
   - **No Arguments Supported** - A URL may be used to request the page, but the URL can not contain Request, Clear Cache, or Name/Value Pair arguments.
   - **No URL Access** - The page may not be accessed using a URL. However, the page may be the target of a Branch to Page branch type, as this does not perform a URL redirect.
4. To save your changes, click **Save**.

21.2.5.3.5 Configuring Session State Protection for Page Items

Configuring Session State Protection for page items in Page Designer.

To configure Session State Protection for items:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In either the Rendering tab or the Layout tab, select the page item.

   The Property Editor displays the page attributes in the right pane. Attributes are organized in groups.
3. In either the Rendering tab, select the page item.
4. Find the **Security** group .
5. Under **Security**, edit the **Session State Protection** attribute. Options include:
   - **Unrestricted** - The item can be set by passing the item in a URL or in a form. No checksum is required in the URL.
   - **Checksum Required - Application Level** - The item can be set by passing the item in a URL that includes a checksum specific to the the workspace and application.
   - **Checksum Required - User Level** - The item can be set by passing the item in a URL that includes a checksum specific to the the workspace, application and user.
   - **Checksum Required - Session Level** - The item can be set by passing the item in a URL that includes a checksum specific to the the session.
   - **Restricted - May not be set from browser** - The item can not be altered using the URL or POSTDATA. Select this option to restrict what can set the item value to internal processes, computations, and so on. This attribute only applies to items that are not used as data entry items and is always observed, even if Session State Protection is disabled. Use this attribute for page or application items that have the following **Display As** types:
     – **Display Only (Save State=No)**
     – **Text Field (Disabled, does not save state)**
6. To save your changes, click **Save**.

21.2.5.3.6 Configuring Session State Protection for Application Items

Configuring Session State Protection for application items in Shared Components.

To configure Session State Protection for an application item:

1. Navigate to the Session State Protection page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Application Logic, select **Application Items**.

The Application Items page appears.

2. Click the name of an **Application Item**.
3. Security, Session State Protection - Select Session State Protection for this item. Options include:
   - **Unrestricted** - The item's session state may be set by passing the item name/value in a URL or in a form. No checksum is required in the URL.
   - **Restricted - May not be set from browser** - The item may not be altered using the URL or POSTDATA. Use this option when you want to restrict the way that the item value can be set to internal processes, computations, and so on. This attribute is only applicable only to items that cannot be used as data entry items and is always observed even if Session State Protection is disabled. This attribute may be used for application items or for page items with any of these **Display As** types:
     - **Display Only (Save State=No)**
     - **Text Field (Disabled, does not save state)**
     - **Stop and Start Grid Layout (Displays label only)**
   - **Checksum Required: Application Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace and application. Use this option when you want to allow the item to be set only by URLs having checksums that were generated by any user running the same application in the current workspace but in a different session.
   - **Checksum Required: User Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the workspace, application, and user. Use this option when you want to allow the item to be set only by URLs having checksums that were generated by the same named user, running the same application in the current workspace but in a different session.
   - **Checksum Required: Session Level** - The item may be set by passing the item name/value in a URL if a checksum is also provided that is specific to the current session. Use this option when you want to allow this item to be set only by URLs having checksums that were generated in the current session.
4. Click **Apply Changes**.

21.2.6 About Securing File Uploads

Learn about developer best practices for securing file uploads.

Oracle APEX enables you to easily build an application that can be used to upload files and to access uploaded files. These files are uploaded into a common file storage table. Although the database view `APEX_APPLICATION_FILES` shows those files associated with your database account (or workspace), programmatic access to the common file storage table does not always require authentication, enabling other users to see your uploaded files. For this reason, Oracle recommends that developers use the methods described in Understanding BLOB Support in Forms and Reports, with the files being uploaded directly to a table in your workspace schema, or if you need programmatic access, uploaded to `APEX_APPLICATION_TEMP_FILES`. When you use the built-in methods of file access, the authorization checks on the page also apply to file access. Storing uploaded files in a table in your workspace schema also gives you more control over the underlying database storage. For example, you can use DDL operations, such as TRUNCATE and LOB segment move, to reclaim unused storage.

Note:

The database view `APEX_APPLICATION_FILES` supports INSERT, UPDATE, and DELETE DML operations. The database view `APEX_APPLICATION_TEMP_FILES` supports DELETE DML operations. You can perform these operations as part of your application or within the context of your workspace, for example in SQL Commands within SQL Workshop.

See Also:

- About the Differences Between Page Items and Application Items
- File Upload to learn more about creating a File Browse page item

21.3 Controlling Access to Applications, Pages, and Page Components

Control access to an application, individual pages, or page components by creating an access control list.

Tip:

To create an access control page, the application schema must have `CREATE TABLE`, `CREATE TRIGGER`, and `CREATE SEQUENCE` privileges.

- Understanding Access Control
  Adding the Access Control feature to an application, creates multiple pages and the following components: an Access Control region, access roles, authorization schemes, a build option, and an Application Setting.
- Creating an Access Control Using the Create Page Wizard
  Create an access control list to associate application roles with application users.
- Managing User Access
  Manage user access on the Administration page.
- Configuring Access Control
  Configure user access on the Administration page.
- About Controlling Access for Pages and Page Components
  Control access to a specific page or page component by editing the page or component and selecting an authorization schemes.
- About Removing an Access Control Created with a Wizard
  Remove an access control created with a wizard using build options.
- Managing Roles and User Assignments
  Manage application access control roles and user role assignments on the Application Access Control page.

> **See Also:**
>
> Managing Roles and User Assignments

21.3.1 Understanding Access Control

Adding the Access Control feature to an application, creates multiple pages and the following components: an Access Control region, access roles, authorization schemes, a build option, and an Application Setting.

- About Adding Access Control
  Learn about adding Access Control.
- About Access Control Authorization Schemes
  Learn about Access Control authorization schemes.
- About Configuring Access Control
  Configure Access Control by running the application and accessing the Access Control region on the Administration page.
- About Exporting an Application with Access Control
  Learn about exporting an application with Access Control.

21.3.1.1 About Adding Access Control

Learn about adding Access Control.

Running the Access Control Wizard creates multiple pages and the following components:

- Adds an Access Control region to the Administration page you specify.
- Creates the access roles: `Administrator`, `Contributor`, and `Reader`.
- Creates the authorization schemes: `Administration Rights`, `Contribution Rights`, and `Reader Rights`.

> **Note:**
>
> When you add a new Access Control to an existing application, these authorization schemes are only be created if the names do not exist. For example, if `Administration Rights` already exists (case sensitive comparison), the wizard will not recreate. Instead, the Access Control page will re-use the existing authorization scheme.

- Creates the build option, `Feature: Access Control`.
- Creates the Application Setting, `ACCESS_CONTROL_SCOPE`.

Developers use the access control list to associate the privileges, `view`, `edit`, and `administration`, with application users. Within the final Access Control UI, each privileges correlates to an access role:

- `View` correlates to the `Reader` role.
- `Edit` correlates to the `Contributor` role.

- `Administration` correlates to the `Administrator` role.

> **See Also:**
>
> - Attaching an Authorization Scheme to an Application, Page, or Components
> - Managing Roles and User Assignments

21.3.1.2 About Access Control Authorization Schemes

Learn about Access Control authorization schemes.

When you add the Access Control feature to an application, the PL/SQL Body Wizard creates the following authorization schemes:

- `Administration Rights` – This authorization scheme checks if the current user in the application is assigned `ADMINISTRATOR` role.

```
return APEX_ACL.HAS_USER_ROLE (
     p_application_id=>:APP_ID,
     p_user_name => :APP_USER,
     p_role_static_id => 'ADMINISTRATOR');
```

- `Contribution Rights` – This authorization scheme checks if the current user in the application is assigned the `ADMINISTRATOR` role or the `CONTRIBUTOR` role.

```
if apex_acl.has_user_role (
  p_application_id=>:APP_ID,
  p_user_name => :APP_USER,
  p_role_static_id => 'ADMINISTRATOR') or
  apex_acl.has_user_role (
    p_application_id=>:APP_ID,
    p_user_name=> :APP_USER,
    p_role_static_id=> 'CONTRIBUTOR') then
    return true;
else
    return false;
end if;
```

- `Reader Rights` – This authorization scheme returns TRUE if the access control is configured to allow any authenticated user access the application. If this behavior is not allowed, it checks if the current user in the application is assigned to any application role.

```
if nvl(apex_app_setting.get_value(
   p_name => 'ACCESS_CONTROL_SCOPE'),'x') = 'ALL_USERS' then
    -- allow user not in the ACL to access the application
    return true;
else
    -- require user to have at least one role
    return apex_acl.has_user_any_roles (
        p_application_id => :APP_ID,
```

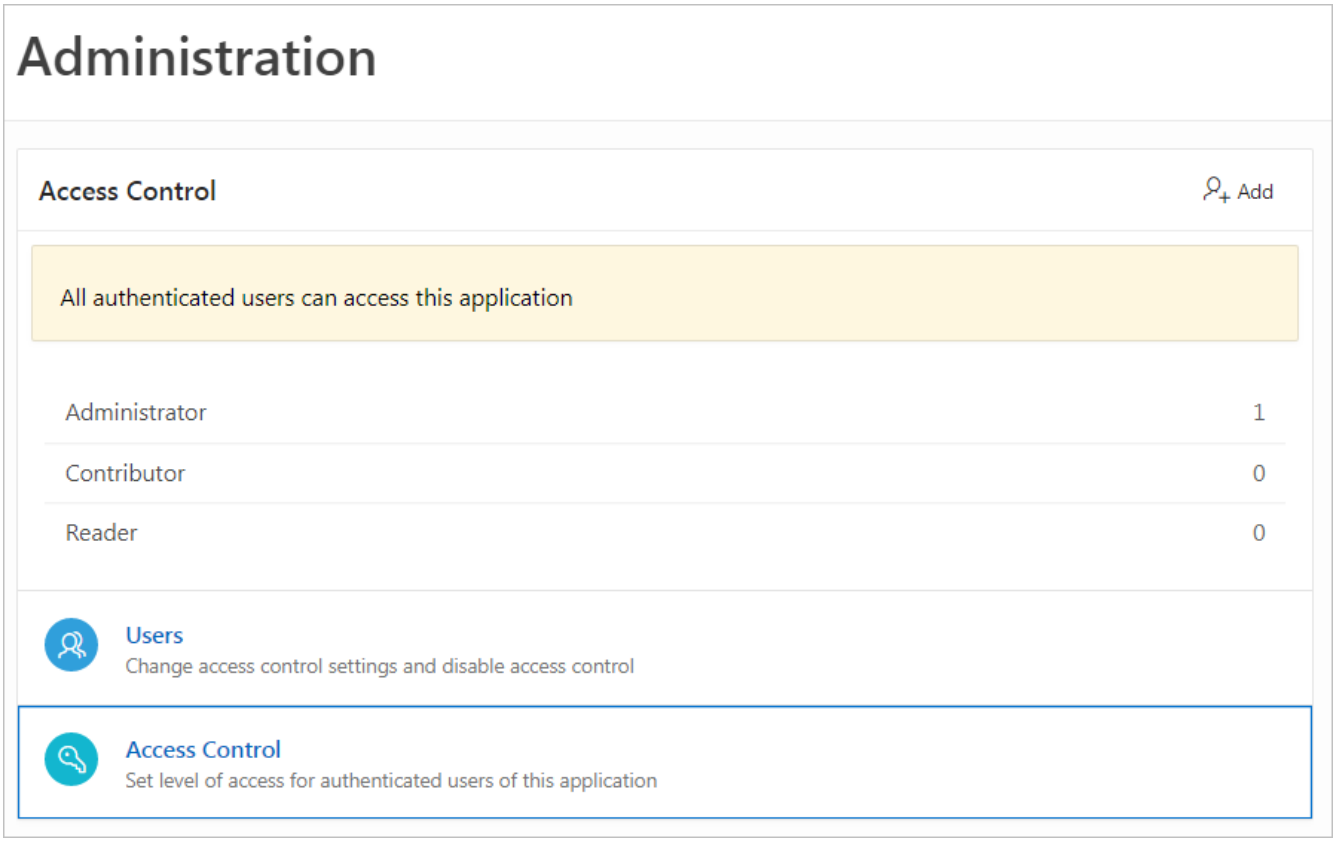
```
        p_user_name      => :APP_USER);
end if;
```

> **See Also:**
>
> Attaching an Authorization Scheme to an Application, Page, or Components

21.3.1.3 About Configuring Access Control

Configure Access Control by running the application and accessing the Access Control region on the Administration page.

Once you add the Access Control feature, you configure it by running the application and accessing the Access Control region on the Administration page.



The Access Control region lists currently defined access roles and contains two sections: **Users** and **Access Control**.

Users

Click **Users** to add new users, change a user's role, or disable access control by locking an account.

> **Tip:**
>
> You add additional roles and configure role assignments on the Shared Components, Application Access Control page. See Managing Roles and User Assignments.

Access Control

Click **Access Control** to specify the behavior when authenticated users access the application.

For **Any authenticated user may access this application**, select one of the following:

- **Off** - Choose **Off** if all defined users are included in the access control list.
- **On** - Choose **On** if authenticated users not in the access control list may also use this application.

> **See Also:**
>
> - Managing User Access
> - Configuring Access Control

21.3.1.4 About Exporting an Application with Access Control

Learn about exporting an application with Access Control.

When your export an application with the Access Control feature, the application roles, Administrator, Contributor, and Reader, are exported. However, the users assigned to these roles are not exported. If you deploy a exported application with the Access Control feature, navigation menu entry for Administration page will not display. When you deploy an application with Access Control feature, your can add user roles as needed by going to Shared Components, Application Access Control. If the application is being deployed in a runtime environment, you can add user roles using `APEX_ACL` API. For example, the following example adds the user name 'SCOTT' as Administrator in application 255:

```
begin
    APEX_ACL.ADD_USER_ROLE (
        p_application_id => 255,
        p_user_name      => 'SCOTT',
        p_role_static_id => 'ADMINISTRATOR' );
end;
```

You can also execute the `APEX_ACL` API from the command line or create an install script in application supporting objects.

See Also:

- Attaching an Authorization Scheme to an Application, Page, or Components
- Managing Roles and User Assignments
- APEX_ACL in *Oracle APEX API Reference*

21.3.2 Creating an Access Control Using the Create Page Wizard

Create an access control list to associate application roles with application users.

Create an access control list by running the Create Page Wizard and selecting **Features** and then **Access Control**. The wizard also adds a **reader**, **contributor** and **administrator** role and corresponding authorization scheme to your application. Apply these authorization schemes to pages and page components to manage access by user and role.

Tip:

This section describes how to create an access control list by running the Create Page Wizard from the Application home page. You can also run this wizard in Page Designer by clicking the **Create** menu and selecting **Page**.

To create an access control list using the Create Page Wizard:

1. On the Workspace home page, click **App Builder**.
2. Select an application.
3. Click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Click **Feature** and select **Access Control**.
5. For Page Type, select **Feature** and then **Access Control**.
6. On the Create Access Control Pages:
   a. Starting Page Number - Enter an unused starting page number. This feature creates multiple pages starting with the number specified. The default is calculated as the first available page number that is divisible by 10, which is greater than the maximum page number within this application..
   b. Page Group - Identify the name of the page group you to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list.
   c. Build Option - The wizard creates the displayed build option to support this feature. You can use the build option to control whether the feature appears or to remove it later on.
   d. Administration Page Preference - Specify the administration page to which this access control list will be added. Options include:
      - **Create a new page**

- **Identify an existing page**

  What appears next depends upon your selection. Follow the on-screen instructions.

e. Existing Administration Page - Select an existing page used as the administration page in this application.

f. Click **Next**.

A Confirmation page appears.

7. Click **Create**.

21.3.3 Managing User Access

Manage user access on the Administration page.

Use Manage User Access page to add new users, change a user's role, or disable access control by locking an account.

> **Tip:**
>
> You can also add users by clicking the **Add** button adjacent to the Access Control region heading.

To manage user access:

1. Run the application and navigate to Access Control.

   By default, navigate to the **Administration** page and locate **Access Control** region.

2. Under Access Control, click **Users.**

   The Manage User Access appears.

3. To add a user:

   a. Click the **Add User** .

   b. On Manage User Access, edit the attributes:

      - Username
      - Email
      - Role - Select role (for example, **Administrator**, **Contributor**, or **Reader**).

   c. Click **Add User**.

4. To edit an existing user:

   a. Click the **Edit** icon adjacent to the Username.

   b. On Manage User Access, edit the following:

      - Username
      - Email
      - Role - Select role (for example, **Administrator**, **Contributor**, or **Reader**).
      - Account Locked - Select **Yes** or **No**.

   c. Click **Apply Changes**.

5. To add multiple users:

   a. Role - Select a role (such as **Administrator**, **Contributor**, or **Reader**).
   b. Username - Enter usernames separated by commas, semicolons, or white space. Existing or duplicate usernames are automatically be ignored.
   c. Confirm the additions and click **Apply Changes**.
6. To delete a user:
   a. Click the **Edit** icon adjacent to the Username.
   b. Click **Delete**
7. To revert you changes, click **Reset**.

21.3.4 Configuring Access Control

Configure user access on the Administration page.

Use Configure Access Control to specify the behavior when authenticated users access the application, determine the default role for users not in the access control list, and determine the userame format (email or string).

To configure access control:

1. Run the application and navigate to Access Control.

   By default, navigate to the **Administration** page and locate **Access Control** section.
2. Under Access Control, click **Access Control**.
3. In the Configure Access Control dialog, edit **Any authenticated user may access this application**:
   - **Off** - Choose **Off** if all defined users are included in the access control list.
   - **On** - Choose **On** if authenticated users not in the access control list may also use this application.
4. To save your changes, click **Apply Changes**.

21.3.5 About Controlling Access for Pages and Page Components

Control access to a specific page or page component by editing the page or component and selecting an authorization schemes.

The Access Control Wizard creates authorization schemes that correspond to the application mode list options and the privileges available in the Access Control List.

You can control access to a specific page or page component by editing the page or component and selecting one of the following authorization schemes:

- **Administration Rights** - Only users with Administrator privileges can view the page or component.
- **Contribution Rights** - Users with both Edit and Administrator privileges can view the page or component. Users with View privileges cannot view the page or component.
- **Reader Rights** - Users with Administrator, Edit, or View privileges can view the page or component.
- **Not Administration Rights** - Users with Administrator privileges cannot view the page or component.
- **Not Contribution Rights** - Users with both Edit and Administrator privileges cannot view the page or component. Users with View privileges can view the page or component.

- **Not Reader Rights** - Users with Administrator, Edit, or View privileges cannot view the page or component.

> **See Also:**
>
> Attaching an Authorization Scheme to an Application, Page, or Components

21.3.6 About Removing an Access Control Created with a Wizard

Remove an access control created with a wizard using build options.

When you add an access control list using a wizard, the wizard creates one or more pages and other components and processes to seamlessly integrate it into your application. Each feature is associated with a build option which contains one or more components. You use the associated build option to enable, disable, or permanently remove features.

Using Build Options to Include or Exclude Features

Build options have two possible values: **Include** and **Exclude**. If you select the build option status of **Include**, then the Oracle APEX engine considers the associated components (in this case features) as part of the application definition at runtime. Conversely, if you specify the build options status as **Exclude**, then the Oracle APEX engine treats it and any associated components as if it did not exist.

Removing Features Permanently

You remove features created with a wizard by first removing the components associated with the build option and then deleting the build option.

> **See Also:**
>
> - Including or Excluding Build Options
> - Deleting Build Options and Associated Components

21.3.7 Managing Roles and User Assignments

Manage application access control roles and user role assignments on the Application Access Control page.

> **Tip:**
>
> You also use the `APEX_ACL` API. See APEX_ACL in *Oracle APEX API Reference*

- About Application Access Control
  Learn about managing access control application users and roles.

- Creating Access Control Roles
  Create application access control roles.
- Copying or Subscribing to Access Control Roles
  Copy access control roles from the current application or from another application. When copying an access control role from another app, you can also subscribe to it.
- Subscribing to Access Control Roles
  Subscribe to an access control role in the Role dialog.
- Deleting Access Control Roles
  Learn about deleting application access control roles.
- Adding User Role Assignments
  Define additional user role assignments on the Application Access Control page.
- Editing User Role Assignments
  Edit or delete user role assignments.

21.3.7.1 About Application Access Control

Learn about managing access control application users and roles.

You create an access control list by running the Access Control Wizard from either the Create Application Wizard or Create Page Wizard. The Access Control Wizard creates a page to manage an access control list and creates two tables within the application's default parsing schema to manage the access control list. Use the access control list within the application to associate the privileges (`view`, `edit`, and `administration`, with application users. Each privileges correlates to an access level role:

- `View` correlates to the `READER` role.
- `Edit` correlates to the `CONTRIBUTOR` role.
- `Administration` correlates to the `ADMINISTRATOR` role.

To control access to application pages and components, you need to create an Authorization Scheme and associate it with the application.

About Defining Additional Roles

You can define additional roles on the Application Access Control page. Since roles are applied to users you must create the roles before adding users. Roles and users defined on the Application Access Control page can be reviewed using the following view:

- `APEX_APPL_ACL_USERS`
- `APEX_APPL_ACL_USER_ROLES`
- `APEX_APPL_ACL_ROLES`

> **See Also:**
>
> Attaching an Authorization Scheme to an Application

21.3.7.2 Creating Access Control Roles

Create application access control roles.

> **Tip:**
>
> Since roles are applied to users, you must create the roles before adding users.

To create an application access control role:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Security, select **Application Access Control**.

   The Application Access Control page appears.
3. Under Roles, click **Add Role**.

   The Role dialog appears.
4. On Role:
   a. Name - Enter a descriptive name for this role. Name may only contain alphanumeric characters and underscores (_).
   b. Static Identifier - Alternate application identifier for this role.
   c. Description - Enter an optional description of this role.
5. Click **Create Role**.

   The new role displays under Roles on the Application Access Control page.

21.3.7.3 Copying or Subscribing to Access Control Roles

Copy access control roles from the current application or from another application. When copying an access control role from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to access control roles:

1. Navigate to the Application Access Control page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.

c. Select an application.

d. On the Application home page, click **Shared Components**.

e. Under Security, click **Application Access Control**.

The Application Access Control page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. To copy a role within the current application:

a. Search for the role and click **Copy** in the Copy column.

The Copy Role Wizard appears.

b. New Role Name - Enter the name of the role. This name must be unique within the application.

c. Click **Copy Role**.

3. To copy a role from another application:

a. From the Tasks region, click **Copy Role from another app**.

The Copy Role Wizard appears.

b. Application - Select the application from which you want to copy.

c. Copy Role - Select the role to copy.

d. Subscribe - Set this flag to **On** to subscribe to the role.

e. Click **Copy Role**.

21.3.7.4 Subscribing to Access Control Roles

Subscribe to an access control role in the Role dialog.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Note:**
>
> You can also subscribe to a role by copying it and then subscribing. See Copying or Subscribing to Access Control Roles.

To subscribe to an access control role:

1. Navigate to the Shared Components page:

a. On the Workspace home page, click **App Builder**.

b. Select an application.

c. On the Application home page, click **Shared Components**.

d. Under Security, click **Application Access Control**.

The Application Access Control page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. Under Roles, select the role.

The Role dialog appears.

3. To subscribe to an access control role:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master role to subscribe to and click **Apply Changes**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscribed access control role:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Role - If the access control role is already subscribed, click **Refresh Role**.
5. To unsubscribe from a access control role:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

21.3.7.5 Deleting Access Control Roles

Learn about deleting application access control roles.

To delete a role:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Security, select **Application Access Control**.

   The Application Access Control page appears.
3. Under Roles, select the role.

   The Role dialog appears.
4. Click **Delete**.

21.3.7.6 Adding User Role Assignments

Define additional user role assignments on the Application Access Control page.

To add user role assignments:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Security, select **Application Access Control**.

   The Application Access Control page appears.
3. Under User Role Assignments, click **Add User Role Assignment**.

   The User Assignment dialog appears.

4. On User Assignment:
   a. **User Name** - Enter a descriptive name for this role. Name may only contain alphanumeric characters and underscores (_).
   b. **Application Role** - Select a role.
5. Click **Create Assignment**.

   The new user assignment displays under User Role Assignments.

> **Tip:**
>
> Application users are not exported as part of your application. When you deploy your application you will need to manually manage your user to role assignments. Roles are exported as part of an application export and imported with application imports.

21.3.7.7 Editing User Role Assignments

Edit or delete user role assignments.

To edit user role assignments:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Security, select **Application Access Control**.

   The Application Access Control page appears.
3. To edit an existing user role assignment:
   a. Under User Role Assignments, select a user name.

      The User Assignment dialog appears.
   b. For Application Role, celect a new role.
   c. Click **Save**.
4. To delete a user role assignment:
   a. Under User Role Assignments, select a user name.

      The User Assignment dialog appears.
   b. Click **Delete**.

> **Tip:**
>
> Application users are not exported as part of your application. When you deploy your application you will need to manually manage your user to role assignments. Roles are exported as part of an application export and imported with application imports.

> **See Also:**
>
> Exporting Workspaces, Applications, and App Components

21.4 Establishing User Identity Through Authentication

Use authentication to establish a user's identity to control access to an application. Authentication may require a user identify a user name and password or could involve the use of digital certificates or a secure key.

- Understanding Authentication
  Learn about authentication.
- Understanding Preconfigured Authentication Schemes
  Learn about the preconfigured authentication schemes.
- Viewing the Authentication Scheme Associated with an Application
  View the current authentication scheme associated with an application.
- Creating an Authentication Scheme
  To associate an authentication scheme with an application, you create an authentication scheme on the Shared Components page.
- Configuring an Existing Authentication Scheme
  Configure various settings and program hooks in an existing authentication scheme.
- Changing the Authentication Scheme Associated with an Application
  Change the authentication scheme associated with an application.
- Copying or Subscribing to an Authentication Scheme
  Copy an authentication scheme from the current application or from another applications. When copying an authentication scheme from another app, you can also subscribe to it.
- Subscribing to an Authentication Scheme
  Subscribe to an authentication scheme on the Authentication Schemes Create / Edit page.
- Creating a Login Page
  Create an application login page.
- Using a Procedure to Configure Authentication at Runtime
  Configure authentication at runtime by specifying a procedure on the Security page.
- Viewing Authentication Scheme Reports
  View authentication scheme reports.

21.4.1 Understanding Authentication

Learn about authentication.

- How Authentication Works
  Lean how authentication works.
- About Support for Deep Linking
  Learn about support for deep linking.
- About Determining Whether to Include Authentication
  Learn about determining whether to include application authentication.

21.4.1.1 How Authentication Works

Lean how authentication works.

You determine how your application interacts with users. If all users have the same rights and privileges, they are referred to as public users. However, if your application must track each user individually, you must specify an authentication method.

Authentication establishes the identity of each user who accesses your application. Many authentication processes require that a user provide some type of credentials such as a user name and password. These credentials are then evaluated and they either pass or fail. If the credentials pass, the user has access to the application. Otherwise, access is denied.

Once a user has been identified, the Oracle APEX engine keeps track of each user by setting the value of the built-in substitution string `APP_USER`. As a user navigates from page to page, the APEX engine sets the value of `APP_USER` to identify the user. The APEX engine uses `APP_USER` as one component of a key for tracking each user's session state.

From a programming perspective, you can access `APP_USER` using the following syntax:

- As a bind variable from either PL/SQL or SQL:

  ```
  :APP_USER
  ```

- From PL/SQL packages and triggers:

  ```
  V('APP_USER')
  ```

- As an attribute of the context APEX$SESSION:

  ```
  sys_context('APEX$SESSION', 'APP_USER')
  ```

You can use `APP_USER` to perform your own security checks and conditional processing. For example, suppose you created the following table:

```
CREATE TABLE my_security_table (
  user_id   VARCHAR2(30),
  privilege VARCHAR2(30));
```

Once created, you could populate this table with user privilege information and then use it to control the display of pages, tabs, navigation bars, buttons, regions, or any other control or component.

> **See Also:**
>
> - APP_USER
> - Configuring Security Attributes

21.4.1.2 About Support for Deep Linking

Learn about support for deep linking.

APEX applications that use authentication schemes support deep linking. Deep linking refers to the ability to link to an APEX page out of context (for example, from a hyperlink in an email or workflow notification). When you link to a page out of context and the application requires the user be authenticated, the user is taken to the Sign In page. After credentials verification,

the APEX engine automatically displays the page that was referenced in the original link. Deep linking is disabled by default. Application developers can enable deep linking at the application-level on the Security Attributes page in Shared Components or for individual pages in Page Designer.

21.4.1.3 About Determining Whether to Include Authentication

Learn about determining whether to include application authentication.

As you create your application, you must determine whether to include authentication. Options include:

- **Choose to not require authentication**. APEX does not check any user credentials. All pages of your application are accessible to all users.
- **Select a built-in authentication scheme**. Create an authentication method based on available preconfigured authentication schemes. Depending on which scheme you choose, you may also have to configure the corresponding components of Oracle 10*gi*AS, Oracle Internet Directory, or other external services.
- **Create custom authentication scheme**. Create a custom authentication method to have complete control over the authentication interface. To implement this approach, you must provide a PL/SQL function the APEX engine executes before processing each page request. This function's Boolean return value determines whether the APEX engine processes the page normally or displays a failure page.

> **See Also:**
>
> - Understanding Preconfigured Authentication Schemes
> - Changing the Authentication Scheme Associated with an Application
> - Editing an Authorization Scheme

21.4.2 Understanding Preconfigured Authentication Schemes

Learn about the preconfigured authentication schemes.

When you create an authentication scheme from the gallery, you can select a preconfigured authentication scheme which follows a standard behavior for authentication and session management. To associate an authentication scheme with an application, you select the application and create a new authentication scheme. A newly created authentication scheme is not enabled by default. To enable the new authentication scheme for an application, edit the authentication scheme and click **Make Current Scheme**.

This section describes all preconfigured authentication schemes that ship with Oracle APEX.

> **Tip:**
>
> Authentication schemes support plug-ins. To learn more, see Implementing Plug-ins.

- Builder Extension Sign-in
  The Extension Builder Sign-in enables users to log into an Extension App without having to sign-in again when already signed into a APEX session. The Extension Sign-in Authentication checks for a valid APEX session.
- Custom Authentication
  Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.
- Database Accounts
  Database Account Credentials authentication utilizes database schema accounts to authenticate users.
- HTTP Header Variable
  Authenticate users externally by storing the username in a HTTP Header variable set by the web server.
- LDAP Directory
  Authenticate a user and password with an authentication request to a LDAP server.
- No Authentication (using DAD)
  Adopts the current database user. This approach can be used in combination with a mod_plsql Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.
- Open Door Credentials
  Enable anyone to access your application using a built-in login page that captures a user name.
- Oracle APEX Accounts
  Oracle APEX Accounts are user accounts that are created within and managed in the APEX user repository. When you use this method, your application is authenticated against these accounts.
- Oracle Application Server Single Sign-On Server
  Delegates authentication to the Oracle AS Single Sign-On (SSO) Server. To use this authentication scheme, your site must have been registered as a partner application with the SSO server.
- SAML Sign-In
  Delegates authentication to the Security Assertion Markup Language (SAML) Sign In authentication scheme.
- Social Sign-In
  Social Sign-In supports authentication with Google, Facebook, and other social networks and enterprise identity providers that support OpenID Connect or OAuth2 standards.

See Also:

- Changing the Authentication Scheme Associated with an Application
- Configuring Authentication Schemes for an Instance in *Oracle APEX Administration Guide*

21.4.2.7 Open Door Credentials

Enable anyone to access your application using a built-in login page that captures a user name.

Open Door Credentials enables anyone to access your application using a login page that captures a user name. This authentication scheme is useful during application development.

- Setting Up Open Door Credentials
  Learn how to set up Open Door Credentials authentication.

21.4.2.7.1 Setting Up Open Door Credentials

Learn how to set up Open Door Credentials authentication.

To set up Open Door Credentials authentication:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Open Door Credentials**.
8. Click **Create Authentication Scheme**.

21.4.2.1 Builder Extension Sign-in

The Extension Builder Sign-in enables users to log into an Extension App without having to sign-in again when already signed into a APEX session. The Extension Sign-in Authentication checks for a valid APEX session.

- About Builder Extension Sign-in
  Learn about Builder Extension Sign-in.
- Setting Up Builder Extension Sign-in Authentication
  Learn how to set up Builder Extension Sign-in authentication.

21.4.2.1.1 About Builder Extension Sign-in

Learn about Builder Extension Sign-in.

The Builder Extension Sign-in Authentication Scheme enables users to open the Extension app from a APEX session without having to sign-in again with a username and password. Use this authentication scheme with Builder Extension apps published from a workspace that has been configured to host and publish Builder Extension apps.

Administrator Controls

Administrators can enable or disable support for hosting of Builder Extensions at the workspace-level by configuring the **Allow Hosting Extensions** setting on the Edit Workspace Information page. To learn more, see Editing Workspace Information in *Oracle APEX Administration Guide*.

> **See Also:**
>
> Managing Links on the Extension Menu in *Oracle APEX Administration Guide*

21.4.2.1.2 Setting Up Builder Extension Sign-in Authentication

Learn how to set up Builder Extension Sign-in authentication.

To set up Builder Extension Sign-in:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Builder Extension Sign-in**.
   c. Click **Create Authentication Scheme**.

21.4.2.2 Custom Authentication

Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.

- About Custom Authentication
  Learn about Custom authentication.
- Setting Up Custom Authentication
  Learn how to set up Custom authentication.
- About Session Management Security
  Learn about session management security.

21.4.2.2.1 About Custom Authentication

Learn about Custom authentication.

Custom authentication is the best approach for applications when any of the following is true:

- Database authentication or other methods are not adequate.
- You want to develop your own login form and associated methods.
- You want to control security aspects of session management.
- You want to record or audit activity at the user or session level.
- You want to enforce session activity or expiry limits.

- You want to program conditional one-way redirection logic before Oracle APEX page processing.
- You want to integrate your application with non-APEX applications using a common session management framework.
- Your application consists of multiple applications that operate seamlessly (for example, more than one application ID).

> **Tip:**
>
> If you are planning on using the same authentication scheme for multiple applications, consider writing a custom authentication plug-in. See Implementing Plug-ins.

21.4.2.2.2 Setting Up Custom Authentication

Learn how to set up Custom authentication.

To create a custom authentication scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Custom**.
8. Fill in the appropriate fields.

   To learn more about a specific field, see field-level Help.
9. Click **Create Authentication Scheme**.

> **See Also:**
>
> Viewing Field-Level and Page Designer Help

21.4.2.2.3 About Session Management Security

Learn about session management security.

When running custom authentication, APEX attempts to prevent two improper situations:

- Intentional attempts by a user to access session state belonging to someone else. However, users can still type in an arbitrary application session ID into the URL.
- Inadvertent access to a stale session state (probably belonging to the same user from an earlier time). This would commonly result from using bookmarks to application pages.

APEX checks that the user identity token set by the custom authentication function matches the user identity recorded when the application session was first created. If the user has not yet been authenticated and the user identity is not yet known, the session state being accessed does not belong to someone else. These checks determine whether the session ID in the request can be used. If not, the APEX engine redirects back the same page using an appropriate session ID.

21.4.2.3 Database Accounts

Database Account Credentials authentication utilizes database schema accounts to authenticate users.

- About Database Account Credentials
  Learn about Database Account Credentials.
- Setting Up Database Account Credentials
  Learn how to set up Database Account Credentials.

21.4.2.3.1 About Database Account Credentials

Learn about Database Account Credentials.

Database Accounts Credentials authentication requires that a database user (schema) exist in the local database and is unlocked. This authentication method validates end user's credentials against the database accounts repository. Unlike database clients, such as SQL Developer, this authentication method does not establish a new database session for this user. Application code is still executed as the application parsing schema, not as the end user's database schema.

Choose Database Accounts Credentials if having one database account for each named user of your application is feasible and account maintenance using database tools meets your needs (for example, changing end user's password using the `ALTER USER` statement).

21.4.2.3.2 Setting Up Database Account Credentials

Learn how to set up Database Account Credentials.

To set up Database Account Credentials:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.

b. Scheme Type - Select **Database Accounts**.

c. Click **Create Authentication Scheme**.

21.4.2.4 HTTP Header Variable

Authenticate users externally by storing the username in a HTTP Header variable set by the web server.

- About HTTP Header Variable
  Learn about HTTP Header Variable authentication.
- Setting Up HTTP Header Variable
  Learn how to set up HTTP Header Variable authentication.

21.4.2.4.1 About HTTP Header Variable

Learn about HTTP Header Variable authentication.

HTTP Header Variable supports the use of header variables to identify a user and to create an Oracle APEX user session. Use HTTP Header Variable authentication scheme if your company employs a centralized web authentication solution like Oracle Access Manager which provides single sign-on across applications and technologies. User credential verification is performed by these systems and they pass the user's name to APEX using a HTTP header variable such as "REMOTE_USER" (which is the default).

21.4.2.4.2 Setting Up HTTP Header Variable

Learn how to set up HTTP Header Variable authentication.

To set up HTTP Header Variable authentication:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:

   a. Name - Enter the name used to reference the authentication scheme by other application developers.

   b. Scheme Type - Select **HTTP Header Variable**.
8. Under Settings:

   To learn more about a specific field, see field-level Help.

   a. HTTP Header Variable Name - Specifies the name of the HTTP header variable which contains the username. If not specified, `REMOTE_USER` will be used. The HTTP header variable is a variable set by the web server.

   b. Action if Username is Empty - Specifies the action which should be performed if the username stored in the HTTP header variable is empty. Options include:

- **Redirect to Built-In URL** - Initiates a login by the web server. This Built-In URL has to be setup to be protected by the web server, in order to force a login when called. After login, the web server must store the validated username in the HTTP header variable, so it is available to the authentication scheme. The Built-In URL is:

  `/apex/apex_authentication.callback`

  Where The prefix `/apex/` depends on your server configuration.

- **Redirect to URL** - Initiates a login on an external server which then sets the validated username in the HTTP header variable. The external server should redirect back to the application using the URL generated by the #CALLBACK# placeholder to complete authentication in the application.
- **Display Error** - Displays the specified error message and no login will be performed in the application.

c. Verify Username - Specifies how often the username stored in the HTTP header variable is verified. Options include:

- **Each Request** - Will expect that the CGI variable is always set as soon as the login has been performed by the web server. If the username is not identical to the one stored in the current APEX session, the session will be invalidated and a new login will be initiated. This is the most secure option because it detects logouts or username changes.
- **After Login** - Will only verify and store the username in the APEX session after the web server has performed the login and calls the callback to complete authentication in the application. For callback see "Action if Username is Empty" configuration.

d. Logout URL of SSO Server - If the authentication scheme is based on Oracle Access Manager or similar servers, you can use this attribute to specify a URL to log out of the central single sign-on server. The substitution parameter `%POST_LOGOUT_URL%` will be replaced by an encoded URL to the login page of your application.

   Oracle Access Manager based SSO example:

   `/oamsso/logout.html?end_url=%POST_LOGOUT_URL%`

9. Click **Create Authentication Scheme**.

21.4.2.5 LDAP Directory

Authenticate a user and password with an authentication request to a LDAP server.

- About LDAP Directory Verification
  Learn about LDAP Directory Verification.
- Setting Up LDAP Directory Verification
  Learn how to set up LDAP Directory Verification.

21.4.2.5.1 About LDAP Directory Verification

Learn about LDAP Directory Verification.

You can configure any authentication scheme that uses a login page to use Lightweight Directory Access Protocol (LDAP) to verify the user name and password submitted on the login page.

App Builder includes wizards and edit pages that explain how to configure this option. These wizards assume that an LDAP directory accessible to your application for this purpose already

exists and that it can respond to a `SIMPLE_BIND_S` call for credentials verification. When you create an LDAP Credentials authentication scheme, the wizard requests and saves the LDAP host name, LDAP port, DN string, and determines whether to use SSL, exact DN, and optionally a search filter if not using exact DN. An optional preprocessing function can be specified to adjust formatting of the user name passed to the API.

21.4.2.5.2 Setting Up LDAP Directory Verification

Learn how to set up LDAP Directory Verification.

To set up LDAP Directory:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **LDAP Directory**.
8. Under Settings:
   a. Host - Enter the hostname of your LDAP directory server.
   b. Port - Enter the port number of your LDAP directory host. The default is 389.
   c. Use SSL - Choose whether to use SSL to bind to the LDAP directory. If SSL with Authentication is chosen, a wallet must be configured for the Oracle APEX instance.
   d. Distinguished Name (DN) String - Enter the pattern used to construct the fully qualified distinguished name (DN) string to `DBMS_LDAP.SIMPLE_BIND_S` if using exact DN or the search base if using non-exact DN. Use `%LDAP_USER%` as a placeholder for the username. For example:

      Exact DN:

      ```
      cn=%LDAP_USER%,l=amer,dc=yourdomain,dc=com
      ```

      Non-Exact DN (Search Base):

      ```
      Non-Exact DN (Search Base)
      ```
   e. Use Distinguished Name (DN) - Choose whether the LDAP Distinguished Name (DN) String is exact or non-exact. If non-exact, LDAP Distinguished Name (DN) is the search base and you must supply a Search Filter.
   f. LDAP Username Edit Function - You may provide additional code to be executed to transform the username into a format perfectly suited to the LDAP directory entry or LDAP username. The bind variable `:USERNAME` contains the name the end user specified. For example, the following code calls a function which replaces all periods (`.`) with underscores (_) in the DN string:

      ```
      return apex_custom_auth.ldap_dnprep(p_username => :USERNAME);
      ```

g. Username Escaping - Choose how Oracle APEX escapes special characters in `%LDAP_USER%`. The correct value for this attribute depends on the used LDAP server and whether a LDAP Username Edit Function already escapes special characters.

To learn more about a specific field, see field-level Help.

9. Click **Create Authentication Scheme**.

> **Note:**
>
> If you choose SSL with Authentication as the SSL mode, a wallet must be set up using the Oracle APEX instance settings. Additionally, the root certification of the LDAP server must be imported into that wallet as a trusted certificate.

> **See Also:**
>
> - Configuring Authentication Schemes for an Instance
> - Configuring Wallet Information in *Oracle APEX Administration Guide*
> - Using Oracle Wallet Manager in *Oracle Database Enterprise User Security Administrator's Guide*

21.4.2.6 No Authentication (using DAD)

Adopts the current database user. This approach can be used in combination with a mod_plsql Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.

- About DAD Credentials Verification
  Learn about DAD Credentials Verification.
- Setting Up DAD Credentials Verification
  Learn how to set up DAD Credentials Verification.

21.4.2.6.1 About DAD Credentials Verification

Learn about DAD Credentials Verification.

DAD credentials verification uses the Oracle database native authentication and user mechanisms to authenticate users using a basic authentication scheme. This authentication scheme gets the user name from the DAD either as the value stored in the DAD configuration or, if the account information is not stored in the DAD configuration, as the user name captured using the basic authentication challenge.

To use DAD credentials verification:

- Each application user must have a user account in the Oracle database.
- You must configure a PL/SQL DAD for basic authentication (without account information).

  This results in one user name/password challenge for browser session for your application users. The user identity token is then made available in the `APP_USER` item.

DAD database authentication is useful when you must implement an authentication method that requires minimal setup for a manageable number of users. Ideally these users would have self-managed accounts in the database and your use of this authentication method would be short lived (for example, during the demonstration or prototyping stages of development).

The main drawback of this approach is burdensome account maintenance, especially if users do not administer their own passwords, or if their database accounts exist only to facilitate authentication to your application.

21.4.2.6.2 Setting Up DAD Credentials Verification

Learn how to set up DAD Credentials Verification.

To set up DAD Credentials Verification authentication:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **No Authentication**.
8. Settings, Username - Enter the username for the APEX session. If empty, APEX uses the database session user (typically `APEX_PUBLIC_USER`).
9. Click **Create Authentication Scheme**.

21.4.2.8 Oracle APEX Accounts

Oracle APEX Accounts are user accounts that are created within and managed in the APEX user repository. When you use this method, your application is authenticated against these accounts.

- About Oracle APEX Accounts
  Learn about Oracle APEX accounts.
- Setting Up Oracle APEX Accounts Authentication
  Learn how to set up Oracle APEX accounts authentication.

21.4.2.8.1 About Oracle APEX Accounts

Learn about Oracle APEX accounts.

Oracle APEX Accounts authentication uses a native user repository within APEX. When using this authentication method, the user name and password are validated against this repository.

Oracle APEX Accounts is a good solution when:

- You want control of the user account repository.
- User name and password-based approach to security is sufficient.

- You do not need to integrate into a single sign-on framework.

Oracle APEX Accounts is an especially good approach when you must get a group of users up and running on a new application quickly.

> **Tip:**
>
> Oracle does not recommend Oracle APEX Accounts authentication in production environments. The credentials used to log in to an application may be the same as the credentials used to log in to a workspace. In production environments, Oracle recommends using a full-featured external identity provider such as Oracle Identity Cloud Service (IDCS).

> **See Also:**
>
> Managing APEX Users in *Oracle APEX Administration Guide*

21.4.2.8.2 Setting Up Oracle APEX Accounts Authentication

Learn how to set up Oracle APEX accounts authentication.

To set up Oracle APEX Accounts:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Oracle APEX Accounts**.
   c. Click **Create Authentication Scheme**.

21.4.2.9 Oracle Application Server Single Sign-On Server

Delegates authentication to the Oracle AS Single Sign-On (SSO) Server. To use this authentication scheme, your site must have been registered as a partner application with the SSO server.

> **Note:**
>
> This authentication scheme only appears if Oracle 9iAS SSO Software Developer Kit (SDK) is installed in your database. Oracle recommends using a modern Single Sign-On authentication scheme such as SAML Sign-In or Social Sign-In.

- About Application Server Single Sign-On Server
  Learn about Application Server Single Sign-On Server authentication.
- Setting Up Oracle Application Server Single Sign-On
  Learn how to set up Single Sign-On authentication.

21.4.2.9.1 About Application Server Single Sign-On Server

Learn about Application Server Single Sign-On Server authentication.

Oracle APEX applications can operate as partner applications with Oracle Application Server's Single Sign-On (SSO) infrastructure. You must register your application (or register the APEX engine) as the partner application by following the Oracle Application Server instructions for registering partner applications and install the Oracle 9iAS SSO Software Developer Kit (SDK).

If you choose this approach, your application will not use an integrated login page. Instead, when a user accesses your application in a new browser session, the APEX engine redirects to the Single Sign-On login page. After the user is authenticated by SSO, the SSO components redirect back to your application, passing the user identity and other information to the APEX engine. The user can then continue to use the application until they log off, terminate their browser session, or until some other session-terminating event occurs.

21.4.2.9.2 Setting Up Oracle Application Server Single Sign-On

Learn how to set up Single Sign-On authentication.

To set up Oracle Application Server Single Sign-On:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Oracle Application Server Single Sign-On**.
8. Under Settings:
   a. Partner Application Name - If you authenticate with Oracle Application Server Single Sign-On and your application is a registered partner application with SSO, then enter exactly the same name you used when registering the partner application with SSO.

      If you are relying on APEX to be the registered partner application, leave this field blank.

b. Logout URL of SSO Server - Enter the logout path of the Single Sign-On Server (for example `https://login.example.com/sso/logout`). On logout of your application, APEX will use this value, concatenate '`?p_done_url=`' and the application's login url, and use the resulting URL to log out of Single Sign-On before redirecting back to your application.

9. Click **Create Authentication Scheme**.

21.4.2.10 SAML Sign-In

Delegates authentication to the Security Assertion Markup Language (SAML) Sign In authentication scheme.

> **Note:**
>
> SAML support requires Oracle Database 19c (Database Release Update 19.9.0.0.0 or newer) or Oracle Database 23ai. If your database does not meet these requirements, SAML Sign-In authentication scheme will not display.

- About SAML Sign In
  Learn about SAML Sign In authentication.
- Setting Up SAML Sign In
  Learn how to set up SAML Sign In authentication.

21.4.2.10.1 About SAML Sign In

Learn about SAML Sign In authentication.

Oracle APEX supports the use of Security Assertion Markup Language (SAML). SAML is an XML-based protocol for exchanging security information between software entities on the Web. SAML security is based on the interaction of asserting and relying parties. SAML provides single sign-on capabilities; users can authenticate at one location and then access service providers at other locations without having to log in multiple times.

> **Note:**
>
> SAML support requires Oracle Database 19c (Database Release Update 19.9.0.0.0 or newer) or Oracle Database 23ai. If your database does not meet these requirements, SAML Sign-In authentication scheme will not display.

21.4.2.10.2 Setting Up SAML Sign In

Learn how to set up SAML Sign In authentication.

> **Tip:**
>
> Your Instance Administrator must configure the SAML Sign In authentication scheme in Oracle APEX Administration Services before you can use it in applications. See Editing SAML Sign-In in *Oracle APEX Administration Guide*.

To set up a SAML Sign In authentication scheme in your application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **SAML Sign In**.
8. Under Settings:
   a. Use SAML Attributes of - Select **Instance**.
   b. Convert Username To Upper Case - Configure whether the attribute values for the username should be converted to upper case.
9. Click **Create Authentication Scheme**.

21.4.2.11 Social Sign-In

Social Sign-In supports authentication with Google, Facebook, and other social networks and enterprise identity providers that support OpenID Connect or OAuth2 standards.

- About Social Sign-In
  Learn about Social Sign-In authentication.
- Setting Up Social Sign-In Authentication
  Learn how to set up Social Sign-In authentication.

21.4.2.11.1 About Social Sign-In

Learn about Social Sign-In authentication.

Social Sign-In authentication is primarily useful for the following use cases:

- Your application is internet facing and you expect an unknown number of users from social networks to use your application.
- Your company has standardized on one of these identity providers for authentication: `Oracle Identity Cloud Service`, an internal `OpenID Connect` or `OAuth2`.

  Since these identity providers perform user credential verification, be aware that anyone who registers with the provider can use your application, unless you use authorization schemes for protection.

When registering your application with the identity provider, you need to use the full URI for one of the predefinedOracle APEX callback URLs. For example, suppose your application URI is:

```
https://www.example.com/apex/f?p=100
```

Then, you can register one of the following:

```
https://www.example.com/apex/apex_authentication.callback
```

or

```
https://www.example.com/apex/apex_authentication.callback2
```

Social network authentication providers store a multitude of information about users. You can configure the Social Login authentication scheme to request this information, using the Scope attribute (see Item Help). In the authentication scheme's Post-Authentication procedure, you can access this information using `APEX_JSON.GET_%` function calls.

You can use multiple authentication schemes in your application, to support more than one Social Login provider and other schemes.

> **See Also:**
>
> Authorization for the authentication scheme attribute `Switch in Session` for details.

21.4.2.11.2 Setting Up Social Sign-In Authentication

Learn how to set up Social Sign-In authentication.

> **Note:**
>
> As a prerequisite for creating the Social Sign-In Authentication , you have to request `OAuth2` credentials from the authentication provider and configure your account to support redirects to your application. Check your authentication provider's documentation for the details. The steps below assume that this has been done and that you saved the `OAuth2` credentials in the application's shared components.

To set up Social Sign-In Authentication:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. On the Authentication Schemes page, click **Create**.
6. Select **Based on a pre-configured scheme from the gallery** and click **Next**.
7. Under Name:
   a. Name - Enter the name used to reference the authentication scheme by other application developers.
   b. Scheme Type - Select **Social Sign-In**.

8. Under Settings:
   a. Credential Store - Select the credential store which contains Client ID and Secret for the Authentication Provider.
   b. Authentication Provider - Options include:
      - OpenID Connect Provider - Use an OpenID Connect based authentication provider. You have to enter the provider's OpenID Discovery URL below.
      - Generic OAuth2 - Provider Use an OAuth2 authentication provider. You have to enter URLs for the authorization, token and userinfo endpoints below.
      - Google - Use pre-defined settings for Google Login.
      - Facebook - Use pre-defined settings for Facebook Login.
   c. Discovery URL - Enter the OpenID Connect provider's discovery URL.

      Example:

      ```
      https://accounts.example.com/.well-known/openid-configuration
      ```

   d. Scope - Enter a comma separated list of permissions to request for the user who is logging in. The acceptable values depend on your authentication provider. For OpenID Connect and Google, APEX automatically adds the "`openid`" scope. The authentication provider returns user attributes based on these permissions. You can map the attributes to application items, or use a Post Authentication procedure to process them, using the `APEX_JSON.GET_%` functions.

      OpenID Connect scopes and attributes (also called claims) are listed here: https://openid.net/specs/openid-connect-basic-1_0.html#Scopes%20Examples%20email,profile

   e. Authentication URI Parameters - Enter optional parameters for the authentication URI. Consult the authentication provider's documentation for supported parameters.

      Example:
      `prompt=consent`

   f. Username - Enter the attribute which contains the username, or free text with one or more attribute substitutions that are enclosed by "#". You can reference the authentication scheme's name with `#APEX_AUTH_NAME#`. For example, `#email# (#APEX_AUTH_NAME#)` will result in a username that consists of the email address, followed by " (", followed by the authentication scheme name, followed by ")", like "jane.doe@example.com (Google)".

      Consult the authentication provider's documentation for supported attributes. Note that you need to set the required Scope, otherwise the authentication provider will not send all attributes. OpenID Connect scopes and attributes (also called claims) are listed here:

      Examples:

      ```
      #sub# (#APEX_AUTH_NAME#)
      ```

   g. Convert Username To Upper Case - Configure whether the attribute values for the username should be converted to upper case.
   h. Additional User Attributes - Enter a comma separated list of additional user data attributes.

      Consult the authentication provider's documentation for supported attributes. Note that you need to set the required Scope, otherwise the authentication provider will not send all attributes. OpenID Connect scopes and attributes (also called claims) are listed here:

https://openid.net/specs/openid-connect-basic-1_0.html#Scopes

i. Map Additional User Attributes - Provide a comma-separated list of application item names that the Additional User Attributes map to.

   For example, if you set the following attributes, then APEX automatically saves the attribute values in session state for the corresponding items.:

   - Additional User Attributes - `email,profile,picture`
   - Map Additional User Attributes To - `G_EMAIL,G_PROFILE,G_PICTURE`

j. Verify Attributes - If enabled, APEX looks for an attribute "*#name#*_verified" (for example, "email_verified" for Attribute "email"). OpenID defines that the identity provider sends "#name#_verified":false for unverified "email" and "phone_number" attributes. APEX ignores such unverified attributes. This will result in a failed authentication if "#name#" is the Username attribute. If the identity provider lets attackers enter unverified data that the application uses for authentication, they could impersonate other users in the application. The Verify Attributes check prevents this.

9. Click **Create Authentication Scheme**.

21.4.3 Viewing the Authentication Scheme Associated with an Application

View the current authentication scheme associated with an application.

To view the current authentication scheme for an application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under Security, click **Security Attributes**.
5. Locate the Authentication section. The current authentication scheme displays next to **Authentication Scheme**.
6. To edit the Authentication Scheme, **Define Authentication Schemes**.

21.4.4 Creating an Authentication Scheme

To associate an authentication scheme with an application, you create an authentication scheme on the Shared Components page.

To create an authentication scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.

   The Authentication Schemes page appears.
5. You can customize the appearance the page using the Search bar at the top of the page.
6. To create a new authentication scheme, click **Create**.
7. Specify how the scheme should be created by selecting one of the following:

- **Based on a pre-configured scheme from the gallery**.
- **As a copy of an existing authentication scheme**.

8. Follow the on-screen instructions.

> **Tip:**
>
> A newly created authentication scheme is not enabled by default. To enable the new authentication scheme for an application, edit the authentication scheme and click **Make Current Scheme**. See Changing the Authentication Scheme Associated with an Application.

> **See Also:**
>
> - Understanding Preconfigured Authentication Schemes
> - Custom Authentication

21.4.5 Configuring an Existing Authentication Scheme

Configure various settings and program hooks in an existing authentication scheme.

The authentication scheme designated as **Current** determines how an application identifies and verifies the user. Depending on the selected Authentication Scheme, you can configure various settings and program hooks to fine-tune your application's authentication such as login processing, post-logout URL, and session sharing.

To configure an existing authentication scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.

   Customize the appearance the page using the Search bar at the top of the page.
5. To configure an authentication scheme, click the authentication scheme name.

   The Authentication Scheme page appears.
6. Edit the appropriate attributes.

   To learn more about a specific attribute, see field-level Help.
7. Edit the appropriate attributes and click **Apply Changes**.

> **See Also:**
>
> Configuring Authentication Schemes for an Instance in *Oracle APEX Administration Guide*

21.4.6 Changing the Authentication Scheme Associated with an Application

Change the authentication scheme associated with an application.

A newly created authentication scheme is not enabled by default. To enable the new authentication scheme for an application, edit the authentication scheme and click **Make Current Scheme**.

To change the authentication scheme for an application:

1. Navigate to the Authentication Schemes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Security, select **Authentication Schemes**.
2. Select the authentication scheme you want to activate.

   The Create/Edit page appears.
3. Click **Make Current Scheme**.

> **See Also:**
>
> Understanding Preconfigured Authentication Schemes

21.4.7 Copying or Subscribing to an Authentication Scheme

Copy an authentication scheme from the current application or from another applications. When copying an authentication scheme from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to an authentication scheme:

1. Navigate to the Authentication Schemes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Security, click **Authentication Schemes**.

   The Authentication Schemes appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. From the Tasks list, click **Copy from another app**.
3. Scheme:
   a. Copy From Application - Select an application from which to copy the authentication scheme and click **Next**.
   b. Click **Next**.
4. Scheme:
   a. To Name - Optionally edit the name.
   b. Copy - Select an action:
      - **Yes** - Copy the authentication scheme.
      - **No** - Do not copy the authentication scheme.
      - **Copy and Subscribe** - Copy and subscribe to this authentication scheme. Subscribing to an authentication scheme adds a copy of it in your application and sets the authentication scheme master reference.
5. Click **Copy Scheme**.

21.4.8 Subscribing to an Authentication Scheme

Subscribe to an authentication scheme on the Authentication Schemes Create / Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an authentication scheme by copying it and then subscribing. See Copying or Subscribing to an Authentication Scheme.

To subscribe to an authentication scheme

1. Navigate to the Authentication Schemes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Security, click **Authentication Schemes**.

   The Authentication Schemes page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an authentication scheme.

   The Create / Edit page appears.
3. To subscribe to an authentication scheme:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select an application whose authentication scheme you want this scheme to subscribe to and click **Apply Changes**.

   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.

4. To refresh an authentication scheme:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Scheme - If the authentication scheme is already subscribed, click **Refresh Scheme**.
   c. Click **Apply Changes**.
5. To unsubscribe an authentication scheme:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

21.4.9 Creating a Login Page

Create an application login page.

When you create an application in Oracle APEX, a login page is created. The alias for the page is `'LOGIN'`. You can use this page as the "invalid session page" in an authentication scheme. The page is constructed with processes that call the APEX login API to perform credentials verification and session registration.

You can also build your own login pages using the pre-built pages as models and tailoring all of the user interface and processing logic to your requirements.

To create a login page for your application:

1. On the Workspace home page, click the App Builder icon.
2. Select an application.
3. Click **Create Page**.
4. For Create a Page:
   a. User Interface - Select a user interface for the page.
   b. Select a page type - Select **Login Page**.
   c. Click **Next**.
5. Select **Login Page**.
6. Specify Login page attributes and click **Create**.

21.4.10 Using a Procedure to Configure Authentication at Runtime

Configure authentication at runtime by specifying a procedure on the Security page.

To configure authentication at runtime by specifying a procedure:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, you can access the Security page in two ways:
   - **Edit Application Definition** button:
     a. Click **Edit Application Definition** to the right of the application name.

b. Click the **Security** tab.

- From Shared Components:

  a. Click **Shared Components**.

  b. Under Security, click **Security Attributes**.

The Edit Security Attributes page appears.

4. Authentication, Configuration Procedure- Enter the name of a procedure which configures authentication at runtime. See the example that follow.

   In applications with an authentication configuration procedure, the authentication scheme cannot be changed using the URL (for example, `f?p=100:1:1234:APEX_AUTHENTICATION=myauth`).

   The authentication configuration procedure runs at least once per session, on the first request. However, Oracle APEX may need to also call it on other occasions. In such cases, it must set the same values on each call.

5. Click **Apply Changes** to save your changes.

Example 21-1 procedure my_auth_config

This example is for an application that implements multi-tenancy and supports different variations of Social Sign-In for the tenants. This example uses the domain name in the URL to fetch configuration data (for example, `https://cust-01.example.com`, `https://cust-02.example.com`, and so on).

```
procedure my_auth_config (
    p_conf in out nocopy apex_authentication.t_configuration )
is
    l_host varchar2(32767) := sys.owa_util.get_cgi_env('HTTP_HOST');
begin
    for i in ( select discovery_url,
                      auth_scheme_name,
                      credential,
                      tenant_id
                 from customer_tenants
                where hostname = l_host )
    loop
        p_conf.authentication_name := i.auth_scheme_name;
        p_conf.substitutions := apex_t_varchar2 (
                                    'DISCOVERY_URL'       , i.discovery_url,
                                    'CREDENTIAL_STATIC_ID', i.credential );
        p_conf.tenant_id := i.tenant_id;
    end loop;
end my_auth_config;
```

The procedure can change three attributes of the in/out parameter `p_conf`, all of them are optional:

- `authentication_name` - Assign the name of an authentication scheme in your application, which will be used instead of the default scheme. Note that **Switch In Session** must be enabled for that scheme.
- `substitutions` - Assign name/value pairs using `apex_t_varchar2`. Oracle APEX substitutes each `#NAME#` in the authentication scheme attributes with the associated value.

The built-in Social Sign-In scheme uses `CREDENTIAL_STATIC_ID` to use the corresponding credential store instead of the one that was configured in the scheme.

- `tenant_id` - Set this tenant id in the session (see `APEX_SESSION.SET_TENANT_ID`).

> **Note:**
>
> SET_TENANT_ID Procedure in *Oracle APEX API Reference*

21.4.11 Viewing Authentication Scheme Reports

View authentication scheme reports.

To view authentication scheme reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authentication Schemes**.
5. Click the appropriate tab at the top of the page:
   - **Subscription** - Use the Authentication Scheme Subscription report to view subscribed authentication schemes in your application.
   - **History** - Use the Authentication Scheme History report to view recent modifications made to items in this application.

21.5 Providing Security Through Authorization

Extend the security of your application by creating an authorization scheme.

Authorization is a broad term for controlling access to resources based on user privileges. While conditions control the rendering and processing of specific page controls or components, authorization schemes control user access to specific controls or components.

- How Authorization Schemes Work
  Learn about how an authorization scheme extends the security of your application's authentication scheme.
- About Authorization Scheme Types
  Select an authorization scheme type when you create an authorization scheme.
- Creating an Authorization Scheme
  Create an authorization scheme on the Shared Components page.
- Attaching an Authorization Scheme to an Application, Page, or Components
  Attach an authorization scheme to an entire application, page, control, or component.
- Editing an Authorization Scheme
  Learn how to edit an authorization scheme.
- Copying or Subscribing to an Authorization Scheme
  Copy an authorization scheme from the current application or from another applications. When copying an authorization scheme from another app, you can also subscribe to it.

- Viewing Authorization Reports
  View the Authorization Scheme Subscription and Authorization Scheme Utilization reports to better manage authorization schemes within your application.

21.5.1 How Authorization Schemes Work

Learn about how an authorization scheme extends the security of your application's authentication scheme.

You can specify an authorization scheme for an entire application, page, or specific control such as a region, item, or button. For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bars a user sees. An authorization scheme either succeeds or fails. Common authorization scheme types include Exists, Not Exists SQL Queries, and PL/SQL Function Returning Boolean. If a component or control level authorization scheme succeeds, the user can view the component or control. If it fails, the user cannot view the component or control. If an application or page-level authorization scheme fails, then Oracle APEX displays a previously defined message.

When you define an authorization scheme, you give it a unique name. Once defined, you can attach it to any component or control in your application. To attach an authorization scheme to a component or control in your application, simply navigate to the appropriate attributes page and select an authorization scheme from the Authorization Scheme list.

21.5.2 About Authorization Scheme Types

Select an authorization scheme type when you create an authorization scheme.

The authorization scheme type determines how an authorization scheme is applied. Developers can create new authorization type plug-ins to extend this list.

Authorization Scheme Types	Description
Exists SQL Query	Enter a query that causes the authorization scheme to pass if it returns at least one row and causes the scheme to fail if it returns no rows
NOT Exists SQL Query	Enter a query that causes the authorization scheme to pass if it returns no rows and causes the scheme to fail if it returns one or more rows
PL/SQL Function Returning Boolean	Enter a function body. If the function returns true, the authorization succeeds.
Item in Expression 1 is NULL	Enter an item name. If the item is null, the authorization succeeds.
Item in Expression 1 is NOT NULL	Enter an item name. If the item is not null, the authorization succeeds.
Value of Item in Expression 1 Equals Expression 2	Enter and item name and value. The authorization succeeds if the item's value equals the authorization value.
Value of Item in Expression 1 Does NOT Equal Expression 2	Enter an item name and a value. The authorization succeeds if the item's value is not equal to the authorization value.
Value of Preference in Expression 1 Does NOT Equal Expression 2	Enter an preference name and a value. The authorization succeeds if the preference's value is not equal to the authorization value.
Value of Preference in Expression 1 Equals Expression 2	Enter an preference name and a value. The authorization succeeds if the preference's value equal the authorization value.
Is In Group	Enter a group name. The authorization succeeds if the group is enabled as a dynamic group for the session. See APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS in *Oracle APEX API Reference*. If the application uses APEX Accounts Authentication, this check also includes workspace groups that are granted to the user. If the application uses Database Authentication, this check also includes database roles that are granted to the user.

Authorization Scheme Types	Description
Is Not In Group	Enter a group name. The authorization succeeds if the group is not enabled as a dynamic group for the session.

21.5.3 Creating an Authorization Scheme

Create an authorization scheme on the Shared Components page.

To create an authorization scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select Authorization Schemes.
5. Click **Create**.
6. Specify how to create an authorization scheme by selecting one of the following:
   - **From Scratch**
   - **As a Copy of an Existing Authorization Scheme**
7. On Create Authorization Scheme - Details:
   a. Name - Enter an unique name that identifies this authorization scheme.
   b. Scheme Type - Select how this authorization scheme will be applied. See About Authorization Scheme Types.
   c. Identify error message displayed when scheme violated - Enter error text that displays if the authorization scheme fails (that is, the current user fails the security check).
   d. Validate Authorization Scheme - Authorization schemes are evaluated on first use in a session. Use this option to controls if future uses cause re-evaluations and when a memorized result can be taken instead.

   For more details, see field-level Help.
8. Click **Create Authorization Scheme**.

See Also:

- Attaching an Authorization Scheme to an Application, Page, or Components
- Changing the Evaluation Point Attribute
- About Resetting Authorization Scheme State

21.5.4 Attaching an Authorization Scheme to an Application, Page, or Components

Attach an authorization scheme to an entire application, page, control, or component.

> **Tip:**
>
> Before you can attach an authorization scheme to an application, page, control, or component, you must first create it. See Creating an Authorization Scheme.

- Attaching an Authorization Scheme to an Application
  Learn how to attach an authorization scheme to an application.
- Attaching an Authorization Scheme to a Page
  Learn how to attach an authorization scheme to a page.
- Attaching an Authorization Scheme to a Control or Component
  Learn how to attach an authorization scheme to a control or component.

21.5.4.1 Attaching an Authorization Scheme to an Application

Learn how to attach an authorization scheme to an application.

To attach an authorization scheme to an application:

1. On the Workspace home page, click the App Builder icon.
2. Select an application.
3. Click the **Shared Components** icon.

   The Shared Components page appears.
4. Under Security, click **Security Attributes**.
5. Scroll down to Authorization and make a selection from the Authorization Scheme list.
6. For Run on Public Pages, select **On** or **Off** to specify whether the application-level authorization scheme is checked on public pages (that is, pages that do not require authorization). To learn more, see Authorization.
7. To define a new authorization scheme, click **Define Authorization Schemes**.

21.5.4.2 Attaching an Authorization Scheme to a Page

Learn how to attach an authorization scheme to a page.

To attach an authorization scheme to a page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Select a page.

   Page Designer appears.
4. In the Rendering tab on the left side, click the page title at the top of the tree.
5. In the Property Editor, locate Security and make a selection from the Authorization Scheme attribute.

> **Tip:**
>
> To find a group or attribute:
>
> - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
> - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

6. Click **Save**.

21.5.4.3 Attaching an Authorization Scheme to a Control or Component

Learn how to attach an authorization scheme to a control or component.

To attach an authorization scheme to a page component or control:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Select a page.

   Page Designer appears.
4. Select the component or control to which you want to apply the authorization scheme.
5. In the Property Editor, locate Security and make a selection from the Authorization Scheme attribute.

> **Tip:**
>
> To find a group or attribute:
>
> - **Search for the group or attribute** - Enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. To return to the default display, delete the keywords.
> - **Use Go to Group** - Click **Go to Group** and select the group. To return the default display, click **Go to Group** again and select **Expand All**.

6. Click **Save**.

21.5.5 Editing an Authorization Scheme

Learn how to edit an authorization scheme.

- Editing Attributes of an Existing Authorization Scheme
  Edit attributes of an existing authorization scheme.
- Changing the Evaluation Point Attribute
  Control when an authorization scheme is validated using the Evaluation Point, Validate authorization scheme attribute.
- Subscribing to an Authorization Scheme
  Subscribe to an authorization scheme on the Authorization Schemes, Edit page.

- About Resetting Authorization Scheme State
  Call an API to reset a session's authorization scheme state.

21.5.5.1 Editing Attributes of an Existing Authorization Scheme

Edit attributes of an existing authorization scheme.

To edit attributes of an existing authorization scheme:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authorization Schemes**.

   The Authorization Schemes page appears. By default, each scheme displays as an icon. You can customize the appearance the page using the Search bar at the top of the page.
5. Select an authorization scheme.
6. Edit the appropriate attributes.

   To learn more, see field-level Help.
7. To save your changes, click **Apply Changes**.

21.5.5.2 Changing the Evaluation Point Attribute

Control when an authorization scheme is validated using the Evaluation Point, Validate authorization scheme attribute.

Authorization schemes are evaluated on first use in a session. The Validate authorization scheme attribute controls if future uses cause re-evaluations and when a memorized result can be taken instead.

To change the authorization scheme evaluation point:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authorization Schemes**.

   The Authorization Schemes page appears.

   By default, each scheme displays as an icon. You can customize the appearance the page using the Search bar at the top of the page.
5. Select an authorization scheme.
6. Scroll down to Evaluation Point and edit the Validate authorization scheme attribute. Options include:
   - **Once per session** - Evaluate only once and always use the memorized result afterwards.
   - **Once per page view** - Evaluate once for each request that is processed. Use the memorized result if the authorization scheme is referenced in more than one component on the page.

- **Once per component** - Evaluate once for each component that references the authorization scheme, but save the result in the session. Further requests to render or process the scheme use the result that was memorized for the component.
- **Always (No Caching)** - The authorization scheme will always be evaluated.

To learn more, see field-level Help.

7. To save your changes, click **Apply Changes**.

> **Tip:**
>
> The default value **Once per session** is the most efficient. You should choose another value if the authorization check depends on changing session state or other factors that are not consistent over an entire session.

21.5.5.3 Subscribing to an Authorization Scheme

Subscribe to an authorization scheme on the Authorization Schemes, Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to an authorization schemes by copying it and then subscribing. See Copying or Subscribing to an Authorization Scheme.

To subscribe to an authorization scheme

1. Navigate to the Authorization Schemes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Security, click **Authorization Schemes**.

   The Authorization Schemes page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select an authorization scheme.

   The Edit page appears.
3. To subscribe to an authorization scheme:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select an application whose authorization scheme you want this scheme to subscribe to and click **Apply Changes**.

      When you select a master authorization scheme and click **Apply Changes**, the subscribed shared component are be automatically refreshed from the selected master

> **Tip:**
>
> You can also subscribe to a shared component during the copy process.

4. To refresh an authorization scheme:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Scheme - If the authentication scheme is already subscribed, click **Refresh Scheme**.
   c. Click **Apply Changes**.
5. To unsubscribe an authorization scheme:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

21.5.5.4 About Resetting Authorization Scheme State

Call an API to reset a session's authorization scheme state.

If an authorization scheme is validated once for each session, Oracle APEX caches the validation results in each user's session cache. You can reset a session's authorization scheme state by calling the `APEX_AUTHORIZATION.RESET_CACHE` API.

> **See Also:**
>
> `APEX_AUTHORIZATION.RESET_CACHE` Procedure in *Oracle APEX API Reference*

21.5.6 Copying or Subscribing to an Authorization Scheme

Copy an authorization scheme from the current application or from another applications. When copying an authorization scheme from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to an authorization scheme:

1. Navigate to the Authorization Schemes page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Security, click **Authorization Schemes**.

The Authorization Schemes page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. From the Tasks list, click **Copy from another app**.
3. Copy From:
   a. Copy From Application - Select an application from which to copy the authorization scheme.
   b. Click **Next**.
4. Names:
   a. To Name - Optionally edit the name.
   b. Copy - Select an action:
      - **Yes** - Copy the authorization scheme.
      - **No** - Do not copy the authorization scheme.
      - **Copy and Subscribe** - Copy and subscribe to the authorization scheme. Subscribing to an authorization scheme adds a copy of it in your application and sets the authorization scheme master reference.
5. Click **Copy** .

21.5.7 Viewing Authorization Reports

View the Authorization Scheme Subscription and Authorization Scheme Utilization reports to better manage authorization schemes within your application.

To view authorization scheme reports:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.

   The Shared Components page appears.
4. Under Security, select **Authorization Schemes**.
5. Click the appropriate tab at the top of the page:
   - **Subscription** - Use the Authorization Scheme Subscription report to view details about authorization schemes subscription.
   - **Utilization** - Use the Authorization Scheme Utilization report to view details about authorization schemes utilization.

     To view additional reports indicating which pages having authorization schemes and which do not, select one of the following from the Tasks list:
     – **Report Pages With Authorization Schemes**
     – **Report Pages Without Authorization Schemes**
   - **History** - Use the Authorization Scheme History report to view recent modifications made to Authorization Schemes in this application.

22

Managing Application Globalization

You can run applications built in App Builder concurrently in different languages.

This section describes how to translate an application built in App Builder.

- Understanding Application Translation and Globalization Support
  Learn about developing APEX applications that can run concurrently in different languages.
- Specifying the Primary Language for an Application
  Edit Globalization attributes to specify how the Oracle APEX engine determines the primary language of an application.
- Applying Format Masks to Items
  The Oracle APEX engine applies globalization settings for each rendered page. This default behavior can impact the display of certain items such as numbers and dates.
- About Translating Applications for Multibyte Languages
  If your application must run in several languages simultaneously (such as Chinese and Japanese), consider configuring your database with a character set to support all of the languages.
- Understanding the Translation Process
  To translate an APEX application, map the primary and target language, seed and export text to a translation file, translate the text, apply the translation file, and publish the translated application.
- Translating Default Reports for Interactive Reports and Interactive Grids
  Learn how developers can translate default reports for interactive reports and interactive grids.
- Translating Messages
  Learn when and how to translate messages in an Oracle APEX application.
- Translating Data That Supports List of Values
  You create a dynamic translation to translate dynamic pieces of data. For example, you might use a dynamic translation on a list of values based on a database query.
- Understanding Supported Globalization Codes
  If you are building a multilingual application, it is important to understand how globalization codes affect the way in which your application runs. These codes are set automatically based on the application-level Globalization attributes you select.

See Also:

- Installing Translated Versions of APEX in *Oracle APEX Installation Guide*
- Viewing Installed Translations in *Oracle APEX Administration Guide*

22.1 Understanding Application Translation and Globalization Support

Learn about developing APEX applications that can run concurrently in different languages.

A single Oracle database instance and Oracle APEX can support multiple database sessions customized to support different languages.

- About the Translation Process
  Learn about translating an application built in App Builder.
- About Language Identification
  Learn how the APEX engine determines the application language.
- About the Rules for Translating Applications
  Learn about rules for translating applications.
- How Translated Applications Are Rendered
  Learn about how APEX renders translated applications.
- Making Application Attributes Translatable
  Learn about best practices to make application attributes within your application translatable.

22.1.1 About the Translation Process

Learn about translating an application built in App Builder.

In general, translating an application built in App Builder involves the following steps:

- **Step 1: Define Application Languages** - Create a mapping which identifies the target primary and target application language.
- **Step 2: Seed and Download to a Translation File** - Seed the translation table and then export the translation text to a translation file. Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you generate and export an XLIFF file for translation.
- **Step 3: Translate the XLIFF File** - Download the XLIFF files for translation or manually edit the translation repository.
- **Step 4: Upload and Apply a Translated XLIFF Document** - Upload XLIFF files with translated text and apply translations to the Translation Text repository.
- **Step 5: Publish the Application** - Make the translated application available to users.

> **Tip:**
>
> To learn more, see the translations steps described in Understanding the Translation Process.

22.1.2 About Language Identification

Learn how the APEX engine determines the application language.

After you create an application, you specify a language preference by editing the **Application Primary Language** attribute. Globalization attributes control how the how the APEX engine determines the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.

> **Tip:**
>
> To learn more, see Specifying the Primary Language for an Application

22.1.3 About the Rules for Translating Applications

Learn about rules for translating applications.

Use the following rules to determine which translated version to use:

- Look for an exact match between the user language preference and the language code of the translated application.
- Look for a truncated match. That is, see if the language and locale exist. For example, if the user language preference is `en-us` and the translated version of `en-us` does not exist, look for a translated application that has the language code `en`.
- Use the primary application language.

For example, suppose you create an application with the primary language of German, `de`, and you create a translated version of the application with a language code of `en-us`. Users accessing this application with a browser language of `en-us` execute the English `en-us` version of the application. Users accessing the application with a browser language of `en-gb` view the application in the application's primary language, that is, in German. For this example, you should create the translated English version using language code `en` to encompass all variations of `en`.

22.1.4 How Translated Applications Are Rendered

Learn about how APEX renders translated applications.

After APEX determines the language for an application, the APEX engine alters the database language for a specific page request. It then looks for a translated application in the appropriate language. If the APEX engine finds that language, it renders the application using that definition. Otherwise, it renders the application in the base (or primary) application language.

Note that the text that displays within an application is not translated on the fly. APEX dynamically collects page attributes from either a base language application definition or an alternative application definition.

> **See Also:**
>
> - About Dynamic Translation Text Strings
> - Translating Data That Supports List of Values

22.1.5 Making Application Attributes Translatable

Learn about best practices to make application attributes within your application translatable.

When you build an application in App Builder, you define a large number of declarative attributes such as field labels, region headings, page header text, and so on.

- About Shortcuts that Support Translatable Messages
  Learn about shortcuts that support translatable messages.
- About Messages
  Learn about messages.
- About Dynamic Translation Text Strings
  Learn about dynamic translation text strings.
- Marking a Region as Not Translatable
  Learn about marking a region as not translatable.
- Identifying a Template as Translatable
  Learn how to identify a template as translatable.

22.1.5.1 About Shortcuts that Support Translatable Messages

Learn about shortcuts that support translatable messages.

App Builder includes two shortcut types that enable you to reference translatable messages:

- **Message**. Use this shortcut to reference a translatable message at runtime. Note that the name of the shortcut must match the corresponding message name. At runtime, the name of the shortcut expands to the text of the translatable message for the current language.
- **Message with JavaScript Escaped Single Quotes**. Use this shortcut to reference a shortcut inside of a JavaScript literal string and reference a translatable message at runtime. This shortcut defines a text string. When the shortcut is referenced, it escapes the single quotation marks required for JavaScript.

> **See Also:**
>
> Using Shortcuts

22.1.5.2 About Messages

Learn about messages.

If your application includes PL/SQL regions or PL/SQL processes, you must translate any generated HTML or text. You may also need to translate messages used in reports if your application uses a language that is not one of the ten languages into which APEX is translated.

> **See Also:**
>
> Translating Messages

22.1.5.3 About Dynamic Translation Text Strings

Learn about dynamic translation text strings.

Dynamic translations are used for database data that must be translated at runtime. For example, you might use a dynamic translation to translate a list of values based on a database query. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string. You can also use the `APEX_LANG.LANG` API to retrieve dynamic translations programmatically.

> **See Also:**
>
> Translating Data That Supports List of Values

22.1.5.4 Marking a Region as Not Translatable

Learn about marking a region as not translatable.

By default, page region titles are included in the generated translation file. However, you can mark a region title as not translatable.

To mark a region title as not translatable:

1. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
2. In the Rendering tab, select the region.

   The Property Editor displays the region attributes. Attributes are organized in groups.
3. Find and expand the **Advanced** group.
4. For **For Exclude Title from Translation**, select **On** to exclude this title from translation. This attribute has no effect, except when translating the application into another language.
5. Click **Save**.

22.1.5.5 Identifying a Template as Translatable

Learn how to identify a template as translatable.

By default, templates are not translatable, and therefore are not included in the generated translation file. Generally, templates do not and should not contain translatable text. However, if you must mark a template as translatable, select the **Translatable** checkbox on the Edit Template page.

To identify a template as translatable:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

3. On the Application home page, click **Shared Components**.
4. Under User Interface, select **Templates**.

   The Templates page appears.
5. Select a template.
6. Under Name, select **Translatable**.

   You can include translatable text at the application level by defining the translatable text using static substitution strings. Because application-level attributes are translated, any text defined as a static substitution string is included in the generated translation file.

> **See Also:**
>
> - Creating Custom Themes
> - Substitutions

22.2 Specifying the Primary Language for an Application

Edit Globalization attributes to specify how the Oracle APEX engine determines the primary language of an application.

To edit globalization attributes:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Shared Components**.
4. Under Globalization, select **Globalization Attributes**.
5. For **Application Primary Language**, select the language in which the application is being developed.
6. For **Application Language Derived From**, specify how the APEX engine determines (or derives) the application language. The application primary language can be static, derived from the Web browser language, or determined from a user preference or item.
   - **No NLS (Application not translated)** - Select this option if the application will not be translated.
   - **Application Primary Language** - Determines the translated application language based on the Application Primary Language attribute.
   - **Browser (use browser language preference)** - Determines the translated application language based on the user's Web browser language.
   - **Application Preference (use FSP_LANGUAGE_PREFERENCE)** - Determines the translated application language based on a value defined using the APEX_UTIL.SET_PREFERENCE API. This option can be used to maintain a user's language preference across multiple logins.
   - **Item Preference (use item containing preference)** - Determines the translated application language based on an application-level item called FSP_LANGUAGE_PREFERENCE. Using this option requires Oracle APEX to determine the appropriate language preference every time the user logs in.

- **Session** - Determines the translated application language from the session setting. The Oracle APEX session language can be set using either the `APEX_UTIL.SET_SESSION_LANG` procedure or via the `P_LANG` parameter of the **F procedure** in the URL.

7. Configure other options as appropriate. To learn more about an attribute, see field-level Help or see Globalization Page.
8. Click **Apply Changes**.

> **See Also:**
>
> - Editing the Application Definition
> - Configuring Security Attributes
> - Understanding Supported Globalization Codes

22.3 Applying Format Masks to Items

The Oracle APEX engine applies globalization settings for each rendered page. This default behavior can impact the display of certain items such as numbers and dates.

For example, suppose your application determines the application language based on the user's browser language preference. If the APEX engine determines the user's browser language preference is French, it displays dates and numbers in a format that conforms to French standards. You can override this default behavior and explicitly control how items display by applying a format mask. You apply a format mask by making a selection from the Display As list:

- When you create the item.
- After you create the item by editing the item attributes.

22.4 About Translating Applications for Multibyte Languages

If your application must run in several languages simultaneously (such as Chinese and Japanese), consider configuring your database with a character set to support all of the languages.

`UTF8` and `AL32UTF8` are the character sets you can use to support almost all languages around the world.

22.5 Understanding the Translation Process

To translate an APEX application, map the primary and target language, seed and export text to a translation file, translate the text, apply the translation file, and publish the translated application.

This section describes the process of translating an application.

> **Tip:**
>
> If you make *any* changes to the primary application, such as refreshing the theme, modifying the page title, adding a new page item or modifying an existing one, you will need to repeat Steps 2 to 5 and publish the strings to the translated application. Otherwise, you run the risk of having the translated application out of sync with the primary application.

- Step 1: Define Application Languages
  The first step in translating an application is to map the primary and target application language. The primary application is the application to be translated. The target application is the resulting translated application.
- Step 2: Seed and Download to a Translation File
  Seed the translation table and then export the translation text to a translation file.
- Step 3: Translate the XLIFF File
  After you export a translatable file to XLIFF format, you can translate it into the appropriate languages.
- Step 4: Upload and Apply a Translated XLIFF Document
  After your XLIFF document has been translated, the next step is to upload and then apply it.
- Step 5: Publish the Application
  Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.
- Manually Editing a Translation
  Once you have mapped the target language and seeded the translatable text, you manually edit a translation.

> **See Also:**
>
> - Translating Messages
> - Translating Data That Supports List of Values

22.5.1 Step 1: Define Application Languages

The first step in translating an application is to map the primary and target application language. The primary application is the application to be translated. The target application is the resulting translated application.

To map the primary and target application language:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

The Translate page appears.

2. Click **Define application languages**.

   The Language Mappings page appears.

3. Click **Create**.

4. On Create/Edit Language Application Mapping:

   - Translation Application - Enter a unique integer value that identifies an application. The application ID must be unique on your instance of Oracle APEX.
   - Language - Select the language to be translated.
   - Document Direction - Set document direction. Options: **Left-To-Right** or **Right-To-Left**.
   - Image Directory - Enter the directory where images are obtained. Unless you are using static files from the images directory which are translated, leave this field blank. This is the default and most common option.

5. Click **Create**.

22.5.2 Step 2: Seed and Download to a Translation File

Seed the translation table and then export the translation text to a translation file.

- Seeding Translatable Text
  Learn about seeding translatable text.
- Downloading an XLIFF File
  Learn how to download an XLIFF file.

22.5.2.1 Seeding Translatable Text

Learn about seeding translatable text.

Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you can then generate and export an XLIFF file for translation.

The seeding process keeps your primary language application synchronized with the Translation Text repository. You should run the seed process any time your primary language application changes.

To seed translatable text:

1. Navigate to the Translate Application page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Click **Shared Components**.

   d. Under Globalization, click **Application Translations**.

   The Translate page appears.

2. Click **Seed translatable text**.

3. Select the appropriate target application.

4. Click **Seed**.

   When completed, the following message displays:

```
Successfully seeded translatable text.
```

22.5.2.2 Downloading an XLIFF File

Learn how to download an XLIFF file.

Once the translation mappings are established the translatable text within the application is seeded into a translation repository. This repository is then exported to an XML Localization Interchange File Format (XLIFF) file for translation.

- About the XLIFF Export Page
  Learn about the XLIFF Export page.
- Downloading a XLIFF file for an Application
  Learn how to download a XLIFF file for an application.
- Downloading an XLIFF File for an Application Page
  Learn how to download an XLIFF File for an application page.
- About Including XLIFF Target Elements
  Learn about including XLIFF target elements.

22.5.2.2.1 About the XLIFF Export Page

Learn about the XLIFF Export page.

The XLIFF Export page is divided into two sections. Use **Download XLIFF file for complete Application** to export translatable text for an entire application (that is, all pages, lists of values, messages, and so on). Use **Download XLIFF file for Application Page** to export translatable text for a specific page.

> **Note:**
>
> XML Localization Interchange File Format (XLIFF) is an XML-based format for exchanging localization data. For more information about XLIFF and the XLIFF 1.0 specification, see:
>
> https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

22.5.2.2.2 Downloading a XLIFF file for an Application

Learn how to download a XLIFF file for an application.

To download an XLIFF file for an complete application:

1. Seed the translatable text.
2. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
3. On the Translate page, click **Download XLIFF translation files**.

4. Under **Download XLIFF file for complete Application**:
   a. Language - Choose the language to be downloaded in the XLIFF file.
   b. Include XLIFF Target Elements - Specify whether to include the target elements and corresponding language code in the generated XLIFF file. To include, select **Include XLIFF Target Elements**.
   c. Export - Choose to export all translatable elements of the application or only those elements which are new or have been updated.
   d. Click **Export XLIFF for Application**.
5. Follow the on-screen instructions.

> **See Also:**
>
> Seeding Translatable Text

22.5.2.2.3 Downloading an XLIFF File for an Application Page

Learn how to download an XLIFF File for an application page.

To download an XLIFF file for a specific page:

1. Seed the translatable text.
2. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
3. On the Translate page, click **Download XLIFF translation files**.
4. Under **Download XLIFF file for Application Page**:
   a. **Language** - Choose the language to be downloaded in the XLIFF file.
   b. **Page** - Select a page.
   c. **Include XLIFF Target Elements** - Specify whether to include the target elements and corresponding language code in the generated XLIFF file. To include, select **Include XLIFF Target Elements**.
   d. **Export** - Choose to export all translatable elements of the page or only those elements which are new or have been updated.
   e. Click **Export XLIFF for Page**.
5. Follow the on-screen instructions.

> **See Also:**
>
> Seeding Translatable Text

22.5.2.2.4 About Including XLIFF Target Elements

Learn about including XLIFF target elements.

When APEX generates an XLIFF document, each document contains multiple translation units. Each translation unit consists of a source element and a target element. The XLIFF document can be generated with both the source and target elements for each translation unit. You have the option of generating a file containing only source elements. The updated translations will be applied from the target elements of the translation units.

22.5.3 Step 3: Translate the XLIFF File

After you export a translatable file to XLIFF format, you can translate it into the appropriate languages.

Because XLIFF is an open standard XML file for exchanging translations, most translation vendors should support it. APEX only supports XLIFF files encoded in UTF-8 character sets. In other words, it exports XLIFF files for translation in UTF-8 and assumes that the translated XLIFF files are in the same character set.

Translation is a time-consuming task. APEX supports incremental translation so that application development can be done in parallel with the translation. An XLIFF file can be translated and uploaded to APEX even when only part of the XLIFF file is translated. For strings that have no translation in the corresponding translated application,APEX uses the corresponding ones in the primary language.

> **See Also:**
>
> For more information about the XLIFF and the XLIFF 1.0 specification, see:
>
> `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff`

22.5.4 Step 4: Upload and Apply a Translated XLIFF Document

After your XLIFF document has been translated, the next step is to upload and then apply it.

- Uploading a Translated XLIFF Document
  Learn how to upload a translated XLIFF document.
- Applying an Uploaded XLIFF Document
  Learn about applying an uploaded XLIFF document.
- Deleting an Uploaded XLIFF Document
  Learn how to delete an uploaded XLIFF document.

22.5.4.1 Uploading a Translated XLIFF Document

Learn how to upload a translated XLIFF document.

To upload a translated XLIFF document:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.

2. Click **Apply XLIFF translation files**.

   The XLIFF Translation Files page appears.

3. Click **Upload Files**.
4. On the XLIFF Upload page:
   a. Locate the file to be uploaded.
   b. Click **Upload**.

      The uploaded document appears on the XLIFF Translation Files page.

22.5.4.2 Applying an Uploaded XLIFF Document

Learn about applying an uploaded XLIFF document.

After you upload an XLIFF document, the next step is to apply the XLIFF document and then publish the translated application. When you apply an XLIFF document, the APEX engine parses the file and then updates the translation tables with the new translatable text.

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.

Remember that to run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

To apply a translated XLIFF document:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.

2. Click **Apply XLIFF translation files**.

   The XLIFF Translation Files page appears.

3. To apply the XLIFF files:
   a. Select the XLIFF files you want to apply.
   b. From Apply to Translation, choose a translation mapping.
   c. Click **Apply Checked**.

   The XLIFF file is parsed and the contents is applied to the translation repository. Once this process is complete, you must publish the application for each translation.

> **See Also:**
>
> Specifying the Primary Language for an Application

22.5.4.3 Deleting an Uploaded XLIFF Document

Learn how to delete an uploaded XLIFF document.

To delete an uploaded XLIFF document:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
2. Click **Apply XLIFF translation files**.

   The XLIFF Translation Files page appears.
3. In the XLIFF Files repository, select the checkbox to the left of the Filename column.
4. Click **Delete Checked**.

> **See Also:**
>
> Specifying the Primary Language for an Application

22.5.5 Step 5: Publish the Application

Publishing your application creates a copy of the base language application, substituting the translated text strings from your translations table. This published application can then be used to render your application in alternate languages.

Remember that to run an application in an alternative language, you must run it with globalization settings that cause an alternative language version to display. For example, if the language is derived from the browser language, you must set the browser language to the same language as the translated application.

To publish an application:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
2. Click **Publish translated applications**.

The XLIFF Translation Files page appears

3. Select the items you want to publish and click **Publish**.

You should verify the existence of the translated application after it is published. Translated applications do not display in the Available Applications list on the App Builder home page. In order for a translated application to appear in App Builder, you must ensure that you have correctly configured the application Globalization attributes.

> **See Also:**
>
> Specifying the Primary Language for an Application

22.5.6 Manually Editing a Translation

Once you have mapped the target language and seeded the translatable text, you manually edit a translation.

To manually edit a translation:

1. Define the application language.
2. Seed the translatable text.
3. Navigate to the Translatable Text:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
4. Under Translation Utilities, click **Translation Repository**.

   The Translatable Text page appears.
5. Select a Language Mapping or page and click **Set**.
6. To further customize the report, use the Search bar at the top of the page.
7. To edit translatable text:
   a. Click the **Edit** icon.

      The Translatable Text page appears.
   b. Translate From Text - Identifies the primary language text.
   c. Translate to Text - Identifies the translated text.
   d. Update all occurrences of this string - Enable this option to update all occurrences of the string in the Translation Repository.
   e. Click **Apply Changes**.
8. To access a Grid Edit page:
   a. Click **Grid Edit**.

      The Translatable Text - Grid Edit page appears.

b. Click the **Edit** button.

c. In the **Translate To** column, select and then edit the string.

d. Click the **Save** button.

> **See Also:**
>
> - Step 1: Define Application Languages
> - Seeding Translatable Text

22.6 Translating Default Reports for Interactive Reports and Interactive Grids

Learn how developers can translate default reports for interactive reports and interactive grids.

Both interactive reports and interactive grids enable developers to save default reports. Developers can save Primary and Alternative default reports from both report regions. APEX supports the translation of these default reports saved by developers.

> **Note:**
>
> User saved reports (including public or private reports) are not translatable.

APEX supports the translation of the following report settings:

- For both for interactive report and interactive grid default reports:
  - Report name and description
  - Filter name
  - Highlight name
  - Chart axis labels
- Only interactive report default reports:
  - Computation label and format mask
  - Pivot Aggregate label and format mask
  - Group By function label and format mask
- Only interactive grid default reports:
  - Report aggregate tooltip

To translate interactive report and interactive grid default reports:

1. A developer saves the default reports in the application primary language.
2. Translate the saved default reports using application translation process. Steps in translation process include:
   - **Define Application Languages** - Create a mapping which identifies the target primary and target application language.

See Step 1: Define Application Languages.

- **Seed and Download to a Translation File** - Seed the translation table and then export the translation text to a translation file. Seeding the translation copies all translatable text into the Translation Text repository. After you specify the language and seed the Translation Text, you generate and export an XLIFF file for translation.

  See Step 2: Seed and Download to a Translation File.

- **Translate the XLIFF File** - Download the XLIFF files for translation or manually edit the translation repository.

  The XLIFF file includes default report translatable texts (such as report name, computation label, chart label, group by aggregate label, pivot aggregate label, and so on).

  See Step 3: Translate the XLIFF File.

- **Upload and Apply a Translated XLIFF Document** - Upload XLIFF files with translated text and apply translations to the Translation Text repository.

  See Step 4: Upload and Apply a Translated XLIFF Document.

- **Publish the Application** - Make the translated application available to users.

  Once the translation is published, the default reports and their settings are translated. A default report that does not have translation displays using the default Application Primary Language setting.

  See Step 5: Publish the Application.

See Also:

- Saving Interactive Reports
- Saving Interactive Grid Reports

22.7 Translating Messages

Learn when and how to translate messages in an Oracle APEX application.

- About Translating Messages
  Learn about when to translate text messages in your application.
- About Translating Messages Used in PL/SQL Procedures
  Learn about translating messages used in PL/SQL procedures.
- Translating Messages Used Internally by APEX
  If your application uses a language that is not among the standard languages into which APEX is translated, you must translate messages displayed by the APEX runtime engine.
- Managing Text Messages
  Create, edit, copy, or subscribe to text messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.

22.7.1 About Translating Messages

Learn about when to translate text messages in your application.

You may need to translate text messages if your application:

- Includes PL/SQL regions or PL/SQL processes, or calls a PL/SQL package, procedure, or function. If it does, you may need to translate the generated HTML. See About Translating Messages Used in PL/SQL Procedures.
- Uses a language that is not one of the ten languages into which APEX is translated. If it does, you may need to translate messages used in reports. See Translating Messages Used Internally by APEX.

22.7.2 About Translating Messages Used in PL/SQL Procedures

Learn about translating messages used in PL/SQL procedures.

If your application includes PL/SQL regions or PL/SQL processes or calls PL/SQL package, procedures, or functions, you may need to translate generated HTML.

First, you define each message on the Translatable Messages page. Second, you use the `APEX_LANG.MESSAGE` API to translate the messages from PL/SQL stored procedures, functions, triggers, or packaged procedures and functions.

> **See Also:**
>
> - APEX_LANG in *Oracle APEX API Reference*
> - Managing Text Messages

22.7.3 Translating Messages Used Internally by APEX

If your application uses a language that is not among the standard languages into which APEX is translated, you must translate messages displayed by the APEX runtime engine.

The APEX runtime engine which is used by developers to create applications is available in the following languages: Arabic, Brazilian Portuguese, Croatian, Czech, Danish, Dutch, Finnish, French, French - Canada, German, Greek, Hebrew, Hungarian, Italian, Japanese, Korean, Norwegian, Polish, Portuguese (Portugal) (pt), Romanian, Russian, Serbian - Cyrillic, Serbian - Latin, Simplified Chinese, Slovak, Slovenian, Spanish, Swedish, Thai, Traditional Chinese, Turkish, Ukrainian, and Vietnamese.

To support additional languages not covered in the above list, developers must provide their own translations. For example, if you develop a Bulgarian application and want to include report messages, such as pagination, in Bulgarian, you must translate the strings used in messages displayed in reports.

- Internal Messages Requiring Translation
  Lists internal messages that require translation.
- Interactive Grid Messages Requiring Translation
  Lists the interactive grid messages that require translation.
- Interactive Report Messages Requiring Translation
  Lists interactive reports messages that require translation.
- Progressive Web App (PWA) Messages Requiring Translation
  Lists Progressive Web Application (PWA) messages that require translation.

- Search and Smart Filters Messages Requiring Translation
  Lists faceted search and smart filter messages that require translation.

> **See Also:**
>
> - About Installing Translated Versions of Oracle APEX in *Oracle APEX Installation Guide*
> - Managing Text Messages

22.7.3.1 Internal Messages Requiring Translation

Lists internal messages that require translation.

Message Name	English Text
APEX.ACTIONS.TOGGLE	Toggle %0
APEX.ACTIVE_STATE	(Active)
APEX.AUTHENTICATION.LOGIN_THROTTLE.COUNTER	Please wait %0 seconds to login again.
APEX.AUTHENTICATION.LOGIN_THROTTLE.ERROR	The login attempt has been blocked.
APEX.COMBOBOX.LIST_OF_VALUES	List of Values
APEX.COMBOBOX.SHOW_ALL_VALUES	Open list for: %0
APEX.COMPLETED_STATE	(Completed)
APEX.CORRECT_ERRORS	Correct errors before saving.
APEX.CS.MATCHES_FOUND	%0 matches found
APEX.CS.NO_MATCHES	No matches found
APEX.DATA_HAS_CHANGED	Current version of data in database has changed since user initiated update process. current checksum = "%0" application checksum = "%1".
APEX.DATEPICKER.VALUE_MUST_BE_ON_OR_AFTER	#LABEL# must be on or after %0.
APEX.DATEPICKER.VALUE_MUST_BE_ON_OR_BEFORE	#LABEL# must be on or before %0.
APEX.DATEPICKER.VALUE_MUST_BE_BETWEEN	#LABEL# must be between %0 and %1.
APEX.DATEPICKER.VALUE_INVALID	#LABEL# must be a valid date, for example %0.
APEX.DIALOG.CANCEL	Cancel
APEX.DIALOG.CLOSE	Close
APEX.DIALOG.HELP	Help
APEX.DIALOG.OK	OK
APEX.DIALOG.SAVE	Save
APEX.ERROR.PAGE_NOT_AVAILABLE	Sorry, this page isn't available
APEX.ERROR_MESSAGE_HEADING	Error Message

Message Name	English Text
APEX.FILE_BROWSE.DOWNLOAD_LINK_TEXT	Download
APEX.GO_TO_ERROR	Go to error
APEX.ITEM.HELP_TEXT	Help Text: %0
APEX.ITEM.NOT_FOUND	Item %0 not found.
APEX.ITEM_TYPE.SLIDER.VALUE_NOT_BETWEEN_MIN_MAX	#LABEL# is not between the valid range of %0 and %1.
APEX.ITEM_TYPE.SLIDER.VALUE_NOT_MULTIPLE_OF_STEP	#LABEL# is not a multiple of %0.
APEX.ITEM_TYPE.YES_NO.INVALID_VALUE	#LABEL# must match to the values %0 and %1.
APEX.ITEM_TYPE.YES_NO.NO_LABEL	No
APEX.ITEM_TYPE.YES_NO.YES_LABEL	Yes
APEX.MENU.CURRENT_MENU	current
APEX.MENU.OVERFLOW_LABEL	More...
APEX.MENU.PROCESSING	Loading
APEX.NUMBER_FIELD.VALUE_GREATER_MAX_VALUE	#LABEL# is greater than specified maximum %0.
APEX.NUMBER_FIELD.VALUE_INVALID	#LABEL# must be Numeric.
APEX.NUMBER_FIELD.VALUE_INVALID2	#LABEL# does not match number format %0 (For example, %1).
APEX.NUMBER_FIELD.VALUE_LESS_MIN_VALUE	#LABEL# is less than specified minimum %0.
APEX.NUMBER_FIELD.VALUE_NOT_BETWEEN_MIN_MAX	#LABEL# is not between the valid range of %0 and %1.
APEX.PAGE.DUPLICATE_SUBMIT	This page was already submitted and can not be re-submitted.
APEX.PAGE_ITEM_IS_REQUIRED	#LABEL# must have some value.
APEX.POPUP_LOV.ICON_TEXT	Popup List of Values: %0.
APEX.PROCESSING	Processing
APEX.REGION.JQM_LIST_VIEW.LOAD_MORE	Load more
APEX.REGION.JQM_LIST_VIEW.SEARCH	Search
APEX.RICH_TEXT_EDITOR.ACCESSIBLE_LABEL	%0, rich text editor
APEX.SINCE.SHORT.DAYS_AGO	%0d
APEX.SINCE.SHORT.DAYS_FROM_NOW	in %0d
APEX.SINCE.SHORT.HOURS_AGO	%0h
APEX.SINCE.SHORT.HOURS_FROM_NOW	in %0h
APEX.SINCE.SHORT.MINUTES_AGO	%0m
APEX.SINCE.SHORT.MINUTES_FROM_NOW	in %0m
APEX.SINCE.SHORT.MONTHS_AGO	%0mo
APEX.SINCE.SHORT.MONTHS_FROM_NOW	in %0mo
APEX.SINCE.SHORT.SECONDS_AGO	%0s
APEX.SINCE.SHORT.SECONDS_FROM_NOW	in %0s

Message Name	English Text
APEX.SINCE.SHORT.WEEKS_AGO	%0w
APEX.SINCE.SHORT.WEEKS_FROM_NOW	in %0w
APEX.SINCE.SHORT.YEARS_AGO	%0y
APEX.SINCE.SHORT.YEARS_FROM_NOW	in %0y
APEX.SUCCESS_MESSAGE_HEADING	Success Message
APEX.TABS.NEXT	Next
APEX.TABS.PREVIOUS	Previous
APEX.TB.TOOLBAR	Toolbar
APEX.TEMPLATE.EXPAND_COLLAPSE_NAV_LABEL	Expand / Collapse Navigation
APEX.TEMPLATE.EXPAND_COLLAPSE_SIDE_COL_LABEL	Expand / Collapse Side Column
APEX.THEMES.INVALID_THEME_NUMBER	Theme number is invalid or theme is not a current UI theme.
APEX.UI.BACK_TO_TOP	Start of page
APEX.VALUE_REQUIRED	Value Required
CHART_SERIES_ERROR	Chart series error %0 for %1.
FLOW.SINGLE_VALIDATION_ERROR	1 error has occurred.
FLOW.VALIDATION_ERROR	%0 errors have occurred.
INVALID_CREDENTIALS	Invalid Login Credentials .
LAYOUT.CHART	Chart
LAYOUT.T_#EXPAND_COLLAPSE_NAV_LABEL#	Label for Expand / Collapse Navigation
LAYOUT.T_#EXPAND_COLLAPSE_SIDE_COL_LABEL#	Label for Expand / Collapse Side Column
OUT_OF_RANGE	Invalid set of rows requested, the source data of the report has been modified.
PAGINATION.NEXT	Next
PAGINATION.NEXT_SET	Next Set
PAGINATION.PREVIOUS	Previous
PAGINATION.PREVIOUS_SET	Previous Set
REGION_NAME.NATIVE_JET_CHART	Chart
REPORT_TOTAL	report total
RESET	reset pagination
SHOW_ALL	Show All
SINCE_DAYS_AGO	%0 days ago
SINCE_DAYS_FROM_NOW	%0 days from now
SINCE_HOURS_AGO	%0 hours ago
SINCE_HOURS_FROM_NOW	%0 hours from now
SINCE_MINUTES_AGO	%0 minutes ago
SINCE_MINUTES_FROM_NOW	%0 minutes from now
SINCE_MONTHS_AGO	%0 months ago

Message Name	English Text
SINCE_MONTHS_FROM_NOW	%0 months from now
SINCE_NOW	Now
SINCE_SECONDS_AGO	%0 seconds ago
SINCE_SECONDS_FROM_NOW	%0 seconds from now
SINCE_WEEKS_AGO	%0 weeks ago
SINCE_WEEKS_FROM_NOW	%0 weeks from now
SINCE_YEARS_AGO	%0 years ago
SINCE_YEARS_FROM_NOW	%0 years from now
TOTAL	Total
UI_PAGE_SKIP_TO_CONTENT	Skip to Main Content
UPGRADE_CANDLESTICK_CHART	After upgrading, ensure the series attribute Label Column is mapped to a Date / Timestamp column.
WWV_FLOW_CREATE_MODEL_APP.CREATE_IG	Unable to create interactive grid page. %0
WWV_FLOW_CUSTOMIZE.T_MESSAGE3	You can personalize the appearance of this application by changing the Theme Style. Please select a Theme Style from the list below and click on Apply Changes.
WWV_FLOW_CUSTOMIZE.T_REGION_DISP	Region Display
WWV_FLOW_CUSTOMIZE.T_REMOVE_STYLE	Use Application Default Style
WWV_FLOW_CUSTOMIZE.T_THEME_STYLE	Appearance
WWV_FLOW_DATA_EXPORT.AGG_COLUMN_IDX_NOT_EXIST	The column index referenced in the aggregate %0 does not exist.
WWV_FLOW_DATA_EXPORT.COLUMN_BREAK_MUST_BE_IN_THE_BEGGINING	The column break needs to be in the beggining of the columns array.
WWV_FLOW_DATA_EXPORT.COLUMN_GROUP_IDX_NOT_EXIST	The column group index referenced in %0 does not exist.
WWV_FLOW_DATA_EXPORT.HIGHLIGHT_COLUMN_IDX_NOT_EXIST	The column index referenced in the highlight %0 does not exist.
WWV_FLOW_DATA_EXPORT.PARENT_GROUP_IDX_NOT_EXIST	The parent group index referenced in %0 does not exist.
WWV_FLOW_UTILITIES.CAL	Calendar
WWV_FLOW_UTILITIES.CLOSE	Close
WWV_FLOW_UTILITIES.OK	Ok
WWV_FLOW_WEB_SERVICES.AUTHENTICATION_FAILED	Authentication failed.
WWV_FLOW_WEB_SERVICES.NO_VALID_OAUTH_TOKEN	OAuth access token not available or expired.
WWV_FLOW_WEB_SERVICES.UNSUPPORTED_OAUTH_TOKEN	Server responded with unsupported OAuth token type.
WWV_RENDER_REPORT3.FOUND_BUT_NOT_DISPLAYED	Minimum row requested: %0, rows found but not displayed: %1
WWV_RENDER_REPORT3.SORT_BY_THIS_COLUMN	Sort by this column.

Message Name	English Text
WWV_RENDER_REPORT3.UNSAVED_DATA	This form contains unsaved changes. Press Ok to proceed without saving your changes.
WWV_RENDER_REPORT3.X_Y_OF_MORE_THAN_Z	row(s) %0 - %1 of more than %2
WWV_RENDER_REPORT3.X_Y_OF_Z	row(s)%0 - %1 of %2
WWV_RENDER_REPORT3.X_Y_OF_Z_2	%0 - %1 of %2

22.7.3.2 Interactive Grid Messages Requiring Translation

Lists the interactive grid messages that require translation.

Message Name	English Text
APEX.GV.BREAK_COLLAPSE	Collapse control break
APEX.GV.BREAK_EXPAND	Expand control break
APEX.GV.DELETED_COUNT	%0 rows deleted
APEX.GV.DUP_REC_ID	Duplicate identity
APEX.GV.FIRST_PAGE	First
APEX.GV.FOOTER_LANDMARK	Grid Footer
APEX.GV.LAST_PAGE	Last
APEX.GV.LOAD_ALL	Load All
APEX.GV.LOAD_MORE	Show More
APEX.GV.NEXT_PAGE	Next
APEX.GV.PAGE_RANGE	Page rows
APEX.GV.PAGE_RANGE_ENTITY	Page %0
APEX.GV.PAGE_RANGE_XY	%0 - %1
APEX.GV.PAGE_RANGE_XYZ	%0 - %1 of %2
APEX.GV.PAGE_SELECTION	Page Selection
APEX.GV.PAGINATION_LANDMARK	Pagination
APEX.GV.PREV_PAGE	Previous
APEX.GV.ROW_ADDED	Added
APEX.GV.ROW_CHANGED	Changed
APEX.GV.ROW_DELETED	Deleted
APEX.GV.ROW_HEADER	Row header
APEX.GV.SELECT_ALL	Select All
APEX.GV.SELECT_PAGE_N	Page %0
APEX.GV.SELECT_ROW	Select Row
APEX.GV.SELECTED_ENTITY_COUNT	%0 %1 selected
APEX.GV.SELECTION_CELL_COUNT	%0 cells selected
APEX.GV.SELECTION_COUNT	%0 rows selected
APEX.GV.SORT_ASCENDING	Sort Ascending

Message Name	English Text
APEX.GV.SORT_ASCENDING_ORDER	Sort Ascending %0
APEX.GV.SORT_DESCENDING	Sort Descending
APEX.GV.SORT_DESCENDING_ORDER	Sort Descending %0
APEX.GV.SORT_OFF	Don't Sort
APEX.GV.STATE_ICONS_LANDMARK	State Icons
APEX.GV.STATUS_LANDMARK	Grid Status
APEX.GV.TOTAL_ENTITY_PLURAL	%0 %1
APEX.GV.TOTAL_ENTITY_SINGULAR	1 %0
APEX.GV.TOTAL_PAGES	Total %0
APEX.IG.ACC_LABEL	Interactive Grid %0
APEX.IG.ACTIONS	Actions
APEX.IG.ADD	Add
APEX.IG.ADD_ROW	Add Row
APEX.IG.AGGREGATE	Aggregate
APEX.IG.AGGREGATION	Aggregation
APEX.IG.ALL	All
APEX.IG.ALL_TEXT_COLUMNS	All Text Columns
APEX.IG.ALTERNATIVE	Alternative
APEX.IG.AND	and
APEX.IG.APPROX_COUNT_DISTINCT	Approx. Count Distinct
APEX.IG.APPROX_COUNT_DISTINCT_OVERALL	Overall Approx. Count Distinct
APEX.IG.AREA	Area
APEX.IG.ASCENDING	Ascending
APEX.IG.AUTHORIZATION	Authorization
APEX.IG.AUTO	Auto
APEX.IG.AVG	Average
APEX.IG.AVG_OVERALL	Overall Average
APEX.IG.AXIS_LABEL_TITLE	Label Axis Title
APEX.IG.AXIS_VALUE_DECIMAL	Decimal Places
APEX.IG.AXIS_VALUE_TITLE	Value Axis Title
APEX.IG.BACKGROUND_COLOR	Background Color
APEX.IG.BAR	Bar
APEX.IG.BETWEEN	between
APEX.IG.BOTH	Both
APEX.IG.BUBBLE	Bubble
APEX.IG.CANCEL	Cancel
APEX.IG.CASE_SENSITIVE	Case Sensitive
APEX.IG.CASE_SENSITIVE_WITH_BRACKETS	(Case Sensitive)

Message Name	English Text
APEX.IG.CHANGE_VIEW	Change View
APEX.IG.CHANGES_SAVED	Changes saved
APEX.IG.CHART	Chart
APEX.IG.CHART_VIEW	Chart View
APEX.IG.CLOSE_COLUMN	Close
APEX.IG.COLOR_BLUE	Blue
APEX.IG.COLOR_GREEN	Green
APEX.IG.COLOR_ORANGE	Orange
APEX.IG.COLOR_RED	Red
APEX.IG.COLOR_YELLOW	Yellow
APEX.IG.COLORS	Colors
APEX.IG.COLUMN	Column
APEX.IG.COLUMN_CONTEXT	Column %0
APEX.IG.COLUMN_TYPE	Column Purpose
APEX.IG.COLUMNS	Columns
APEX.IG.COPY_CB	Copy to Clipboard
APEX.IG.COMPLEX	Complex
APEX.IG.COMPUTE	Compute
APEX.IG.CONTAINS	contains
APEX.IG.CONTROL_BREAK	Control Break
APEX.IG.COUNT	Count
APEX.IG.COUNT_DISTINCT	Count Distinct
APEX.IG.COUNT_DISTINCT_OVERALL	Overall Count Distinct
APEX.IG.COUNT_OVERALL	Overall Count
APEX.IG.CREATE_X	Create %0
APEX.IG.DATA	Data
APEX.IG.DATA_TYPE	Data Type
APEX.IG.DATE	Date
APEX.IG.DAYS	days
APEX.IG.DEFAULT_SETTINGS	Default Settings
APEX.IG.DEFAULT_TYPE	Default Type
APEX.IG.DELETE	Delete
APEX.IG.DELETE_REPORT_CONFIRM	Are you sure you would like to delete this report?
APEX.IG.DELETE_ROW	Delete Row
APEX.IG.DELETE_ROWS	Delete Rows
APEX.IG.DESCENDING	Descending
APEX.IG.DETAIL	Detail
APEX.IG.DETAIL_VIEW	Detail View

Message Name	English Text
APEX.IG.DIRECTION	Direction
APEX.IG.DISABLED	Disabled
APEX.IG.DOES_NOT_CONTAIN	does not contain
APEX.IG.DOES_NOT_START_WITH	does not start with
APEX.IG.DONUT	Donut
APEX.IG.DOWNLOAD	Download
APEX.IG.DOWNLOAD_FORMAT	Choose Format
APEX.IG.DUPLICATE_AGGREGATION	Duplicate Aggregation
APEX.IG.DUPLICATE_CONTROLBREAK	Duplicate Control Break
APEX.IG.DUPLICATE_ROW	Duplicate Row
APEX.IG.DUPLICATE_ROWS	Duplicate Rows
APEX.IG.EDIT	Edit
APEX.IG.EDIT_CHART	Edit Chart
APEX.IG.EDIT_GROUP_BY	Edit Group By
APEX.IG.EMAIL_BCC	Blind Copy (bcc)
APEX.IG.EMAIL_BODY	Message
APEX.IG.EMAIL_CC	Copy (cc)
APEX.IG.EMAIL_SENT	Email sent.
APEX.IG.EMAIL_SUBJECT	Subject
APEX.IG.EMAIL_TO	Recipient (to)
APEX.IG.ENABLED	Enabled
APEX.IG.EQUALS	equals
APEX.IG.EXAMPLE	Example
APEX.IG.EXPRESSION	Expression
APEX.IG.FD_TYPE	Type
APEX.IG.FILTER	Filter
APEX.IG.FILTER_WITH_DOTS	Filter...
APEX.IG.FILTERS	Filters
APEX.IG.FIRST	First
APEX.IG.FLASHBACK	Flashback
APEX.IG.FORMAT	Format
APEX.IG.FORMAT_CSV	CSV
APEX.IG.FORMAT_HTML	HTML
APEX.IG.FORMATMASK	Format Mask
APEX.IG.FREEZE	Freeze
APEX.IG.FUNCTIONS_AND_OPERATORS	Functions and Operators
APEX.IG.FUNNEL	Funnel
APEX.IG.GO	Go

Message Name	English Text
APEX.IG.GREATER_THAN	greater than
APEX.IG.GREATER_THAN_OR_EQUALS	greater than or equals
APEX.IG.GRID	Grid
APEX.IG.GRID_VIEW	Grid View
APEX.IG.GROUP	Group
APEX.IG.GROUP_BY	Group By
APEX.IG.GROUP_BY_VIEW	Group By View
APEX.IG.HD_TYPE	Condition Type
APEX.IG.HEADING	Heading
APEX.IG.HEADING_ALIGN	Heading Alignment
APEX.IG.HELP	Help
APEX.IG.HELP.ACTIONS.EDITING	You can insert, update, and delete data directly within this interactive grid. Insert a new row by clicking the Add Row button. Edit existing data by double-clicking a specific cell. For larger editing work, click Edit to enter editing mode. In editing mode, you can single-click or use the keyboard to edit specific cells. Use the Change menu to duplicate and delete rows. To enable the Change menu, use the checkboxes to select one or more rows. Duplicate a selected row by clicking the Change menu and selecting Duplicate Rows. Delete a selected row by clicking the Change menu and selecting Delete Row.
APEX.IG.HELP.ACTIONS.EDITING_HEADING	Editing Capabilities
APEX.IG.HELP.ACTIONS.INTRO	An interactive grid presents a set of data in a searchable, customizable report. You can perform numerous operations to limit the records returned, and change the way the data is displayed. Use the Search field to filter the records returned. Click Actions to access numerous options for modifying the report layout, or use the Column Heading menus on displayed columns. Use Report Settings to save your customizations to a report. You can also download the data from the report to an external file or email the data to yourself or others. To learn more, see Using Interactive Grids" in Oracle Application Express End User's Guide
APEX.IG.HELP.ACTIONS.INTRO_HEADING	Overview

Message Name	English Text
APEX.IG.HELP.ACTIONS.REPORTING	You can customize the interactive grid to display data in various different ways using the built-in capabilities. Use the Column Heading menus or the Actions menu to determine which columns to display, in what sequence, and freeze columns. You can also define various data filters and sort the data returned. Use the View button (adjacent to the Search field) to access other data views that may have been defined by the application developer. You can also create a chart or view an existing chart. Note: Click Help in the interactive grid dialogs to obtain more detailed information on the selected function.
APEX.IG.HELP.ACTIONS.REPORTING_HEADING	Reporting Capabilities
APEX.IG.HELP.ACTIONS_TITLE	Interactive Grid Help
APEX.IG.HELP.AGGREGATE	Use this dialog to aggregate columns. Aggregated values display at the bottom of the data, or if Control Breaks are defined, at the bottom of each break. Aggregation List The Aggregation list displays defined aggregations. Disable an existing aggregation by deselecting it. Click Add (+) to create a new aggregation, or Delete (−) to remove an existing aggregation. Aggregation Settings Use the form on the right to define the aggregation. Select the Column name and Aggregation type. Optionally, enter a tooltip for the aggregation. If you have defined a Control Break, selecting Show Overall Value displays the overall average, total, or similar value at the bottom of the data. Note: Access the Aggregation dialog in the Actions menu or by clicking the column heading and sum(∑).
APEX.IG.HELP.AGGREGATE_TITLE	Aggregation Help

Message Name	English Text
APEX.IG.HELP.CHART	Use this dialog to define a chart which displays as a separate data view. Select a chart Type, configure the chart settings, and click Save. Chart Settings</br> The chart attributes that display vary depending on the chart type. A number of attributes can be entered to define the chart. Attributes marked with a red asterisk are mandatory. Below are all available attributes across the different chart types (in alphabetical order): - Aggregation - Select how to aggregate the associated chart values. - Close - Select the column that contains the daily stock close price (Stock chart only). - Decimal Places - Enter the number of decimal places to which the values are rounded. - Direction - In relation to the Sort By attribute, specify whether the data is sorted in ascending or descending values. - High - Select the column that contains the high value (Range and Stock charts only). - Label - Select the column that contains the text for each data point. - Label Axis Title - Enter the title that displays on the label axis. - Low - Select the column that contains the low value (Range and Stock charts only). - Nulls - In relation to the Sort By attribute, specify how you want records with null values to be sorted in relation to records with non null values. - Open - Select the column that contains the daily stock opening price (Stock chart only). - Orientation - Select whether the chart elements, such as bars, display vertically or horizontally. - Series - Select the column used for defining your multi-series dynamic query. - Stack - Specify whether the data items are stacked. - Sort By - Select whether the chart is sorted by the label or the value(s). - Target - Select the column to be used for defining the target value on this chart. When set, the Value attribute defines the filled area within the slice and the Target represents the value of the whole slice (Funnel chart only). - Value - Select the column that contains the data to be plotted. - Value Axis Title - Enter the title that displays on the value axis. - Volume - Select the column that contains the daily stock volume (Stock chart only).

Message Name	English Text
	- X - Select the column that contains the x-axis value for this chart (Bubble and Scatter charts only). - Y - Select the column that contains the y-axis value for this chart (Bubble and Scatter charts only). - Z - Select the column that contains the bar width or bubble radius (Bar, Bubble, and Range charts only)
APEX.IG.HELP.CHART_TITLE	Chart Help
APEX.IG.HELP.COLUMNS	Use this dialog to choose which columns display and in what order. Hide a column by deselecting it. Reorder columns by clicking Move Up (↑) or Move Down (↓). Use the drop down selector to list All columns, Displayed columns, or Not Displayed columns. Optionally, use the form to specify the minimum width of a column in pixels. Note: You can also reorder displayed columns by clicking the drag handle (at the start of the column heading) and dragging the column left or right. You can also change the column width of displayed columns by selecting the column separator, between headings, and moving it left or right.
APEX.IG.HELP.COLUMNS_TITLE	Columns Help
APEX.IG.HELP.COMPUTE	Use this dialog to define additional columns based on mathematical and functional computations performed against existing columns. Computation List The Computation list displays defined computations. Disable an existing computation by deselecting it. Click Add (&plus) to add a new computation, or Delete (−) to remove an existing computation. Computation Settings Use the form to define the computation. Enter the column details such as heading, label, and select alignment settings. Use the Expression textarea to enter the column(s) and associated functions for the computation. Select the appropriate data type, and optionally a format mask, for the new column.
APEX.IG.HELP.COMPUTE_TITLE	Compute Help

Message Name	English Text
APEX.IG.HELP.CONTROL_BREAK	Use this dialog to define a control break on one or more columns. Control Break List The Control Break list displays defined control breaks. Disable an existing control break column by deselecting it. Click Add (&plus) to include a new column in the control break, or Delete (−) to remove an existing column from the control break. To reorder columns, click Move Up (↑) or Move Down (↓) to move the selected column up and down relative to other columns. Control Break Settings Use the form to define the control break column. Selected a control break column, the sort direction, and how to order null columns (columns with no value). Note: When viewing the interactive grid, you can define a control break by clicking a Column Heading and selecting the control break icon.
APEX.IG.HELP.CONTROL_BREAK_TITLE	Control Break Help
APEX.IG.HELP.DOWNLOAD	Use this dialog to download all the current rows to an external file. The file will contain only the currently displayed columns, using any filters and sorts applied to the data. Select the file format and click Download. Note: CSV will not include text formatting such as aggregates and control breaks. To email the file, select Send as Email and enter the email details (Recipient, Subject and Message).
APEX.IG.HELP.DOWNLOAD_TITLE	Download Help

APEX.IG.HELP.FILTER	Use this dialog to configure data filters which limit the rows returned.
	 Filter List
	
	The Filter list displays defined filters. Disable an existing filter by deselecting it.
	
	Click Add (&plus) to create a new filter, or Delete (−) to remove an existing filter.
	 Filter Settings
	
	Use the form to define the filter properties.
	
	Select the appropriate filter type:
	
	  ; Row - filter for a term in any filterable column.
	
	  ; Column - filter a specific column with a specified operator and value.
	 Note: When viewing the interactive grid, you can define row filters by typing directly into the Search field. Click Select Columns to Search to limit the search to a specific column. Alternately, open a Column Heading menu and select a value to create a column filter.

APEX.IG.HELP.FILTER_TITLE	Filter Help

Message Name	English Text
APEX.IG.HELP.FLASHBACK	Use this dialog to view the data as it existed at a previous point in time. Enter the number of minutes in the past to execute the flashback query.
APEX.IG.HELP.FLASHBACK_TITLE	Flashback Help
APEX.IG.HELP.GROUP_BY_TITLE	Group By Help

APEX.IG.HELP.HIGHLIGHT	Use this dialog to highlight rows or columns of data based on the condition entered.
	
	 Highlight List
	
	The Highlight list displays defined highlights. Disable an existing highlight by deselecting it.
	
	Click Add (&plus) to create a new highlight, or Delete (−) to remove an existing highlight.
	
	 Highlight Settings
	
	Use the form to define the highlight properties.
	
	Enter the name, select Row or Column, and select the HTML color codes for the background and text.
	
	Select the appropriate Condition Type to highlight specific data:
	
	  ; Row - highlight the term in any column.
	
	  ; Column - highlight within a specific column based on the specified operator and value.

APEX.IG.HELP.HIGHLIGHT_TITLE	Highlight Help
APEX.IG.HELP.REPORT	Use this dialog to save changes you have made to the current grid layout and configuration. Application developers can define multiple alternate report layouts. Where permissible, you and other end users can save a report as Public, which makes the report available to all other users of the grid. You can also save a report as a Private report that only you can view. Select from the available types and enter a name for the saved report.
APEX.IG.HELP.REPORT_TITLE	Report Help

Message Name	English Text
APEX.IG.HELP.SORT	Use this dialog to set the display order. Sort List The Sort dialog displays a list of configured sorting rules. Click Add (&plus) to create a sort column, or Delete (−) to remove a sort column. Click Move Up (↑) and Move Down (↓) to move the selected sort column up and down relative to the other sort columns. Sort Settings Select a sort column, the sort direction, and how to order null columns (columns with no value). Note: Data can be sorted by columns which are not displayed; however, not all columns may be sortable. Displayed columns can be sorted by pressing the up (ascending) or down (descending) arrows at the end of the column headings. To add a subsequent column to an existing sort, hold the Shift key and click the up or down arrow.
APEX.IG.HELP.SORT_TITLE	Sort Help
APEX.IG.HELP.SUBSCRIPTION_TITLE	Subscription Help
APEX.IG.HIDE	Hide
APEX.IG.HIGH_COLUMN	High
APEX.IG.HIGHLIGHT	Highlight
APEX.IG.HORIZONTAL	Horizontal
APEX.IG.HOURS	hours
APEX.IG.ICON	Icon
APEX.IG.ICON_VIEW	Icon View
APEX.IG.IN	in
APEX.IG.IN_THE_LAST	in the last
APEX.IG.IN_THE_NEXT	in the next
APEX.IG.INACTIVE_SETTING	Inactive Setting
APEX.IG.INACTIVE_SETTINGS	Inactive Settings
APEX.IG.INTERNAL_ERROR	An internal error has occurred while processing the Interactive Grid request.
APEX.IG.INVALID_DATE_FORMAT	Invalid Date Format
APEX.IG.INVALID_SETTING	Invalid Setting
APEX.IG.INVALID_SETTINGS	Invalid Settings
APEX.IG.INVALID_SORT_BY	Sort By has been set to %0, but no column as been selected for %0.
APEX.IG.INVALID_VALUE	Invalid Value
APEX.IG.INVISIBLE	Not Displayed
APEX.IG.IS_NOT_NULL	is not empty

Message Name	English Text
APEX.IG.IS_NULL	is empty
APEX.IG.LABEL	Label
APEX.IG.LABEL_COLUMN	Label
APEX.IG.LAST	Last
APEX.IG.LAST.DAY	Last Day
APEX.IG.LAST.HOUR	Last Hour
APEX.IG.LAST.MINUTE	Last Minute
APEX.IG.LAST.MONTH	Last Month
APEX.IG.LAST.WEEK	Last Week
APEX.IG.LAST.X_DAYS	Last %0 Days
APEX.IG.LAST.X_HOURS	Last %0 Hours
APEX.IG.LAST.X_MINUTES	Last %0 Minutes
APEX.IG.LAST.X_MONTHS	Last %0 Months
APEX.IG.LAST.X_WEEKS	Last %0 Weeks
APEX.IG.LAST.X_YEARS	Last %0 Years
APEX.IG.LAST.YEAR	Last Year
APEX.IG.LAYOUT_ALIGN	Cell Alignment
APEX.IG.LAYOUT_USEGROUPFOR	Use Group For
APEX.IG.LESS_THAN	less than
APEX.IG.LESS_THAN_OR_EQUALS	less than or equals
APEX.IG.LINE	Line
APEX.IG.LINE_WITH_AREA	Line with Area
APEX.IG.LISTAGG	Listagg
APEX.IG.LOW_COLUMN	Low
APEX.IG.MAILADDRESSES_COMMASEP	Separate multiple Addresses with commas
APEX.IG.MATCHES_REGULAR_EXPRESSION	matches regular expression
APEX.IG.MAX	Maximum
APEX.IG.MAX_OVERALL	Overall Maximum
APEX.IG.MEDIAN	Median
APEX.IG.MEDIAN_OVERALL	Overall Median
APEX.IG.MIN	Minimum
APEX.IG.MIN_OVERALL	Overall Minimum
APEX.IG.MINUTES	minutes
APEX.IG.MINUTES_AGO	Minutes ago
APEX.IG.MONTHS	months
APEX.IG.MORE_DATA_FOUND	The data contains more than %0 rows which exceeds the maximum allowed. Please apply additional filters in order to view the results.
APEX.IG.NAME	Name

Message Name	English Text
APEX.IG.NAMED_REPORT	Named Report
APEX.IG.NEXT.DAY	Next Day
APEX.IG.NEXT.HOUR	Next Hour
APEX.IG.NEXT.MINUTE	Next Minute
APEX.IG.NEXT.MONTH	Next Month
APEX.IG.NEXT.WEEK	Next Week
APEX.IG.NEXT.X_DAYS	Next %0 Days
APEX.IG.NEXT.X_HOURS	Next %0 Hours
APEX.IG.NEXT.X_MINUTES	Next %0 Minutes
APEX.IG.NEXT.X_MONTHS	Next %0 Months
APEX.IG.NEXT.X_WEEKS	Next %0 Weeks
APEX.IG.NEXT.X_YEARS	Next %0 Years
APEX.IG.NEXT.YEAR	Next Year
APEX.IG.NO_DATA_FOUND	No data found
APEX.IG.NOT_BETWEEN	not between
APEX.IG.NOT_EQUALS	not equals
APEX.IG.NOT_EXIST	Region with ID %0 is not an Interactive Grid region or does not exist in application %1.
APEX.IG.NOT_IN	not in
APEX.IG.NOT_IN_THE_LAST	not in the last
APEX.IG.NOT_IN_THE_NEXT	not in the next
APEX.IG.NULLS	Nulls
APEX.IG.NUMBER	Numeric
APEX.IG.OFF	Off
APEX.IG.ON	On
APEX.IG.ONE_MINUTE_AGO	1 minute ago
APEX.IG.OPEN_COLORPICKER	Open Color Picker: %0
APEX.IG.OPEN_COLUMN	Open
APEX.IG.OPERATOR	Operator
APEX.IG.ORIENTATION	Orientation
APEX.IG.PIE	Pie
APEX.IG.PIVOT	Pivot
APEX.IG.PIVOT_VIEW	Pivot View
APEX.IG.PLACEHOLDER_INVALUES	Separate values with "%0"
APEX.IG.POLAR	Polar
APEX.IG.POSITION_CENTER	Center
APEX.IG.POSITION_END	End
APEX.IG.POSITION_START	Start
APEX.IG.PRIMARY	Primary

Message Name	English Text
APEX.IG.PRIMARY_DEFAULT	Primary Default
APEX.IG.PRIMARY_REPORT	Primary Report
APEX.IG.RADAR	Radar
APEX.IG.RANGE	Range
APEX.IG.REFRESH	Refresh
APEX.IG.REFRESH_ROW	Refresh Row
APEX.IG.REFRESH_ROWS	Refresh Rows
APEX.IG.REMOVE_CONTROL	Remove %0
APEX.IG.REPORT	Report
APEX.IG.REPORT.DELETED	Report deleted
APEX.IG.REPORT.SAVED.ALTERNATIVE	Alternative report saved for all users
APEX.IG.REPORT.SAVED.DEFAULT	Default report saved for all users
APEX.IG.REPORT.SAVED.PRIVATE	Private report saved
APEX.IG.REPORT.SAVED.PUBLIC	Public report saved for all users
APEX.IG.REPORT_DATA_AS_OF.X.MINUTES_AGO	Report data as of %0 minutes ago
APEX.IG.REPORT_DATA_AS_OF_ONE_MINUTE_AGO	Report data as of 1 minute ago
APEX.IG.REPORT_EDIT	Report - Edit
APEX.IG.REPORT_SAVE_AS	Report - Save As
APEX.IG.REPORT_SETTINGS	Report Settings
APEX.IG.REPORT_STATIC_ID_DOES_NOT_EXIST	Saved Interactive Grid with static ID %0 does not exist.
APEX.IG.REPORT_VIEW	Report View
APEX.IG.RESET	Reset
APEX.IG.REVERT_CHANGES	Revert Changes
APEX.IG.REVERT_ROWS	Revert Rows
APEX.IG.ROW	Row
APEX.IG.ROW_ACTIONS	Row Actions
APEX.IG.ROW_ACTIONS_FOR	Actions for row %0
APEX.IG.ROW_COLUMN_CONTEXT	Row %0 Column %1
APEX.IG.ROW_CONTEXT	Row %0
APEX.IG.ROWS_PER_PAGE	Rows Per Page
APEX.IG.SAVE	Save
APEX.IG.SAVE_AS	Save As
APEX.IG.SAVE_REPORT_SETTINGS	Save Report Settings
APEX.IG.SAVED_REPORT_DEFAULT	Default
APEX.IG.SAVED_REPORT_PRIVATE	Private
APEX.IG.SAVED_REPORT_PUBLIC	Public
APEX.IG.SAVED_REPORTS	Saved Reports
APEX.IG.SCATTER	Scatter

Message Name	English Text
APEX.IG.SEARCH	Search
APEX.IG.SEARCH.ALL_COLUMNS	Search: All Text Columns
APEX.IG.SEARCH.COLUMN	Search: %0
APEX.IG.SEARCH.ORACLE_TEXT	Search: Full Text
APEX.IG.SEARCH_FOR.X	Search for '%0'
APEX.IG.SEL_ACTIONS	Selection Actions
APEX.IG.SELECT	- Select -
APEX.IG.SELECT_1_ROW_IN_MASTER	Select 1 row in the parent region
APEX.IG.SELECT_COLUMNS_TO_SEARCH	Select columns to search
APEX.IG.SELECTION	Cell Selection
APEX.IG.SEL_MODE_CELL	Selection
APEX.IG.SEND_AS_EMAIL	Send as Email
APEX.IG.SERIES_COLUMN	Series
APEX.IG.SHOW_OVERALL_VALUE	Show Overall Value
APEX.IG.SINGLE_ROW_VIEW	Single Row View
APEX.IG.SORT	Sort
APEX.IG.SORT_BY	Sort By
APEX.IG.SORT_ONLY_ONE_PER_COLUMN	You can define only one sort per column.
APEX.IG.SRV_CHANGE_MENU	Change Menu
APEX.IG.STACK	Stack
APEX.IG.STARTS_WITH	starts with
APEX.IG.STOCK	Stock
APEX.IG.STRETCH_COLUMNS	Stretch Column
APEX.IG.SUBSCRIPTION	Subscription
APEX.IG.SUM	Sum
APEX.IG.SUM_OVERALL	Overall Sum
APEX.IG.SUMMARY	Interactive Grid. Report: %0, View: %1.
APEX.IG.TARGET_COLUMN	Target
APEX.IG.TEXT_COLOR	Text Color
APEX.IG.TOGGLE	Toggle
APEX.IG.TOOLTIP	Tooltip
APEX.IG.TYPE	Type
APEX.IG.UNFREEZE	Unfreeze
APEX.IG.UNIT	Unit
APEX.IG.UNSAVED_CHANGES_CONTINUE_CONFIRM	There are unsaved changes. Do you want to continue?
APEX.IG.VALUE	Value
APEX.IG.VALUE_COLUMN	Value
APEX.IG.VALUE_REQUIRED	A value is required.

Message Name	English Text
APEX.IG.VARCHAR2	Text
APEX.IG.VERTICAL	Vertical
APEX.IG.VIEW	View
APEX.IG.VISIBLE	Displayed
APEX.IG.VOLUME_COLUMN	Volume
APEX.IG.WEEKS	weeks
APEX.IG.WIDTH	Minimum Column Width (Pixel)
APEX.IG.X.BETWEEN.Y.AND.Z	%0 between %1 and %2
APEX.IG.X.CONTAINS.Y	%0 contains %1
APEX.IG.X.DOES_NOT_CONTAIN.Y	%0 does not contain %1
APEX.IG.X.EQUALS.Y	%0 equals %1
APEX.IG.X.GREATER_THAN.Y	%0 greater than %1
APEX.IG.X.GREATER_THAN_OR_EQUALS.Y	%0 greater than or equal to %1
APEX.IG.X.IN.Y	%0 in %1
APEX.IG.X.IN_THE_LAST.Y.DAYS	%0 in the last %1 days
APEX.IG.X.IN_THE_LAST.Y.HOURS	%0 in the last %1 hours
APEX.IG.X.IN_THE_LAST.Y.MINUTES	%0 in the last %1 minutes
APEX.IG.X.IN_THE_LAST.Y.MONTHS	%0 in the last %1 months
APEX.IG.X.IN_THE_LAST.Y.WEEKS	%0 in the last %1 weeks
APEX.IG.X.IN_THE_LAST.Y.YEARS	%0 in the last %1 years
APEX.IG.X.IN_THE_LAST_DAY	%0 in the last day
APEX.IG.X.IN_THE_LAST_HOUR	%0 in the last hour
APEX.IG.X.IN_THE_LAST_MINUTE	%0 in the last minute
APEX.IG.X.IN_THE_LAST_MONTH	%0 in the last month
APEX.IG.X.IN_THE_LAST_WEEK	%0 in the last week
APEX.IG.X.IN_THE_LAST_YEAR	%0 in the last year
APEX.IG.X.IN_THE_NEXT.Y.DAYS	%0 in the next %1 days
APEX.IG.X.IN_THE_NEXT.Y.HOURS	%0 in the next %1 hours
APEX.IG.X.IN_THE_NEXT.Y.MINUTES	%0 in the next %1 minutes
APEX.IG.X.IN_THE_NEXT.Y.MONTHS	%0 in the next %1 months
APEX.IG.X.IN_THE_NEXT.Y.WEEKS	%0 in the next %1 weeks
APEX.IG.X.IN_THE_NEXT.Y.YEARS	%0 in the next %1 years
APEX.IG.X.IN_THE_NEXT_DAY	%0 in the next day
APEX.IG.X.IN_THE_NEXT_HOUR	%0 in the next hour
APEX.IG.X.IN_THE_NEXT_MINUTE	%0 in the next minute
APEX.IG.X.IN_THE_NEXT_MONTH	%0 in the next month
APEX.IG.X.IN_THE_NEXT_WEEK	%0 in the next week
APEX.IG.X.IN_THE_NEXT_YEAR	%0 in the next year

Message Name	English Text
APEX.IG.X.IS_NOT_NULL	%0 is not empty
APEX.IG.X.IS_NULL	%0 is empty
APEX.IG.X.LESS_THAN.Y	%0 less than %1
APEX.IG.X.LESS_THAN_OR_EQUALS.Y	%0 less than or equal to %1
APEX.IG.X.LIKE.Y	%0 like %1
APEX.IG.X.MATCHES_REGULAR_EXPRESSION.Y	%0 matches regular expression %1
APEX.IG.X.MINUTES_AGO	%0 minutes ago
APEX.IG.X.NOT_BETWEEN.Y.AND.Z	%0 not between %1 and %2
APEX.IG.X.NOT_EQUALS.Y	%0 not equals %1
APEX.IG.X.NOT_IN.Y	%0 not in %1
APEX.IG.X.NOT_IN_THE_LAST.Y.DAYS	%0 not in the last %1 days
APEX.IG.X.NOT_IN_THE_LAST.Y.HOURS	%0 not in the last %1 hours
APEX.IG.X.NOT_IN_THE_LAST.Y.MINUTES	%0 not in the last %1 minutes
APEX.IG.X.NOT_IN_THE_LAST.Y.MONTHS	%0 not in the last %1 months
APEX.IG.X.NOT_IN_THE_LAST.Y.WEEKS	%0 not in the last %1 weeks
APEX.IG.X.NOT_IN_THE_LAST.Y.YEARS	%0 not in the last %1 years
APEX.IG.X.NOT_IN_THE_LAST_DAY	%0 not in the last day
APEX.IG.X.NOT_IN_THE_LAST_HOUR	%0 not in the last hour
APEX.IG.X.NOT_IN_THE_LAST_MINUTE	%0 not in the last minute
APEX.IG.X.NOT_IN_THE_LAST_MONTH	%0 not in the last month
APEX.IG.X.NOT_IN_THE_LAST_WEEK	%0 not in the last week
APEX.IG.X.NOT_IN_THE_LAST_YEAR	%0 not in the last year
APEX.IG.X.NOT_IN_THE_NEXT.Y.DAYS	%0 not in the next %1 days
APEX.IG.X.NOT_IN_THE_NEXT.Y.HOURS	%0 not in the next %1 hours
APEX.IG.X.NOT_IN_THE_NEXT.Y.MINUTES	%0 not in the next %1 minutes
APEX.IG.X.NOT_IN_THE_NEXT.Y.MONTHS	%0 not in the next %1 months
APEX.IG.X.NOT_IN_THE_NEXT.Y.WEEKS	%0 not in the next %1 weeks
APEX.IG.X.NOT_IN_THE_NEXT.Y.YEARS	%0 not in the next %1 years
APEX.IG.X.NOT_IN_THE_NEXT_DAY	%0 not in the next day
APEX.IG.X.NOT_IN_THE_NEXT_HOUR	%0 not in the next hour
APEX.IG.X.NOT_IN_THE_NEXT_MINUTE	%0 not in the next minute
APEX.IG.X.NOT_IN_THE_NEXT_MONTH	%0 not in the next month
APEX.IG.X.NOT_IN_THE_NEXT_WEEK	%0 not in the next week
APEX.IG.X.NOT_IN_THE_NEXT_YEAR	%0 not in the next year
APEX.IG.X.NOT_LIKE.Y	%0 not like %1
APEX.IG.X.STARTS_WITH.Y	%0 starts with %1
APEX.IG.X_COLUMN	X
APEX.IG.Y_COLUMN	Y

Message Name	English Text
APEX.IG.YEARS	years
APEX.IG.Z_COLUMN	Z
APEX.IG_FORMAT_SAMPLE_1	Monday, 12 January, 2016
APEX.IG_FORMAT_SAMPLE_2	January
APEX.IG_FORMAT_SAMPLE_3	16 hours ago
APEX.IG_FORMAT_SAMPLE_4	in 16h
APEX.RV.NOT_GROUPED_LABEL	Other Columns
APEX.RV.REC_XY	Row %0 of %1
APEX.RV.REC_X	Row %0
APEX.RV.EXCLUDE_HIDDEN	Displayed Columns
APEX.RV.EXCLUDE_NULL	Exclude Null Values
APEX.RV.PREV_RECORD	Previous
APEX.RV.NEXT_RECORD	Next
APEX.RV.SETTINGS_MENU	Settings Menu
APEX.RV.INSERT	Add
APEX.RV.DELETE	Delete
APEX.TMV.SELECTION_COUNT	%0 items selected
APEX.TMV.SELECTOR_LABEL	Select Item
APEX.TMV.SELECTOR_LABEL_1	Select %0

22.7.3.3 Interactive Report Messages Requiring Translation

Lists interactive reports messages that require translation.

Message Name	English Text
4150_COLUMN_NUMBER	Column %0
APEX.DIALOG.CLOSE	Close
APEXIR_3D	3D
APEXIR_ACTIONS	Actions
APEXIR_ACTIONS_MENU	Actions Menu
APEXIR_ADD	Add
APEXIR_ADD_FUNCTION	Add Function
APEXIR_ADD_GROUP_BY_COLUMN	Add Group By Column
APEXIR_ADD_PIVOT_COLUMN	Add Pivot Column
APEXIR_ADD_ROW_COLUMN	Add Row Column
APEXIR_ADD_SUBSCRIPTION	Add Subscription
APEXIR_AGG_AVG	Average
APEXIR_AGG_COUNT	Count
APEXIR_AGG_MAX	Maximum

Message Name	English Text
APEXIR_AGG_MEDIAN	Median
APEXIR_AGG_MIN	Minimum
APEXIR_AGG_MODE	Mode
APEXIR_AGG_SUM	Sum
APEXIR_AGGREGATE	Aggregate
APEXIR_AGGREGATE_DESCRIPTION	Aggregates are displayed after each control break and at the end of the report.
APEXIR_AGGREGATION	Aggregation
APEXIR_ALL	All
APEXIR_ALL_COLUMNS	All Columns
APEXIR_ALL_ROWS	All Rows
APEXIR_ALTERNATIVE	Alternative
APEXIR_ALTERNATIVE_DEFAULT_NAME	Alternative Default: %0
APEXIR_AND	and
APEXIR_APPLY	Apply
APEXIR_AS_OF	As of %0
APEXIR_ASCENDING	Ascending
APEXIR_AVERAGE_X	Average %0
APEXIR_BETWEEN	between
APEXIR_BGCOLOR	Background Color
APEXIR_BLUE	blue
APEXIR_BOTTOM	Bottom
APEXIR_BTNS_NEXT_TO_SEARCH_OF	Buttons next to search bar of %0
APEXIR_CALENDAR	Calendar
APEXIR_CANCEL	Cancel
APEXIR_CATEGORY	Category
APEXIR_CELL	Cell
APEXIR_CHART	Chart
APEXIR_CHART_INITIALIZING	Initializing...
APEXIR_CHART_LABEL_NOT_NULL	Chart label must be specified
APEXIR_CHART_MAX_ROW_CNT	The maximum row count for a chart query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query.
APEXIR_CHART_TYPE	Chart Type
APEXIR_CHECK_ALL	Check All
APEXIR_CHOOSE_DOWNLOAD_FORMAT	Choose report download format
APEXIR_CLEAR	clear
APEXIR_COLUMN	Column

Message Name	English Text
APEXIR_COLUMN_ALIASES	Column Aliases
APEXIR_COLUMN_FILTER	Filter...
APEXIR_COLUMN_HEADING_MENU	Column Heading Menu
APEXIR_COLUMN_INFO	Column Information
APEXIR_COLUMN_LABEL	Column Label
APEXIR_COLUMN_N	Column %0
APEXIR_COLUMNS	Columns
APEXIR_COMPARISON_CONTAINS	contains
APEXIR_COMPARISON_DOESNOT_CONTAIN	does not contain
APEXIR_COMPARISON_IN	in
APEXIR_COMPARISON_IS_IN_LAST	is in the last
APEXIR_COMPARISON_IS_IN_NEXT	is in the next
APEXIR_COMPARISON_IS_NOT_NULL	is not null
APEXIR_COMPARISON_IS_NULL	is null
APEXIR_COMPARISON_ISNOT_IN_LAST	is not in the last
APEXIR_COMPARISON_ISNOT_IN_NEXT	is not in the next
APEXIR_COMPARISON_LIKE	like
APEXIR_COMPARISON_NOT_IN	not in
APEXIR_COMPARISON_NOT_LIKE	not like
APEXIR_COMPARISON_REGEXP_LIKE	matches regular expression
APEXIR_COMPUTATION	Computation
APEXIR_COMPUTATION_EXPRESSION	Computation Expression
APEXIR_COMPUTATION_FOOTER	Create a computation using column aliases.
APEXIR_COMPUTATION_FOOTER_E1	(B+C)*100
APEXIR_COMPUTATION_FOOTER_E2	INITCAP(B)||', '||INITCAP(C)
APEXIR_COMPUTATION_FOOTER_E3	CASE WHEN A = 10 THEN B + C ELSE B END
APEXIR_COMPUTE	Compute
APEXIR_CONTROL_BREAK	Control Break
APEXIR_CONTROL_BREAK_COLUMNS	Control Break Columns
APEXIR_CONTROL_BREAKS	Control Breaks
APEXIR_COUNT_DISTINCT	Count Distinct
APEXIR_COUNT_DISTINCT_X	Count Distinct
APEXIR_COUNT_X	Count %0
APEXIR_DAILY	Daily
APEXIR_DATA	Data
APEXIR_DATA_AS_OF	Report data as of %0 minutes ago.
APEXIR_DATE	Date
APEXIR_DAY	Day

Message Name	English Text
APEXIR_DEFAULT	Default
APEXIR_DEFAULT_REPORT_TYPE	Default Report Type
APEXIR_DELETE	Delete
APEXIR_DELETE_CHECKED	Delete Checked
APEXIR_DELETE_CONFIRM	Would you like to delete these report settings?
APEXIR_DELETE_CONFIRM_JS_DIALOG	Would you like to perform this delete action?
APEXIR_DELETE_DEFAULT_REPORT	Delete Default Report
APEXIR_DELETE_REPORT	Delete Report
APEXIR_DESCENDING	Descending
APEXIR_DESCRIPTION	Description
APEXIR_DETAIL_VIEW	Single Row View
APEXIR_DIRECTION	Direction
APEXIR_DISABLE	Disable
APEXIR_DISABLED	Disabled
APEXIR_DISPLAY	Display
APEXIR_DISPLAY_IN_REPORT	Display in Report
APEXIR_DISPLAYED	Displayed
APEXIR_DISPLAYED_COLUMNS	Displayed Columns
APEXIR_DO_NOT_AGGREGATE	- Do not aggregate -
APEXIR_DO_NOT_DISPLAY	Do Not Display
APEXIR_DOWN	Down
APEXIR_DOWNLOAD	Download
APEXIR_DUPLICATE_PIVOT_COLUMN	Duplicate pivot column. Pivot column list must be unique.
APEXIR_EDIT	Edit
APEXIR_EDIT_ALTERNATIVE_DEFAULT	Edit Alternative Default
APEXIR_EDIT_CHART	Edit Chart Settings
APEXIR_EDIT_CHART2	Edit Chart
APEXIR_EDIT_FILTER	Edit Filter
APEXIR_EDIT_GROUP_BY	Edit Group By
APEXIR_EDIT_HIGHLIGHT	Edit Highlight
APEXIR_EDIT_PIVOT	Edit Pivot
APEXIR_EMAIL	Email
APEXIR_EMAIL_ADDRESS	Email Address
APEXIR_EMAIL_BCC	Bcc
APEXIR_EMAIL_BODY	Body
APEXIR_EMAIL_CC	Cc
APEXIR_EMAIL_FREQUENCY	Frequency
APEXIR_EMAIL_NOT_CONFIGURED	Email has not been configured for this application. Please contact your administrator.

Message Name	English Text
APEXIR_EMAIL_REQUIRED	Email Address must be specified.
APEXIR_EMAIL_SEE_ATTACHED	See attached.
APEXIR_EMAIL_SUBJECT	Subject
APEXIR_EMAIL_SUBJECT_REQUIRED	Email Subject must be specified.
APEXIR_EMAIL_TO	To
APEXIR_ENABLE	Enable
APEXIR_ENABLE_DISABLE_ALT	Enable/Disable
APEXIR_ENABLED	Enabled
APEXIR_ERROR	Error
APEXIR_EXAMPLES	Examples
APEXIR_EXAMPLES_WITH_COLON	Examples:
APEXIR_EXCLUDE_NULL	Exclude Null Values
APEXIR_EXPAND_COLLAPSE_ALT	Expand/Collapse
APEXIR_EXPRESSION	Expression
APEXIR_FILTER	Filter
APEXIR_FILTER_EXPRESSION	Filter Expression
APEXIR_FILTER_EXPR_TOO_LONG	The filter expression is too long.
APEXIR_FILTER_TYPE	Filter Type
APEXIR_FILTERS	Filters
APEXIR_FINDER_ALT	Select columns to search.
APEXIR_FLASHBACK	Flashback
APEXIR_FLASHBACK_DESCRIPTION	A flashback query enables you to view the data as it existed at a previous point in time.
APEXIR_FLASHBACK_ERROR_MSG	Unable to perform flashback request.
APEXIR_FORMAT	Format
APEXIR_FORMAT_MASK	Format Mask
APEXIR_FUNCTION	Function
APEXIR_FUNCTION_N	Function %0
APEXIR_FUNCTIONS	Functions
APEXIR_FUNCTIONS_OPERATORS	Functions / Operators
APEXIR_GO	Go
APEXIR_GREEN	green
APEXIR_GROUP_BY	Group By
APEXIR_GROUP_BY_COL_NOT_NULL	Group by column must be specified
APEXIR_GROUP_BY_COLUMN	Group By Column
APEXIR_GROUP_BY_MAX_ROW_CNT	The maximum row count for a Group By query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query.

Message Name	English Text
APEXIR_GROUP_BY_SORT	Group By Sort
APEXIR_GROUP_BY_SORT_ORDER	Group By Sort Order
APEXIR_HCOLUMN	Horizontal Column
APEXIR_HELP	Help
APEXIR_HELP_01	An Interactive Report displays a predetermined set of columns. The report may be further customized with an initial filter clause, a default sort order, control breaks, highlighting, computations, aggregates and a chart. Each Interactive Report can then be further customized and the results can be viewed, or downloaded, and the report definition can be stored for later use. <p/> An Interactive Report can be customized in three ways: the search bar, actions menu and column heading menu.
APEXIR_HELP_ACTIONS_MENU	The actions menu is used to customize the display of your Interactive Report.
APEXIR_HELP_AGGREGATE	Aggregates are mathematical computations performed against a column. Aggregates are displayed after each control break and at the end of the report within the column they are defined. <p/> - Aggregation allows you to select a previously defined aggregation to edit. - Function is the function to be performed (e.g. SUM, MIN). - Column is used to select the column to apply the mathematical function to. Only numeric columns will be displayed.
APEXIR_HELP_CHART	You can include one chart per Interactive Report. Once defined, you can switch between the chart and report views using links below the search bar. <p/> - Chart Type identifies the chart type to include. Select from horizontal bar, vertical bar, pie or line. - Label allows you to select the column to be used as the label. - Axis Title for Label is the title that will display on the axis associated with the column selected for Label. This is not available for pie chart. - Value allows you to select the column to be used as the value. If your function is a COUNT, a Value does not need to be selected. - Axis Title for Value is the title that will display on the axis associated with the column selected for Value. This is not available for pie chart. - Function is an optional function to be performed on the column selected for Value.

Message Name	English Text
APEXIR_HELP_COLUMN_HEADING_MENU	Clicking on any column heading exposes a column heading menu. <p/> - Sort Ascending icon sorts the report by the column in ascending order. - Sort Descending icon sorts the report by the column in descending order. - Hide Column hides the column. - Break Column creates a break group on the column. This pulls the column out of the report as a master record. - Column Information displays help text about the column, if available. - Text Area is used to enter case insensitive search criteria (no need for wild cards). Entering a value will reduce the list of values at the bottom of the menu. You can then select a value from the bottom and the selected value will be created as a filter using '=' (e.g. column = 'ABC'). Alternatively, you can click the flashlight icon and the entered value will be created as a filter with the 'LIKE' modifier (e.g. column LIKE '%ABC%'). - List of Unique Values contains the first 500 unique values that meet your filters. If the column is a date, a list of date ranges is displayed instead. If you select a value, a filter will be created using '=' (e.g. column = 'ABC').

Message Name	English Text
APEXIR_HELP_COMPUTE	Computations allow you to add computed columns to your report. These can be mathematical computations (e.g. NBR_HOURS/24) or standard Oracle functions applied to existing columns (some have been displayed for example, others, like TO_DATE, can also be used). <p/> - Computation allows you to select a previously defined computation to edit. - Column Heading is the column heading for the new column. - Format Mask is an Oracle format mask to be applied against the column (e.g. S9999). - Format Mask is an Oracle format mask to be applied against the column (e.g. S9999). - Computation is the computation to be performed. Within the computation, columns are referenced using the aliases displayed. <p/> Below computation, the columns in your query are displayed with their associated alias. Clicking on the column name or alias will write them into the Computation. Next to Columns is a Keypad. These are simply shortcuts of commonly used keys. On the far right are Functions. <p/> An example computation to display Total Compensation is: <p/> <pre>CASE WHEN A = 'SALES' THEN B + C ELSE B END</pre> (where A is ORGANIZATION, B is SALARY and C is COMMISSION)
APEXIR_HELP_CONTROL_BREAK	Used to create a break group on one or several columns. This pulls the columns out of the Interactive Report and displays them as a master record.
APEXIR_HELP_DETAIL_VIEW	To view the details of a single row at a time, click the single row view icon on the row you want to view. If available, the single row view will always be the first column. Depending on the customization of the Interactive Report, the single row view may be the standard view or a custom page that may allow update.
APEXIR_HELP_DOWNLOAD	Allows the current result set to be downloaded. The download formats will differ depending upon your installation and report definition but may include CSV, XLS, PDF, or RTF.
APEXIR_HELP_FILTER	Used to add or modify the where clause on the query. You first select a column (it does not need to be one that is displayed), select from a list of standard Oracle operators (=, !=, not in, between), and enter an expression to compare against. The expression is case sensitive and you can use % as a wildcard (for example, STATE_NAME like A%).

Message Name	English Text
APEXIR_HELP_FLASHBACK	Performs a flashback query to allow you to view the data as it existed at a previous point in time. The default amount of time that you can flashback is 3 hours (or 180 minutes) but the actual amount will differ per database.
APEXIR_HELP_FORMAT	Format enable you to customize the display of the report. Format contains the following submenu: - Sort - Control Break - Highlight - Compute - Aggregate - Chart - Group By - Pivot
APEXIR_HELP_GROUP_BY	You can define one Group By view per saved report. Once defined, you can switch between the group by and report views using view icons on the Search bar. To create a Group By view, you select: - the columns on which to group - the columns to aggregate along with the function to be performed (average, sum, count, etc.)
APEXIR_HELP_HIGHLIGHT	Highlighting allows you to define a filter. The rows that meet the filter are highlighted using the characteristics associated with the filter. - Name is used only for display. - Sequence identifies the sequence in which the rules will be evaluated. - Enabled identifies if the rule is enabled or disabled. - Highlight Type identifies whether the Row or Cell should be highlighted. If Cell is selected, the column referenced in the Highlight Condition is highlighted. - Background Color is the new color for the background of the highlighted area. - Text Color is the new color for the text in the highlighted area. - Highlight Condition defines your filter condition.
APEXIR_HELP_PIVOT	You can define one Pivot view per saved report. Once defined, you can switch between the pivot and report views using view icons on the Search bar. To create a Pivot view, you select: - the columns on which to pivot - the columns to display as rows - the columns to aggregate along with the function to be performed (average, sum, count, etc.)

Message Name	English Text
APEXIR_HELP_REPORT_SETTINGS	If you have customized your Interactive Report, the report settings will be displayed below the Search Bar and above the report. If you have saved customized reports, they will be shown as tabs. You can access your alternate views by clicking the tabs. Below the tabs are the report settings for the current report. This area can be collapsed and expanded using the icon on the left. <p/> For each report setting, you can: - Edit by clicking the name. - Disable/Enable by unchecking or checking the Enable/Disable checkbox. This is used to temporarily turn off and on the setting. - Remove by click the Remove icon. This permanently removes the setting. <p/> If you have created a chart, you can toggle between the report and chart using the Report View and Chart View links shown on the right. If you are viewing the chart, you can also use the Edit Chart link to edit the chart settings.
APEXIR_HELP_RESET	Resets the report back to the default settings, removing any customizations that you have made.
APEXIR_HELP_ROWS_PER_PAGE	Sets the number of records to display per page.
APEXIR_HELP_SAVE_REPORT	Saves the customized report for future use. You provide a name and optional description and can make the report accessible to the public (that is, all users who can access the primary default report). You can save four types of interactive reports: • **Primary Default (Developer Only)**. The Primary Default is the report that initially displays. Primary Default reports cannot be renamed or deleted. • **Alternative Report (Developer Only)**. Enables developers to create multiple report layouts. Only developers can save, rename, or delete an Alternative Report. • **Public Report (End user)**. Can be saved, renamed, or deleted by the end user who created it. Other users can view and save the layout as another report. • **Private Report (End user)**. Only the end user that created the report can view, save, rename or delete the report. If you save customized reports, a Reports selector displays in the Search bar to the left of the Rows selector (if this feature is enabled).

Message Name	English Text
APEXIR_HELP_SEARCH_BAR	At the top of each report page is a search region. The region provides the following features: <p/> - Select columns icon allows you to identify which column to search (or all). - Text area allows for case insensitive search criteria (no need for wild cards). - Rows selects the number of records to display per page. [Go] button executes the search. - Actions Menu icon displays the actions menu (discussed next). <p/> Please note that all features may not be available for each report.
APEXIR_HELP_SEARCH_BAR_ACTIONS_MENU	Actions Menu enables you to customize a report. See the sections that follow.
APEXIR_HELP_SEARCH_BAR_FINDER	Select columns icon enables you to identify which column to search (or all).
APEXIR_HELP_SEARCH_BAR_REPORTS	Reports displays alternate default and saved private or public reports.
APEXIR_HELP_SEARCH_BAR_ROWS	Rows sets the number of records to display per page.
APEXIR_HELP_SEARCH_BAR_TEXTBOX	- Text area enables you to enter case insensitive search criteria (wild card characters are implied). - Go button executes the search. Hitting the enter key will also execute the search when the cursor is in the search text area.
APEXIR_HELP_SEARCH_BAR_VIEW	View Icons switches between the icon, report, detail, chart, group by, and pivot views of the report if they are defined.
APEXIR_HELP_SELECT_COLUMNS	Used to modify the columns displayed. The columns on the right are displayed. The columns on the left are hidden. You can reorder the displayed columns using the arrows on the far right. Computed columns are prefixed with **.
APEXIR_HELP_SORT	Used to change the column(s) to sort on and whether to sort ascending or descending. You can also specify how to handle nulls (use the default setting, always display them last or always display them first). The resulting sorting is displayed to the right of column headings in the report.
APEXIR_HELP_SUBSCRIPTION	When you add a subscription, you provide an email address (or multiple email addresses, separated by commas), email subject, frequency, and start and end dates. The resulting emails include an HTML version of the interactive report containing the current data using the report setting that were present when the subscription was added.
APEXIR_HIDE_COLUMN	Hide Column
APEXIR_HIGHLIGHT	Highlight
APEXIR_HIGHLIGHT_CONDITION	Highlight Condition
APEXIR_HIGHLIGHT_TYPE	Highlight Type

Message Name	English Text
APEXIR_HIGHLIGHT_WHEN	Highlight When
APEXIR_HIGHLIGHTS	Highlights
APEXIR_INACTIVE_SETTING	1 inactive setting
APEXIR_INACTIVE_SETTINGS	%0 inactive settings
APEXIR_INTERACTIVE_REPORT_HELP	Interactive Report Help
APEXIR_INVALID	Invalid
APEXIR_INVALID_COMPUTATION	Invalid computation expression. %0
APEXIR_INVALID_END_DATE	The end date must be greater than the start date.
APEXIR_INVALID_END_DATE	The end date must be greater than the start date.
APEXIR_INVALID_FILTER	Invalid filter expression. %0
APEXIR_INVALID_FILTER_QUERY	Invalid filter query
APEXIR_INVALID_SETTING	1 invalid setting
APEXIR_INVALID_SETTINGS	%0 invalid settings
APEXIR_IS_IN_THE_LAST	%0 is in the last %1
APEXIR_IS_IN_THE_NEXT	%0 is in the next %1
APEXIR_IS_NOT_IN_THE_LAST	%0 is not in the last %1
APEXIR_IS_NOT_IN_THE_NEXT	%0 is not in the next %1
APEXIR_KEYPAD	Keypad
APEXIR_LABEL	Label
APEXIR_LABEL_AXIS_TITLE	Axis Title for Label
APEXIR_LABEL_PREFIX	Label Prefix
APEXIR_LAST_DAY	Last Day
APEXIR_LAST_HOUR	Last Hour
APEXIR_LAST_MONTH	Last Month
APEXIR_LAST_WEEK	Last Week
APEXIR_LAST_X_DAYS	Last %0 Days
APEXIR_LAST_X_HOURS	Last %0 Hours
APEXIR_LAST_X_YEARS	Last %0 Years
APEXIR_LAST_YEAR	Last Year
APEXIR_LINE	Line
APEXIR_MAX_QUERY_COST	The query is estimated to exceed the maximum allowed resources. Please modify your report settings and try again.
APEXIR_MAX_ROW_CNT	This query returns more then %0 rows, please filter your data to ensure complete results.
APEXIR_MAX_X	Maximum %0
APEXIR_MEDIAN_X	Median %0
APEXIR_MIN_AGO	%0 minutes ago
APEXIR_MIN_X	Minimum %0
APEXIR_MONTH	Month

Message Name	English Text
APEXIR_MONTHLY	Monthly
APEXIR_MORE_DATA	More Data
APEXIR_MOVE	Move
APEXIR_MOVE_ALL	Move All
APEXIR_MULTIIR_PAGE_REGION_STATIC_ID_REQUIRED	Region Static ID must be specified as the page contains multiple interactive reports.
APEXIR_NAME	Name
APEXIR_NEW_AGGREGATION	New Aggregation
APEXIR_NEW_CATEGORY	- New Category -
APEXIR_NEW_CATEGORY_LABEL	New Category
APEXIR_NEW_COMPUTATION	New Computation
APEXIR_NEXT	>
APEXIR_NEXT_DAY	Next Day
APEXIR_NEXT_HOUR	Next Hour
APEXIR_NEXT_MONTH	Next Month
APEXIR_NEXT_WEEK	Next Week
APEXIR_NEXT_X_DAYS	Next %0 Days
APEXIR_NEXT_X_HOURS	Next %0 Hours
APEXIR_NEXT_X_YEARS	Next %0 Years
APEXIR_NEXT_YEAR	Next Year
APEXIR_NO	No
APEXIR_NO_AGGREGATION_DEFINED	No aggregation defined.
APEXIR_NO_COLUMN_INFO	No column information available.
APEXIR_NO_COMPUTATION_DEFINED	No computation defined.
APEXIR_NO_END_DATE	- No End Date -
APEXIR_NONE	- None -
APEXIR_NOT_VALID_EMAIL	Not a valid email address.
APEXIR_NULL_SORTING	Null Sorting
APEXIR_NULLS_ALWAYS_FIRST	Nulls Always First
APEXIR_NULLS_ALWAYS_LAST	Nulls Always Last
APEXIR_NUMERIC_FLASHBACK_TIME	Flashback time must be numeric.
APEXIR_NUMERIC_SEQUENCE	Sequence must be numeric.
APEXIR_OPERATOR	Operator
APEXIR_ORANGE	orange
APEXIR_OTHER	Other
APEXIR_PDF_ORIENTATION_VERTICAL	Portrait
APEXIR_PDF_PAGE_SIZE_LEGAL	Legal
APEXIR_PDF_ORIENTATION	Page orientation
APEXIR_PDF_PAGE_SIZE	Page size

Message Name	English Text
APEXIR_PDF_PAGE_SIZE_LETTER	Letter
APEXIR_PDF_ORIENTATION_HORIZONTAL	Landscape
APEXIR_PERCENT_OF_TOTAL_COUNT_X	Percent of Total Count %0 (%)
APEXIR_PERCENT_OF_TOTAL_SUM_X	Percent of Total Sum %0 (%)
APEXIR_PERCENT_TOTAL_COUNT	Percent of Total Count
APEXIR_PERCENT_TOTAL_SUM	Percent of Total Sum
APEXIR_PIE	Pie
APEXIR_PIVOT	Add Pivot Column
APEXIR_PIVOT_AGG_NOT_NULL	Aggregate must be specified.
APEXIR_PIVOT_AGG_NOT_ON_ROW_COL	You cannot aggregate on a column selected to as row column.
APEXIR_PIVOT_COLUMN_N	Pivot Column %0
APEXIR_PIVOT_COLUMN_NOT_NULL	Pivot column must be specified.
APEXIR_PIVOT_COLUMNS	Pivot Columns
APEXIR_PIVOT_MAX_ROW_CNT	The maximum row count for a Pivot query limits the number of rows in the base query, not the number of rows displayed. Your base query exceeds the maximum row count of %0. Please apply a filter to reduce the number of records in your base query.
APEXIR_PIVOT_ROW_COLUMN_INVALID	Select different row column. The HTML expression or link in the row column contains column defined as pivot or aggregate column.
APEXIR_PIVOT_SORT	Pivot Sort
APEXIR_PREVIOUS	<
APEXIR_PRIMARY	Primary
APEXIR_PRIMARY_REPORT	Primary Report
APEXIR_PRIVATE	Private
APEXIR_PUBLIC	Public
APEXIR_RED	red
APEXIR_REGION_STATIC_ID_DOES_NOT_EXIST	Region Static ID %0 does not exist.
APEXIR_REMOVE	Remove
APEXIR_REMOVE_AGGREGATE	Remove Aggregate
APEXIR_REMOVE_ALL	Remove All
APEXIR_REMOVE_CHART	Remove Chart
APEXIR_REMOVE_CONTROL_BREAK	Remove Control Break
APEXIR_REMOVE_FILTER	Remove Filter
APEXIR_REMOVE_FLASHBACK	Remove Flashback
APEXIR_REMOVE_GROUP_BY	Remove Group By
APEXIR_REMOVE_HIGHLIGHT	Remove Highlight
APEXIR_REMOVE_PIVOT	Remove Pivot
APEXIR_REMOVE_REPORT	Remove Report

Message Name	English Text
APEXIR_RENAME_DEFAULT_REPORT	Rename Default Report
APEXIR_RENAME_REPORT	Rename Report
APEXIR_REPORT	Report
APEXIR_REPORT_ALIAS_DOES_NOT_EXIST	Saved Interactive report with alias %0 does not exist.
APEXIR_REPORT_DISPLAY_COLUMN_LIMIT_REACHED	The number of display columns in the report reached the limit. Please click Select Columns under Actions menu to minimize the report display column list.
APEXIR_REPORT_DOES_NOT_EXIST	Report does not exist.
APEXIR_REPORT_ID_DOES_NOT_EXIST	Saved Interactive Report ID %0 does not exist.
APEXIR_REPORT_SETTINGS	Report Settings
APEXIR_REPORT_VIEW	< Report View
APEXIR_REPORTS	Reports
APEXIR_RESET	Reset
APEXIR_RESET_CONFIRM	Restore report to the default settings.
APEXIR_ROW	Row
APEXIR_ROW_COL_DIFF_FROM_PIVOT_COL	Row column must be different from the pivot column.
APEXIR_ROW_COLUMN_N	Row Column %0
APEXIR_ROW_COLUMN_NOT_NULL	Row column must be specified.
APEXIR_ROW_COLUMNS	Row Columns
APEXIR_ROW_FILTER	Row Filter
APEXIR_ROW_OF	Row %0 of %1
APEXIR_ROW_ORDER	Row Order
APEXIR_ROW_TEXT_CONTAINS	Row text contains
APEXIR_ROWS	Rows
APEXIR_ROWS_PER_PAGE	Rows Per Page
APEXIR_RPT_DISP_COL_EXCEED	The number of display columns in the report reached the limit. Please click Select Columns under Actions menu to minimize the report display column list.
APEXIR_SAVE	Save
APEXIR_SAVE_AS_DEFAULT	Save as Default
APEXIR_SAVE_DEFAULT_CONFIRM	The current report settings are used as the default for all users.
APEXIR_SAVE_DEFAULT_REPORT	Save Default Report
APEXIR_SAVE_REPORT	Save Report
APEXIR_SAVE_REPORT_DEFAULT	Save Report *
APEXIR_SAVED_REPORT	Saved Report
APEXIR_SAVED_REPORT_MSG	Saved Report = "%0"
APEXIR_SEARCH	Search
APEXIR_SEARCH_BAR	Search Bar
APEXIR_SEARCH_COLUMN	Search: %0

Message Name	English Text
APEXIR_SEARCH_REPORT	Search Report
APEXIR_SELECT_CATEGORY	- Select Category -
APEXIR_SELECT_COLUMN	- Select Column -
APEXIR_SELECT_COLUMNS	Select Columns
APEXIR_SELECT_COLUMNS_FOOTER	Computed columns are prefixed with **.
APEXIR_SELECT_FUNCTION	- Select Function -
APEXIR_SELECT_GROUP_BY_COLUMN	- Select Group By Column -
APEXIR_SELECT_PIVOT_COLUMN	- Select Pivot Column -
APEXIR_SELECT_ROW	Select Row
APEXIR_SELECT_ROW_COLUM	- Select Row Column -
APEXIR_SELECT_SORT_COLUMN	- Select Sort Column -
APEXIR_SELECT_VALUE	Select Value
APEXIR_SELECTED_COLUMNS	Selected Columns
APEXIR_SEND	Send
APEXIR_SEQUENCE	Sequence
APEXIR_SORT	Sort
APEXIR_SORT_ASCENDING	Sort Ascending
APEXIR_SORT_COLUMN	Sort Column
APEXIR_SORT_DESCENDING	Sort Descending
APEXIR_SORT_ORDER	Sort Order
APEXIR_SPACE_AS_IN_ONE_EMPTY_STRING	space
APEXIR_STATUS	Status
APEXIR_SUBSCRIPTION	Subscription
APEXIR_SUBSCRIPTION_ENDING	Ending
APEXIR_SUBSCRIPTION_STARTING_FROM	Starting From
APEXIR_SUM_X	Sum %0
APEXIR_TABLE_SUMMARY	%0, Report = %1, View = %2
APEXIR_TEXT_COLOR	Text Color
APEXIR_TIME_DAYS	days
APEXIR_TIME_HOURS	hours
APEXIR_TIME_MINS	minutes
APEXIR_TIME_MONTHS	months
APEXIR_TIME_WEEKS	weeks
APEXIR_TIME_YEARS	years
APEXIR_TOGGLE	Toggle
APEXIR_TOP	Top
APEXIR_UNAUTHORIZED	Unauthorized
APEXIR_UNGROUPED_COLUMN	Ungrouped Column

Message Name	English Text
APEXIR_UNIQUE_COLUMN_HEADING	Column Heading must be unique.
APEXIR_UNIQUE_HIGHLIGHT_NAME	Highlight Name must be unique.
APEXIR_UNSUPPORTED_DATA_TYPE	unsupported data type
APEXIR_UP	Up
APEXIR_VALID_COLOR	Please enter a valid color.
APEXIR_VALID_FORMAT_MASK	Please enter a valid format mask.
APEXIR_VALUE	Value
APEXIR_VALUE_AXIS_TITLE	Axis Title for Value
APEXIR_VALUE_REQUIRED	Value Required
APEXIR_VCOLUMN	Vertical Column
APEXIR_VIEW_CHART	View Chart
APEXIR_VIEW_DETAIL	View Detail
APEXIR_VIEW_GROUP_BY	View Group By
APEXIR_VIEW_ICONS	View Icons
APEXIR_VIEW_PIVOT	View Pivot
APEXIR_VIEW_REPORT	View Report
APEXIR_WEEK	Week
APEXIR_WEEKLY	Weekly
APEXIR_WORKING_REPORT	Working Report
APEXIR_X_DAYS	%0 days
APEXIR_X_HOURS	%0 hours
APEXIR_X_MINS	%0 minutes
APEXIR_X_MONTHS	%0 months
APEXIR_X_WEEKS	%0 weeks
APEXIR_X_YEARS	%0 years
APEXIR_YEAR	Year
APEXIR_YELLOW	yellow
APEXIR_YES	Yes
IR_AS_DEFAULT_REPORT_SETTING	As Default Report Settings
IR_AS_NAMED_REPORT	As Named Report
IR_STAR	Only displayed for developers
LAYOUT.T_CONDITION_EXPR2	Expression 2
OUT_OF_RANGE	Invalid set of rows requested, the source data of the report has been modified.
REPORT	Report
REPORTING_PERIOD	Reporting Period
RESET	Reset Pagination
SAVED_REPORTS.PRIMARY.DEFAULT	Primary Default
WWV_RENDER_REPORT3.X_Y_OF_Z_2	%0 - %1 of %2

22.7.3.4 Progressive Web App (PWA) Messages Requiring Translation

Lists Progressive Web Application (PWA) messages that require translation.

Message Name	English Text
APEX.PWA.DIALOG.TITLE	Add to Home Screen

APEX.PWA.INSTRUCTIONS	\
	  Your device or browser does not appear to support the installation of Progressive Web Apps right now. \
	  If this application is already installed on your device, open it from your home screen. \

APEX.PWA.INSTRUCTIONS.ANDROID.FIREFOX	\
	  This website has app functionality. Add it to your home screen to use it in fullscreen. \
	 \
	 \
	  \
	  \
	  1. Tap the Menu icon in the bottom navigation \
	  \
	  \
	  \
	  2. Tap Install \
	  \
	  \
	  \
	  3. Tap Add to confirm \
	  \

Message Name	English Text
APEX.PWA.INSTRUCTIONS.ANDROID.WEBVIEW	\ \ This website has app functionality that is best experienced in Chrome or Edge. \ \ \ \\ \ 1. Tap the Menu icon next to the address bar \ \ \\ \ 2. Tap Install
APEX.PWA.INSTRUCTIONS.IOS.GSA	\ \ This website has app functionality that is best experienced in Safari. \ \ \ \\ \ 1. Tap the Share icon in the bottom navigation \ \ \\ \ 2. Tap Open in Safari

Message Name	English Text

APEX.PWA.INSTRUCTIONS.IOS.SAFARI.IPAD	\
	  This website has app functionality. Add it to your home screen for the best experience. \
	 \
	 \
	  \
	  \
	  1. Tap the Share icon next to the address bar \
	  \
	  \
	  \
	  2. Tap Add to Home Screen \
	  \
	  \
	  \
	  3. Tap Add to confirm \
	  \

Message Name	English Text

APEX.PWA.INSTRUCTIONS.IOS.SAFARI.IPHONE	\
	  This website has app functionality. Add it to your home screen for the best experience. \
	 \
	 \
	  \
	  \
	  1. Tap the Share icon in the bottom navigation \
	  \
	  \
	  \
	  2. Scroll down and tap Add to Home Screen \
	  \
	  \
	  \
	  3. Tap Add to confirm \
	  \

Message Name	English Text

APEX.PWA.INSTRUCTIONS.IOS.SAFARI.IPHONE.15	\
	  This website has app functionality. Add it to your home screen for the best experience. \
	 \
	 \
	  \
	  \
	  1. Tap the More icon next to the address bar \
	  \
	  \
	  \
	  2. Tap Add to Home Screen \
	  \
	  \
	  \
	  3. Tap Add to confirm \
	  \

Message Name	English Text

APEX.PWA.INSTRUCTIONS.IOS.WEBVIEW.IPAD	\
	  This website has app functionality. Add it to your home screen for the best experience.
	
	 \
	  \
	  \
	  Tap the Share icon next to the address bar
	
	  \
	  \
	  Tap Open in Safari

APEX.PWA.INSTRUCTIONS.IOS.WEBVIEW.IPHONE	\
	  This website has app functionality. Add it to your home screen for the best experience.
	
	 \
	  \
	  \
	  Tap the Share icon in the bottom navigation
	
	  \
	  \
	  Tap Open in Safari

Message Name	English Text
APEX.PWA.INSTRUCTIONS.LINUX.FIREFOX	\ \ This website has app functionality that is best experienced in Chrome.\ \ \ Firefox does not currently support web apps installation.\ \ \
APEX.PWA.INSTRUCTIONS.MAC_OS.FIREFOX	\ \ This website has app functionality that is best experienced in Chrome or Edge.\ \ \ Firefox does not currently support web apps installation on macOS.\ \ \
APEX.PWA.INSTRUCTIONS.MAC_OS.SAFARI	\ \ This website has app functionality that is best experienced in Chrome or Edge.\ \ \ Safari does not currently support web apps installation on macOS.\ \ \
APEX.PWA.INSTRUCTIONS.WINDOWS.FIREFOX	\ \ This website has app functionality that is best experienced in Chrome or Edge.\ \ \ Firefox does not currently support web apps installation on Windows.\ \ \

Message Name	English Text
APEX.PWA.OFFLINE.BODY	<main> <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 400 280" role="presentation"> <g fill="none"> <path d="M316.846 213.183c39.532 0 63.154-30.455 63.154-62.764 0-30.943-22.158-56.615-51.441-62.179v-1.17c0-48.123-38.947-87.07-87.07-87.07-87.07-39.044 0-72.036 25.672-83.066 61.007-8.492-3.612-17.863-5.564-27.722-5.564-34.261 0-62.764 24.11-69.694 56.322a51.007 51.007 0 0 0-9.468-.879C23.036 110.79 0 133.825 0 162.327c0 28.405 23.036 51.441 51.441 51.441l265.405-.585z" fill="currentColor" opacity=".2"/> <circle fill="#D63B25" cx="336" cy="216" r="64"/> <path d="M367.357 198.439c-.395-.395-.947-.632-1.657-.632-.710-1.184.237-1.657.632L351.97 210.51l-10.494-10.493 12.072-12.072c.395-.395.71-.947.71-1.657A2.29 2.29 0 0 0 351.97 184c-.631 0-1.183.237-1.657.631l-12.071 12.072-7.496-7.496c-.394-.394-.947-.71-1.657-.71a2.29 2.29 0 0 0-2.288 2.288c0 .632.237 1.184.71 1.657l2.604 2.604-13.176 13.176a13.781 13.781 0 0 0-4.024 9.705c0 3.787 1.499 7.18 4.024 9.705l2.13 2.13-14.36 14.36c-.394.394-.71.947-.71 1.657a2.29 2.29 0 0 0 2.288 2.288c.631 0 1.184-.237 1.657-.71l14.36-14.36 1.736 1.736a13.781 13.781 0 0 0 9.704 4.024c3.787 0 7.18-1.5 9.705-4.024l13.176-13.177 2.92 2.92c.394.394.946.71 1.656.71a2.29 2.29 0 0 0 2.289-2.288c0-.632-.237-1.184-.71-1.657l-7.575-7.496 12.072-12.071c.394-.395.71-.947.710-1.657.079-.632-.237-1.184-.631-1.578zm-27.142 33.059a9.398 9.398 0 0 1-6.47 2.603c-2.525 0-4.813-.946-6.47-2.603l-7.1-7.101a9.124 9.124 0 0 1-2.683-6.47 9.124 9.124 0 0 1 2.682-6.47l13.177-13.176 3.156 3.156c.079.079.079.158.158.158l.157.157 13.413 13.413c.08.08.08.158.158.158l.158.158 2.761 2.762-13.097 13.255z" fill="#FFF"/> </g> </svg> <h1>Can't connect</h1> <p>You need an internet connection to

Message Name	English Text

	use this app.</p>
	  <button type="button">Retry</button>
	</main>
	
	<script>
	
	document.querySelector("button").addEventListener("click", () => {
	  window.location.reload();
	  });
	</script>

APEX.PWA.OFFLINE.TITLE	Can't connect

22.7.3.5 Search and Smart Filters Messages Requiring Translation

Lists faceted search and smart filter messages that require translation.

Message Name	English Text
APEX.FS.APPLIED_FACET	Applied Filter %0
APEX.FS.BATCH_APPLY	Apply
APEX.FS.CHART_BAR	Bar Chart
APEX.FS.CHART_PIE	Pie Chart
APEX.FS.CHART_OTHERS	All Others
APEX.FS.CHART_TITLE	Chart
APEX.FS.CHART_VALUE_LABEL	Count
APEX.FS.CLEAR	Clear
APEX.FS.CLEAR_ALL	Clear All
APEX.FS.CLEAR_VALUE	Clear %0
APEX.FS.CONFIG_TITLE	Choose filters to show
APEX.FS.COUNT_RESULTS	%0 results
APEX.FS.COUNT_SELECTED	%0 selected
APEX.FS.CUR_FILTERS_LM	Current filters
APEX.FS.FACETS_LIST	Filters list
APEX.FS.FILTER	Filter %0
APEX.FS.GO	Go
APEX.FS.INPUT_CURRENT_LABEL	equals %0
APEX.FS.OPEN_CONFIG	More Filters
APEX.FS.RANGE_BEGIN	Range Begin
APEX.FS.RANGE_END	Range End

Message Name	English Text
APEX.FS.RANGE_CURRENT_LABEL	%0 to %1
APEX.FS.RANGE_CURRENT_LABEL_OPEN_LO	Below %0
APEX.FS.RANGE_CURRENT_LABEL_OPEN_HI	Above %0
APEX.FS.RANGE_TEXT	to
APEX.FS.REMOVE_CHART	Remove Chart
APEX.FS.RESET	Reset
APEX.FS.SEARCH_LABEL	Search
APEX.FS.SHOW_CHART	Show Chart
APEX.FS.SHOW_LESS	Show Less
APEX.FS.SHOW_MORE	Show More
APEX.FS.SEARCH_PLACEHOLDER	Search...
APEX.FS.SELECT_PLACEHOLDER	- Select -
APEX.FS.STAR_RATING_LABEL	%0 stars and up
APEX.FS.SUGGESTIONS	Filter suggestions
APEX.FS.TOTAL_ROW_COUNT_LABEL	Total Row Count

22.7.4 Managing Text Messages

Create, edit, copy, or subscribe to text messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.

- Workflow for Translating Messages into Multiple Languages
  Learn about the general workflow for translating text messages into multiple languages.
- Creating Text Messages
  Create text messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.
- Editing or Deleting Text Messages
  Edit or delete translation messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.
- Selecting Translation Messages in Page Designer
  Learn about searching for and selecting translation text messages in Page Designer.
- Copying or Subscribing to Text Messages
  Copy text messages from the current application or from another application. When copying a text message from another app, you can also subscribe to it.
- Subscribing to Text Messages
  Subscribe to text messages on the Create/Edit Text Message page.

22.7.4.1 Workflow for Translating Messages into Multiple Languages

Learn about the general workflow for translating text messages into multiple languages.

The general workflow when translating text messages into multiple languages:

1. Select the application and navigate to **Shared Components**, **Globalization**, **Text Messages**.

2. Create the first text message for the application's primary language.
3. Copy the text message for the application's primary language, entering the same message name for **New Text Message Name** and specifying an alternate **Language**.
4. Edit the new copied text message and enter the new translation in the **Text** attribute. Click **Apply Changes** to save your changes.

See the topics that follow for more details on each step.

22.7.4.2 Creating Text Messages

Create text messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.

To create a text message:

1. Navigate to the Text Messages page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Text Messages**.
2. On the Translate Messages page, click **Create Text Message**.
3. On Create/Edit Text Message:
   a. Name - Enter a unique name for the message you are creating in the base language (for example, `MY_MESSAGE`).
   b. Language - Select the language for which the message will be used.
   c. Used in JavaScript - Enable this flag if the message is used in `APEX.LANG.*` API calls in JavaScript code.
   d. Text - Enter the text to be returned when the text message is called.

      Text Messages support named substitution values. Named substitution values must be unique within the message. Named substitution values can contain any characters except whitespace and "=". For example:

      ```
      ACCOUNT_MSG Hello %name, your balance is %balance.
      ```

      Text messages can contain either named or positional substitution values (for example, `%name`, `%amount`, or `%0,%1`), ensure that your defined message also contains the same named and number of positional substitution values.

      For example, you could define the message `GREETING_MSG` in English as:

      ```
      Good morning %name
      ```

      Or, you could define the message `GREETING_MSG` in German as:

      ```
      Guten Tag %name
      ```

   e. Comment - Enter any developer comments or notes here. These comments only display within App Builder and never display when running the application.
4. Click **Create And Create Another** or **Create Text Message**.

22.7.4.3 Editing or Deleting Text Messages

Edit or delete translation messages by selecting the application and navigating to Shared Components, Globalization, Text Messages.

To edit or delete text messages:

1. Navigate to the Text Messages page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Text Messages**.

      The Text Messages page appears.
2. To edit text messages:
   a. On the Text Messages page, click the message Name.

      Create/Edit Text Message appears.
   b. Name - Enter a unique name that expresses the purpose of the message.

      For example, a good name for an error message text for a date validation could be `NO_WEEKEND_DATES_ALLOWED`.
   c. Language - Select the language for which the message will be used.
   d. Used in JavaScript - Enable this flag if the message is used in `APEX.LANG.*` API calls in JavaScript code.
   e. Text - Enter the text to be returned when the text message is called.

      If the English text message contains positional substitution values (for example, `%0`, `%1`), ensure that your defined message also contains the same named and number of positional substitution values.

      For example, you could define the message `GREETING_MSG` in English as:

      ```
      Good morning %0
      ```

      Or, you could define the message `GREETING_MSG` in German as:

      ```
      Guten Tag %0
      ```
   f. Comment - Enter any developer comments or notes here. These comments only display within App Builder and never display when running the application.
   g. Click **Apply Changes**.
3. To delete a text message:
   a. On the Text Messages page, click the message Name.

      Create/Edit Text Message appears.
   b. Click **Delete**.
4. To access a Grid Edit page:
   a. Under Tasks, click **Grid Edit**.

      The Grid Edit page appears.
   b. Make the appropriate edits directly in the report.

For **Text**, if the English text message contains positional substitution values (for example, `%0`, `%1`), ensure that your defined message also contains the same named and number of positional substitution values.

c. Click **Save**.

> **Tip:**
>
> To add a new message from the Grid Edit page, click **Add Row**.

22.7.4.4 Selecting Translation Messages in Page Designer

Learn about searching for and selecting translation text messages in Page Designer.

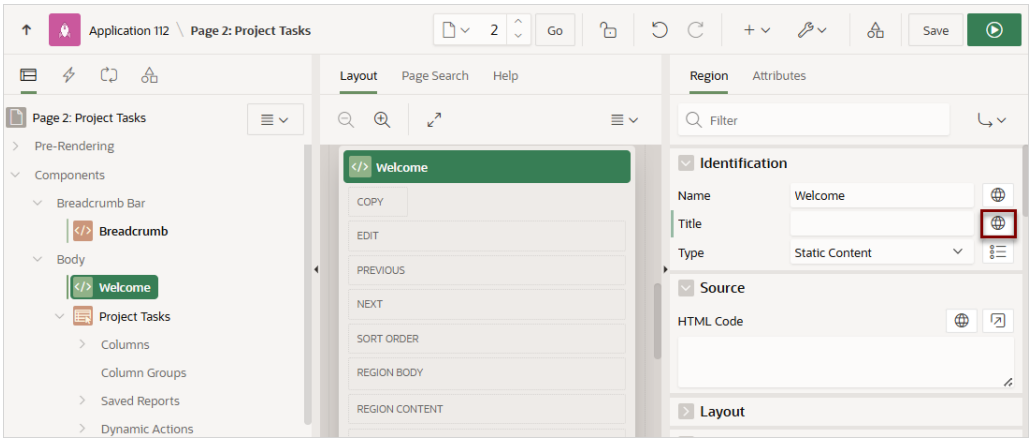
Select translation text messages directly in Page Designer by clicking the Text Messages icon or by triggering the Suggestion Box HTML Code Editors.

To search for or select translation text message in Page Designer:

1. Create translation text messages in Shared Components. See Creating Text Messages.
2. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
3. Enable the Text Message Picker:
   a. On the Page Designer toolbar, click **Utilities**.
   b. Click **Show** and then **Text Message Picker**.

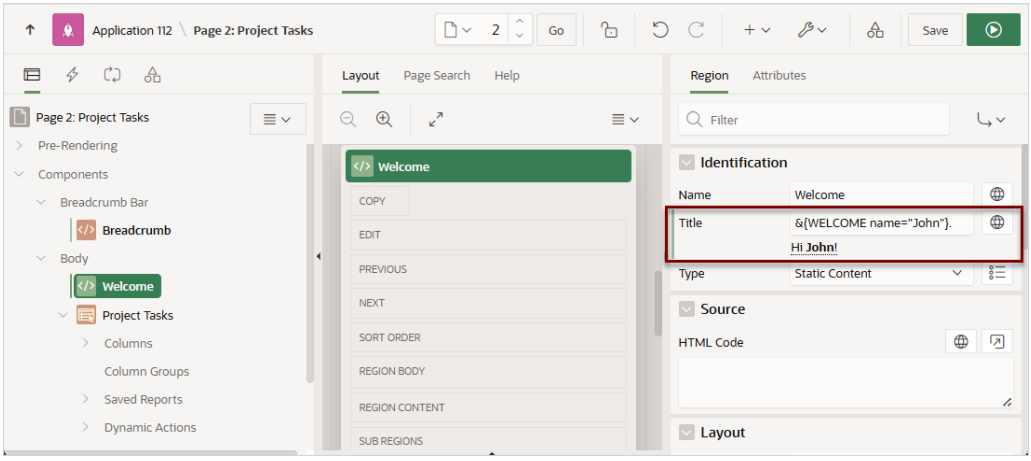
   When Text Message Picker is enabled, the **Text Messages** icon displays next to all text fields as shown in the following example.



4. To search for or select text messages in Page Designer:
   a. Click the **Text Message** icon ( ).
   b. To search for a text message, enter keywords in the Search field and click **Go**.

c. Select the text message and edit the attribute.

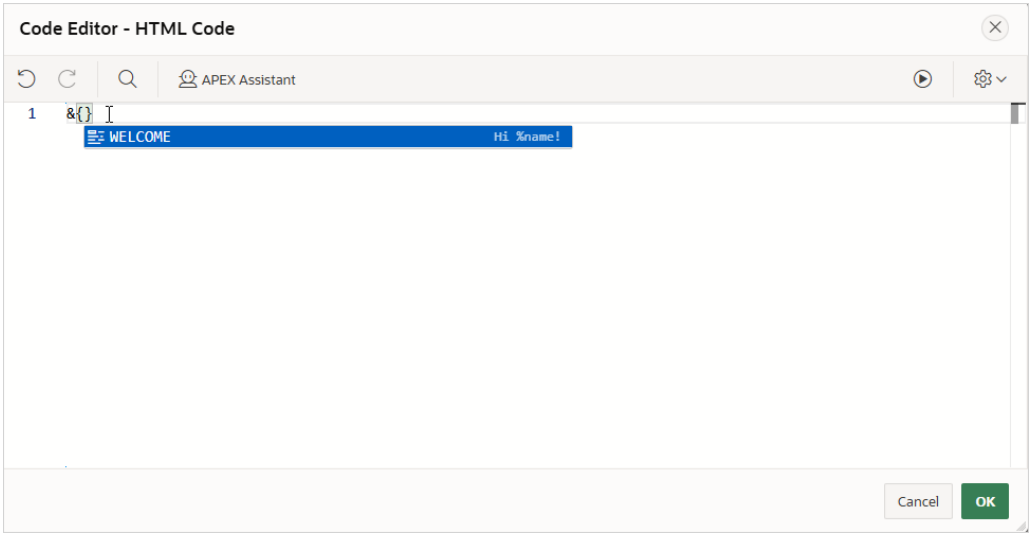
When a text message is used in an attribute, a preview of the content displays below it as shown in the following example:



5. To search for and select text messages in a Code Editor:

a. Expand the Code Editor by clicking the **Code Editor** icon ( ).

b. In the Code Editor enter `&{` or `&APP_TEXT$` and a suggestion box displays.



> **Note:**
>
> `&{` only works for applications with a Compatibiltiy Mode of 24.2 or higher. See Editing the Application Definition.

22.7.4.5 Copying or Subscribing to Text Messages

Copy text messages from the current application or from another application. When copying a text message from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to a text message:

1. Navigate to the Text Messages page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Text Messages**.

      The Text Messages page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. To copy a text message within the current application:
   a. Search for the text message and click **Copy** in the Copy column.

      The Copy Text Message Wizard appears.
   b. Language - Select a language for which this message will be used.
   c. New Text Message Name - Enter the name of the text message. For a selected language, the text message name must be unique within the application.

      As a best practice, Oracle recommends using the existing message name but include a language identifier (for example, `MY_MESSAGE_FR` for English and `MY_MESSAGE_FR` for the French.)
   d. Click **Copy Text Message**.
3. To copy a text message from another application:
   a. From the Tasks region, click **Copy from another app**.

      The Copy Text Message Wizard appears.
   b. Application - Select the application from which you want to copy.
   c. Copy Text Message - Select the text message to copy.
   d. Subscribe - Set this flag to **On** to subscribe to the text message.
   e. Click **Copy Text Message**.

22.7.4.6 Subscribing to Text Messages

Subscribe to text messages on the Create/Edit Text Message page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a text messages by copying it and enabling the Subscribe option. See Copying or Subscribing to Text Messages.

To subscribe to a text message in the current application:

1. Navigate to the Text Messages page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Text Messages**.

      The Text Messages page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. Select a text message.

   The Create/Edit Text Message page appears.
3. To subscribe to a text message:
   a. Find the **Subscription** region.
   b. Subscription, Subscribe From - Select a master text message to subscribe to and click **Apply Changes**.
   c. Confirm your selection and click **Subscribe**.

      The subscribed component is automatically refreshed from the selected master.
4. To refresh a subscription:
   a. Find the **Subscription** region.
   b. Subscription, Refresh Text Message - If the text message is already subscribed, click **Refresh Text Message**.
   c. Click **Apply Changes**.
5. To unsubscribe from an email template:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

22.8 Translating Data That Supports List of Values

You create a dynamic translation to translate dynamic pieces of data. For example, you might use a dynamic translation on a list of values based on a database query.

Dynamic translations differ from messages in that you query a specific string rather than a message name. You define dynamic translations on the Dynamic Translations page. You then use the `APEX_LANG.LANG` API to return the dynamic translation string identified by the `p_primary_text_string` parameter.

- Defining a Dynamic Translation
  You define dynamic translations on the Dynamic Translations page. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string.

> **See Also:**
>
> APEX_LANG in *Oracle APEX API Reference*

22.8.1 Defining a Dynamic Translation

You define dynamic translations on the Dynamic Translations page. A dynamic translation consists of a translate-from language string, a language code, and a translate-to string.

To define a dynamic translation:

1. Navigate to the Translate Application page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. Under Globalization, click **Application Translations**.

      The Translate page appears.
2. Under Translation Utilities, select **Dynamic Translations**.
3. On the Dynamic Translations page, click **Create** and specify the following:
   a. Language - Select a target language.
   b. Translate From Text - Enter the source text to be translated.
   c. Translate To Text - Enter the translated text.
4. Click **Create**.

22.9 Understanding Supported Globalization Codes

If you are building a multilingual application, it is important to understand how globalization codes affect the way in which your application runs. These codes are set automatically based on the application-level Globalization attributes you select.

- Oracle APEX Globalization Codes
  `NLS_LANGUAGE` and `NLS_TERRITORY` determine the default presentation of numbers, dates, and currencies.

> **See Also:**
>
> Specifying the Primary Language for an Application

22.9.1 Oracle APEX Globalization Codes

`NLS_LANGUAGE` and `NLS_TERRITORY` determine the default presentation of numbers, dates, and currencies.

The following table describes the globalization codes in Oracle APEX.

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
Afrikaans	af	ENGLISH	SOUTH AFRICA
Arabic	ar	ARABIC	UNITED ARAB EMIRATES
Arabic (Algeria)	ar-dz	ARABIC	ALGERIA
Arabic (Bahrain)	ar-bh	ARABIC	BAHRAIN
Arabic (Egypt)	ar-eg	EGYPTIAN	EGYPT
Arabic (Iraq)	ar-iq	ARABIC	IRAQ
Arabic (Jordan)	ar-jo	ARABIC	JORDAN
Arabic (Kuwait)	ar-kw	ARABIC	KUWAIT
Arabic (Lebanon	ar-lb	ARABIC	LEBANNON
Arabic (Libya)	ar-ly	ARABIC	LIBYA
Arabic (Morocco)	ar-ma	ARABIC	MOROCCO
Arabic (Oman)	ar-om	ARABIC	OMAN
Arabic (Qatar)	ar-qa	ARABIC	QATAR
Arabic (Saudi Arabia)	ar-sa	ARABIC	SAUDI ARABIA
Arabic (Syria)	ar-sy	ARABIC	SYRIA
Arabic (Tunisia)	ar-tn	ARABIC	TUNISIA
Arabic (U.A.E.)	ar-ae	ARABIC	UNITED ARAB EMIRATES
Arabic (YEMEN)	ar-ye	ARABIC	YEMEN
Assamese	as	ASSAMESE	INDIA
Basque	eu	FRENCH	FRANCE
Belarusian	be	RUSSIAN	RUSSIA
Bengali	bn	BANGLA	BANGLADESH
Bulgarian	bg	BULGARIAN	BULGARIA
Catalan	ca	CATALAN	CATALONIA
Chinese	zh	SIMPLIFIED CHINESE	CHINA
Chinese (China)	zh-cn	SIMPLIFIED CHINESE	CHINA
Chinese (Hong Kong SAR)	zh-hk	TRADITIONAL CHINESE	HONG KONG
Chinese (Macau SAR)	zh-mo	TRADITIONAL CHINESE	HONG KONG
Chinese (Singapore)	zh-sg	SIMPLIFIED CHINESE	SINGAPORE
Chinese (Taiwan)	zh-tw	TRADITIONAL CHINESE	TAIWAN
Croatian	hr	CROATIAN	CROATIA
Czech	cs	CZECH	CZECH REPUBLIC
Danish	da	DANISH	DENMARK
Dutch (Belgium)	nl-be	DUTCH	BELGIUM
Dutch (Netherlands)	nl	DUTCH	THE NETHERLANDS
English	en	AMERICAN	AMERICA
English (Australia)	en-au	ENGLISH	AUSTRALIA
English (Belize)	en-bz	ENGLISH	UNITED KINGDOM

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
English (Canada)	en-ca	ENGLISH	CANADA
English (Ireland)	en-ie	ENGLISH	IRELAND
English (Jamaica)	en-jm	ENGLISH	UNITED KINGDOM
English (New Zealand)	en-nz	ENGLISH	NEW ZEALAND
English (Philippines)	en-ph	ENGLISH	PHILIPPINES
English (South Africa)	en-za	ENGLISH	SOUTH AFRICA
English (Trinidad)	en-tt	ENGLISH	UNITED KINGDOM
English (United Kingdom)	en-gb	ENGLISH	UNITED KINGDOM
English (United States)	en-us	AMERICAN	AMERICA
English (Zimbabwe)	en-zw	ENGLISH	UNITED KINGDOM
Estonian	et	ESTONIAN	ESTONIA
Faeroese	fo	ENGLISH	UNITED KINGDOM
Farsi	fa	ENGLISH	UNITED KINGDOM
Finnish	fi	FINNISH	FINLAND
French (Belgium)	fr-be	FRENCH	BELGIUM
French (Canada)	fr-ca	CANADIAN FRENCH	CANADA
French (France)	fr	FRENCH	FRANCE
French (Luxembourg)	fr-lu	FRENCH	LUXEMBOURG
French (Monaco)	fr-mc	FRENCH	FRANCE
French (Switzerland)	fr-ch	FRANCH	SWITZERLAND
FYRO Macedonian	mk	MACEDONIAN	FYR MACEDONIA
Gaelic	gd	ENGLISH	UNITED KINGDOM
Galician	gl	SPANISH	SPAIN
German (Austria)	de-at	GERMAN	AUSTRIA
German (Germany)	de	GERMAN	GERMANY
German (Liechtenstein)	de-li	GERMAN	GERMANY
German (Luxemgourg)	de-lu	GERMAN	LUXEMBOURG
German (Switzerland)	de-ch	GERMAN	SWITZERLAND
Greek	el	GREEK	GREECE
Gujarati	gu	GUJARATI	INDIA
Hebrew	he	HEBREW	ISRAEL
Hindi	hi	HINDI	INDIA
Hungarian	hu	HUNGARIAN	HUNGARY
Icelandic	is	ICELANDIC	ICELAND
Indonesian	id	INDONESIAN	INDONESIA
Italian (Italy)	it	ITALIAN	ITALY
Italian (Switzerland)	it-ch	ITALIAN	SWITZERLAND
Japanese	ja	JAPANESE	JAPAN

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
Kannada	kn	KANNADA	INDIA
Kazakh	kk	CYRILLIC KAZAKH	KAZAKHSTAN
Konkani	kok	KOREAN	KOREA
Korean	ko	KOREAN	KOREA
Kyrgyz	kz	RUSSIAN	RUSSIA
Latvian	lv	LATVIAN	LATVIA
Lithuanian	lt	LITHUANIAN	LITHUANIANA
Malay (Malaysia)	ms	MALAY	MALAYSIA
Malayalam	ml	MALAYALAM	INDIA
Maltese	mt	ENGLISH	UNITED KINGDOM
Marathi	mr	ENGLISH	INDIA
Nepali (India)	ne	ENGLISH	UNITED KINGDOM
Norwegian (Bokmal)	nb-no	NORWEGIAN	NORWAY
Norwegian (Bokmal)	no	NORWEGIAN	NORWAY
Norwegian (Nynorsk)	nn-no	NORWEGIAN	NORWAY
Oriya	or	ORIYA	INDIA
Polish	pl	POLISH	POLAND
Portuguese (Brazil)	pt-br	BRAZILIAN PORTUGUESE	BRAZIL
Portuguese (Portugal)	pt	PORTUGUESE	PORTUGAL
Punjabi	pa	PUNJABI	INDIA
Romanian	ro	ROMANIAN	ROMANIA
Russian	ru	RUSSIAN	RUSSIA
Russian (Moldova)	ru-md	RUSSIAN	RUSSIA
Serbia	sr	CYRILLIC SERBIAN	SERBIA AND MONTENEGRO
Slovak	sk	SLOVAK	SLOVAKIA
Slovenian	sl	SLOVENIAN	SLOVENIA
Spanish (Argentina)	es-ar	LATIN AMERICAN SPANISH	ARGENTINA
Spanish (Bolivia)	es-bo	LATIN AMERICAN SPANISH	ARGENTINA
Spanish (Chile)	es-cl	LATIN AMERICAN SPANISH	CHILE
Spanish (Colombia)	ec-co	LATIN AMERICAN SPANISH	COLOMBIA
Spanish (Costa Rica)	es-cr	LATIN AMERICAN SPANISH	COSTA RICA
Spanish (Dominican Republic)	es-do	LATIN AMERICAN SPANISH	PUERTO RICO
Spanish (Ecuador)	es-ec	LATIN AMERICAN SPANISH	ECUDOR
Spanish (El Salvador)	es-sv	LATIN AMERICAN SPANISH	EL SALVADOR
Spanish (Guatemala)	es-gt	LATIN AMERICAN SPANISH	GUATEMALA
Spanish (Honduras)	es-hn	LATIN AMERICAN SPANISH	GUATEMALA
Spanish (Mexico)	es-mx	MEXICAN SPANISH	MEXICO
Spanish (Nicaragua)	es-ni	LATIN AMERICAN SPANISH	Nicaragua

Language Name	Language Code	NLS_LANGUAGE	NLS_TERRITORY
Spanish (Panama)	es-pa	LATIN AMERICAN SPANISH	Panama
Spanish (Paraguay)	es-py	LATIN AMERICAN SPANISH	ARGENTINA
Spanish (Peru)	es-pe	LATIN AMERICAN SPANISH	PERU
Spanish (Puerto Rico)	es-pr	LATIN AMERICAN SPANISH	PUERTO RICO
Spanish (Traditional Sort)	es	LATIN AMERICAN SPANISH	SPAIN
Spanish (United States)	es-us	LATIN AMERICAN SPANISH	AMERICAN
Spanish (Uruguay)	es-uy	LATIN AMERICAN SPANISH	ARGENTINA
Spanish (Venezuela)	es-ve	LATIN AMERICAN SPANISH	VENEZUELA
Swedish	sv	SWEDISH	SWEDEN
Swedish	sv-fi	SWEDISH	FINLAND
Tamil	ta	TAMIL	INDIA
Telugu	te	TELUGU	INDIA
Thai	th	THAI	THAILAND
Turkish	tr	TURKISH	TURKEY
Ukrainian	uk	UKRAINIAN	UKRAINE
Urdu	ur	ENGLISH	UNITED KINGDOM
Uzbek	uz	LATIN UZBEK	UZBEKISTAN
Vietnamese	vi	VIETNAMESE	VIETNAM

23
Managing Application Performance

Learn about how to improve application performance.

This section describes how to improve application performance.

- About Best Practices to Improve Performance
  Improve performance by using bind variables and monitor performance by using dashboards in Workspace Administration or querying APEX dictionary views.
- Identifying Performance Issues
  Consider the following key strategies to help identify and resolve performance issues with your Oracle APEX application.
- Utilizing Logs and Reports
  Review application logs and reports to help identify problematic pages.
- About Utilizing Database Reporting
  Improve application performance by utilizing database reporting.
- About Database Parameters that Impact Performance
  Evaluate whether or not database parameters are impacting performance.
- About Limiting Resources
  Use Resource Manager to limit the types of system resources available to each user request.
- About Uploading Static Files to Your Web Server
  To improve performance, you should place static files on the Web server.
- Creating Custom Activity Reports Using APEX_ACTIVITY_LOG
  Learn how to use the `APEX_ACTIVITY_LOG` view to query activity for the current workspace.

Related Topics

- Debugging an Application
  Learn about best practices for debugging an Oracle APEX application.

23.1 About Best Practices to Improve Performance

Improve performance by using bind variables and monitor performance by using dashboards in Workspace Administration or querying APEX dictionary views.

For applications having a large number of concurrent users, maintaining optimal performance is critical. Best practices to consider include:

- Use bind variables within your application whenever possible. You can reference session state values using bind variable syntax in SQL queries and application logic such as PL/SQL executed from processes and validations. Accessing session state using bind variables is the most efficient way to reference session state.
- Monitor application performance including page rendering time using reports in Workspace Administration, Monitor Activity. You can also query the `APEX_WORKSPACE_ACTIVITY_LOG` dictionary view. Application performance data is available when application activity logging is enabled.

See Also:

- Referencing Session State Using Bind Variable Syntax
- Using Substitution Strings
- Using the Attribute Dictionary
- Monitoring Activity Within a Workspace and Viewing Workspace Dashboards in *Oracle APEX Administration Guide*

23.2 Identifying Performance Issues

Consider the following key strategies to help identify and resolve performance issues with your Oracle APEX application.

- About Troubleshooting Slow-Running Applications
  Learn about troubleshooting slow-running applications.
- Diagnosing Performance Issues
  Learn how to diagnose performance issues.
- About Debugging Problematic SQL Queries
  Learn about debugging problematic SQL queries.
- About Addressing Slow Queries
  Learn about addressing slow queries.

23.2.1 About Troubleshooting Slow-Running Applications

Learn about troubleshooting slow-running applications.

The first step in troubleshooting a slow-running application is to determine the location of the bottleneck. You can evaluate application performance by reviewing the time it takes for the database to return a specific page request or submission. If this time is shorter than the time it takes to render the page on the user's browser, then other components are likely causing the perception that the application is slow. As a best practice, always check all of the components involved including client machines, the network, the middle-tier, the database, and disk and storage devices. Regarding the database, check the System Global Area (SGA), CPU utilization, locks, and database parameters. It is also important to determine if anything else is running within the database such as legacy applications, data warehouse, batch programs, and database jobs.

The majority of the time, performance issues with APEX applications are due to developer-authored SQL and PL/SQL. You should utilize standard database performance tuning techniques to address these issues. Performance issues are rarely related to a bug with APEX or the Oracle database.

See Also:

Oracle Database Performance Tuning Guide

23.2.2 Diagnosing Performance Issues

Learn how to diagnose performance issues.

To diagnose performance issues, follow these steps:

1. Identify the time window.
2. Isolate the largest consumers of database time.
3. Identify the APEX workspace and application.
4. Correlate the offending SQL with a specific location within the application.

Once you have identified the slow running pages, you should run the application in debug mode to identify the specific components consuming the most time.

23.2.3 About Debugging Problematic SQL Queries

Learn about debugging problematic SQL queries.

If your query does not seem to be running correctly, try running it in SQLcl, SQL Developer, or in SQL Commands. Any of these approaches will test your query outside the context of your application, making it easier to define the problem.

23.2.4 About Addressing Slow Queries

Learn about addressing slow queries.

For optimized queries that still take more than a second to process, consider implementing a progress bar and preventing multiple page submissions. Users expect instant results and often resubmit a page if it does not return immediately. Unless managed properly, each page submission instigates a new session in the database and continues until completion. Multiple page submissions often degrade performance and may lead to database locks. This is especially true with interactive reports. As an alternative approach, try using APEX collections to hold query results. By using collections, expensive and time consuming queries are only executed once for each user rather than each time they perform pagination or create a filter.

> **Tip:**
>
> You control whether end users may submit the same page more than once by configuring the page attribute, **Advanced**, **Enable duplicate page submissions**.

> **See Also:**
>
> - Editing Page Attributes
> - About Using Collections

23.3 Utilizing Logs and Reports

Review application logs and reports to help identify problematic pages.

The Oracle APEX engine records activity logs to capture usage and performance data. This information displays in various reports. If a particular page is accessed extensively or a page renders slowly then these reports can help you identify problematic pages.

You can then further analyze the problematic pages. For example, you can run a report query in SQL Developer to review the Explain Plan.

> **Tip:**
>
> Another method of determining expensive elements within a page or process is to review the APEX Debug mode.

- Accessing the Monitor Activity Page in Workspace Administration
  Use the Monitor Activity page in Workspace Administration to view developer activity and application change information.
- About Managing Log Files
  An Instance administrator can delete log files and manage the interval at which log file information is retained.
- Viewing the Dashboard Report
  The Dashboard report displays metrics about applications in your workspace.
- Instance Activity Reports
  Instance administrators are superusers that manage an entire hosted instance using the APEX Administration Services application.

> **See Also:**
>
> - APEX_DEBUG in *Oracle APEX API Reference*
> - Utilizing Debug Mode

23.3.1 Accessing the Monitor Activity Page in Workspace Administration

Use the Monitor Activity page in Workspace Administration to view developer activity and application change information.

To access the Monitor Activity page in Workspace Administration:

1. Navigate to the Workspace Administration page:
   a. Sign in to APEX.
   b. At the top of the Workspace home page, click the **Administration** menu and select **Administration**.

   The Workspace Administration home page appears.

2. Click **Monitor Activity**.

   The Monitor Activity page appears and is divided into the following sections:

   - **Page Views** - Contains reports of page views including: By View, By User, By User and Hour, By Application, By Application and Page, By Day, and By Hour.
   - **Developer Activity** - Offers reports of developer activity including: By Developer, By Developer Bar Chart, By Developer by Hour, By Day, By Application, Application Changes (detailed), By Day (Monthly View), and Component Utilitzation Chart.
   - **Sessions** - Lists active sessions with the current workspace such as Active Sessions and Bar Chart of Active Session by Hour.
   - **Page View Analysis** - Contains reports analyzing page views, such as Most Viewed Pages over All Applications, Monthly Calendar of Page Views by Day, Line Chart of Usage by Day, and By Weighted Page Performance.
   - **Environment** - Contains reports of environments including By User Agent, By Browser, By External Click, and By Operating System.
   - **Login Attempts** - Offers reports listing login attempts such as Login Attempts, Login Attempts by Authentication Result, and Developer Login Summary.
   - **Application Errors** - Links to the Application Errors report.
   - **Workspace Schema Reports and Other Reports** - Includes the following reports: Schema Tablespace Utilization, Database Privileges by Schema, Workspace Schemas, Report Tablespace Utilization, Web Service Activity Log, and Archive of Purged Task Files.
   - **Archived Activity Report** - Includes the following reports: By Application by Month, By Application by Day, Application Hash by Day, By Application, and Chart of Workspace Activity by Day.

3. Select a report to review.

23.3.2 About Managing Log Files

An Instance administrator can delete log files and manage the interval at which log file information is retained.

If you wish to keep the activity log files indefinitely, you must create a batch job to copy records from the activity log table to another table.

> **See Also:**
>
> Configuring and Deleting Logs and Log Entries in *Oracle APEX Administration Guide*

23.3.3 Viewing the Dashboard Report

The Dashboard report displays metrics about applications in your workspace.

To view the Dashboard:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Click **Dashboard**.

The Dashboard appears and features the following regions:

- **Applications** - Count of applications in the current workspace.
- **Pages** - Count of pages in the current workspace.
- **Developer Activity (Last 2 Days)**
- **Page Events**
- **Page Count by Application**
- **Most Active Pages**
- **Native Region Component Utilization**

23.3.4 Instance Activity Reports

Instance administrators are superusers that manage an entire hosted instance using the APEX Administration Services application.

> **See Also:**
>
> - Monitoring Activity Across a Development Instance in *Oracle APEX Administration Guide*
> - About Oracle APEX Administrator Roles

23.4 About Utilizing Database Reporting

Improve application performance by utilizing database reporting.

Oracle APEX establishes a physical connection from the database pool as `APEX_PUBLIC_USER`, which calls the APEX engine. The APEX engine utilizes `SQL.DBMS_SYS_SQL` to parse SQL as another user, the parsing schema. Database `V$SESSION` records contain useful information for each SQL executed including APEX specific information for the client information, client identifier, and module. You can use this information to identify the associated SQL when you generate Oracle Trace files and use `TKPROF` to analyze them. You can also add `&p_trace=YES` to the end of the APEX URL or use `ALTER SESSION`.

Automatic Workload Repository (AWR) reports collect performance statistics every hour, by default. Active Session History (ASH) reports are a system-wide record of database activity. You can use these reports to identify resource intensive SQL statements. For PL/SQL packages, procedures and functions that in turn call other PL/SQL programs, you can use PL/SQL hierarchical profiler in Oracle Database 11*g*. These subprograms account for execution times separately.

APEX also includes views which catalog everything related to APEX. You can use these views to access information from APEX using tools such as SQL Developer, SQLcl, or SQL Commands (Oracle APEX SQL Workshop). If you are granted the `APEX_ADMINISTRATOR_READ_ROLE` or `APEX_ADMINISTRATOR_ROLE` then you can query across the entire instance, rather than just the workspace your schema user is associated with. You should grant `APEX_ADMINISTRATOR_READ_ROLE` to monitoring users, while `APEX_ADMINISTRATOR_ROLE` should be used for instance administrators, who manage instance parameters, workspaces, and so on.

See Also:

- Enabling SQL Tracing and Using TKPROF
- *Oracle Database Performance Tuning Guide*
- *Oracle APEX SQL Workshop Guide*

23.5 About Database Parameters that Impact Performance

Evaluate whether or not database parameters are impacting performance.

There are certain database parameters that can impact performance. In particular, you should review the memory related parameters (`MEMORY_TARGET`, `SGA_TARGET`, `PGA_AGGREGATE_TARGET`, `SHARED_POOL_SIZE`) parameters to ensure they are sized correctly. You should also check that the database System Global Area (SGA) fits in real memory. An undersized SGA can dramatically impact Oracle APEX performance. There needs to be sufficient memory in the Shared Pool for PL/SQL and in Buffer Cache for the APEX metadata. If the SGA is sized correctly, it should not be necessary to pin PL/SQL programs from the APEX engine as they are accessed regularly and should not be swapped out of memory. Because the APEX engine resides in the database, it is strongly recommended that you gather statistics on the `APEX_240200` schema. Accurate statistics allow the cost-based optimizer to develop better query plans for accessing the application metadata.

23.6 About Limiting Resources

Use Resource Manager to limit the types of system resources available to each user request.

Database Resource Manager, an Oracle Database Enterprise Edition option, is exceedingly useful for maintaining an Oracle APEX instance. Using Resource Manager you can limit the types of system resources (`CPU_PER_CALL` and `LOGICAL_READS_PER_CALL`) available to each user request. By setting up multiple resource profiles, with automatic switches between the profiles, as a user request exceeds a certain threshold you can reduce the CPU available to that transaction. For example you may define a default profile APEX_HIGH which can utilize 70% of CPU for 10 seconds, APEX_MEDIUM with 8% CPU for 120 seconds, APEX_LOW with 2% CPU for 1800 seconds, and OTHERS with 20% CPU. When a specific transaction exceeds 10 seconds, it is switched to the APEX_MEDIUM profile which drastically reduces the CPU available. If the transaction switches to APEX_LOW and exceeds 1800 seconds then you can cancel the SQL or terminate the session. This approach ensures that one specific user or poorly written application cannot monopolize the CPU causing other users and applications to respond poorly.

Tip:

Instance administrators can define Resource Manager profiles for individual workspaces.

> **See Also:**
>
> - Isolating a Workspace to Prevent Browser Attacks and Configuring Instance-Level Workspace Isolation Attributes in *Oracle APEX Administration Guide*
> - Managing Resources with Oracle Database Resource Manager in *Oracle Database Administrator's Guide*

23.7 About Uploading Static Files to Your Web Server

To improve performance, you should place static files on the Web server.

As a best practice, you should also enable gzip compression on the Web server so that the file size is minimized when transmitting the static files to the browser. Furthermore, you should also enable file caching for browsers by sending expiration headers.

> **See Also:**
>
> - Managing Static Application Files
> - Managing Static Workspace Files

23.8 Creating Custom Activity Reports Using APEX_ACTIVITY_LOG

Learn how to use the `APEX_ACTIVITY_LOG` view to query activity for the current workspace.

- Enabling Logging for an Application
  Enable logging for an application on the Edit Application Definition page.
- APEX_ACTIVITY_LOG
  Use `APEX_ACTIVITY_LOG` to view and query all activity for the current workspace.
- APEX_ACTIVITY_LOG Sample Query
  View a `APEX_ACTIVITY_LOG` sample query.

23.8.1 Enabling Logging for an Application

Enable logging for an application on the Edit Application Definition page.

To edit the application definition:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click the **Edit Application Definition** button to the right of the application name.

   The Edit Application Definition page appears.
4. Under Properties, locate Logging. Select **Yes** or **No**.

When set to **Yes**, every page view is logged, enabling an administrator to monitor user activity for each application. Disabling logging may be advisable for high volume applications.

> **Tip:**
>
> You can only modify the Logging attribute if the Application Activity Logging attribute in Oracle APEX Administration Services is set to **Use Application Setting**.

> **See Also:**
>
> Enabling Application Activity Logging in *Oracle APEX Administration Guide*

23.8.2 APEX_ACTIVITY_LOG

Use `APEX_ACTIVITY_LOG` to view and query all activity for the current workspace.

The `APEX_ACTIVITY_LOG` view records all activity in a workspace, including developer activity and application runtime activity. You can use `APEX_ACTIVITY_LOG` to view and query all activity for the current workspace. For example, you can use this view to develop monitoring reports within a specific application to provide real-time performance statistics.

The following table describes the columns in the `APEX_ACTIVITY_LOG` view.

Column	Type	Description
`time_stamp`	DATE	Date and time that activity was logged at the end of the page view.
`component_type`	VARCHAR2(255)	Reserved for future use.
`component_name`	VARCHAR2(255)	Reserved for future use.
`component_attribute`	VARCHAR2(4000)	Title of page.
`information`	VARCHAR2(4000)	Reserved for future use.
`elap`	NUMBER	Elapsed time of page view in seconds.
`num_rows`	NUMBER	Number of rows processed on page.
`userid`	VARCHAR2(255)	User ID performing page view.
`ip_address`	VARCHAR2(4000)	IP address of client.
`ir_report_id`	NUMBER	Interactive report ID
`ir_search`	VARCHAR2	Interactive report search criteria entered by users.
`user_agent`	VARCHAR2(4000)	web browser user agent of client.
`flow_id`	NUMBER	Application ID.
`step_id`	NUMBER	Page number.
`session_id`	NUMBER	Oracle APEX session identifier.
`sqlerrm`	VARCHAR2(4000)	SQL Error message.

Column	Type	Description
sqlerrm_component_type	VARCHAR2(255)	Reserved for future use.
sqlerrm_component_name	VARCHAR2(255)	Reserved for future use.

To conserve space in the activity log, only the first log entry of each unique session contains the IP address and web browser user agent.

23.8.3 APEX_ACTIVITY_LOG Sample Query

View a APEX_ACTIVITY_LOG sample query.

The following example demonstrates how to create a report that displays the total number of page views and the average page view time in the past 24 hours for application 9529, and grouped by userid:

```
SELECT COUNT(*), AVG(elap), userid
    FROM APEX_ACTIVITY_LOG
 WHERE time_stamp > (SYSDATE-1)
   AND flow_id = 9529
GROUP BY userid
```

Keep in mind that activity logging in an APEX instance is rotated between two different log tables. Because there are two log tables, logging information is only as current as the oldest available entry in the logs. To persist your application-specific log information for all time, you must either copy the log information into your own application table or implement logging directly in your application.

24
Debugging an Application

Learn about best practices for debugging an Oracle APEX application.

- Reviewing Session State
  View current session state for your application by clicking the Session link on the Runtime Developer Toolbar.
- Utilizing Debug Mode
  Use Debug mode to track down unexpected application behavior.
- Removing Controls and Components to Isolate a Problem
  If you have problems running a page, try removing controls and components one at a time.
- About Monitoring Application and Page Resource Use
  Query the `V$SESSION` and `V$SQLAREA` views to monitor application and page resource use..
- Enabling SQL Tracing and Using TKPROF
  Enable SQL tracing and then analyze the temporary file Oracle APEX creates using the TKPROF utility.
- Running Advisor to Check Application Integrity
  Use Oracle APEX Advisor (Advisor) to check the integrity and quality of your APEX application.

Related Topics

- Managing Application Performance
  Learn about how to improve application performance.
- APEX_DEBUG

24.1 Reviewing Session State

View current session state for your application by clicking the Session link on the Runtime Developer Toolbar.

Session state enables developers to store and retrieve values for a user as the user navigates between different application pages. The appearance or behavior of many applications is based on data contained within application controls. For example, buttons can display conditionally based on a value stored in session state.

See Also:

- Runtime Developer Toolbar
- Viewing Session State
- Managing Session State Values
- Managing Session State for a Workspace and Managing Session State for an Instance in *Oracle APEX Administration Guide*

24.2 Utilizing Debug Mode

Use Debug mode to track down unexpected application behavior.

- Enabling and Disabling Debug Mode for an Application
  Configure whether end users can run the application in debug mode by editing the Debugging attribute in the Application Definition.
- Available Debug Levels
  Learn about debug levels available in Oracle APEX.
- Controlling Debug Mode for a Page at Runtime
  Enable and disable Debug mode for the current page and session by clicking the Debug on the Runtime Developer toolbar.
- Configuring Debug Mode Using the Session Detail Report
  View the Session Detail report to review session information and configure Debug Level and Trace Mode for an active session.
- Configuring the Debug Level Programatically
  Use the `SET_DEBUG` procedure to set the debug level for all future requests in a session.
- Using URL Parameters to Control Debug Mode
  Learn about using URL parameters to control Debug mode.
- Controlling the Debug Level for an Instance
  Change the default instance-wide Debug Level using the `APEX_INSTANCE_ADMIN` API
- Viewing Debug Messages
  Learn about viewing debug messages.
- Using APEX_DEBUG When Developing Your Application
  Learn about APIs in the `APEX_DEBUG` package.

24.2.1 Enabling and Disabling Debug Mode for an Application

Configure whether end users can run the application in debug mode by editing the Debugging attribute in the Application Definition.

Running an application in debug mode is useful when an application is under development. For production applications, Oracle recommends disabling debugging and thus preventing users from viewing application logic.

Developer who are logged into workspace where the application resides can always run the application in debug mode. Debug can also be enabled programmatically regardless of this debug setting.

To enable or disable debug mode for end users:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.

   The Application home page appears.
3. From the Application home page, access the Application Definition in one of the following ways:
   - **Edit Application Definition** button:
     a. Click **Edit Application Definition** to the right of the application name.

   b. Click the **Definition** tab.
- From Shared Components:
   a. Click **Shared Components**.
   b. Under Application Logic, click **Application Definition**.

4. Under Properties, edit the Debugging attribute:
   - **Yes** - Enables the application debug mode using the browser at runtime.
   - **No**- Disables the application debug mode using the browser at runtime.

> **See Also:**
>
> Managing Application Attributes

24.2.2 Available Debug Levels

Learn about debug levels available in Oracle APEX.

Oracle APEX supports seven different debug levels. Each level enables a new set of messages on top of the ones enabled in the previous level. For example, using level 2 will enable the messages in level 1 and level 2. Using level 9 will enable all messages from all the previous levels plus the ones in level 9.

Debug levels available in APEX include:

- **Error (LEVEL1)** - Logs error messages.
- **Warn (LEVEL2)** - Logs warning messages.
- **Info (LEVEL4)** - Logs informational messages. This is the default level when debugging is enabled but not set to a specific level.
- **Application Enter (LEVEL5)** - Logs messages when entering (starting) and leaving (finishing) specific methods and operations specific to the application.
- **Application Trace (LEVEL6)** - Logs detailed information about methods as well as operations specific to the application.
- **APEX Engine Enter (LEVEL8)** - Logs messages when entering (starting) and leaving (finishing) specific methods as well as operations from the APEX engine.
- **APEX Engine Trace (LEVEL9)** - Logs detailed information about methods and operations from the APEX engine.

> **Note:**
>
> APEX engine will generate a substantial amount of messages at higher debug levels, which will impact application performance and require more time to review and analyze. Oracle recommends debugging applications at the **Application Trace (LEVEL6)** level or lower unless directed by Oracle Support.

24.2.3 Controlling Debug Mode for a Page at Runtime

Enable and disable Debug mode for the current page and session by clicking the Debug on the Runtime Developer toolbar.

To enable or disable Debug mode for a page at runtime:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the desired application.
3. Run the application.

   The Runtime Developer toolbar displays at the bottom of any editable page.

4. On the Runtime Developer Toolbar, click **Debug**. Available options include:
   - **Enable Debug** - Specifies the level of detail included when you click **View Debug** .
     - **Off** - Enables and disables debug mode
     - **Info (default)** - The debug log contains the default or normal amount of information. This corresponds with a log `LEVEL4` and a Debug URL argument of `YES`.
     - **App Trace** - The debug log contains messages up to level 6. This corresponds with a Debug URL argument of `LEVEL6`.
     - **Full Trace** - The log will contain the maximum amount of information. This corresponds with a Debug URL argument of `LEVEL9`. **Full Trace** can slow request processing but provides the most amount of detail.

     If debug mode is enabled, debug data is captured in the background. As long as the browser tab or window stays open, uses the current debug mode and level.
   - **View Debug** - Displays debug information for the current page.

> **See Also:**
>
> - Runtime Developer Toolbar
> - APEX_DEBUG in *Oracle APEX API Reference*

24.2.4 Configuring Debug Mode Using the Session Detail Report

View the Session Detail report to review session information and configure Debug Level and Trace Mode for an active session.

You can configure the Debug Level and Trace Mode for an active session by viewing the Session Detail report. From the Administration menu, select Monitor Activity, Active Sessions, and select an active session.

To learn more, see Viewing the Active Sessions Report in *Oracle APEX Administration Guide*.

24.2.5 Configuring the Debug Level Programatically

Use the `SET_DEBUG` procedure to set the debug level for all future requests in a session.

To learn more about the `SET_DEBUG` procedure, see SET_DEBUG Procedure in *Oracle APEX API Reference*.

24.2.6 Using URL Parameters to Control Debug Mode

Learn about using URL parameters to control Debug mode.

- Using Friendly URL Syntax to Access Debug Mode
  Use Friendly URL syntax to access debug mode.
- Using f?p Syntax to Access Debug Mode
  Use `f?p` syntax to access debug mode.

24.2.6.1 Using Friendly URL Syntax to Access Debug Mode

Use Friendly URL syntax to access debug mode.

Use Friendly URL syntax to run an application in debug mode, call the page and set the debug query parameter to `YES` or `LEVELn`, where *n* is either 1 (least detailed), 2, 4, 5, 6, 8, and 9 (most detailed). Setting the parameter to `YES` is equivalent to using `LEVEL4`. For example:

```
https://<HOSTNAME>:<PORT>/ords/r/workspace_name/application_name/page_name?
session=&APP_SESSION.&debug=YES
```

```
https://<HOSTNAME>:<PORT>/ords/r/workspace_name/application_name/page_name?
session=&APP_SESSION.&debug=LEVEL9
```

> **Note:**
>
> Even when debugging is turned off, error messages are logged.

> **See Also:**
>
> - Understanding Friendly URL Syntax
> - Available Debug Levels

24.2.6.2 Using f?p Syntax to Access Debug Mode

Use `f?p` syntax to access debug mode.

Use `f?p` syntax to run an application in debug mode, call the page and set the debug query parameter to `YES` or `LEVELn`, where *n* is either 1 (least detailed), 2, 4, 5, 6, 8, and 9 (most detailed). Setting the parameter to `YES` is equivalent to using `LEVEL4`. For example:

```
f?p=100:1:&APP_SESSION.::YES
```

> **Note:**
>
> Even when debugging is turned off, error messages are logged.

> **Note:**
>
> - Understanding f?p URL Syntax
> - Available Debug Levels

24.2.7 Controlling the Debug Level for an Instance

Change the default instance-wide Debug Level using the `APEX_INSTANCE_ADMIN` API

For example, to set the Debug Level to `Application Trace (LEVEL6)`, execute a statement similar to the following as `SYS` or the database user who has been granted the `APEX_ADMINISTRATOR_ROLE` database role.

```
BEGIN
    APEX_INSTANCE_ADMIN.SET_PARAMETER(
        p_parameter => 'SYSTEM_DEBUG_LEVEL',
        p_value     => '6');
    COMMIT;
END;
```

> **See Also:**
>
> SET_PARAMETER Procedure in *Oracle APEX API Reference*

24.2.8 Viewing Debug Messages

Learn about viewing debug messages.

- Viewing Debug Messages for an Application
  View debug messages for an application.
- Viewing Debug Messages from Utilities
  Use the Debug Messages page in Utilities to view or purge debug messages generated by an application.
- Querying Debug Messages
  Learn about querying debug messages.

- Available Debug Utility Scripts
  Learn about available debug utility scripts.
- Debugging an APEX Session from the User Interface
  Learn about debugging an APEX session from the user interface.
- Tips for Sharing Debug Output
  Learn about best practices for sharing debug output.

24.2.8.1 Viewing Debug Messages for an Application

View debug messages for an application.

> **Tip:**
>
> You can also access Debug messages by navigating to the application and clicking the **Find** icon on the toolbar. When the Item Finder appears, click the **Debug** tab.

To view debug messages:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the desired application.
3. Run the application.
4. Find the Runtime Developer toolbar.

5. On the Runtime Developer toolbar, click **Debug** and then select **View Debug**.

   The Debug report appears.
6. Find the session to view and click the **View Identifier** link.

   The Debug page appears.
7. To view steps that took the longest, hover over that step on the graph above the report.

   The graph with step details displays.
8. To go to that step in the report, click the bar.

> **Note:**
>
> Debug sessions listed in the report exist for at least 2 weeks and age out along with the activity log. This timeframe is usually more than adequate for debugging purposes.

24.2.8.2 Viewing Debug Messages from Utilities

Use the Debug Messages page in Utilities to view or purge debug messages generated by an application.

Use the Debug Messages page to view all debug messages generated by an application and purge debug messages data by age, current session or View Identifier.

To view the Debug Messages page:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the desired application.
3. Click **Utilities**.
4. On the Utilities page, click **Debug Messages**.

   The Debug Messages page appears.
5. Click the **View Identifier** for the session you want to view.

   The Debug Message Data page appears.
6. To view steps that took the longest, hover over that step on the graph above the report.

   The step details display.
7. To go to a step in the report, click the bar graph.

> **Note:**
>
> Debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

8. To purge debug messages:
   a. Click **Purge Debug Messages**.
   b. On Purge Debug Messages Data - Select a purge option:
      - **Purge all messages**
      - **Purge messages by age** - ,In **Older than**, select how many days or weeks of debug message data you wish to purge for the application.
      - **Purge current session messages**
      - **Purge by View Identifier** - If selected, enter the **View Identifier**.
   c. Click **Purge**.

24.2.8.3 Querying Debug Messages

Learn about querying debug messages.

> **Tip:**
>
> This topic applies to both development and runtime-only instances.

A page view ID makes it easier to query debug messages. In order to obtain it, you can query the `APEX_WORKSPACE_ACTIVITY_LOG` as shown in the following example:

```
select *
  from APEX_WORKSPACE_ACTIVITY_LOG
 where application_id = <APPLICATION_ID>
   and page_id = <PAGE_ID>
 order by view_timestamp desc;
```

Locate and make note of the `DEBUG_PAGE_VIEW_ID` associated with the debug messages you are looking for.

You can query the debug messages from the `APEX_DEBUG_MESSAGES` view, which shows all messages logged in the current workspace. Messages can be filtered and sorted as shown in the following example:

```
select *
  from APEX_DEBUG_MESSAGES
 where page_view_id = <DEBUG_PAGE_VIEW_ID>
 order by message_timestamp asc;
```

In the previous example, *`<DEBUG_PAGE_VIEW_ID>`* is the ID you noted earlier. The messages column in the above query will contain the most relevant information for debugging purposes.

24.2.8.4 Available Debug Utility Scripts

Learn about available debug utility scripts.

> **Tip:**
>
> The scripts described in this topic are not available for Oracle APEX running on Autonomous Database.

The APEX software distribution includes SQL utility scripts to aid with debugging. You can find these scripts in the `apex/utilities/debug` folder. To use the scripts described in this topic, you must have `SYSDBA` privileges.

activity.sql

`activity.sql` prints recent activity log entries. This script takes one parameter, which can contain a comma-separated list of settings. To see script usage instructions, execute:

```
SQL> @activity help
```

d0.sql

`d0.sql` toggles between disabling Debug mode and a LEVEL9 debug log. A LEVEL9 debug level logs every request and generates a substantial amount of messages. A LEVEL9 debug log will impact application performance for all users and App Builder performance for developers. For example:

```
-- Switch from disabled debug to LEVEL9
SQL> @d0
-- Switch from LEVEL9 debug to disabled debug
SQL> @d0
```

> **Note:**
>
> Oracle recommends disabling instance-wide debugging as soon as you collect the necessary debug information since it will impact application performance.

d1.sql

`d1.sql` displays a report of the last 30 APEX requests that had debug enabled. Use the displayed `PAGE_VIEW_ID` with the `d2.sql` script.

ds.sql

`ds.sql` displays a report of APEX requests with debug data for the specified APEX session. Use the displayed `PAGE_VIEW_ID` with the `d2.sql` script. For example:

```
SQL> @ds <SESSION_ID>
```

d2.sql

`d2.sql` saves the details for the specified `PAGE_VIEW_ID` into a file and opens this file in a text editor. You can define the `_editor` variable in SQLcl to your preferred editor. For example:

```
SQL> define _editor=vim
SQL> @d2 <PAGE_VIEW_ID>
```

Example

The following is a typical usage example of APEX debug utility scripts:

```
$ cd apex/utilities/debug
$ sql / as SYSDBA
SQL> alter session set current_schema=APEX_240200;
SQL> define _editor=vim
SQL> @d0
Changed debug level from "" to "9"
-- Reproduce the problem
SQL> @d0
Changed debug level from "9" to ""
SQL> @d1

PAGE_VIEW_ID STARTED  SECS   LVL  CNT PATH_INFO   FLOW_PAGE   SESSION_ID
WORKSPACE APEX_USER
___________ ________ ______ ___ ____ ___________ ___________ _______________
_________ _________
@d2 1002     14:02:48  10.84 WRN  261             100-4750:50 8488213505163
DEMO      DEMO
@d2 1003     14:12:22   0.87     2603 show        100:9       0-5628985429897
DEMO      DEMO
@d2 1006     14:12:23   0.75      252 ajax plugin 100:9       0-5628985429897
DEMO      DEMO
@d2 1007     14:12:41   2.04      297 ajax plugin 100:9       0-5628985429897
DEMO      DEMO

4 rows selected.

SQL> @d2 1003

-- This will open the editor program you defined above, in this case vim with
content similar to the following:
```

```
SECS MESSAGE_LEVEL MESSAGE FLOW_PAGE SID_SGID APEX_USER CALL_STACK1
____ _____________ _______ _________ ________ _________ ___________
...
```

Alternatively, you can execute `ds.sql` instead of `d1.sql` to narrow down the search of a `PAGE_VIEW_ID` to a specific APEX Session ID. For example:

```
$ cd apex/utilities/debug
$ sql / as SYSDBA
SQL> alter session set current_schema=APEX_240100;
SQL> define _editor=vim
SQL> @d0
Changed debug level from "" to "9"
-- Reproduce the problem and obtain the APEX session ID, in this case
5628985429897
SQL> @d0
Changed debug level from "9" to ""
SQL> @ds 5628985429897

PAGE_VIEW_ID STARTED  SECS    CNT PATH_INFO
FLOW_PAGE SESSION_ID      WORKSPACE APEX_USER
___________ ________ ______ ____ ____________________________________
_________ _______________ _________ _________
@d2 1003     14:12:22   0.87 2603 bar?clear=9&session=5628985429897&cs=391
100:9     0-5628985429897 DEMO      DEMO
@d2 1006     14:12:23   0.75  252 wwv_flow.ajax?p_context=sample-charts%2F
100:9     0-5628985429897 DEMO      DEMO
@d2 1007     14:12:41   2.04  297 wwv_flow.ajax?p_context=sample-charts%2F
100:9     0-5628985429897 DEMO      DEMO
@d2 1008     14:12:41   1.95  297 wwv_flow.ajax?p_context=sample-charts%2F
100:9     0-5628985429897 DEMO      DEMO

4 rows selected.

SQL> @d2 1003

-- This will open the editor program you defined above, in this case vim with
content similar to the following:

SECS MESSAGE_LEVEL MESSAGE FLOW_PAGE SID_SGID APEX_USER CALL_STACK1
____ _____________ _______ _________ ________ _________ ___________
...
```

24.2.8.5 Debugging an APEX Session from the User Interface

Learn about debugging an APEX session from the user interface.

If you are unable to run the debug utility scripts described in Available Debug Utility Scripts, you can debug an APEX session using existing reports in the user interface.

> **Note:**
>
> One requirement of this procedure is that you must access an APEX session by including a public page in your application. If you create a public page specially for debugging, remember to delete it once you have resolved the error.

To perform session debugging from the APEX user interface:

1. If your application does not have a public page, create one by running the Create Page Wizard.
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. Click the **Create Page** button.
   d. In the Create Page Wizard:
      - Component - Select **Blank Page**.
      - Page Definition, Name - Specify a text name for this page.
      - Click **Create Page**.
2. Edit the page attributes and make the page public:
   a. On the Application home page, select the new page.

      Page Designer appears.
   b. In the Rendering tab, select the page.
   c. In the Property Editor:
      - Find the **Security** group.
      - Authentication - Select **Page is Public**
3. Run the page. Click **Save and Run Page**.
4. Return to your development environment tab and view the Session Detail report and set the Debug Level:
   a. Click the **Administration** menu and select **Monitor Activity**.
   b. Under Sessions, select **Active Sessions**.
   c. Click a session ID to view the Session Details page.
   d. Session Attributes, Debug Level - Select **APEX Trace** (LEVEL9). This option logs detailed information about methods and operations from the APEX engine.
   e. Click **Apply Changes**.

      APEX will now log debug data for this session on each request, including `APEX_AUTHENTICATION.CALLBACK`.
5. Reproduce the error:
   a. In you application, change the URL to navigate to a protected page and press **Enter**.

      APEX will detect that authentication is required and redirect to your SSO login server.
   b. Enter your credentials and continue, until you see the APEX error page.
6. Return to your development environment tab and view the Session Detail report in Workspace Administration again.

a. Click the **Administration** menu and select **Monitor Activity**.

b. Under Sessions, select **Active Sessions**.

c. Click a session ID to view the Session Details page.

d. On the Session Details page, find the Page View region and click **Go**.

e. Click the Debug ID for the request that ran into an error to view the details. This report captures the same information as if you ran `d2.sql` script.

> **Tip:**
>
> Not all available columns display by default in the report. To change the default columns that appear, click the **Actions** menu and then **Columns**. Then, select additional columns and click **Apply**. For example, consider adding **Call Stack**.

24.2.8.6 Tips for Sharing Debug Output

Learn about best practices for sharing debug output.

If you wish to share the debug output from your application for offline review, consider the following best practices:

- Enable the capture log at the appropriate level.

  For example, if you are debugging the behavior of your application, use **Application Trace (LEVEL6)**. If you are debugging the behavior of the APEX engine, use **APEX Engine Trace (LEVEL9)**.

  See Available Debug Utility Scripts to learn more about debug levels.

- If you are reviewing debug output at the **APEX Engine Trace (LEVEL9)** level, consider including the Call Stack column in the report. This is particularly useful if you are planning to share debug output with Oracle Support. If you are reviewing debug output in an interactive report, use the Actions menu to select the Call Stack column.

- Pick the appropriate Page View ID. Best practices to consider:

  – When debugging an operation that takes place during the rendering of the page, select the last page view with the Rendering page view type.

  – If the operation takes place in the submission of the page (for example, after clicking a button) select the last page view with the Processing page view type.

  – If the operation happens in-between (for example, a partial page refresh or rendering of a chart) select the last page view with the AJAX page view type.

  – Remember that modal dialogs are separate pages so when debugging dialogs, you will need to look for the dialog's page number rather than the page number of the parent page.

- You can export debug output in multiple formats. The most useful formats for offline review are the HTML format and the Excel format. If you are reviewing debug output in an interactive report, use the Actions menu, click **Download**, and select the **HTML** or **Excel** format.

To share debug output:

1. Find the desired Page View ID. To learn more about available debug reports, see 24.5.6 Viewing debug messages.

2. From the Actions menu of the interactive report, click **Download**, select the desired format, and save the debug report to your file system.
3. If App Builder is not available in your environment, use the `d2.sql` script to print debug output and share the resulting text file. To learn more, see 24.5.6.4 Using the Debug Utility Scripts to learn more.

24.2.9 Using APEX_DEBUG When Developing Your Application

Learn about APIs in the `APEX_DEBUG` package.

The `APEX_DEBUG` package contains APIs that you can utilize when developing your application and add application-specific debug messages to the APEX Debug Log. Custom application debug messages will be recorded alongside messages emitted by the APEX engine.

- Programmatically Enabling or Disabling Debugging for a Call
  Learn about enabling or disabling debugging for a call.
- Programmatically Enabling or Disabling Debugging for a Specific Session
  Learn about enabling or disabling debugging on a specific session.
- Logging Messages
  Learn about sever-side and client-side logging.

24.2.9.1 Programmatically Enabling or Disabling Debugging for a Call

Learn about enabling or disabling debugging for a call.

You can use the following methods to enable or disable debugging:

- `APEX_DEBUG.ENABLE` - This procedure enables debug messaging. You can specify the types of debug messages that are logged by level of importance.
- `APEX_DEBUG.DISABLE` - This procedure disables debug messaging.

Note you only need to call `APEX_DEBUG.ENABLE` once to enable debugging for the duration of the APEX request.

> **See Also:**
>
> ENABLE Procedure and DISABLE Procedure in *Oracle APEX API Reference*.

24.2.9.2 Programmatically Enabling or Disabling Debugging for a Specific Session

Learn about enabling or disabling debugging on a specific session.

You can use `APEX_SESSION.SET_DEBUG` to enable or disable debugging on a specific session. This is useful when debugging issues only reproducible for other users. Note that the higher the debug level, the more resources pages will consume to be rendered and pages will noticeable slow down.

> **See Also:**
>
> SET_DEBUG Procedure in *Oracle APEX API Reference*.

24.2.9.3 Logging Messages

Learn about sever-side and client-side logging.

Server-side Logging

You can use the following APIs to aid with debugging your application. You can use these APIs in application components such as processes and dynamic content regions as well as in your PL/SQL packages. These APIs will record debug information only when debugging is enabled for given APEX session or at the APEX instance level.

- `APEX_DEBUG.ERROR` - Logs messages on LEVEL1. It can be used to log unrecoverable application error messages to the debug log.
- `APEX_DEBUG.WARN` - Logs messages on LEVEL2. It can be used to log recoverable application error messages to the debug log.
- `APEX_DEBUG.INFO` - Logs messages on LEVEL4. It can be used to log messages about the status of a process or operation to the debug log.
- `APEX_DEBUG.ENTER` - Logs messages on LEVEL5. It is used to log the names and parameters of methods that are "entered" and is tyìcally found to be the first statement in a procedure's or function's begin block
- `APEX_DEBUG.TRACE` - Logs messages on LEVEL6. It is used to log very detailed information about a process or operation to the debug log.

If you want to log a debug message at a debug level that is not known until runtime, use the general `APEX_DEBUG.MESSAGE` procedure. If you want to log a message that is longer than 4,000 characters, use `APEX_DEBUG.LOG_LONG_MESSAGE` procedure.

To view or query the collected custom debug messages, see Viewing Debug Messages.

> **See Also:**
>
> APEX_DEBUG in *Oracle APEX API Reference*

Client-side Logging

You can also use the `apex.debug` JavaScript API to instrument your JavaScript code so that messages can show in the browser developer console only when the appropriate debug level is set. For example, you can incorporate *apex.debug* calls in your JavaScript code that is executed from Dynamic Actions.See apex.debug in the APEX JavaScript APIs Documentation (Link)

> **See Also:**
>
> apex.debug in the *Oracle APEX JavaScript API Reference*

24.3 Removing Controls and Components to Isolate a Problem

If you have problems running a page, try removing controls and components one at a time.

Using this approach, you can quickly determine which control or component may be the source of your problem.

- Excluding and Including Page Components
  Exclude or include page components at runtime using the Comment Out and Uncomment context menu option in Page Designer.
- Using Server-side Conditions
  Use Server-side conditions to prevent a component from rendering or executing.

24.3.1 Excluding and Including Page Components

Exclude or include page components at runtime using the Comment Out and Uncomment context menu option in Page Designer.

Starting with APEX release 23.2, new applications include a Comment Out and Uncomment context menu in Page Designer. To exclude an application component at runtime, right-click the component in the left tab and select **Comment Out** from the context menu. To include a component at runtime, select the page component again but this time select **Uncomment** from the context menu. The Comment Out and Uncomment context menu option appears in the Rendering tab, the Dynamic Actions tab, and the Processing Tab.

If you do not see Comment Out/Uncomment context menu , you can change the build option manually in Property Editor under Configuration, Build Options.

> **Tip:**
>
> The **Comment Out** and **Uncomment** context menu options are created from a build option named **Commented Out**. To learn more, see Using Build Options to Control Configuration.

> **See Also:**
>
> Commenting Out Page Components in Page Designer

24.3.2 Using Server-side Conditions

Use Server-side conditions to prevent a component from rendering or executing.

You can configure the Server-side Condition of **Never** to prevent a component from rendering or executing.

The following are typical steps for creating Server-side condition of **Never**:

> **Note:**
>
> The user interface may differ slightly depending upon the selected page component.

1. In Page Designer, select the page component.
2. Find the **Server-side Condition** group.

3. From Type, select **Never**.

> **Note:**
>
> Changing a condition to **Never** will remove any existing condition. To temporarily disabli a component, Oracle recommends using the Build Option attribute.

> **See Also:**
>
> - Using Build Options to Control Configuration
> - Conditions Available in App Builder

24.4 About Monitoring Application and Page Resource Use

Query the `V$SESSION` and `V$SQLAREA` views to monitor application and page resource use..

APEX facilitates the monitoring of resources used by applications and pages by calling the package `DBMS_APPLICATION_INFO`. Whenever the APEX engine renders or processes a page, the module is set to `APEX` and includes the application ID and page number. Once set, you can query the `V$SESSION` and `V$SQLAREA` views to monitor transactions.

24.5 Enabling SQL Tracing and Using TKPROF

Enable SQL tracing and then analyze the temporary file Oracle APEX creates using the TKPROF utility.

Tracing your session can be a very effective way to debug an application. From a database perspective, each page request is a single database session. If you enable SQL tracing, then Oracle APEX creates a temporary file you can then analyze using the TKPROF utility.

You enable SQL tracing in APEX by using `f?p` syntax to set the argument `p_trace=YES`. For example, to trace the display of page 1 in application 100, you would use the syntax:

```
http:/.../f?p=100:1&p_trace=YES
```

To use the TKPROF utility:

1. Log in to SQLcl as a privileged user.
2. Execute the following statement:

   ```
   show parameter USER_DUMP_DEST
   ```

3. Navigate to the directory in which the trace file is created.
4. Run the TKPROF utility from the operating system prompt using the following syntax:

   ```
   tkprof filename1 filename2 [waits=yes|no] [sort=option] [print=n]
       [aggregate=yes|no] [insert=filename3] [sys=yes|no] [table=schema.table]
       [explain=user/password] [record=filename4] [width=n]
   ```

   The input and output files are the only required arguments.

5. To view online Help, invoke TKPROF without arguments.

> **See Also:**
>
> Tools for End-to-End Application Tracing in *Oracle Database SQL Tuning Guide* for information about using the TKPROF program

24.6 Running Advisor to Check Application Integrity

Use Oracle APEX Advisor (Advisor) to check the integrity and quality of your APEX application.

- About APEX Advisor
  Learn About APEX Advisor.
- Running Advisor on an Entire Application
  Learn how to run Advisor on an entire application.
- Running Advisor on a Single Page
  Learn how to running Advisor on a single page.

24.6.1 About APEX Advisor

Learn About APEX Advisor.

Before deploying your application, you can use the APEX Advisor to perform various sanity checks on your application. The Advisor checks for errors, security issues, performance bottlenecks, quality assurance, and other best practices.

Oracle APEX Advisor functions like a compiler or LINT and flags suspicious behavior or errors. Running Advisor checks the integrity of your application based on the underlying metadata.

> **Tip:**
>
> APEX Advisor includes checks specific to ensuring application accessibility. To learn more, see *Oracle APEX Accessibility Guide*.

24.6.2 Running Advisor on an Entire Application

Learn how to run Advisor on an entire application.

To run Advisor on an entire application:

1. Navigate to the appropriate application:
   a. On the Workspace home page, click the **App Builder** icon.

      The App Builder home page appears.
   b. Select the application.

      The Application home page appears.
2. Click **Utilities**.
3. Click **Advisor**.

The Advisor page appears.

4. Under Checks to Perform:
   a. Expand Checks to Perform.
   b. Review the selected options. Select and deselect options as appropriate.
5. In Pages(s), enter a comma separated list of pages in the field provided. To check all pages, leave this option blank.
6. Click **Perform Check** at the top of the page.

   A Results page appears.

   > **Tip:**
   >
   > Current Advisor settings are used next time the check is performed.

7. To alter the existing preferences:
   a. Under Filter Result, deselect the appropriate options.
   b. Click **Apply Filter**.
8. To run Advisor again, click **Perform Check**.

24.6.3 Running Advisor on a Single Page

Learn how to running Advisor on a single page.

To run Advisor on a single page:

1. Navigate to the appropriate page.
2. Click the **Utilities** menu and select **Advisor**.

   Advisor appears.
3. Under Checks to Perform, review the selected options. Select and deselect options as appropriate.
4. Click **Perform Check** at the top of the page.

   A Results page appears.
5. To alter the existing preferences:
   a. Under Filter Result, deselect the appropriate options.
   b. Click **Apply Filter**.
6. To re-run the Advisor and see if reported problems are fixed, click **Perform Check**.

25
Deploying an Application

Learn about best practices for deploying an application. Deployment is the process of moving an application from the testing phase to the production phase.

- System Development Life Cycle Methodologies to Consider
  The system development life cycle (SDLC) is the overall process of developing software using a series of defined steps. There are several system SDLC models that work well for developing applications in Oracle APEX.
- Understanding the Deployment Process
  Learn about different approaches and best practices when deploying applications.
- Exporting and Importing from App Builder
  Move a workspace, an application, application components, or related files by creating an export script file and then importing and installing it into another APEX instance.
- How to Create a Custom Application
  Simplify the steps needed to deploy an application by creating a custom application using the Supporting Objects utility.
- About Publishing an Application URL
  Once you deploy your application, load the data, and create users, you can publish your production URL.
- Using Build Options to Control Configuration
  Use build options to conditionally include or exclude application components and functionality.
- Understanding One-Click Remote Application Deployment
  Deploy an application to remote Oracle APEX instances using REST Enabled SQL references.

25.1 System Development Life Cycle Methodologies to Consider

The system development life cycle (SDLC) is the overall process of developing software using a series of defined steps. There are several system SDLC models that work well for developing applications in Oracle APEX.

- Differences Between Iterative and Planned Development
  Learn about the differences between iterative and planned development.
- About the Advantages of Creating Prototypes
  Learn about the advantages of creating prototypes.
- Waterfall
  Learn about the Waterfall methodology.
- Spiral
  Learn about the Spiral methodology.
- Rapid Application Development
  Learn about the Rapid Application Development (RAD) methodology.

25.1.1 Differences Between Iterative and Planned Development

Learn about the differences between iterative and planned development.

When developing applications using App Builder, you must find a balance between two dramatically different development methodologies:

- Iterative, rapid application development
- Planned, linear style development

Iterative, rapid application development offers so much flexibility that you run the risk of never completing your project. In contrast, planned, linear style development can yield applications that do not meet the needs of end users even if they meet the stated requirements on paper.

25.1.2 About the Advantages of Creating Prototypes

Learn about the advantages of creating prototypes.

The APEX development environment enables developers to take a more iterative approach to development. Unlike many other development environments, creating prototypes is easy. With APEX, developers can:

- Use built-in wizards to quickly design an application user interface.
- Make prototypes available to users and gather feedback.
- Implement changes in real time, creating new prototypes instantly.

Methodologies that work well with APEX include Spiral and Rapid Application Development (RAD).

> **See Also:**
>
> - Spiral
> - Rapid Application Development

25.1.3 Waterfall

Learn about the Waterfall methodology.

The **Waterfall** is probably the best known SDLC model. In this methodology, the development process is broken down into the following stages:

1. Project Planning
2. Requirements Definition
3. Design
4. Development
5. Integration and Testing
6. Installation and Acceptance
7. Maintenance

This methodology is referred to as a waterfall because the output from one stage is the input for the next stage.

A primary problem with this approach is that it is assumed that all requirements can be established in advance. Unfortunately, requirements often change and evolve during the development process.

25.1.4 Spiral

Learn about the Spiral methodology.

A **Spiral** methodology is actually a series of short waterfall cycles. Each waterfall cycle yields new requirements and enables the development team to create a robust series of prototypes. One advantage of this approach is that it accommodates changing requirements. Disadvantages include complex project management and the risk development goes on indefinitely.

25.1.5 Rapid Application Development

Learn about the Rapid Application Development (RAD) methodology.

A **Rapid Application Development** (RAD) methodology has a heavy emphasis on creating a prototype that closely resembles the final product. The prototype is an essential part of the requirements phase. Advantages of this model include the ability to accommodate changing requirements, rapid development cycles, and progress can be easily measured. The major disadvantage of this model is that the emphasis on prototyping can result in scope creep. As a result, developers can lose sight of their initial goals in the attempt to create the perfect application.

25.2 Understanding the Deployment Process

Learn about different approaches and best practices when deploying applications.

- About Suggested Environments
  Learn about the best practices in development, testing, and production environments.
- Deployment Scenarios to Consider
  Review common deployment scenarios for APEX applications.

25.2.1 About Suggested Environments

Learn about the best practices in development, testing, and production environments.

Oracle recommends developers follow standard system development life cycle practices when developing applications by having three environments: **development**, **testing**, and **production**. As a best practice, developers should only make changes to applications and related database objects in their **development** environment. To further enforce this policy Oracle recommends using a runtime environment for both **testing** and **production** environments. This approach will prohibit developers from accessing the App Builder and SQL Workshop in these environments. As a general rule, your administrators (DBAs) should be the only ones with permission to update the test and production environments. DBAs should use the appropriate provided APIs and import applications from a SQL interface, such as SQLcl or Oracle SQL Developer Command Line.

> **See Also:**
>
> About the Differences Between Runtime and Full Development Environments

25.2.2 Deployment Scenarios to Consider

Review common deployment scenarios for APEX applications.

When you develop an application, you create the application within a specific workspace. Each workspace has a unique ID and name. A common scenario is to create the application in a development instance and then deploy it to a production instance.

Deployment options to consider include:

1. **Use the same workspace and same schema.** Export and then import the application and install it using a different application ID. This approach works well when there are few changes to the underlying objects, but frequent changes to the application functionality.
2. **Use a different workspace and same schema.** Export and then import the application into a different workspace. This is an effective way to prevent a production application from being modified by developers.
3. **Use a different workspace and different schema.** Export and then import the application into a different workspace and install it so that it uses a different schema. This new schema needs to have the database objects required by your application.
4. **Use a different database with all its variations.** Export and then import the application into a different APEX instance and install it using a different workspace, schema, and database.

Whether to Copy the Workspace

Deciding whether to copy an existing workspace is a matter of preference. Keep in mind that the production version must have access to all the appropriate objects. For example, you might want to copy a workspace in the following situations:

- When you want to keep the same application identifiers between development and other environments.
- When the application relies on APEX authentication. Copying the workspace automatically migrates all the required user data.

Whether to Copy the Database

When deciding whether to copy the database, remember that the schema against which the application runs must have access to the same objects as the development instance. The actual name of the schema is unimportant. You can change it during the import process.

About the Application ID

It is not necessary to have matching application IDs for a development version and production version of an application. In fact, as a best practice, never hard code the application ID into your application. Instead, use the application alias (defined on the Edit Application page), or use a built-in substitution string (such as `APP_ID` and `APP_ALIAS`). Using a substitution string is the better approach because it enables you to change the application ID without affecting any application functionality.

See Also:

- Name for information about defining an application alias
- Using Built-in Substitution Strings for information about using `APP_ID` and `APP_ALIAS`

25.3 Exporting and Importing from App Builder

Move a workspace, an application, application components, or related files by creating an export script file and then importing and installing it into another APEX instance.

- Understanding the Export and Import Process
  Move a workspace, application, or related files by creating an export script file and then importing and installing it into another APEX instance.
- Exporting Workspaces, Applications, and App Components
  Export a workspace, application, application components, or related file by creating an export script file.
- Importing Export Files
  After you create Export files, you then import them into the target Oracle APEX instance.
- Managing Export Files from the Export Repository
  Import, install, or delete export files stored in the Export Repository.

See Also:

- Exporting and Importing Using SQLcl in *Oracle APEX Administration Guide*
- Understanding One-Click Remote Application Deployment

25.3.1 Understanding the Export and Import Process

Move a workspace, application, or related files by creating an export script file and then importing and installing it into another APEX instance.

The export process consists of following steps:

1. **Create the Export File**.

   Exporting a workspace, application, page, or application components creates an export file. An **export file** is a readable script file with a `.SQL` extension.

   You can create export files using the Export pages available App Builder or from a command line.

   For details on creating an export file in App Builder, see Exporting Workspaces, Applications, and App Components.

2. **Import and Install the Export File**.

   Import the export file into the target APEX instance and then install it. In App Builder, you can install an export file directly after importing, or install it from the Export Repository.

Alternatively, you can import and install from a command line. As a best practice, Oracle recommends importing an application first and then the related files.

For details on importing and installing an export file in App Builder, see Importing Export Files.

About Exporting, Importing, and Installing from a Command Line

To export and import from a command line, Oracle recommends using Oracle SQL Developer Command Line (SQLcl) as described in Exporting and Importing Using SQLcl in *Oracle APEX Administration Guide*.

> **See Also:**
>
> - Managing Export Files from the Export Repository
> - How to Create a Custom Application

25.3.2 Exporting Workspaces, Applications, and App Components

Export a workspace, application, application components, or related file by creating an export script file.

> **Tip:**
>
> You can also import and export from a command line using Oracle SQL Developer Command Line (SQLcl) release 22.1 and later. See Exporting and Importing Using SQLcl in *Oracle APEX Administration Guide*

- Splitting Export Files into Separate Scripts
  Learn about splitting application, page, and component export files into separate SQL scripts.
- About Exporting Static Files
  Learn about exporting static application and workspace files.
- Exporting a Workspace
  Move or copy a workspace by exporting it.
- Exporting an Application
  Move or copy an application by exporting and then importing it into the target instance.
- Exporting an Application Page
  Move or copy a single application page by exporting it. You can export or page from the Export page or from Page Designer.
- Exporting Application Components
  Create a Component Export to back up a component before editing it, or create an export that functions as a patch to another Oracle APEX instance.
- Exporting Themes
  Move or copy a theme by exporting it.

- Exporting Plug-ins
  Move or copy plug-ins by exporting them.
- Exporting UI Defaults
  Move or copy a UI Defaults by exporting them.
- Exporting Team Development Feedback
  Move or copy user-provided feedback from the current APEX instance.
- Exporting a REST Source Catalog
  Move a REST Source Catalog to another instance on the Export page.
- Exporting Script Files from the Script Repository
  Move scripts from your current Script Repository by exporting them.

> **See Also:**
>
> - Understanding the Export and Import Process
> - How to Create a Custom Application

25.3.2.1 Splitting Export Files into Separate Scripts

Learn about splitting application, page, and component export files into separate SQL scripts.

Splitting application, page, and component exports into separate files can be very useful if you are managing files using a version control system such as Apache Subversion (SVN) or Git. This approach enables you to check in files to the repository as you make changes. For example, suppose a developer modifies Page 1 and a list of values called Categories. If you export the application as separate files, you can commit only the two files with changes.

Another benefit of splitting exports into separate files is that it enables you to take an incremental approach to deployment. Rather than having to re-install an entire application for a minor change (for example, fixing a typo on a label), only the file containing the fix needs to be run on the destination system.

You can split application, page, and component exports into separate files as follows:

- **Applications** - Select the **Split into Multiple Files** flag to export the application as a ZIP archive containing separate files for each page, shared component, and so on. You can unzip the Zip file on your local file system and then commit updated files to your source code repository.
- **Pages and other Components** - Select the **Export as Zip** flag to export the selected components as a ZIP archive containing separate SQL files for each component.

> **See Also:**
>
> Exporting and Importing from a Command Line in *Oracle APEX Administration Guide*

Export Directory Structure

Exporting an application, page, or components into separate scripts creates a directory structure organized by component type.

> **See Also:**
>
> Directory Structure When Splitting Export Files into Separate Scripts

Export File Names

Export files use the following naming conventions:

- **Application Export** - Named using the application ID. For example, an application with the ID 326 would be named: `f326.sql` or `f326.zip`.
- **Page Export** - Named after the application ID and page number. For example, exporting page 4 in an application with the ID 326 would be named: `f326_page_4.sql` or `f326_page_4.zip`.
- **Component Export** - If you select multiple components, the name uses the application ID and components. For, example, multiple components in an application with the ID 326 would be named: `f326_components.sql` or `f326_components.zip`.

  Component file names are based on the name specified in App Builder. In general, special characters are removed and spaces are replaced with underscores.

  If you rename a component in App Builder, the next time you create a Component Export, a new file name is used. For example, if you have breadcrumb named `Menu` and you create a Component Export, the file would be named `menu.sql`. If you rename that breadcrumb `Main Menu`, the new Component Export is named `main_menu.sql`. If you are using a version control system, you are responsible for applying the proper commands to rename or move the original file to maintain the history.

Generated Installation Scripts

Exporting create a number of installation scripts. Depending upon the type of export, the installation scripts are named one of the following:

- **Application Export** - `install.sql`
- **Page Export** - `install_page.sql`
- **Component Export** - `install_component.sql`

In addition to installation files, each export includes two files which are useful when developing custom patch scripts:

- `set_environment.sql`
- `end_environment.sql`

These files contain internal code that help improve performance. These scripts set internal variables and perform security checks at one time for the entire import. Without these scripts, the internal variables check will be executed for each file you run.

A sample custom installation file might resemble following:

```
@@application/set_environment.sql
@@application/pages/page_00001.sql
@@application/pages/page_00023.sql
@@application/pages/page_00095.sql
@@application/end_environment.sq
```

25.3.2.2 About Exporting Static Files

Learn about exporting static application and workspace files.

Static application and workspace files include images, CSS files, and other files which must be managed independently. Export static application and workspace files by either:

- Creating a ZIP file in Shared Components. See Downloading Static Application Files.
- Creating installation scripts using component export. See Exporting Application Components.

25.3.2.3 Exporting a Workspace

Move or copy a workspace by exporting it.

Exporting a workspace creates a SQL script file which includes user groups, users, Team Development data, and workspace artifacts including SQL Scripts, SQL Command History, saved SQL, user preferences, developer login history, email logs, and UI Defaults. Workspace exports do not include applications.

Oracle does not recommend exporting an entire workspace unless you want to migrate workspace users or replicate Shared Component subscriptions in the target instance.

To export a workspace:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**.

2. Click the **Workspace** tab.
3. Under **Export Workspace**:
   a. **Include Team Development** - Set to **On** to include Team Development data. Only select **Off** if the target instance for this workspace is a runtime only instance.
   b. **Include Extension Links** - Enable this option to include Extension Links data in your workspace export file.
   c. **Export Type** - In most cases, do not change the default value **Minimal**. Only select **Full** if you want to replicate this workspace and all workspace artifacts including SQL Scripts, SQL Command History, saved SQL, user preferences, developer login history, email logs, and UI Defaults.
   d. **File Character Set** - Indicates the character set in which the export file will be encoded.
4. Click **Export**.
5. Import the Workspace Export in Administration Services.

> **See Also:**
>
> Importing a Workspace in Administration Services in *Oracle APEX Administration Guide*

25.3.2.4 Exporting an Application

Move or copy an application by exporting and then importing it into the target instance.

Exporting an application in Oracle APEX generates a readable script file with a `.SQL` extension. An application export includes the application definition, including templates, pages, regions, items, buttons, and so on.

As a best practice, Oracle recommends you export complete applications rather than individual components. If you wish to export an application and there are components such as pages that are not yet ready for testing, Oracle recommends you use Build Options to include or exclude various application components.

> **Tip:**
>
> In addition to exporting the actual application, you may also need to export other related static files such as cascading style sheets, images, and script files. See Managing Static Application Files.

To export an application:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**.
2. Click the **Application** tab.
3. Under **Export Application**:
   a. **Application** - Select the the application to export.
   b. **Readable Format** - Specify whether the export file should contain a human-readable (YAML) version of the application meta-data. Use the YAML version to identify and review differences between versions of an application when checked into a Version Control System.
      - **On** - Include a human-readable version of the application.
      - **Off** - Exclude the human-readable version.

> **Tip:**
>
> Enabling **Readable Format** automaticlly enables the **Original IDs** attribute. For more information, see the **Original IDs** discussion that follows.

c. **Build Status Override** - Select the build status of the exported application. Options include:

- **Run Application Only** - Users can only run an application. Selecting this option is an effective way to protect an application from modifications from other developers.

  If you select **Run Application Only**, you cannot set the argument `p_trace` to `Yes`. The only way to change this setting after you import the application is to log in to APEX Administration Services

- **Run and Build Application** - Developers and users can both run and edit an application.

> **Note:**
>
> The **Build Status Override** setting can be changed when importing using the Web interface in App Builder but not when using the command line interface.

d. **Supporting Object Definitions** - Specify whether to export supporting object definitions with your application. Supporting object definitions include all configuration options and scripts and enable an application export to include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file.

   Options include:

- **Yes** - Includes supporting object definitions in the application export. Does not automatically load supporting objects when invoked from a command line.
- **No** - Does not include the supporting object definitions in the export.
- **Yes and Install on Import Automatically** - Includes supporting object definitions and a call to install supporting objects in the application export. This option is valid only for command line installs. When application imports from the command line, it automatically installs or upgrades the supporting objects.

e. **Split into Multiple Files** - Options include:

   Options include:

- **On** - Exports the application as a ZIP archive containing separate files for each page, shared component, and so on.
- **Off** - Exports as a single SQL file containing the entire application.

> **Tip:**
>
> To learn more about the Zip archive, see Splitting Export Files into Separate Scripts.

f. **Public Reports** - Specify whether to include public reports with your application. Public reports are the public shared report settings saved by users from an interactive report or interactive grid.

   Options include:

   - **On** - Include public reports in the application export.
   - **Off** - Do not include public reports in the application export.

g. **Private Reports** - Specify whether to include Private reports with your application. Private reports are the private customized settings saved by users from an interactive report or interactive grid.

   Options include:

   - **On** - Include public reports in the application export.
   - **Off** - Do not include public reports in the application export.

h. **Report Subscriptions** - Specify whether to include interactive report or interactive grid subscription settings with your application. This will be limited to the unexpired subscription settings of all users for all interactive reports or interactive grids in the application.

   Options include:

   - **On** - Include report subscriptions for the reports you are exporting in the application export.
   - **Off** - Do not include report subscriptions for the reports in the application export.

i. **Developer Comments** - Specify whether to include Developer Comments in your application export. Options include:

   - **On** - Include Developer Comments in the application export.
   - **Off** - Do not include Developer Comments in the application export.

j. **Translations** - Specify whether to include translations with your application export. Translation Text Messages and Dynamic Translations are always included in the application export. Options include:

   - **On** - Include the translation mappings and all text from the translation repository.
   - **Off** - Do not include translation mappings or the translation repository.

k. **Debugging** - Specify if debugging is enabled or disabled. Options include:

   - **Yes** - Exports the application with debugging enabled.
   - **No** - Exports the application with debugging disabled.

   Enabling debugging is useful for applications under development. As a best practice, Oracle recommends turn off debugging for production applications to prevent users from viewing application logic.

l. **Original IDs** - Specify whether the export file should contain the application component IDs as of now or as of the last import of this application. Options include:

   - **On** - Include the IDs as of the last import.
   - **Off** - Include only the current IDs of the components.

   Examples of when enabling **Original IDs** can be beneficial:

   - **Readable Format** - Enabling **Original IDs** simplifies comparisons between different versions of app components and pages. It also minimizes changes to version controlled export files.

- **Shared Component Subscriptions** - Enabling **Original IDs** ensure subscribed components stay linked to a source app when it is replaced with an updated version.

m. **Owner Override** - By default, the application is owned by the owner identified in the application attribute owner. Use this attribute to import this application as a different schema owner then you are exporting.

n. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:

- **None** - (Default) No audit information is included.
- **Names and Dates** - Include names and dates in all audit information.
- **Dates Only** - Only include date information and exclude user details.

o. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases can have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.

p. **File Character Set** - Indicates the character set in which the export file will be encoded. When importing this file you will need to specify the same character set.

4. Under Recent Exports to Export Repository, review any recent exports.
5. Click **Export**.
6. Import the Application Export into the target instance.

> **See Also:**
>
> - Importing an Application Export
> - Changing Application Build Status in Administration Services in *Oracle APEX Administration Guide*

25.3.2.5 Exporting an Application Page

Move or copy a single application page by exporting it. You can export or page from the Export page or from Page Designer.

> **WARNING:**
>
> Exporting and importing a single page to a different system is an advanced deployment technique with the following restrictions:
>
> - Both systems have to have the same application ID and workspace ID.
> - Both systems have to have the same IDs for Shared Components.
> - If the page references new or updated Shared Components, those have to be exported and imported first.
>
> If these restrictions are not fulfilled, the import will fail.

- Exporting an Application Page from the Export Page
  Move or copy a single application page by exporting it from the Export Page.
- Exporting an Application Page from Page Designer
  Move or copy a single application page by exporting it from Page Designer.

25.3.2.5.1 Exporting an Application Page from the Export Page

Move or copy a single application page by exporting it from the Export Page.

To export a page from the Export page:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**.
2. Click the **Page** tab.
3. On the Export Page, configure the following:
   a. **Application** - Select the application that contains the page to be exported.
   b. **Page** - Select the page you wish to export. You can select only one page for export at a time.
   c. **Include Installable SQL File** - Specify whether the export file should include the installable SQL file for the page. Options include:
      - **On** - Includes the installable SQL file.
      - **Off** - Excludes the installable SQL file.
   d. **Include Readable Format** - Specify whether the export file should include the human-readable (YAML) version of the page meta-data. Use the YAML version to identify and review differences between versions of a page when checked into a Version Control System. Options include:

- **On** - Include the human-readable file.
- **Off** - Exclude the human-readable file.

e. **Split into Multiple File** - Select this option to export separate SQL files for the page, shared components, and so on. Options include:

- **On** - Export as a ZIP archive with separate SQL files.

  To learn more about the Zip archive, see Splitting Export Files into Separate Scripts.

- **Off** - Export as a single SQL file for the page.

f. **Original IDs** - Specify whether the export file should contain the application component IDs as of now or as of the last import of this application. Options include:

- **On** - Include the IDs as of the last import.
- **Off** - Include the current IDs of the components.

g. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:

- **None** - (Default) No audit information is included.
- **Names and Dates** - Include names and dates in all audit information.
- **Dates Only** - Only include date information and exclude user details.

h. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

   This utility uses the `DBMS_FLASHBACK` package. Because the timestamps to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the start up parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo table space. While two databases may have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo table space, forcing older data to be archived.

i. **File Character Set** - Identifies the character set in which the export file will be encoded.

4. Click **Export** .
5. Import the Page Export into the target instance.

> **See Also:**
>
> Importing an Application Export

25.3.2.5.2 Exporting an Application Page from Page Designer

Move or copy a single application page by exporting it from Page Designer.

To export a page from Page Designer:

1. View the page to be exported in Page Designer:

a. On the Workspace home page, click the **App Builder** icon.

b. Select an application.

c. Select a page.

   Page Designer appears.

2. On the Page Developer toolbar, click the **Utilities** menu (

) and then select **Export**.

3. On Export Page, configure the following:

   a. **Include Installable SQL File** - Specify whether the export file should include the installable SQL file for the page. Options include:

      - **On** - Includes the installable SQL file.
      - **Off** - Excludes the installable SQL file.

   b. **Include Readable Format** - Specify whether the export file should include the human-readable (YAML) version of the page meta-data. Use the YAML version to identify and review differences between versions of a page when checked into a Version Control System. Options include:

      - **On** - Include the human-readable file.
      - **Off** - Exclude the human-readable file.

   c. **Split into Multiple File** - Select this option to export separate SQL files for the page, shared components, and so on. Options include:

      - **On** - Export as a ZIP archive with separate SQL files.

        To learn more about the Zip archive, see Splitting Export Files into Separate Scripts.

      - **Off** - Export as a single SQL file for the page.

   d. **Original IDs** - Specify whether the export file should contain the application component IDs as of now or as of the last import of this application. Options include:

      - **On** - Include the IDs as of the last import.
      - **Off** - Include the current IDs of the components.

   e. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:

      - **None** - (Default) No audit information is included.
      - **Names and Dates** - Include names and dates in all audit information.
      - **Dates Only** - Only include date information and exclude user details.

   f. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

      This utility uses the `DBMS_FLASHBACK` package. Because the timestamps to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the start up parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo table space. While two databases may have the same `UNDO_RETENTION`

parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo table space, forcing older data to be archived.

g. **File Character Set** - Identifies the character set in which the export file will be encoded.

4. Click **Export** .
5. Import the Page Export into the target instance.

> **See Also:**
>
> Importing an Application Export

25.3.2.6 Exporting Application Components

Create a Component Export to back up a component before editing it, or create an export that functions as a patch to another Oracle APEX instance.

> **Tip:**
>
> If you wish to export an application and some components, such as pages, that are not yet ready for testing, Oracle recommends you use Build Options to include or exclude various application components.

To export shared components or page components:

1. Navigate to the Component Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/ Import**, and then **Export**.

2. Click the **Components** tab.
3. Under **Export Components**:
   a. **Application** - Select the application that contains the components to be exported.
   b. **Export Format** - Select a page format. Options include:
      - **SQL** - Export as a single SQL file containing all the selected components.
      - **ZIP** - Export the selected components as a ZIP archive containing separate SQL files for each components.

To learn more about the Zip archive, see Splitting Export Files into Separate Scripts.

c. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:

- **None** - (Default) No audit information is included.
- **Names and Dates** - Include names and dates in all audit information.
- **Dates Only** - Only include date information and exclude user details.

d. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

This utility uses the `DBMS_FLASHBACK` package. Because the timestamps to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the start up parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo table space. While two databases may have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo table space, forcing older data to be archived.

e. **File Character Set** - Indicates the character set in which the export file will be encoded. When importing this file you will need to specify the same character set.

4. Add Application Components:
   a. Click the **Application Components** tab.
   b. Use the navigation bar to search for components.
   c. Select the appropriate components and click **Add to Export**.

   The selected components display in the **Components to Export** region.
5. Add Build Options:
   a. Click the **Build Options tab**.
   b. Use the navigation bar to search for build options.
   c. Select the appropriate build options and click **Add to Export**.

   The selected build options display in the **Components to Export** region.
6. Review your selections.
7. To remove selected components from the export, deselect the component in the **Components to Export** region.
8. Click **Export**.
9. Import the Component Export into the target instance.

See Also:

- Importing an Application Export
- Managing Shared Components
- Exporting Build Options or Build Option Status

25.3.2.7 Exporting Themes

Move or copy a theme by exporting it.

Exporting a theme to create a a readable script file with a .SQL extension.

To export an application theme:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, clicking **Export/Import**, and selecting then **Export**.

2. Click the **Themes** tab.
3. Under Themes:
   a. **Application** - Select an application.
   b. **Export Theme** - A **theme** a collection of templates which define the look and feel of the application. Select the theme to export.
   c. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:
      - **None** - (Default) No audit information is included.
      - **Names and Dates** - Include names and dates in all audit information.
      - **Dates Only** - Only include date information and exclude user details.
   d. **File Character Set** - Identifies the character set in which the export file will be encoded.
4. Click **Export** .
5. Import the Theme Export into the target instance.

> **See Also:**
>
> - Importing Themes
> - Using Themes

25.3.2.8 Exporting Plug-ins

Move or copy plug-ins by exporting them.

Use a Plug-in Export to export plug-ins from one Oracle APEX development instance to a file.

To export a plug-in:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**. Alternatively, navigate to the Plug-ins page and select **Export Plug-in** from the **Tasks** list.

2. Click **Plug-ins** tab.
3. Under **Export Plug-in**:
   a. **Application** - Select an application
   b. **Plug-in** - Select the plug-in to export.
   c. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:
      - **None** - (Default) No audit information is included.
      - **Names and Dates** - Include names and dates in all audit information.
      - **Dates Only** - Only include date information and exclude user details.
   d. **File Character Set** - Indicates the character set in which the export file will be encoded.
4. Click **Export** .
5. Import the Plug-in Export into the target instance.

> **See Also:**
>
> Importing Plug-ins

25.3.2.9 Exporting UI Defaults

Move or copy a UI Defaults by exporting them.

- About Exporting UI Defaults
  Learn about exporting UI Defaults.
- Exporting UI Defaults from the Export Page
  Export UI Defaults from the Export page.
- Exporting UI Defaults from the UI Defaults Page
  Export UI Defaults from the UI Defaults page.

25.3.2.9.1 About Exporting UI Defaults

Learn about exporting UI Defaults.

Exporting UI Defaults is useful when you plan to develop on a target system. Developers use UI Defaults to populate initial values for region and item properties providing consistency across multiple pages in an application or multiple applications. UI Defaults are divided into two categories:

- The **Table Dictionary** consists is specific to tables and columns within a selected schema. These defaults are used over those defined in the Attribute Dictionary.
- The **Attribute Dictionary** contains a set of attributes about a column that is used in creating forms and reports. The definitions are matched by column name and a particular definition can be shared among several columns by using synonyms.

When you export UI Defaults, all UI Defaults for the selected dictionary and schema are exported to a single script. When prompted, save this file to your hard drive. The file contains an API call to create table hints by making calls to the application PL/SQL API. You can use this file to import UI Defaults to another database and Oracle APEX instance.

> **See Also:**
>
> - Importing UI Defaults
> - Managing UI Defaults in *Oracle APEX SQL Workshop Guide*

25.3.2.9.2 Exporting UI Defaults from the Export Page

Export UI Defaults from the Export page.

To export UI Defaults:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

      The Export page appears.

      > **Tip:**
      >
      > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**.
2. Click **UI Defaults** tab.
3. Under **Table Dictionary**:
   a. **Schema** - Select the name of the schema to be exported.
   b. **File Character Set** - Displays the character set in which the export file will be encoded.

4. **Attribute Dictionary**, **Count of Columns** - Displays the number of entries in the Attribute Dictionary
5. Click **Export**.
6. Import the UI Defaults Export into the target instance.

> **See Also:**
>
> Importing UI Defaults

25.3.2.9.3 Exporting UI Defaults from the UI Defaults Page

Export UI Defaults from the UI Defaults page.

To export UI Defaults from the UI Defaults page:

1. On the Workspace home page, click the **SQL Workshop** icon.
2. Click **Utilities**.
3. Click **UI Defaults**.
4. Review the UI Defaults page.
   - The **Table Dictionary** is specific to tables and columns within a selected schema. These defaults are used over any defined within the Attribute Dictionary.
   - The **Attribute Dictionary** contains a set of attributes about a column and are used in creating forms and reports.
5. To export the **Table Dictionary**:
   a. Click **Table Dictionary**.
   b. Under **Tasks**, click **Export**.
   c. **File Format** - Select a file format.
      - **DOS** - Select **DOS** to have the lines in the resulting file terminated by carriage returns and line feeds..
      - **UNIX** - Select **UNIX** to have the lines in the resulting file contain terminated by line feeds..
   d. Click **Export**.
6. Import the UI Defaults Export into the target instance.

> **See Also:**
>
> Importing UI Defaults

25.3.2.10 Exporting Team Development Feedback

Move or copy user-provided feedback from the current APEX instance.

To export Team Development Feedback:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

   The Export page appears.

   > **Tip:**
   >
   > You can also access the Export page by selecting the application, **Export/Import**, and then **Export**.

2. Click the **Feedback** tab.
3. Under **Export Feedback**:
   a. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:
      - **None** - (Default) No audit information is included.
      - **Names and Dates** - Include names and dates in all audit information.
      - **Dates Only** - Only include date information and exclude user details.
   b. **Changes Since** - Select the date from which to export feedback.
4. Click **Export** .
5. Import the Feedback Export into the target instance.

> **See Also:**
>
> - Importing Team Development Feedback
> - Managing Feedback

25.3.2.11 Exporting a REST Source Catalog

Move a REST Source Catalog to another instance on the Export page.

To export a REST Source Catalog:

1. Navigate to the Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Workspace Utilities**.
   c. Click **Export**.

   The Export page appears.

> **Tip:**
>
> You can also access the Export page by navigating to the REST Source Catalogs page and selecting **Export Catalog** from the Tasks list on the right side of the page.

2. Click the **Catalog** tab.
3. Under **Export REST Source Catalog**:
   a. **REST Source Catalog** - Select the catalog to export.
   b. **Audit Information** - Specify the audit information to include in the export file. Audit information includes the Created By User, Created On Date, Changed By User and Changed On Date. Options include:
      - **None** - (Default) No audit information is included.
      - **Names and Dates** - Include names and dates in all audit information.
      - **Dates Only** - Only include date information and exclude user details.
   c. **File Character Set** - Identifies the character set in which the export file will be encoded.
4. Click **Export** .
5. Next, import the REST Source Catalog Export into the target instance. See Importing a REST Source Catalog.

> **See Also:**
>
> Managing REST Source Catalogs

25.3.2.12 Exporting Script Files from the Script Repository

Move scripts from your current Script Repository by exporting them.

> **See Also:**
>
> Exporting and Importing SQL Scripts in *Oracle APEX SQL Workshop Guide*

25.3.3 Importing Export Files

After you create Export files, you then import them into the target Oracle APEX instance.

After you import the files, you have the option to immediately install them. You can also install them at a later time by navigating to the Export Repository.

Note that you cannot import an application created in a specific release into an earlier version. As a general rule, always import the application first and then the related files.

> **Tip:**
>
> To simplify the steps needed to deploy an application, Oracle recommends creating a custom application. See How to Create a Custom Application.

- Importing an Application Export
  Import an Application Export on the Import page.
- Importing a Page
  Import page on the Import page.
- Importing a Component
  Import a Component on the Import page.
- Reimporting an Application Containing Background Processes
  Learn about the workflow when reimporting an application with background processes.
- About Importing Applications and Application Groups
  Application groups are exported with an application, Application Exports include a group ID, name, and comments.
- Importing Themes
  Import a Theme Export on the Import page.
- Importing Plug-ins
  Import the Plug-in Export on the Import page.
- Importing UI Defaults
  Import the UI Defaults Export on the Import page.
- Importing Team Development Feedback
  Import a Team Development Feedback Export on the Import page.
- Importing a REST Source Catalog
  Import a REST Source Catalog Export on the Import page.

> **See Also:**
>
> - Understanding the Export and Import Process
> - Managing Export Files from the Export Repository
> - Exporting and Importing from a Command Line in *Oracle APEX Administration Guide*

25.3.3.1 Importing an Application Export

Import an Application Export on the Import page.

> **Tip:**
>
> You can also import an export file from the Export Repository. Navigate to the Export Repository and click the **Import File** and then go to step 3.

To import an application:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Indicate the type of export file to be imported. Select **Application, Page or Component Export**.
   c. **File Character Set** - Verify that the File Character Set is correct.
   d. Click **Next** to import the file.

   Once you import a file, you have the option to install it. You can also install the file at a later time by navigating to the Export Repository.
4. On Install Application:
   a. **Parsing Schema** - Select a schema. After the installation finishes successfully, this will be used to set the owner of the application.
   b. **Build Status** - Set the build status of the application. Options include:
      - **Run Application Only** - Users can only run an application.

        Selecting **Run Application Only** is an effective way to protect an application from modifications from other developers.
      - **Run and Build Application** - Users can run an application and developers can both run and edit an application.

      > **Tip:**
      >
      > If you select **Run Application Only**, you can only change this setting after you import the application by signing in to Administration Services.

   c. **Install As Application** - Select how to install this application. You must specify an application ID that is unique across all workspaces. Options include:
      - **Auto Assign New Application ID** - A new application ID will be created for you.
      - **Reuse Application ID XX From Imported Application** - The existing application ID will be used. When you install an application with the same ID as an existing application in the current workspace, the existing application is deleted and then replaced by the new application.
      - **Change Application ID** - Specify a new application ID.

      You must specify an application ID that is unique across all APEX workspaces.
   d. Click **Install Application**.

      The file is installed. What appears next, depends upon the type of export. For example, the export could include Credentials, Remote Servers, or supporting objects.
5. If Credentials or Remote Servers are detected, follow the on-screen instructions to set up the Credentials or Remote Servers for your workspace.
6. If supporting objects are detected, the wizard guides you through the installation process. Click **Install Supporting Objects** and follow the on-screen instructions.

7. On the confirmation page, select the appropriate button:
   - **Upgrade Application**
   - **Edit Application**
   - **Run Application**

> **See Also:**
>
> Changing Application Build Status Set During Deployment in *Oracle APEX Administration Guide*

25.3.3.2 Importing a Page

Import page on the Import page.

> **Tip:**
>
> You can also import an export file from the Export Repository. Navigate to the Export Repository and click the **Import File** and then go to step 3.

To import a page:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Indicate the type of export file to be imported. Select **Application, Page or Component Export**.
   c. **File Character Set** - Verify that the File Character Set is correct.
   d. Click **Next** to import the file.

      Once you import a file, you have the option to install it. You can also install the file at a later time by navigating to the Export Repository.
4. On the Install page:
   a. **Install Into Application** - Select the target application. You can install the Page into the same application or into a different application.
   b. Click **Install Page**.

      > **Tip:**
      >
      > When you install the page age into the current application, the Confirm Replace Page appears. To have the new page overwrite the existing page click **Replace Page**.

5. On the confirmation page, select the appropriate button:
   - **Edit Application**
   - **Edit Page**
   - **Run Page**

> **Note:**
>
> If you attempt to import a page that includes a shared component or a table not available in the target workspace, an error will occur because the import process will reference a non-existent component or table. To resolve this, ensure that the necessary components or tables are imported first.

> **See Also:**
>
> Changing Application Build Status Set During Deployment in *Oracle APEX Administration Guide*

25.3.3.3 Importing a Component

Import a Component on the Import page.

> **Tip:**
>
> You can also import an export file from the Export Repository. Navigate to the Export Repository and click **Import File** and then go to step 3.

To import a Component:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Indicate the type of export file to be imported. Select **Application, Page or Component Export**.
   c. **File Character Set** - Verify that the File Character Set is correct.
   d. Click **Next** to import the file.

   The Install Application Components Wizard appears.

   You can install the file now or install it at a later time by navigating to the Export Repository.

To install the file now:

4. For **Install into Application**, select the target application.

You can install a component into the application you exported it from or into a different application.

> **Tip:**
>
> When you install a component into the application it was exported from, the new Component overwrites the existing component with the same component name.

5. Click **Next**.
6. Click **Replace Application Components**.

> **Note:**
>
> Importing templates to different applications or workspaces is not currently supported.

25.3.3.4 Reimporting an Application Containing Background Processes

Learn about the workflow when reimporting an application with background processes.

Reimporting (or redeploying) an application which has background processes has a slightly different workflow compared to an initial Application Export import.

> **Tip:**
>
> You can also import an export file from the Export Repository. Navigate to the Export Repository and click the **Import File** and then go to step 3

To reimport an Application Export with background processes:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Indicate the type of export file to be imported. Select **Application, Page or Component Export**.
   c. **File Character Set** - Verify that the File Character Set is correct.
   d. Click **Next** to import the file.

   The Install Application Wizard appears.

   You can install the export file now or install it at a later time by navigating to the Export Repository.
4. To install the export file:

   On **Install Application Wizard**:

a. **Parsing Schema** - Select a schema. After the installation finishes successfully, this will be used to set the owner of the application.

b. **Build Status** - Set the build status of the application. Options include:

   - **Run and Build Application** - Users can run an application and developers can both run and edit an application.
   - **Run Application Only** - Users can only run an application.

     Selecting **Run Application Only** is an effective way to protect an application from modifications from other developers.

> **Note:**
>
> If you select **Run Application Only**, you can only change this setting after you import the application by signing in to Administration Services.

c. **Install as Application** - Since you are importing an application export again, select **Change Application ID**.

d. **New Application** - Enter the ID of the application from your first import.

e. Click **Install Application**.

A variation on the following message appears:

```
The application to be replaced has page processes with background executions
enabled. No process execution is running right now (9:45:24 PM).
```

- `Use the Disable Background Execution button to prevent new executions to be kicked off (by setting the Maximum Scheduler Jobs attribute to zero) and to return to this page.`
- `Use the Replace Application button to install the imported application now. Running background executions will be terminated; waiting ones will be deleted.`

5. Select the appropriate button:
   - **Disable Background Execution** - Prevents new executions from starting.
   - **Replace Application** - Install the imported application now. No background executions will run and any waiting executions will be deleted.
6. **Install Supporting Objects** - To continue and install supporting objects, enable **Install Supporting Objects** and click **Next**.
7. If Credentials or Remote Servers are detected, follow the on-screen instructions to set up the Credentials or Remote Servers for your workspace.
8. If supporting objects are detected, the wizard guides you through the installation process. Click **Install Supporting Objects** and follow the on-screen instructions.
9. On the confirmation page, select the appropriate button:
   - **Install Summary**
   - **Edit Application**
   - **Run Application**

> **See Also:**
>
> Changing Application Build Status Set During Deployment in *Oracle APEX Administration Guide*

25.3.3.5 About Importing Applications and Application Groups

Application groups are exported with an application, Application Exports include a group ID, name, and comments.

When importing an application, the application import follows these rules:

- On import, an application uses an existing group if the ID matches.
- If no application group exists with the same ID on the target system, then application import looks for another group ID with the same name. If a name match is found, the application is imported with the ID corresponding to this application group name.
- If no ID or name matches existing application groups on the target system then a new application group is created. The new application group uses the application group ID if that ID is not used in that Oracle APEX instance. If the group ID is being used, then a new ID is generated.

25.3.3.6 Importing Themes

Import a Theme Export on the Import page.

To import a Theme Export:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Select **Theme Export**.
   c. **File Character Set** - Verify that File Character Set is correct.
   d. Click **Next**.

      Once you import a file, you have the option to install it. You can also install this file at a later time by navigating to the Export Repository.
4. To install the file, click **Next**.

   The Install Application wizard appears.
5. Click **Install Theme**.

See Also:

- Using Themes
- Exporting Themes

25.3.3.7 Importing Plug-ins

Import the Plug-in Export on the Import page.

To import a Plug-in Export:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Select **Plug-in**.
   c. **File Character Set** - Verify that File Character Set is correct.
   d. Click **Next**.

      Once you import a file, you have the option to install it. You can also install this file at a later time by navigating to the Export Repository.
4. To install an imported file, click **Next**.
5. On Install Plug-in:
   a. **Install Into Application** - Select the target application. You can install the plug-in into the same application or into a different application.

      Tip:

      When you install a plug-in into the current application, the new plug-in will overwrite an existing plug-in having the same plug-in name. If the installation succeeds, the installation of the plug-in becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.
   b. Click **Install Plug-in**.

See Also:

Exporting Plug-ins

25.3.3.8 Importing UI Defaults

Import the UI Defaults Export on the Import page.

You can install a UI Defaults Table Dictionary export into any schema provisioned in the workspace. When you install a Table Dictionary export, the old defaults will be overwritten. UI Defaults Attribute Dictionary imports replace any matching attribute definitions that are found but will not remove any existing attribute definitions that are not contained in the export. If the installation succeeds, the installation of the UI Defaults becomes permanent. If any errors are encountered, the actions are rolled back, resulting in no permanent changes.

To import a UI Defaults Export:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.
   b. **File Type** - Select **UI Defaults**.
   c. **File Character Set** - Verify that File Character Set is correct.
   d. Click **Next**.

      Once you import a file, you have the option to install it. You can also install this file at a later time by navigating to the Export Repository.
4. To install an imported file, click **Next**.
5. Click **Install UI Defaults**.

> **See Also:**
>
> - Managing UI Defaults in *Oracle APEX SQL Workshop Guide*
> - Exporting UI Defaults

25.3.3.9 Importing Team Development Feedback

Import a Team Development Feedback Export on the Import page.

To import a Team Development Feedback Export:

1. Sign in to the APEX instance.
2. Navigate to the Import page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. On the App Builder home page, click **Import**.
3. On Import:
   a. Drag and drop, or navigate to the file.

b. **File Type** - Select **Team Development Feedback**.

c. **File Character Set** - Verify that File Character Set is correct.

d. Click **Next**.

Once you import a file, you have the option to install it. You can also install this file at a later time by navigating to the Export Repository.

4. To install an imported file, click **Next**.
5. Click **Import Feedback**.

> **See Also:**
>
> - Exporting Team Development Feedback
> - Managing Feedback

25.3.3.10 Importing a REST Source Catalog

Import a REST Source Catalog Export on the Import page.

To import a REST Source Catalog Export:

1. Sign in to the APEX instance.
2. Navigate to the Import page:

   a. On the Workspace home page, click the **App Builder** icon.

   b. On the App Builder home page, click **Import**.

   > **Tip:**
   >
   > You can also access the Import page from the REST Source Catalogs page by selecting **Import Catalog** from the Tasks list on the right side of the page.

3. On Import:

   a. Drag and drop or navigate to the file.

   b. **File Type** - Select **REST Source Catalog**.

   c. **File Character Set** - Verify that File Character Set is correct.

   d. Click **Next**.

   Once you import a file, you have the option to install it. You can also install this file at a later time by navigating to the Export Repository.

4. To install an imported file, click **Next**.
5. Click **Import REST Catalog**.

See Also:

- Exporting a REST Source Catalog
- Managing REST Source Catalogs

25.3.4 Managing Export Files from the Export Repository

Import, install, or delete export files stored in the Export Repository.

Tip:

You can also import an export file from the Export Repository. Navigate to the Export Repository and click the **Import File** and then go to step 3 in Importing an Application Export.

- About Installing Export Files
  When you import Export Files into a target Oracle APEX instance, the files are stored in the Export Repository. From the Export Repository, you can import, install, or delete Export Files.
- Installing an Application Export from the Export Repository
  If you did not install your Application Export after importing, you can install it in the Export Repository.
- Installing Other Files from the Export Repository
  If you did not install your Export Files after importing, you can install them in the Export Repository.
- Deleting Files from the Export Repository
  Delete export files from the Export Repository.

See Also:

- Understanding the Export and Import Process
- How to Create a Custom Application
- Exporting and Importing from a Command Line in *Oracle APEX Administration Guide*

25.3.4.1 About Installing Export Files

When you import Export Files into a target Oracle APEX instance, the files are stored in the Export Repository. From the Export Repository, you can import, install, or delete Export Files.

After you create an Export File, you can install it in the following ways:

- **After Importing**:

  After you import the Export Files, you can install them by clicking the **Next** button.

- **From the Export Repository**:

  Import the Export Files and then install the files from the Export Repository.
- **From Oracle SQL Developer Command Line (SQLcl)**:
  Install the export files from SQLcl.

> **Tip:**
>
> You can simplify the steps needed to deploy an application by creating a custom application.

> **See Also:**
>
> - How to Create a Custom Application
> - Exporting Workspaces, Applications, and App Components
> - Importing Export Files
> - Installing Exported Applications in a Runtime Environment in *Oracle APEX Administration Guide*

25.3.4.2 Installing an Application Export from the Export Repository

If you did not install your Application Export after importing, you can install it in the Export Repository.

To install an Application Export from the Export Repository:

1. Navigate to the Export Repository:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Utilities**.
   d. Click **Export Repository**.
2. Find the application export and click **Install** in the Action column.

   The Install Application Wizard appears.
3. To install the file:
   a. **Parsing Schema** - Select a schema. After the installation finishes successfully, this will be used to set the owner of the application.
   b. **Build Status** - Set the build status of the application. Options include:
      - **Run Application Only** - Users can only run an application.

        Selecting **Run Application Only** is an effective way to protect an application from modifications from other developers.
      - **Run and Build Application** - Users can run an application and developers can both run and edit an application.

> **Tip:**
>
> If you select **Run Application Only**, you can only change this setting after you import the application by signing in to Administration Services.

c. **Install As Application** - Options include:

- **Auto Assign New Application ID** - A new application ID will be created for you.
- **Reuse Application ID XX From Imported Application** - The existing application ID will be used. When you install an application with the same ID as an existing application in the current workspace, the existing application is deleted and then replaced by the new application.
- **Change Application ID** - Specify a new application ID.

You must specify an application ID that is unique across all APEX workspaces.

d. Click **Install Application**.

4. If Credentials or Remote Servers are detected, follow the on-screen instructions to set up the Credentials or Remote Servers for your workspace.
5. If supporting objects are detected, the wizard guides you through the installation process. Click **Install Supporting Objects** and follow the on-screen instructions.
6. On the confirmation page, select the appropriate button:
   - **Install Summary**
   - **Edit Application**
   - **Run Application**

25.3.4.3 Installing Other Files from the Export Repository

If you did not install your Export Files after importing, you can install them in the Export Repository.

To install other Export Files stored in the Export Repository:

1. Navigate to the Export Repository:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. On the Application home page, click **Utilities**.
   d. Click **Export Repository**.
2. Select the file to be installed and click **Install** in the Action column.
3. Follow the on-screen instructions. The steps differ slightly depending upon the export type.

25.3.4.4 Deleting Files from the Export Repository

Delete export files from the Export Repository.

To delete export files stored in the Export Repository:

1. Navigate to the Export Repository:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

c. On the Application home page, click **Utilities**.

d. Click **Export Repository**.

2. Select the file to be deleted and click **Delete Checked**.

25.4 How to Create a Custom Application

Simplify the steps needed to deploy an application by creating a custom application using the Supporting Objects utility.

- How Creating a Custom Application Simplifies Deployment
  Learn about how creating a custom application simplifies the application deployment process.
- Creating a Custom Application
  Learn how to create a custom application.
- Installing Supporting Objects
  Learn how to install supporting objects.
- Deleting Supporting Objects Scripts, Messages, and Installation Options
  Learn about deleting supporting objects scripts, messages, and installation options.
- Upgrading a Custom Application
  Define scripts to upgrade a previously published application on the Upgrade page.
- Deinstalling Supporting Objects
  Learn how to deinstall supporting objects.
- Viewing an Install Summary
  Viewing an Install Summary log.

See Also:

- Exporting and Importing from App Builder
- Exporting Workspaces, Applications, and App Components

25.4.1 How Creating a Custom Application Simplifies Deployment

Learn about how creating a custom application simplifies the application deployment process.

From a developer's perspective, importing and installing an application is a complicated process. First, you create the target database objects and seed data. Second, you import and install the application definition and all related files, including images, themes, and any other required static files.

Creating a custom application using the Supporting Objects utility greatly simplifies this process. Instead of performing numerous steps to create the database objects and then import and install the application and all supporting files, you can define the supporting objects so that the application and supporting files can be migrated in a few easy steps.

After users import and install the application definition, a wizard guides them through a few simple configuration steps. Then, the wizard asks whether to install the supporting application objects. Users have the option of installing the supporting application objects then or doing it later.

From a developer's perspective, creating a custom application has the following advantages:

- Ensures that the supporting objects are created in the correct order.
- Provides an automated process for deploying an application quickly using very few steps.
- Gives users the option to install supporting application objects when they import and install the application definition or at a later time.
- Enables users and developers with a convenient method for removing the application definition, supporting files, and all database objects.
- Provides users and developers with an easy way to upgrade a previously released custom application.

Plus, you can also take advantage of the Deinstall and Install features to quickly edit the underlying database objects that support an application. For example, you can deinstall and remove all database objects, edit the underlying database object creation scripts, and reinstall to create the redefined application objects.

See Also:

- Installing Supporting Objects
- Deinstalling Supporting Objects
- Upgrading a Custom Application

25.4.2 Creating a Custom Application

Learn how to create a custom application.

To create a custom application, you must create installation scripts that define your application's supporting objects (including database objects, images, and seed data) and any preinstallation validations. You manage installation, upgrade, and de-installation scripts and their associated messages on the Supporting Objects page.

- Accessing the Supporting Objects Utility
  Create a custom application on the Supporting Objects utility page.
- Supporting Objects Page
  Learn about the Supporting Objects page.

25.4.2.1 Accessing the Supporting Objects Utility

Create a custom application on the Supporting Objects utility page.

To access the Supporting Objects Page utility:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.

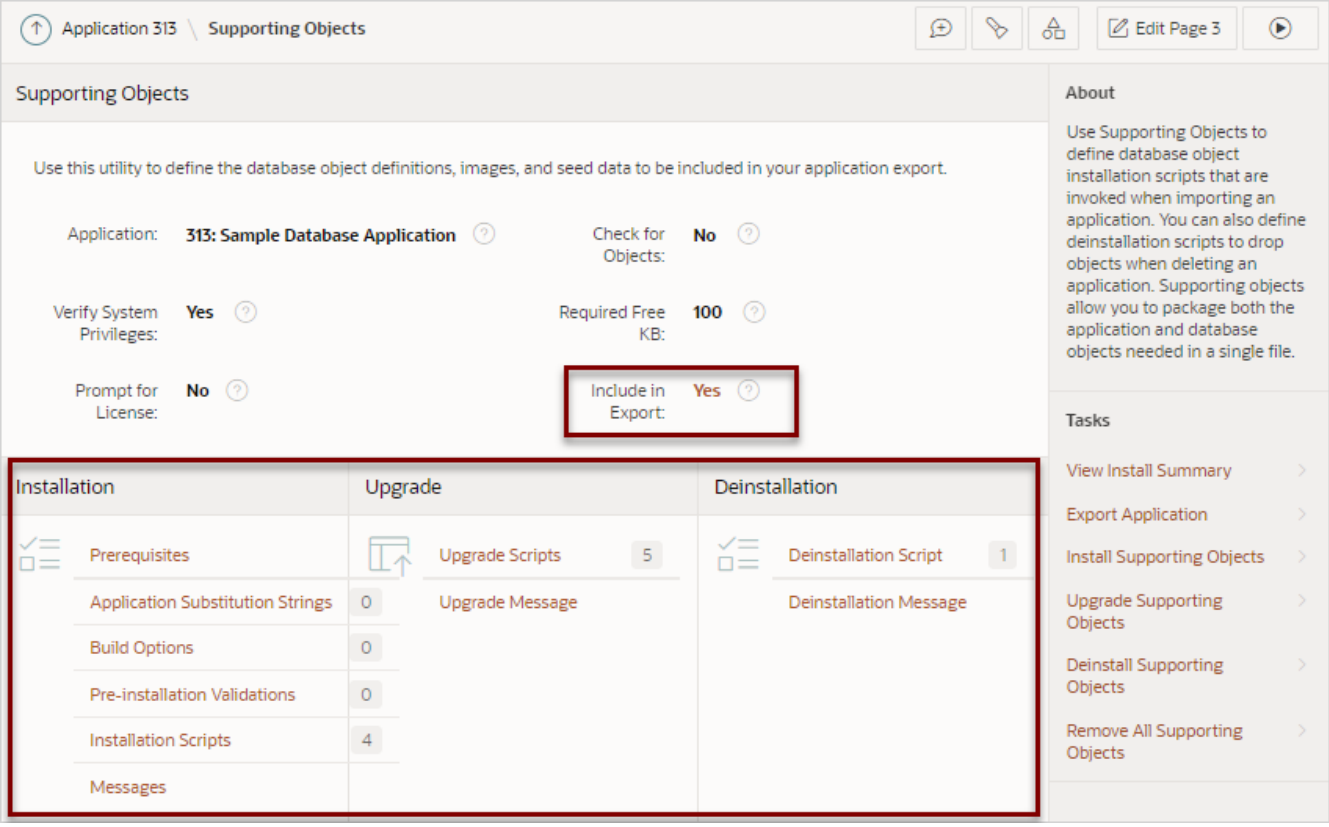
   The Application home page appears.
3. Click the **Supporting Objects** icon.

   The Supporting Objects page appears.

25.4.2.2 Supporting Objects Page

Learn about the Supporting Objects page.

The top of the Supporting Objects page displays the application name and indicates current selections for the following: Check for Objects, Verify System Privileges, Required Free KB, Prompt for License, and Include in Export. To learn more about these options, see field-level Help.



To specify whether to include supporting objects with an application export, find **Include in Export** at the top of the page, click either **Yes** or **No**. If you select **Yes**, a new page appears. From the **Include Supporting Object Definitions in Export** list, make a new selection and click **Apply Changes**.

The rest of the page is divided into the following categories: Installation, Upgrade, and Deinstallation.

- Installation
  Learn about the links under Installation on the Supporting Objects page.
- Upgrade
  Learn about the links under Upgrade on the Supporting Objects page.
- Deinstallation
  Learn about the links under Deinstallation on the Supporting Objects page.

> **See Also:**
>
> Viewing Field-Level and Page Designer Help

25.4.2.2.1 Installation

Learn about the links under Installation on the Supporting Objects page.

Use the links under Installation to define the following types of information:

- **Prerequisites**. Defines built-in checks required before installing the application, such as required free disk space, required system privileges, and schema object restrictions.
- **Application Substitution Strings**. Lists static substitution strings defined for the application. You can define static substitution strings for phrases or labels that occur in many places within an application.

  When packaging an application, you can include prompts for substitution strings which users can specify when they install the custom application.
- **Build Options**. Lists build options defined for this application. You can use build options to conditionally display specific functionality within an application.

  When packaging an application, you can include prompts for specific build options which display when the application is installed.
- **Pre-installation Validations**. Lists validations defined for the custom application. Similar to normal page validations, these validations prevent a user from installing database objects if the user-defined conditions are not satisfied. To create a new validation, click **Create** and follow the on-screen instructions.
- **Installation Scripts**. Define multiple installation scripts that create the application's supporting database objects. To create a new script, click **Create** and follow the on-screen instructions. To edit an existing script, click the **Edit** icon. To prevent another developer from editing a script, click the **Lock** icon. If the script is locked, the Lock icon appears as a closed padlock. If the script is unlocked, the Lock icon appears as an open padlock.

  > **Note:**
  >
  > The Data Package installation script type requires Oracle Database 18c and later.

- **Messages**. Enables you to define messages that display when the user installs or deinstalls the application. Supported HTML tags include `<b>`, `<i>`, `<u>`, `<p>`, `<br>`, `<hr>`, `<ul>`, `<ol>`, `<li>`, and `<pre>`.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

See Also:

- Substitutions
- Using Build Options to Control Configuration
- Exporting Build Options or Build Option Status
- Understanding Cross-Site Scripting Protection

25.4.2.2.2 Upgrade

Learn about the links under Upgrade on the Supporting Objects page.

Use the links under Upgrade to define the following types of information:

- **Upgrade Scripts**. Click **Upgrade Scripts** to define scripts to upgrade database objects, images, and seed data when upgrading an existing application.
- **Upgrade Message**. Enables you to define messages that display when the user upgrades the application. Supported HTML tags include `<b>`, `<i>`, `<u>`, `<p>`, `<br>`, `<hr>`, `<ul>`, `<ol>`, `<li>`, and `<pre>`.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

See Also:

- Upgrading a Custom Application
- Understanding Cross-Site Scripting Protection

25.4.2.2.3 Deinstallation

Learn about the links under Deinstallation on the Supporting Objects page.

Use the links under Deinstallation to define the following types of information:

- **Deinstallation Script**. Click **Deinstallation Script** to define a script to drop database objects and static files created by the installation scripts. To edit an existing script, click the **Edit** icon.
- **Deinstallation Message**. Allows you to define messages that display when the user deinstalls the application. Supported HTML tags include `<b>`, `<i>`, `<u>`, `<p>`, `<br>`, `<hr>`, `<ul>`, `<ol>`, `<li>`, and `<pre>`.

  When these messages display, only a limited set of HTML tags are recognized to prevent a cross site-scripting (XSS) attack.

See Also:

Understanding Cross-Site Scripting Protection

25.4.3 Installing Supporting Objects

Learn how to install supporting objects.

After you edit your supporting objects and create the appropriate scripts, you can run your installation scripts by clicking **Install Supporting Objects** on the Tasks list.

End users can also use this feature if they elect to not install the custom application (or supporting objects) after they import and install the application definition.

To install supporting objects:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.

   The Supporting Objects page appears.
4. From the Tasks list click the **Install Supporting Objects**.

   The Install Supporting Object dialog appears.
5. To view details about the installation script before running it, click **preview the installation details**.

   The Preview Scripts page appears listing summary information, prerequisites, and the actual scripts to be run.
6. To exit the Preview Scripts page and continue, click **Close**.
7. Click **Install Supporting Objects**.
8. Follow the on-screen instructions.

25.4.4 Deleting Supporting Objects Scripts, Messages, and Installation Options

Learn about deleting supporting objects scripts, messages, and installation options.

You can delete the metadata that defines supporting object scripts, messages, and installation options associated with a custom application by clicking **Remove All Supporting Objects** on the Tasks list on the Supporting Objects page.

To delete the metadata that defines supporting object scripts, messages, and installation options:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Supporting Objects**.

   The Supporting Objects page appears.
4. On the Tasks list on the right side of the page, click **Remove All Supporting Objects**.
5. Follow the on-screen instructions.

25.4.5 Upgrading a Custom Application

Define scripts to upgrade a previously published application on the Upgrade page.

- Defining an Upgrade Script
  Create an upgrade script to upgrade database objects, images, and seed data when upgrading an existing application.
- Upgrading a Custom Application
  Learn how to upgrade a custom application.

25.4.5.1 Defining an Upgrade Script

Create an upgrade script to upgrade database objects, images, and seed data when upgrading an existing application.

To create an upgrade script:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. Click **Supporting Objects**.

   The Supporting Objects page appears.
4. Under Upgrade, click **Upgrade Scripts**.
5. Use the Detect Existing Supporting Objects section to determine if the appropriate objects are installed or must be upgraded.
6. In Query to Detect Existing Supporting Objects, enter a query in the field that returns at least one row if the supporting objects exist.

   This query determines whether the user who installs the custom application is prompted to run the installation scripts or the upgrade scripts.
7. To create a script, click **Create**.

   > **Tip:**
   >
   > To enable users to upgrade from various earlier versions of this application, you can add conditions to the upgrade scripts by going to the Script Properties page.

8. To prevent another developer from editing a script, click the **Lock** icon.

   If the script is locked, the Lock icon appears as a closed padlock. If the script is unlocked, the Lock icon appears as an open padlock.
9. To edit an existing script, click the **Edit** icon.
10. Follow the on-screen instructions.

25.4.5.2 Upgrading a Custom Application

Learn how to upgrade a custom application.

After you create your upgrade script, you can test it by clicking **Upgrade Supporting Objects** on Tasks list on the Supporting Object page. End users can also use this feature to upgrade an existing custom application.

To upgrade a custom application:

1. Import a new version of application to be upgraded (if applicable).
2. On the Workspace home page, click the **App Builder** icon.

3. Select the application.
4. Click **Supporting Objects**.

   The Supporting Objects page appears.
5. From the Tasks list, click **Upgrade Supporting Objects**.
6. Follow the on-screen instructions.

> **See Also:**
>
> Importing an Application Export

25.4.6 Deinstalling Supporting Objects

Learn how to deinstall supporting objects.

Once you create or install a custom application, you can deinstall it by either:

- Clicking **Deinstall Supporting Objects** under **Tasks** on the Supporting Objects page.
- Clicking **Delete this Application** on the Application home page.

To deinstall a custom application:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.

   The Supporting Objects page appears.
4. From the Tasks list, click **Deinstall Supporting Objects**.
5. On Deinstall Supporting Objects:
   a. Select **Deinstall Database Objects** to run the deinstallation script defined in the deployment attributes for this application.

      > **Note:**
      >
      > To preview the deinstallation script, expand the **Tasks** region and click **Preview Deinstallation Script**.

   b. Click **Deinstall**
6. Follow the on-screen instructions.

> **See Also:**
>
> Creating a Custom Application

25.4.7 Viewing an Install Summary

Viewing an Install Summary log.

You can view a log of recent installation and deinstallation by clicking **View Install Summary** on the Tasks list on the Supporting Objects page. Note that this log only displays results from the most recent installation or deinstallation that occurred during the current Oracle APEX session.

To view the Install Summary:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. Click **Supporting Objects**.

   The Supporting Objects page appears.
4. On the Tasks list on the right side of the page, click **View Install Summary**.

   A Summary page appears.

25.5 About Publishing an Application URL

Once you deploy your application, load the data, and create users, you can publish your production URL.

Finding Your Production URL

You can determine the production URL for your application by either:

- Selecting the application on the Application home page and right-clicking the **Run** button. Then, select **Copy link address** or **Copy link location** depending on your browser.
- Running the application and then copying the URL.

The Run button gets its value from the Home link attribute on the Edit Security Attributes page. This link is only referenced by this icon and by applications that do not use the Oracle APEX Login API.

Friendly URL Syntax Example

Consider the following example:

```
http://apex.somewhere.com/ords/example/r/hr-app/update-employees?
session=13766599855150
```

Where:

- `apex.somewhere.com` is the URL of the server.
- `ords` is the service name defined when configuring Oracle REST Data Services.
- `example` is the Path Prefix which is URI path prefix used to access RESTful Services. When you create a workspace, this value defaults to workspace name.
- `r` is the router shortcut.
- `hr-app` is the Application Alias. In a new application, the Application Alias defaults to the application Name.

- `update-employees` is the Page Alias. In a new application, the Page Alias defaults to the page Name.
- `?session=13766599855150` is the session ID. When you run an application, the APEX engine generates a session number that serves as a key to the user's session state.

To run this example, a user would use the URL:

```
http://apex.somewhere.com/ords/example/r/hr-app/update-employees
```

f?p URL Syntax Example

Consider the following example:

```
http://apex.somewhere.com/ords/f?p=11563:1:3397731373043366363
```

Where:

- `apex.somewhere.com` is the URL of the server.
- `ords` is the service name defined when configuring Oracle REST Data Services.
- `f?p=` is a prefix used by Oracle APEX.
- `11563` is the application being called.
- `1` is the page within the application to be displayed.
- `3397731373043366363` is the session number.

To run this example, a user would use the URL:

```
http://apex.somewhere.com/pls/apex/f?p=11563:1
```

> **Tip:**
>
> When users sign in, they receive unique session numbers. This number displays at the end of the URL. Do not include the session number as part of the URL. When another user tries to run a URL containing a session ID, an error displays.

> **See Also:**
>
> Understanding URL Syntax

25.6 Using Build Options to Control Configuration

Use build options to conditionally include or exclude application components and functionality.

- About Build Options
  Build options enable developers to enable or disable application components and functionality when the application installs or at runtime using the `APEX_UTIL.GET_BUILD_OPTION_STATUS` and `APEX_UTIL.SET_BUILD_OPTION_STATUS` APIs.

- Creating and Editing Build Options
  Create or edit existing build options on the Build Options, Create / Edit page.
- Including or Excluding Build Options
  Specify **Include** to enable a component and include it with the application. Specify **Exclude** to disable a component and exclude it from the application.
- Copying or Subscribing to Build Options
  Copy build options from the current application or from another application. When copying a build option from another app, you can also subscribe to it.
- Subscribing to Build Options
  Subscribe to build options on the Build Options, Create / Edit page.
- Selecting a Build Option
  Apply build options to a page, component, page control, or shared component in Page Designer.
- Deleting Build Options and Associated Components
  Delete a build option by first removing the associated components and then deleting the build option.
- Viewing Build Option Reports
  Access the Utilization report where build options are utilized in the current application. View the History report to see what modifications have been made to build options in the current application.
- Exporting Build Options or Build Option Status
  Export build option status to toggle build options on or off within another environment.

25.6.4 Copying or Subscribing to Build Options

Copy build options from the current application or from another application. When copying a build option from another app, you can also subscribe to it.

> **Tip:**
>
> Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about subscriptions, see Using Shared Component Subscriptions.

To copy or subscribe to build options:

1. Navigate to the Build Options page.
   a. Navigate to the Workspace home page.
   b. Click the **App Builder** icon.
   c. Select an application.
   d. On the Application home page, click **Shared Components**.
   e. Under Application Logic, click **Build Options**.

      The Build Options page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.
2. To copy a build option within the current application:
   a. Search for the build option and click **Copy** in the Copy column.

      The Copy Build Option Wizard appears.

b. New Build Option Name - Enter the name of the build option. This name must be unique within the application.

c. Click **Copy Build Option**.

3. To copy a build option from another application:

a. From the Tasks region, click **Copy from another app**.

The Copy Build Option Wizard appears.

b. Application - Select the application from which you want to copy.

c. Copy Build Option - Select the build option to copy.

d. Subscribe - Set this flag to **On** to subscribe to the build option.

e. Click **Copy Build Option**.

25.6.5 Subscribing to Build Options

Subscribe to build options on the Build Options, Create / Edit page.

Subscriptions enable developers to reuse shared components across several applications in a workspace. To learn more about shared component subscriptions, see Using Shared Component Subscriptions.

> **Tip:**
>
> You can also subscribe to a search configuration by copying it and then subscribing. See Copying or Subscribing to Build Options.

To subscribe to a c:

1. Navigate to the Build Options page:

a. On the Workspace home page, click **App Builder**.

b. Select an application.

c. On the Application home page, click **Shared Components**.

The Shared Components page appears.

d. Under Application Logic, click **Build Options**.

The Build Options page appears. Note that the report includes the following columns: Subscribed From, Subscription Status, and Subscribers.

2. Select a build option.

The Build Options, Create / Edit page appears.

3. To subscribe to an build option:

a. Find the **Subscription** region.

b. Subscription, Subscribe From - Select a master build option to subscribe to and click **Apply Changes**.

The subscribed component is automatically refreshed from the selected master.

4. To refresh a subscribed a build option:

a. Find the **Subscription** region.

   b. Subscription, Refresh Build Option - If the build option is already subscribed, click **Refresh Build Option**.

5. To unsubscribe from a subscribed build option:
   a. Find the **Subscription** region.
   b. Subscription, Unsubscribe - Click **Unsubscribe**.

25.6.1 About Build Options

Build options enable developers to enable or disable application components and functionality when the application installs or at runtime using the `APEX_UTIL.GET_BUILD_OPTION_STATUS` and `APEX_UTIL.SET_BUILD_OPTION_STATUS` APIs.

You can apply build options to most application components (such as pages, regions, items, validations, and so on) and specify whether to include or exclude them in the runtime application.

Build options have two possible values:

- **Include** - Application components are enabled and included with the application.
- **Exclude** - Application components are disabled and excluded from the application.

If you specify **Include**, then the Oracle APEX engine considers the component as part of the application definition at runtime. Conversely, if you specify **Exclude**, then the APEX engine treats the component as if it did not exist.

> **Note:**
>
> The **Commented Out** build option is included by default in new applications. This build option enables developers to include or exclude page components by selecting a toggle on the tree context menu in Page Designer. Deleting the **Commented Out** will remove this functionality. To learn more, see Commenting Out Page Components in Page Designer.

> **See Also:**
>
> GET_BUILD_OPTION_STATUS Functions and SET_BUILD_OPTION_STATUS Procedure in *Oracle APEX API Reference*

Use Case 1: Enable and Disable Functionality Based on Installation Type

You are developing an application that has many installation types. For each installation type, users may choose to enable or disable certain application functionality. With build options, you can develop one application and when it installs, you can configure build options to include or exclude associated functionality. You may also use an API to expose features within the application. This enables privileged end users to selectively enable or disable application functionality.

Use Case 2: Hiding Incomplete Functionality in Various Environments

You are developing functionality which is not yet complete, so you do not wish to expose it to end users. You can use build options to disable functionality when the application is built for

test and production environments. The benefit of this approach is that the application can still be deployed without including incomplete functionality.

Use Case 3: Determining the Consequences of Removing Functionality

You are removing functionality you think is not needed by the application, but are unsure if removing the functionality is a safe operation. In this case, you can use build options to essentially "comment out" specific functionality. When the application is deployed and you determine the functionality is necessary, you can re-instate it. If the functionality is deemed as unnecessary, you can then safely remove it from the application.

25.6.2 Creating and Editing Build Options

Create or edit existing build options on the Build Options, Create / Edit page.

To create or edit a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. On the Build Options page:
   - To create a new build option, click **Create**.
   - To edit an existing build option, click the build option, **Name**.

   The Create / Edit page appears.
6. Under Attributes:
   a. Application - Identifies the application to which this build option applies.
   b. Build Option - Enter a descriptive name for this build option. Build options are predefined settings that determine whether or not components within an application are enabled.
   c. Status - Options include:
      - **Include** - Include this feature or component. Associated application components are enabled and included with the application.
      - **Exclude** - Do not include this feature or component. Associated application components are disabled and excluded from the application.
   d. Default on Export - When this application is exported, set the build option to this value.
   e. On Upgrade Keep Status - Select whether the build option status in the deployed application should be kept, or overwritten with the specified status when the application is upgraded. **On** keeps the deployed status and is useful if the application supports build option configuration (using the `APEX_UTIL.SET_BUILD_OPTION_STATUS` API) to enable or disable the build option in the deployed application, and where you want to respect those settings upon upgrade.
   f. Comments - Enter any comments or notes. These comments never display when running the application.
7. Confirm your changes:
   - If you are creating a build option, click **Create Build Option**.
   - If editing an existing build option, click **Apply Changes**.

25.6.3 Including or Excluding Build Options

Specify **Include** to enable a component and include it with the application. Specify **Exclude** to disable a component and exclude it from the application.

Build options have two possible values: **Include** and **Exclude**. If you specify an attribute as being included, then the Oracle APEX engine considers it part of the application definition at runtime. Conversely, if you specify an attribute as being excluded, then the APEX engine treats it as if it did not exist.

To include or exclude a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.

   The Build Options page appears.
5. Select a build option Name.

   The Create/Edit Build Option page appears.
6. For Status, select either:
   - **Include** - Include this feature or component. Associated application components are enabled and included with the application.
   - **Exclude** - Do not include this feature or component. Associated application components are disabled and excluded from the application.
7. Edit any additional attributes as needed.

   To learn more about an attribute, see field-level Help.
8. To save your changes, click **Apply Changes**.

> **See Also:**
>
> `APEX_UTIL.GET_BUILD_OPTION_STATUS` and `APEX_UTIL.SET_BUILD_OPTION_STATUS` in Oracle APEX API Reference

25.6.6 Selecting a Build Option

Apply build options to a page, component, page control, or shared component in Page Designer.

Once you create a build option, you can select it for a page, a component (report, chart, or form), a specific page control (button, item, list of value), and another shared component (breadcrumb, list, or tab). You apply build options by viewing to the page in Page Designer and selecting the component, page control, or shared component. In the Property Editor, find the **Configuration** group and select a **Build Option**.

Once you apply a build option, you can toggle it on and off in Page Designer from the Rendering tab context menu.

To apply a build option:

1. Create the build option as described in Creating and Editing Build Options.
2. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
3. In the Rendering tab, select the page component or control.
4. In the Property Editor, find **Configuration**, **Build Options** and select the build option.
5. Click **Save**.

25.6.7 Deleting Build Options and Associated Components

Delete a build option by first removing the associated components and then deleting the build option.

- Removing Components Associated with a Build Options
  Learn how to remove components associated with a build options.
- Deleting Build Options Permanently
  Learn how to delete build options permanently.

25.6.7.1 Removing Components Associated with a Build Options

Learn how to remove components associated with a build options.

Removing components associated with a build option permanently deletes them.

To remove components associated with a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.

   The Build Options page appears.
5. Select the build option .

   The Create/Edit Build Option page appears. Note that Associated Components display at the bottom of the page.
6. Scroll down to Associated Components and click **Remove Components**.

   A verification window appears.
7. Click **Yes - remove components** and then click **Remove Components**.

   The components are removed.

25.6.7.2 Deleting Build Options Permanently

Learn how to delete build options permanently.

> **Tip:**
>
> Deleting a build option does not delete the Associated Components. Before deleting a build option, review the Associated Components to determine if the components should be removed.

To delete a build option:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.

   The Build Options page appears.
5. Select the build option.

   The Create/Edit Build Option page appears.
6. Click **Delete**.

   A verification window appears.
7. Click **Delete Build Option**.

   The build option is removed.

25.6.8 Viewing Build Option Reports

Access the Utilization report where build options are utilized in the current application. View the History report to see what modifications have been made to build options in the current application.

- Viewing Build Option Utilization
  View the Build Option Utilization report.
- Viewing Build Option History
  View the Build Option History report.

25.6.8.1 Viewing Build Option Utilization

View the Build Option Utilization report.

Once you create a build option, a Utilization tab appears on the Build Options page. The Build Option Utilization report details build option utilization in the current application.

> **Note:**
>
> The Utilization tab only appears on the Build Options page after you create a build option.

To view the Build Option Utilization report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. On the Build Options page, click **Utilization**.

   The Build Option Utilization report appears.
6. Select a build option and click **Go**.

25.6.8.2 Viewing Build Option History

View the Build Option History report.

To view the Build Option History report:

1. On the Workspace home page, click the **App Builder** icon.
2. Select an application.
3. On the Application home page, click **Shared Components**.
4. Under Application Logic, click **Build Options**.
5. On the Build Options page, click **History**.

   The Build Option History report appears.

25.6.9 Exporting Build Options or Build Option Status

Export build option status to toggle build options on or off within another environment.

You can export build options or build option status on the Component Export page. For example, you can use this feature to deploy a production application with a hidden feature by associating the components of the hidden feature with a build option having the status of `EXCLUDE`. After deployment, you can enable the hidden feature by changing the status of the build option to `INCLUDE` and then exporting the Build Option Status.

Once you apply the Build Options Status to the production instance, the new feature appears.

To export build options or build option status:

1. Navigate to the Component Export page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Click **Shared Components**.
   d. From the Tasks list, click **Export Application Components**.

   The Components Export page appears with the **Build Options Status** tab selected.

2. On **Build Options Status**:
   a. **Rows** - Select the number of items to display and click **Go**.
   b. Select the Build Options to include and click **Add to Export**.

   The selected Build Options display under **Components to Export**.
3. Click **Next**.
4. For Component Export:
   a. **Export as Zip** - Options include:
      - **On** - Choose **On** to export the application as a ZIP archive containing separate files for each component.
      - **Off** - Choose **Off** to export as a single SQL file containing all the selected components.

      > **Tip:**
      >
      > To learn more about the Zip archive, see Splitting Export Files into Separate Scripts.

   b. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

      This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases may have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer transactions because it is not filling the undo tablespace, forcing older data to be archived.
   c. **File Character Set** - Indicates the character set in which the export file will be encoded. When importing this file you will need to specify the same character set.
   d. Click **Export Components**.

   > **Tip:**
   >
   > For translated applications, the Export Translations field displays. To export the selected shared components for the primary language application and all translated applications, from Export Translations field, select **Yes**. To ensure text strings and markup are up-to-date, remember to synchronize the translated applications prior to exporting the translated components.

> **See Also:**
>
> - Importing Export Files
> - How to Create a Custom Application

25.7 Understanding One-Click Remote Application Deployment

Deploy an application to remote Oracle APEX instances using REST Enabled SQL references.

- About One-click Remote Application Deployment
  One-click remote application deployment leverages the existing REST Enabled SQL references that works with a Oracle REST Data Services (ORDS) REST Enabled SQL Service.
- Remotely Deploying an Application
  Access the Remote Deployment option from the Export/Import option in App Builder.

> **See Also:**
>
> Managing REST Enabled SQL References

25.7.1 About One-click Remote Application Deployment

One-click remote application deployment leverages the existing REST Enabled SQL references that works with a Oracle REST Data Services (ORDS) REST Enabled SQL Service.

One-click remote application deployment is authenticated through first party authentication (Basic Authentication), Schema Authentication (database username and password), or OAuth 2 Client Credentials. When you create a REST Enabled SQL reference in Shared Components, you either specify an existing credential, or create a new credential. Optionally create the credentials as empty credentials to prompt for client ID and client secret each time the REST Enabled SQL reference is used.

> **Note:**
>
> Oracle recommends always using empty credentials for remote application deployment in shared development environments so only those with the knowledge of those credentials can deploy applications or run SQL or PL/SQL on the REST Enabled SQL reference.

25.7.2 Remotely Deploying an Application

Access the Remote Deployment option from the Export/Import option in App Builder.

To remotely deploy an application:

1. Create an application.
2. Optional: Create supporting object installation scripts to create the underlying table and insert the rows.
3. Navigate to the Remote Deployment page:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.

c. On the Application home page, click **Export/Import**.

d. Click **Remote Deployment**.

4. On Deployment Server, select an existing REST Enable SQL Service or click the plus (+) icon to create a new one.

5. To create a new REST Enable SQL Service:

   a. Click the plus (+) icon.

   b. **Name** - Enter a descriptive name for this REST Enabled SQL Service.

   c. **Endpoint URL** - Enter the base URL for this REST Enabled SQL Service. Include the ORDS context root and schema URL prefix, (for example, `https://host:port/ords/schema`, but do not add `/_/sql`.

   d. Click **Next**.

   e. **Credentials** - Select the credential to authenticate against. To edit an existing credential, navigate to Credentials in Shared Components.

      To create new credentials, enter a Credential Name and select an Authentication Type.

   f. Click **Create**.

6. On Remote Deployment:

   a. **Deployment Server** - Select an existing REST Enable SQL Service.

   b. **Client ID** - For Basic Authentication, enter the database user name. For OAuth2, enter the client ID.

      This information will not be encrypted. User names for Basic Authentication are converted to upper case. To include a case-sensitive user name, prefix it with the exclamation mark (!).

   c. **Client Secret** - Enter the password or OAuth2 Client Secret. This information is stored encrypted and cannot be retrieved in clear text.

   d. Click **Next**.

   The Remote Deployment Options page appears.

7. Under Deploy Application:

   a. **Build Status** - Set the build status of the application. Options include:

      - **Run Application Only** - Users can only run an application. This option is an effective way to protect an application from modifications from other developers.
      - **Run and Build Application** - Developers and users can both run and develop the application.

      This setting can be changed when importing using the APEX Web interface but not the command line interface.

   b. **As of** - Specify a time in minutes to go to back to for your export. This option enables you to go back in time in your application, perhaps to get back a deleted object.

      This utility uses the `DBMS_FLASHBACK` package. Because the timestamp to System Change Number (SCN) mapping is refreshed approximately every five minutes, you may have to wait that amount of time to locate the version for which you are looking. The time undo information is retained and influenced by the startup parameter `UNDO_RETENTION` (the default is three hours). However, this only influences the size of the undo tablespace. While two databases can have the same `UNDO_RETENTION` parameter, you are able to go back further in time on a database with fewer

transactions because it is not filling the undo tablespace, forcing older data to be archived.

8. Under Export Preferences:
   a. **Export Supporting Object Definitions** - Supporting object definitions include all configuration options and scripts and enable an application export to include database object definitions, image definitions, and seed data SQL statements encapsulated in a single file. Options include:
      - **Yes** - Includes supporting object definitions in the application export. Does not automatically load supporting objects when invoked from a command line.
      - **No** - Does not include the supporting object definitions in the export.
      - **Yes and Install on Import Automatically** - Includes supporting object definitions and a call to install supporting objects in the application export.
   b. **Export Public Reports** - Specify whether to include public reports with your application. Public reports are the public shared report settings saved by users from an interactive report or interactive grid.
      - **On** - Include public reports in the application export.
      - **Off** - Do not include public reports in the application export.
   c. **Export Private Reports** - Specify whether to include private reports with your application. Private reports are the private customized settings saved by users from an interactive report or interactive grid.
      - **On** - Include private reports in the application export.
      - **Off** - Do not private public reports in the application export.
   d. **Export Report Subscriptions** - Specify whether to include interactive report or interactive grid subscription settings with your application. This will be limited to the unexpired subscription settings of all users for all interactive reports or interactive grids in the application.
      - **On** - Include report subscriptions for the reports you are exporting in the application export.
      - **Off** - Do not include report subscriptions for the reports in the application export.
   e. **Export Developer Comments** - Specify whether to include Developer Comments in your application export.
      - **On** - Include Developer Comments in the application export.
      - **Off** - Do not include Developer Comments in the export.
   f. **Export Translations** - Specify whether to include translations with your application export. Translation Text Messages and Dynamic Translations are always included in the application export. .
      - **On** - Include the translation mappings and all text from the translation repository.
      - **Off** - Do not include translation mappings or the translation repository.
9. On Deployment Overrides:
   a. **Owner Override** - Use this attribute to import this application into the available schemas defined by your REST Enabled SQL connection. If there are no entries to select, then this connection is not mapped to any workspaces, and you will not be able to deploy the application.

b. **Workspace Override** - Select the workspace where the application should be deployed from the list. If there are no entries in the list, then the database user associated with the REST Enabled SQL connection is not mapped to any workspaces.

c. **Overwrite Existing Application** - Specify whether to overwrite the application on the target system. Available options include:

   - **On**- The application will be deployed with the same application ID as this application and the application will be overwritten on the target system.
   - **Off** - You must specify the application ID for the target system.

d. **Application ID Override** - Specify a new application ID for the application on the target system.

e. **Application Alias Override** - Specify a new alternate alphanumeric application identifier. You can use this identifier in place of the application ID. The value is defaulted to the application alias of the current application.

10. Click **Deploy**.

> **Note:**
>
> - Creating Applications
> - Installing Supporting Objects

26

Managing Application Legacy Components

Although Oracle still supports legacy application components, Oracle does not recommend using them.

- Legacy Forms Optimized for Mobile Environments
  Learn about legacy forms optimized for mobile environments.
- Viewing Legacy Components in the Gallery
  View legacy and unsupported components in the Page Designer Gallery by selecting the **Gallery Menu** and then **Show Legacy Components**.
- Managing Legacy Calendars
  Legacy calendars are template-based. In previous releases Legacy Calendar was referred to as Calendar (Template, Classic calendar, Easy calendar, and SQL calendar).
- Managing Applications with Legacy Data Loading Capability
  Applications with legacy data loading capability enable end users to dynamically import data into a table within any schema to which they have access.
- Managing Legacy Tabular Forms
  A legacy tabular form enables users to update multiple rows in a table at once from a single page.
- Managing Legacy Web Services
  Legacy Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment.

26.1 Legacy Forms Optimized for Mobile Environments

Learn about legacy forms optimized for mobile environments.

> **Note:**
>
> Although Oracle still supports legacy application components, Oracle does not recommend using them.

- Mobile Friendly Legacy Forms
  Oracle APEX includes three mobile friendly reports and forms: List View, Column Toggle, and Reflow Report.
- Supported Columns in Legacy Column Toggle and Reflow Reports
  Learn about column types supported by column toggle and reflow legacy reports.
- Creating a Legacy Column Toggle Report
  Create a legacy column toggle report in Page Designer.
- Creating a Legacy List View Report
  Create a legacy list view report in Page Designer.
- Creating a Legacy Reflow Report
  Create a legacy reflow report in Page Designer.

26.1.1 Mobile Friendly Legacy Forms

Oracle APEX includes three mobile friendly reports and forms: List View, Column Toggle, and Reflow Report.

Although the Universal Theme is optimized to work well on mobile devices, not all components are mobile friendly. For example, interactive reports and interactive grids do not work well in mobile environments.

The following mobile friendly reports are optimized for mobile environments:

- **List View** - A **List view** report features a responsive design to display data and provide easy navigation on Smartphones. A list view report contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. Developers edit the report Attributes to determine the database columns that display, including the column used for List view entry.
- **Column Toggle Report** - A **column toggle** report contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. A column toggle report includes a Columns button which enables end users to select which columns they want to view.
- **Reflow Report** - A **reflow report** contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row.

26.1.2 Supported Columns in Legacy Column Toggle and Reflow Reports

Learn about column types supported by column toggle and reflow legacy reports.

Column toggle and reflow reports support the following column types. To view the column type in Page Designer, select the column and find the **Settings**, **Type** attribute in the Property Editor.

> **Note:**
>
> You can also view this information in Page Designer Help. View the page containing the report in Page Designer. Select a column and then **Settings**, **Type** attribute. Click the **Help** tab in the central pane.

Column Type	Description
Plain Text	Displays the column value.
Link	Displays the column as a link.
Percent Graph	Displays the column as a percentage graph. The value retrieved must be between 0 and 100.

26.1.3 Creating a Legacy Column Toggle Report

Create a legacy column toggle report in Page Designer.

A **column toggle** report contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. A column toggle report includes a Columns button which enables end users to select which columns they want to view.

By default, column toggle reports are created with all columns set to the same priority. However, the developer can edit the report column attributes and rank columns by importance. Columns with a lesser priority (larger number) are hidden at narrower screen widths. The report includes a Columns button which enables end users to select which columns they want to view.

To create a column toggle report.

1. Create a Blank page using the Create Page Wizard.
2. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
3. Click the **Layout** tab in the central pane.
4. In the Gallery, click the **Regions** tab. Right-click **Column Toggle** and select **Add To**, and select the appropriate location (for example, **Body**).

   > **Tip:**
   >
   > You can also select the Column Toggle region in the Gallery and drag it to the appropriate location in the Layout tab.

5. Page Designer indicates what to do next. If there are errors, the **Show Messages** icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
6. Select the **New** region (if not already selected).

   Next, edit the **Region** attributes in the Property Editor (or right pane).
7. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.
8. Source - Configure the Data Source for the region:

   What appears next depends upon the attributes you select. For this example, the report is built on a local database on `EBA_PROJECTS` table.

> **Tip:**
>
> To learn more about an attribute and view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

a. Location - Select the location of the data. For example, select **Local Database**.

b. Type - Select how the data is queried. For example, select **Table/View**.

c. Table Name - Enter the case-sensitive table or view name. You can type in the name or pick from the list. For example, select **EBA_PROJECTS**.

9. Edit any additional attributes as needed.

10. View the page, click **Save and Run Page**.

    The toggle columns report appears.

    End users can control what columns display by clicking the **Columns** button.

26.1.4 Creating a Legacy List View Report

Create a legacy list view report in Page Designer.

List view reports feature a responsive design to display data and provide easy navigation on Smartphones. A list view report contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. Developers edit the report Attributes to determine the database columns that display, including the column used for List view entry.

The example in this topic builds a report on the `EBA_PROJECTS` table available in the sample dataset, Project Data.

To create a list view report.

1. If needed, create a Blank page using the Create Page Wizard.

2. View the page in Page Designer:

   a. On the Workspace home page, click the **App Builder** icon.

   b. Select an application.

   c. Select a page.

   Page Designer appears.

3. In the Gallery, click the **Regions** tab. Right-click **List View** and select **Add To**, and select the appropriate location (for example, **Body**).

> **Tip:**
>
> You can also select the **List View** region in the Gallery and drag it to the appropriate location in the Layout tab.

4. Page Designer indicates what to do next. If there are errors, the **Show Messages** icon displays in the Page Developer toolbar.

   a. Click the **Show Messages** icon on the Page Designer toolbar.

   The Messages dialog displays errors that must be addressed.

   b. Select an error to highlight the associated attribute in the Property Editor.

5. Select the **New** region (if not already selected).

   Next, edit the **Region** attributes in the Property Editor (or right pane).

6. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.

7. Source - Configure the Data Source for the region:

   What appears next depends upon the attributes you select. For this example, the report is built on a local database on `EBA_PROJECTS` table.

   > **Tip:**
   >
   > To learn more about an attribute and view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

   a. Location - Select the location of the data. For example, select **Local Database**.

   b. Type - Select how the data is queried. For example, select **Table/View**.

   c. Table Name - Enter the case-sensitive table or view name. You can type in the name or pick from the list. For example, select **EBA_PROJECTS**.

8. Configure the report Attributes:

   a. In the Property Editor, click the **Attributes** tab.

   b. Settings - Configure the attributes such as the following:

      - Features - Select the features to include with list view report. Depending upon what you select additional attributes may appear. For example, select **Enable Search**.
      - Text Column - Select the database column that contains the main content for the list view entry. For example, select **NAME** (required).
      - Supplemental Information Column - -Select the database column that contains supplemental information for the list view entry. For example, select **DESCRIPTION**.
      - Counter Column - Select the database column that contains the value to display in the count bubble. For example, select BUDGET.

9. Edit any additional attributes as needed.

10. View the page, click **Save and Run Page**.

    The list view report appears.

26.1.5 Creating a Legacy Reflow Report

Create a legacy reflow report in Page Designer.

A **reflow report** contains the formatted result of a SQL query based on the data source (local or external) you specify and the table or SQL query you provide. When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row.

The example in this topic builds a report on the `EBA_PROJECTS` table available in the sample dataset, Project Data.

To create a reflow report.

1. If needed, create a Blank page using the Create Page Wizard.
2. View the page in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.
3. In the Gallery, click the **Regions** tab. Right-click **Reflow Report** and select **Add To**, and select the appropriate location (for example, **Body**).

   > **Tip:**
   >
   > You can also select the **Reflow Report** region in the Gallery and drag it to the appropriate location in the Layout tab.

4. Page Designer indicates what to do next. If there are errors, the **Show Messages** icon displays in the Page Developer toolbar.
   a. Click the **Show Messages** icon on the Page Designer toolbar.

      The Messages dialog displays errors that must be addressed.
   b. Select an error to highlight the associated attribute in the Property Editor.
5. Select the **New** region (if not already selected).

   Next, edit the **Region** attributes in the Property Editor (or right pane).
6. Identification, Title - Enter a region title. The region title only displays when it is defined in the region template.
7. Source - Configure the Data Source for the region:

   What appears next depends upon the attributes you select. For this example, the report is built on a local database on `EBA_PROJECTS` table.

   > **Tip:**
   >
   > To learn more about an attribute and view examples, select the attribute in the Property Editor and click the **Help** tab in the central pane.

   a. Location - Select the location of the data. For example, select **Local Database**.
   b. Type - Select how the data is queried. For example, select **Table/View**.
   c. Table Name - Enter the case-sensitive table or view name. You can type in the name or pick from the list. For example, select **EBA_PROJECTS**.
8. Edit any additional attributes as needed.

   For example, to facilitate a better display in a smaller window, you may want to edit the Column attributes, Heading, Alignment and Layout, Column Alignment attributes.
9. View the page, click **Save and Run Page**.

   The reflow report appears.

When there is not enough space available to display the report horizontally, the report responds by collapsing the table columns into a vertical value pairs layout where each column displays on a separate row as shown in the following example.

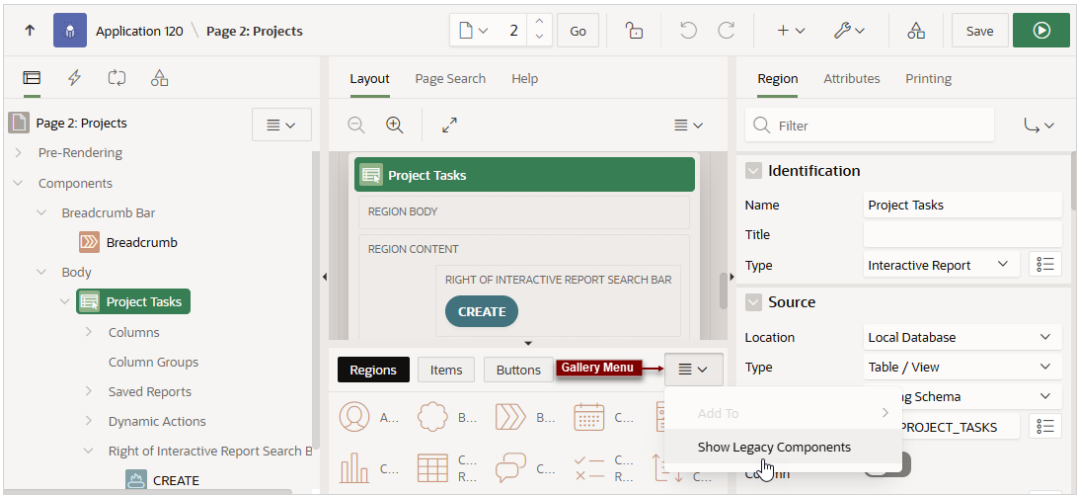
26.2 Viewing Legacy Components in the Gallery

View legacy and unsupported components in the Page Designer Gallery by selecting the **Gallery Menu** and then **Show Legacy Components**.

To view legacy components in the gallery:

1. Navigate to the Workspace home page.
2. Click the **App Builder** icon.
3. Select an application.
4. Select a page.

   The page appears in Page Designer.
5. Click **Gallery Menu** and select **Show Legacy Components**.



26.3 Managing Legacy Calendars

Legacy calendars are template-based. In previous releases Legacy Calendar was referred to as Calendar (Template, Classic calendar, Easy calendar, and SQL calendar).

> **Tip:**
>
> Legacy calendars are legacy application components. Although Oracle still supports legacy components, the creation of new calendars is based on the FullCalendar jQuery library.

- Editing Legacy Calendar Attributes
  Learn about editing legacy calendar attributes.
- About Supported Legacy Calendar Substitution Strings
  Learn about about supported legacy calendar substitution strings.

26.3.1 Editing Legacy Calendar Attributes

Learn about editing legacy calendar attributes.

To edit calendar Attributes:

1. View the page containing the calendar in Page Designer:
   a. On the Workspace home page, click the **App Builder** icon.
   b. Select an application.
   c. Select a page.

   Page Designer appears.

   In the Property Editor, attributes are organized into groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

2. To edit the Region attributes.
   a. In the Rendering tab, locate and select the calendar.
   b. In the Property Editor, review and edit the attributes on the Region tab.
3. To edit the calendar Attributes.
   a. In the Rendering tab, locate and select the calendar.
   b. In the Property Editor, review and edit the attributes on the Attributes tab.
4. To save your changes click **Save**. To save and run the page, click **Save and Run Page**.

26.3.2 About Supported Legacy Calendar Substitution Strings

Learn about about supported legacy calendar substitution strings.

App Builder supports many date format substitution strings. You can view a complete list of supported substitution strings on the Calendar Templates page.

To view a list of supported substitution strings for calendars:

1. Navigate to the appropriate calendar template.
2. View the Substitution Stings list on the right side of the page.

> **See Also:**
>
> Using Themes and Theme Styles

26.4 Managing Applications with Legacy Data Loading Capability

Applications with legacy data loading capability enable end users to dynamically import data into a table within any schema to which they have access.

- Comparing Legacy Data Loading and New Data Loading
  Learn about the differences between legacy data loading and new data loading
- About Creating a Legacy Data Load Wizard
  Create a Data Load Wizard by running the Create Page Wizard and creating a Data Load page.
- Creating a Legacy Data Load Page
  To create a legacy Data Load Wizard, a developer creates a Data Load page with the Create Page Wizard.
- Re-creating Legacy Data Load Pages
  To make changes to existing legacy Data Load pages, Oracle recommends re-creating new pages without deleting the data loading definitions.
- Editing a Legacy Data Load Definition
  Edit the Data Load Definition in Shared Components.

26.4.1 Comparing Legacy Data Loading and New Data Loading

Learn about the differences between legacy data loading and new data loading

Oracle APEX includes two types of data loading: new data loading and legacy data loading. The new data loading has many benefits over legacy data loading.

Benefits of new data loading include:

- New application data loading supports `CSV`, `XLSX`, `XML`, and `JSON` formats.
- Column mapping occurs at design time, removing the burden for end users.
- Flexible column mappings based on simple names or regular expressions.
- Data conversion with transformation rules or lookup queries.
- Easy work flow for end users: upload the file, verify the preview, and load data.
- `CSV`, `XLSX`, `XML`, and `JSON` data formats can be loaded to tables or collections.
- Configure data loading to Append, Merge or Replace data, with or without Error Handling.
- Simple new Process Type Data Loading: Customize Data Loading pages as you wish.
- `APEX_DATA_LOADING` PL/SQL API available for custom processing.
- Maximum number of columns to load is 300. Note that Legacy data load supports up to 45.

> **See Also:**
>
> Creating Applications with Data Loading Capability

26.4.2 About Creating a Legacy Data Load Wizard

Create a Data Load Wizard by running the Create Page Wizard and creating a Data Load page.

During the process of creating the Data Load page, developers can specify the upload table and its unique columns, table lookups, and data transformation rules.

> **Note:**
>
> A Data Load Wizard is not designed or intended to load hundreds of thousands of rows of data. While it is possible to use a Data Load Wizard to load this high volume of data, you may encounter performance issues with both transmitting and loading large data files. Tools like Oracle SQL Developer and Oracle SQL*Loader are better suited to loading large volumes of data.

The Data Load Wizard includes support for the following:

- Table Definitions - This definition specifies the data upload table name with its unique key columns.
- Data Transformation Rules - For formatting transformations such as changing import data to uppercase, lowercase, and so on, you must define data transformation rules. For example, if the import file includes column data with both upper and lowercase and the upload table requires all uppercase, you can define a data transformation rule to insert only uppercase into the target column.
- Table Lookups - If data existing in the import file must be mapped to data in another table, specify a table lookup to perform the mapping. For example, if the import file contains a department name for the `DEPTNO` column but the upload table requires a number for that column, use a table lookup rule to find the corresponding department number for that department name in another table.
- Column Name Aliases - There are many situations when a developer does not wish to expose the table column names to the end user, or to expose all columns to the end user. In those situations, you can create a column aliases for the columns that need to exposed.
- Manage Concurrency - If multiple users are uploading data at the same time, developers can use extra column to track the version of data in the underlying table. The Data Load Wizard can use this column to check and signal the end user if anyone else is working with the same data at the same time. This is particularly important if uploading into a table that is regularly updated.
- Multiple Spreadsheet Columns - There are many situations when a spreadsheet to be uploaded has multiple columns that the developer wants to concatenate and upload in to one table column (for example `FirstName` and `LastName` on spreadsheet can be uploaded into `ENAME` of the `EMP` table).
- Skip Validation - You can improve data loading performance when uploading a large number of records by skipping the validation step. If uploading thousands of records, the end user might not be interested in validating each record. If you are certain that each record will be inserted as new record, the data loading process does not need to check for duplicates.

The newly generated Data Load Wizard consists of four pages that provide users with the ability to upload data from a file or by copy and paste, define data and table mappings, validate

the data, and finally to upload the data to the table. The developer can later edit the Data Load Wizard's definitions such as table lookups and data transformation rules, by accessing Shared Components, Data References, Data Load Definitions.

Supported Data Types

Data Load Wizards support the following data types:

- `VARCHAR2`
- `DATE`
- `TIMESTAMP`
- `NUMBER`

Unsupported Data Types

Data Load Wizards do not support the following data types:

- Large objects (`BLOB` and `CLOB`)
- Complex types (`XMLTYPE` and `SDO_GEOMETRY`)
- `CHAR`

> **See Also:**
>
> - Creating a Legacy Data Load Page
> - Editing a Data Load Definition

26.4.3 Creating a Legacy Data Load Page

To create a legacy Data Load Wizard, a developer creates a Data Load page with the Create Page Wizard.

To create a legacy Data Load Wizard:

1. On the Workspace home page, click the **App Builder** icon.
2. Select the application.
3. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Click **Legacy Pages** and select **Legacy Data Loading**.
5. For Data Load Table:
   a. Legacy Data Load - Choose whether to create a new or to re-use an existing Legacy Data Load definition.
   b. Definition Name - Enter the name of this data load definition.
   c. Owner - Select the owner of the table on which the form will be based..
   d. Table Name - Select the table to use for data loading (also known as the upload table).

   e. Unique Column 1 - Identify the column name(s) to be used as the primary unique key column during the data load process. You can define up to 3 unique key columns.
   f. Case Sensitive - Identify whether the selected unique key column is case sensitive. By default, this is set to **No**.
   g. Define additional Unique Columns. You can define up to 3 unique key columns.
   h. Click **Next**.

6. For Add Transformation Rules (optional) - Transformation Rules enable you to change the data being uploaded before it is inserted into the base table. Select the column to transform and then the desired rule to apply to it.
   a. Select Column(s) to create a transformation rule - Select the column on which the transformation rule definition is to be based and move them to the right.
   b. Rule Name - Enter a name for this transformation rule.
   c. Sequence - Specify the sequence for the transformation rule. The sequence determines the order of execution.
   d. Type - Select the type of transformation rule you want to perform.
   e. Provide additional details based on the transformation type you have chosen.
   f. Click **Add Transformation**.
   g. Click **Next**.
7. For Table Lookups - Match an uploaded value against another table and use the associated key value, instead of the uploaded value.
   a. Add new table lookup for Column (optional) - Identify the column on which the table lookup definition is to be based.
   b. Lookup Table Owner - Select the owner of the lookup table.
   c. Lookup Table Name - Identify the table to be used for this table lookup definition.
   d. Return Column - Select the name of the column returned by the table lookup. This value will be inserted into the load column specified and is generally the key value of the parent in a foreign key relationship (for example: `DEPTNO`).
   e. Upload Column - Select the name of the column end users will upload instead of the return column. This is the column that contains the display value from the lookup table (for example: `DNAME`).
   f. Upload Column 2 - Select the name of the second column to be uploaded to uniquely identify the return column if necessary. For example, to uniquely identify a State Code it may be necessary to upload the State Name and Country.
   g. Upload Column 3 - Select the name of the third column to be uploaded to uniquely identify the return column.
   h. Click **Add lookup** to add the lookup definition. Repeat the previous steps to add additional table lookups.
   i. Click **Next** finish creating lookups.
8. For Page Attributes:
   a. Step 1, Step 2, Step 3, and Step 4 - Enter a page name and page number for each step.
   b. Page Number - Enter a page number for each step.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.

   d. Page Group - Identify the name of the page group you would like to associate with this page.
   e. Breadcrumb - Select whether you want to use a breadcrumb navigation control on your page and which breadcrumb navigation control you want to use.
   f. Click **Next**.
9. For Navigation Menu:
   a. Navigation Preference - Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
   b. Click **Next**.
10. For Buttons and Branching, specify the branching for the buttons on the data load wizard pages:
   a. New Button Label - Enter text to display on the Next button.
   b. Previous Button Label - Enter text to display on the Previous button.
   c. Cancel Button Label - Enter text to display on the Cancel button.
   d. Cancel Button Branch to Page - Specify the page number to branch to when the user clicks Cancel.
   e. Finish Button Label - Enter text to display on the Submit button.
   f. Finish Button Branch to Page - Specify the number of the page to branch to. You can choose to branch back to the same page or any other page in your application.
   g. Click **Create**.
11. Click **Save and Run Page** to test the Data Load Wizard.

> **Tip:**
>
> After creating Data Load Wizard pages, if you wish to make changes, Oracle recommends re-creating new pages without deleting the data loading definitions as described in the next section.

> **See Also:**
>
> - Re-creating Legacy Data Load Pages
> - Editing Pages in Page Designer

26.4.4 Re-creating Legacy Data Load Pages

To make changes to existing legacy Data Load pages, Oracle recommends re-creating new pages without deleting the data loading definitions.

To re-create legacy Data Load Wizard pages without deleting the data loading definitions:

1. Navigate to the Data Load Definitions page:
   a. On the Workspace home page, click **App Builder**.

   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Data Load Definitions, click **Data Load Definitions**.
   e. Expand the **Legacy Data Load** region and click the legacy data load.
2. Click the Legacy Data Load you want to re-create.

   The Data Load Table Details page appears.
3. From Tasks, click **Create Pages using this Legacy Data load**.

   The Page Attributes page of the Create Data Load Wizard appears.
4. For each page, edit the appropriate attributes and click **Next**.

   To learn more about any attributes, see field-level Help.
5. Click **Create**.

26.4.5 Editing a Legacy Data Load Definition

Edit the Data Load Definition in Shared Components.

A Data Load Definition is comprised of a data load table, table rules, and lookup tables used by the Data Load Wizard in your application. A data load table is an existing table in your schema that has been selected for use in the data loading process, to upload data.

To edit a Data Load Definition:

1. Navigate to the Data Load Definitions page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
   d. Under Data References, click **Data Load Definitions**.
   e. Expand the **Legacy Data Load** region and click **Legacy Data Load**.

   The Data Load Table Details page appears. To learn more about any attributes, see field-level Help.
2. For Data Load Definition:
   - Name - Name for the data load definition.
   - Table Name - Displays the name of underlying table to be used for data load definition.
   - Unique column 1 - The column name used as the primary unique key column during the data load process.
   - Case Sensitive - Identify whether the selected unique key column 1 is case sensitive. By default, this is set to No.
   - Unique column 2 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the second unique key column during the data load process.
   - Case Sensitive - Identify whether the selected unique key column 2 is case sensitive. By default, this is set to No.

- Unique column 3 - If the unique key definition of the selected table is a compound key, consisting of 2 or more columns, this column name is used as the third unique key column during the data load process.
- Case Sensitive - Identify whether the selected unique key column 3 is case sensitive. By default, this is set to No.
- Skip Validation - One step in data loading is to validate actions to be taken on records to be uploaded. Select Yes to skip validation.

3. Transformation Rules lists previously defined transformation rules.
   - To create a new rule:
     a. Click **Create Transformation Rule**.
     b. Edit the attributes.

        To learn more about an attribute, see field-level Help.
     c. Click **Create**.
   - To edit an existing rule:
     a. Click the rule name.
     b. Edit the attributes.
     c. Click **Apply Changes**.
4. Table Lookups map data in the import file to data found in another table.
   - To create a new Table Lookups:
     a. Click **Create Table Lookup**.
     b. Edit the attributes.
     c. Click **Create**.
   - To edit an existing Table Lookup:
     a. Click column name.
     b. Edit the attributes.
     c. Click **Apply Changes**.
5. Column Name Aliases define aliases to help users correctly identify the columns to upload.

   To add Column Name Aliases:
   a. Click **Edit List of Values**.
   b. Edit the attributes.
   c. Click **Create**.
6. From Concurrency Column Name, select a column to be used for concurrency management.

   Concurrency gives the developer the option to select a column to check the version of the data in the underlying table. This is particularly important if uploading into a table that is regularly updated.
7. Click **Apply Changes**.

26.5 Managing Legacy Tabular Forms

A legacy tabular form enables users to update multiple rows in a table at once from a single page.

> **Note:**
>
> A tabular form is a legacy application component. Although existing legacy tabular forms are supported, the creation of new legacy tabular forms has been desupported. Oracle recommends developers create interactive grids instead.

A legacy tabular form enables users to update multiple rows in a table at once from a single page. A legacy tabular form enables you to perform update, insert, and delete operations on multiple rows in a database table. Tabular forms include a built-in multiple row update process that performs optimistic locking behind the scenes to maintain the data integrity.

- Best Practices When Managing Tabular Forms
  Learn about best practices when managing tabular forms.
- How Legacy Tabular Forms Work
  Learn how how legacy tabular forms work.
- About Running a Legacy Tabular Form
  Learn about running a legacy tabular form.

> **See Also:**
>
> - Managing Interactive Grids
> - Developing Forms

26.5.1 Best Practices When Managing Tabular Forms

Learn about best practices when managing tabular forms.

Oracle recommends the following best practices when managing tabular forms:

- Do not modify of the select list of a SQL statement of a existing legacy tabular form. If you must modify the query, make sure the values of the updatable columns are not altered after being queried from the database by the Oracle APEX engine.
- Do not add custom JavaScript to display types that use display type specific JavaScript logic (for example, radio groups, simple checkboxes, and popup key LOVs).

26.5.2 How Legacy Tabular Forms Work

Learn how how legacy tabular forms work.

Tabular forms are based on a SQL query in which the query columns are tied to the underlying table columns. Unlike a single record from, tabular forms do not store data in session state.

When a tabular form page is submitted, the tabular form data is processed using built-in data manipulation language (DML), or a custom PL/SQL page processes.

A tabular form with automatic DML has two processes for multi-row operations:

- A Multi Row Update (MRU) process performs create and update operations.
- Multi Row Delete (MRD) process deletes requests on one or more rows.

MRU and MRD processes reference the underlying table name and the primary key columns. The primary key columns must be part of the tabular form SQL query, and the report columns must be either hidden or display-only (save state) in order for the DML processes to be able to identify the correct records. For new records, the primary key values must be `NULL` to be identified as new records. Delete operations are performed by referencing row number of the row to be deleted. Users can check one or more rows on the current page and when they click the Delete button, the row numbers identify the corresponding primary key value(s) and the matching rows are deleted.

Wizard generated tabular forms built-in DML includes a checksum column to prevent lost updates. Using a method called optimistic locking, a checksum is computed for each row when rendering a tabular form. This checksum is stored in an extra hidden tabular form column. Upon submission of the form for update, an additional set of checksums is computed on the then current data stored in the database. The new checksums are compared to those built when the tabular form was first rendered, and if there are any mismatches, the update is rejected, because the data in the database was either modified by another user or anther process in the time between the tabular form was rendered and the time the data was submitted.

26.5.3 About Running a Legacy Tabular Form

Learn about running a legacy tabular form.

When running a tabular form, data is rendered as a report having updatable columns shown using various form elements (including text fields, text areas, date pickers, select lists, radio groups, and so on).

Tabular Form

	Project	Task Name	Start Date	End Date	Status
	APEX Environment Cc	Identify server require	14-JAN-15	15-JAN-15	Closed
	Maintain Support Sys	HR software upgrade	14-JAN-15	11-MAR-15	On-Hold
	Maintain Support Sys	Apply Billing System	14-JAN-15	15-MAR-15	On-Hold
	APEX Environment Cc	Determine Web lister	15-JAN-15	15-JAN-15	Closed
	APEX Environment Cc	Specify security authe	16-JAN-15	18-JAN-15	Closed
	APEX Environment Cc	Select servers for Dev	16-JAN-15	21-JAN-15	Closed
	Email Integration	Complete plan	21-JAN-15	26-FEB-15	Closed
	APEX Environment Cc	Configure Workspace	23-JAN-15	23-JAN-15	Closed
	APEX Environment Cc	Create pilot workspac	23-JAN-15	23-JAN-15	Closed
	Maintain Support Sys	Arrange for vacation	04-MAR-15	27-MAR-15	Open

row(s) 1 - 10 of 73 Next

Add Row

Cancel Delete Apply Changes

For numeric and date fields, you can also pre-define date and number format masks, or apply those format masks after generating the initial form.

Each report also includes a checkbox column (called a row selector). Users use this row selector to select one or more columns and update or delete data. To add a row, users click **Add Row**.

> **Tip:**
>
> You can control the appearance of a specific column by editing column attributes.

> **See Also:**
>
> Managing UI Defaults in *Oracle APEX SQL Workshop Guide*

26.6 Managing Legacy Web Services

Legacy Web services enable applications to interact with one another over the web in a platform-neutral, language independent environment.

> **Note:**
>
> The SOAP 1.1 specification is a W3C note. SOAP Version 1.2 specification is a W3C recommendation.
>
> For information about Simple Object Access Protocol (SOAP) see:
>
> http://www.w3.org/TR/SOAP/

- About Legacy Web Services
  Learn about legacy Web services.
- Creating Legacy Web Service References
  Learn about creating legacy Web service references.
- About Working with SSL Enabled Legacy Web Services
  Learn about working with SSL enabled legacy Web services.
- Creating Legacy Web Service References Based on a WSDL
  Learn about creating legacy Web service references based on a WSDL.
- Creating Web Service References Manually
- Creating Legacy RESTful Web Service References
  Learn how to create legacy RESTful Web service references.
- Using the Legacy Web Service Reference Repository
  Learn about the legacy Web Service Reference Repository.
- Creating an Input Form and Report on a Legacy Web Service
  Learn how to create an input form and report on a legacy Web service.
- Creating a Form on a Legacy Web Service
  Learn how to create a form on a legacy Web service.
- Creating a Report on a Web Service
- How to Invoke a Legacy Web Service as a Process
  Learn how to invoke a legacy Web service as a process.
- Viewing a Legacy Web Service Reference History
  View a history of Web service reference changes.

26.6.1 About Legacy Web Services

Learn about legacy Web services.

In a typical Web services scenario, a business application sends a request to a service at a given URL by using the protocol over HTTP. The service receives the request, processes it, and returns a response. You can incorporate calls with external Web services in applications developed in App Builder.

Web services are typically based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) architectures. SOAP is a World Wide Web Consortium (W3C) standard protocol for sending and receiving requests and responses across the Internet. SOAP messages can be sent back and forth between a service provider and a service user in SOAP envelopes. RESTful Web services are resource oriented. The scope of the Web service is found in the URI and the method of the service is described by the HTTP method that is used such as GET, POST, PUT, HEAD, and DELETE.

SOAP offers two primary advantages:

- SOAP is based on XML, and therefore easy to use.
- SOAP messages are not blocked by firewalls because this protocol uses simple transport protocols, such as HTTP.

REST offers similar advantages:

- REST messages are also not blocked by firewalls because this protocol uses the HTTP protocol.
- REST requests do not require the overhead of XML and SOAP envelopes and inputs are typically provided in the URI.

> **See Also:**
>
> Enabling Network Services in Oracle Database

26.6.2 Creating Legacy Web Service References

Learn about creating legacy Web service references.

> **Note:**
>
> Creating a legacy Web service reference is only supported for applications that already have them.

To use Web services in APEX, you create a Web service reference using a wizard. Web service references can be based on a Web Services Description Language (WSDL) document, RESTful style, or created manually by supplying information about the service.

- Accessing the Web Service References Page
- Specifying an Application Proxy Server Address

> **See Also:**
>
> - Creating Legacy Web Service References Based on a WSDL
> - Creating Web Service References Manually
> - Creating Legacy RESTful Web Service References

26.6.2.1 Accessing the Web Service References Page

You manage Web service references on the Web Service References page.

To access the Web Service References page:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Data Sources, click **Legacy Web Service References**.

   The Web Service References page appears.

26.6.2.2 Specifying an Application Proxy Server Address

If your environment requires a proxy server to access the Internet, you must either specify a proxy server address on the Application Attributes page, or your instance administrator must define an instance proxy, before you can create a Web service reference.

To specify a proxy address for an application:

1. On the Workspace home page, click the **App Builder** icon.

   The App Builder home page appears.
2. Select an application.

   Application home page appears.
3. Click **Edit Application Definition**.
4. Under Properties, enter the proxy server in the Proxy Server field.
5. Click **Apply Changes**.

26.6.3 About Working with SSL Enabled Legacy Web Services

Learn about working with SSL enabled legacy Web services.

Secure Sockets Layer (SSL) is an industry standard protocol that uses RSA public key cryptography with symmetric key cryptography to provide authentication, encryption, and data integrity.

If the Web service that you need to interact with is SSL-enabled (that is, `https` displays in the URL to the Web service), you must create a wallet and configure APEX to use the wallet. A wallet is a password-protected container that stores authentication and signing credentials (including private keys, certificates, and trusted certificates) needed by SSL.

> **See Also:**
>
> Configuring Wallet Information in *Oracle APEX Administration Guide*

26.6.4 Creating Legacy Web Service References Based on a WSDL

Learn about creating legacy Web service references based on a WSDL.

You can create Web service reference based on a Web Services Description Language (WSDL) document.

- About Creating Web Service References Based on a WSDL
- Creating a Web Service Reference by Specifying a WSDL Document
- Testing a Web Service Reference Created from a WSDL

26.6.4.1 About Creating Web Service References Based on a WSDL

Before you create a Web service reference based on a WSDL, you must decide how to locate the WSDL. You locate a WSDL by entering the URL to the WSDL document.

You then run a wizard which analyzes the WSDL and collects all the necessary information to create a valid SOAP message, including:

- The URL used to post the SOAP request over HTTP(S)
- A Universal Resource Identifier (URI) identifying the SOAP HTTP request
- Operations of the Web Service
- Input parameters for each operation
- Output parameters for each operation

26.6.4.2 Creating a Web Service Reference by Specifying a WSDL Document

To create a Web service by specifying a URL to a specific WSDL document:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Click **Create**.
3. For Web Reference Type, select **Based on WSD**L and click **Next**.
4. If prompted to search a UDDI registry to find a WSDL, click **No** then **Next**.
5. In WSDL Location, enter the URL to the WSDL document.
6. For Username, enter the username required to access the WSDL.
7. For Password, enter the password required to access the WSDL.
8. Click **Next**.

   The Web Service Details page appears.
9. For Basic Authentication, select:
   - **No** - Select this option if authentication credentials are not sent as part of the HTTP request.

- **Yes** - Select this option if authentication credentials are sent as part of the HTTP request. Web services that employ basic authentication usually also use the HTTPS (SSL) protocol since the username and password are sent in clear text.

10. Click **Create Reference**.

    The Web service reference is added to the Web Service References Repository.

11. On the Create page, the following selections are available:
    - View Web Service References - Select to go to the Web Service Reference page.
    - Create Form on Web Service - Select to create a Form for this Web Service Reference.
    - Create Form & Report on Web Service - Select to create a Form and Report for this Web Service Reference.

26.6.4.3 Testing a Web Service Reference Created from a WSDL

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Select the **View Report** icon.
3. Click the **Test** icon for the Web Service reference you want to test.

   The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.
4. From Operation, select an operation (that is, the method to be executed).
5. Under Input Parameters, enter the appropriate values.
6. Click **Test**.

   The message request and response appear at the bottom of the page under Messages.

> **See Also:**
>
> Accessing the Web Service References Page

26.6.5 Creating Web Service References Manually

You can create a Web service reference manually.

- About Creating Web Service References Manually
- Creating a Web Service Reference Manually
- Testing a Web Service Reference Created Manually

26.6.5.1 About Creating Web Service References Manually

When you create a Web service reference manually, you supply the necessary information to create a valid SOAP request, including:

- The URL used to post the SOAP request over HTTP(S)
- A Universal Resource Identifier (URI) identifying the SOAP HTTP request
- The SOAP envelope for the request, including any item substitutions
- Optionally the name of a collection to store the response from the Web service

26.6.5.2 Creating a Web Service Reference Manually

Creating a Web service reference manually adds it to Web Service References Repository.

To create a Web service reference manually:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Click **Create**.
3. Select **Manual** and click **Next**.
4. On Create Web Service Reference:
   a. Name - Enter a name to identify the reference.
   b. URL - Enter the URL used to post the SOAP request over HTTP. This corresponds to the soap:address location of a service port in the WSDL.
   c. Action - Enter the intent of the SOAP HTTP request (optional). The value is a URI identifying the intent. SOAP places no restrictions on the format or specificity of the URI or requires that it is resolvable.
   d. Proxy - Enter a proxy to override the application proxy for this service.
   e. SOAP Version - Select **1.1** or **1.2**.
   f. Basic Authentication - Choose whether the Web service requires authentication. Select **Yes** or **No**.
   g. SOAP Envelope - Specify the SOAP envelope to be used for the SOAP request to the Web service..

      > **Note:**
      >
      > You can reference items from session state in the SOAP envelope by using `#ITEM_NAME#` syntax.

   h. Store Response in Collection - Enter the name of a collection to store the Web service response. The response will be stored in the CLOB001 column of the collection entered here.

i. Click **Create**.

> **See Also:**
>
> Accessing the Web Service References Page

26.6.5.3 Testing a Web Service Reference Created Manually

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Select the **View Report** icon.
3. Click the **Test** icon for the Web Service reference you want to test.

   The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.
4. If required, enter the username and password under Basic Authentication.
5. In SOAP Envelope text area, optionally edit the SOAP request envelope.
6. Click **Test**.

   The message request and response appear at the bottom of the page under Messages.

> **See Also:**
>
> Accessing the Web Service References Page

26.6.6 Creating Legacy RESTful Web Service References

Learn how to create legacy RESTful Web service references.

You can create a Representational State Transfer (REST) or RESTful web service.

- About Creating RESTful Web Service References
- Creating a RESTful Web Service Reference
- Testing a REST Web Service Reference

26.6.6.1 About Creating RESTful Web Service References

RESTful Web services are resource oriented. The scope of the Web service is found in the URI and the method of the service is described by the HTTP method that is used such as GET, POST, PUT, HEAD, and DELETE. When you create a RESTful Web service reference, you supply the necessary information about the structure of the request and response including:

- A Universal Resource Identifier (URI) identifying the RESTful request
- The HTTP method identifying the method of the Web service
- HTTP Headers, if required, that are part of the request
- The type of input expected by the Web service
- The format of the response and how to identify the response parameters

26.6.6.2 Creating a RESTful Web Service Reference

To create a RESTful Web service reference:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Click **Create**.
3. Select **REST** and click **Next**.
4. For REST Details, specify the following:
   a. Name - Enter a name to identify the reference.
   b. URL - Enter the URL endpoint of the Web service.
   c. Proxy - Enter a proxy to override the application proxy for this service (optional). This setting is overridden by the proxy setting for this instance.
   d. HTTP Method - Choose the http method used for the request to the Web service. Select **GET**, **HEAD**, **POST**, **PUT** or **DELETE**.
   e. Basic Authentication - Select **Yes** to require HTTP Basic Authentication. Otherwise, select **No**.
   f. REST HTTP Headers - Enter the names of the HTTP headers to send with the request.
   g. Click **Next**.
5. For REST Input parameters, specify the following:
   a. Name - Enter the name of the input parameter expected by the method.
   b. Type - Select the input type.
   c. Click **Add Parameter**.
   d. Repeat steps a though c for each expected input.
   e. Click **Next**.
6. For REST Output parameters, specify the following:

a. Output Format - Select **XML**, **Text** or **JSON** for the response format expected from the Web service.

b. XPath to Output Parameters (XML only) - Enter an XPath expression to the relevant part of the response.

   For example:

   ```
   /ListBucketResult/Contents
   ```

c. Response Namespace (XML only) - Enter the namespace corresponding to the Response XPath.

   For example:

   ```
   http://s3.amazonaws.com/doc/2006-03-01/
   ```

d. Parameter Delimiter (**Text** only) - Enter the character or sequence that separates parameters returned from the Web service. Use `\n` to indicate a new line and `\t` to indicate a tab character.

e. New Record Delimiter (**Text** only) - Enter the character or sequence that determines a new record in a text response from the Web service. Use `\n` to indicate a new line and `\t` to indicate a tab character.

f. Name - Enter the name of the output parameter returned by the method.

g. Path - Enter the path. If the response is XML, the path is an XPath expression to the node. If the response is Text, the path is a number. If the response is JSON, the entire response is stored in the `CLOB001` column of a collection you specify.

h. Type - Select the output type.

i. Click **Add Parameter**.

j. Repeat steps f though h for each returned output parameter.

> **Note:**
>
> Click **Test** to send a request to a RESTful Web service and see the response. This test process helps you specify the appropriate Output parameters.

7. Click **Create**.

The Create Web Service Reference Success page appears. The Web service reference is added to the Web Service References Repository.

> **See Also:**
>
> - Accessing the Web Service References Page
> - Creating a Proxy Server for an Instance in *Oracle APEX Administration Guide*

26.6.6.3 Testing a REST Web Service Reference

After you have created a Web service reference, you can test it on the Test Web Service Reference page.

To test a Web service reference:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.
2. Select the **View Report** icon.
3. Click the **Test** icon for the Web Service reference you want to test.

   The Test Web Service Reference page appears. The Web service name and URL endpoint display at the top of the page.
4. In the URL and Proxy Override fields, optionally edit the values for the test.
5. If required, enter the username and password under Basic Authentication.
6. Under HTTP Headers, enter appropriate values.
7. Under Input Parameters, enter appropriate values.
8. Click **Test**.

   The message response appears at the bottom of the page under Messages.

> **See Also:**
>
> Accessing the Web Service References Page

26.6.7 Using the Legacy Web Service Reference Repository

Learn about the legacy Web Service Reference Repository.

Web service references are stored in the Web Service Reference Repository.

To access the Web Service References Repository:

1. Navigate to the Web Service References page.
   a. On the Workspace home page, click **App Builder**.
   b. Select a existing application.
   c. On the Application home page, click **Shared Components** in the center of the page.
   d. Under Data Sources, select **Legacy Web Service References**.

   The Web Service Reference page appears.

   A Search bar displays at the top of the page. Available controls include:

   - **Search columns icon** - Resembles a magnifying glass. Click this icon to narrow your search. To search all columns, select **All Columns**.
   - **Text area** - Enter case insensitive search criteria (wildcard characters are implied) and click **Go**.
   - **Go button** - Executes a search or applies a filter.

- **View Icons** - Displays each Web service reference as a large icon. To edit a Web service reference, click the appropriate icon.
- **View Report** - Displays each Web service reference as a line in a report.
- **Actions menu** - Displays the Actions menu. Use this menu to customize the report view.

2. Click the **View Report** icon.
3. In report view you can:
   - Edit a reference by clicking the reference name.
   - Test a reference by clicking the **Test** icon.
   - View details about a reference by clicking the **View** icon. Note that this option is not available for manually created or REST Web service references.

> **See Also:**
>
> - Accessing the Web Service References Page
> - About the Actions Menu

26.6.8 Creating an Input Form and Report on a Legacy Web Service

Learn how to create an input form and report on a legacy Web service.

- About the Create Form and Report on Web Service Wizard
- Creating a Form and Report by Adding a New Page

26.6.8.1 About the Create Form and Report on Web Service Wizard

The Create Form and Report on Web Service Wizard creates an input form, a submit button, and a report for displaying results. You can execute this wizard directly after creating the Web service reference from a WSDL or a RESTful style Web service, or by adding a page.

Use this wizard when you expect a nonscalar result from the Web service. The Amazon Product API Web service is a good example. This Web service returns many results based on the search criteria entered in an input form.

26.6.8.2 Creating a Form and Report by Adding a New Page

If you have an existing Web service reference, you can create an input form and report by adding a new page.

To create a form and report by adding a new page:

1. Create the Web service reference.
2. Select the application.
3. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Click **Legacy Pages** and select **Report and Form on Legacy Web Service**.

5. For Choose Service and Operation:
   a. Web Service Reference - Select the Web service reference.
   b. Operation - Select the method to be executed. For RESTful style Web references, doREST is selected automatically. The Operation option does not appear for Manual style Web references.
   c. Click **Next**.
6. For Page and Region Attributes, review and update the page and region attributes and click **Next**.

> **Note:**
>
> If the page you specify does not exist, the wizard creates the page for you.

7. For Input Items:
   a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.
   b. If necessary, edit the Item Name.
   c. If necessary, edit the Item Label.
   d. Click **Next**.
8. If applicable, specify the Item Names and Item Labels for basic authentication, then click **Next**.

> **Note:**
>
> This step only appears if basic authentication was specified for this Web service reference when it was created.

9. Follow the on-screen instructions.
10. Click **Create**.

> **See Also:**
>
> - Creating Legacy Web Service References Based on a WSDL
> - Creating Web Service References Manually
> - Creating Legacy RESTful Web Service References

26.6.9 Creating a Form on a Legacy Web Service

Learn how to create a form on a legacy Web service.

- About the Create Form on Web Service Wizard
- Creating a Form by Adding a New Page

26.6.9.1 About the Create Form on Web Service Wizard

The Create Form on Web Service Wizard creates a form and a submit button. You can execute this wizard after creating the Web service reference from a WSDL or on a RESTful style Web service, or by running the Create Page Wizard.

Use this wizard when you expect a scalar result from the Web service. A Web service that looks up a stock price is a good example because the input is a stock symbol and the output is the scalar value price.

26.6.9.2 Creating a Form by Adding a New Page

If you have an existing Web service reference, you can create a form based on a Web service reference by adding a new page.

To create a form by adding a new page:

1. Create the Web service reference.
2. Select the application.
3. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.
4. Click **Legacy Pages** and select **Form on Legacy Web Service**.
5. For Page Attributes:
   a. Page Number - Select a page to contain the new form, or enter a new page number.
   b. Page Name - Specify a name for the page.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
   d. Page Group - Identify the name of the page group you would like to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list of values.
   e. Region Title - Select whether you want to use a breadcrumb navigation control on your page, and which breadcrumb navigation control you want to use.
   f. Breadcrumb - Enter a title for the region in which the form will appear.
   g. Click **Next**.
6. For Navigation Preference:
   a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
   b. Click **Next**.
7. For Choose Service:
   a. Web Service Reference - Select the Web service reference.
   b. Operation - Select the method to be executed. For RESTful style Web references, doREST is selected automatically. The Operation option does not appear for Manual style Web references.
   c. Click **Next**.
8. For Input Items, make these selections and click **Next**:

a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.

b. If necessary, edit the Item Name.

c. If necessary, edit the Item Label.

9. If applicable, specify the Item Names and Item Labels for basic authentication and click **Next**.

> **Note:**
>
> This step only appears if basic authentication was specified for this Web service reference when it was created.

10. If applicable, specify the Item Names and Item Labels for basic authentication.

    Note that this step only appears if basic authentication was specified for this Web service reference when it was created.

11. For Output Items, make these selections and click **Next**:

    a. Identify which items to add to the form. To include an item, select **Yes** in the Create column. Otherwise, select **No**.

    b. If necessary, edit the Item Name.

    c. If necessary, edit the Item Label.

12. Click **Create**.

> **See Also:**
>
> - Creating Legacy Web Service References Based on a WSDL
> - Creating Web Service References Manually
> - Managing Pages in an Application
> - Editing User Interface Attributes

26.6.10 Creating a Report on a Web Service

If you have an existing Web service reference, you can create a report based on a Web service reference by adding a new page.

To create a report by adding a new page:

1. Create the Web service reference.
2. Select the application.
3. On the Application home page, click **Create Page**.

   Create a Page appears and features three tabs: **Component**, **Feature**, and **Legacy Pages**.

4. Click **Legacy Pages** and select **Report on Legacy Web Service**.
5. For Page Attributes:

   a. Page Number - Select a page to contain the new form, or enter a new page number.
   b. Page Name - Specify a name for the page.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
   d. Breadcrumb - Enter a title for the region in which the form will appear.
   e. Click **Next**.
6. For Navigation Preference:
   a. Select how you want this page integrated into the Navigation Menu. To learn more, see field-level Help.
   b. Click **Next**.
7. If Web Reference Type is **REST**:
   a. Web Reference Type - Choose **REST**.
   b. Web Service Reference - Select the name of the Web service reference and click **Next**.
   c. Result Stored in Collection - Enter the name of the temporary result set name where the Web service result is stored. The temporary result set is stored in an Application Express collection.
   d. Report Template - Choose the report template to be used by this report region.
   e. Rows Per Page- Select the number of rows to display per page.
   f. Select the parameters to be included in the report.
   g. Click **Next**.
8. For Web Reference Type, choose **REST**, **Generated from WSDL** , or **Manually Created**.

   What appears next depends upon your selection.
9. If Web Reference Type is **Generated from WSDL**:
   a. Web Reference Type - Choose **Generated from WSDL**.
   b. Web Service Reference - Select the name of the Web service reference.
   c. Operation - Select the operation. The operation is the method that will be executed..
   d. Result Tree to Report On - Select the portion of the resulting XML document that contains the information you want to include in the report.
   e. Report Parameters and Options:
      - Temporary Result Set Name - Enter a name for the collection that stores the Web service result.
      - Select and deselect the appropriate parameters.
   f. Click **Next**.
10. If Web Reference Type is **Manually Created**:
    a. Web Reference Type - Choose **Manually Created**.
    b. Web Service Reference - Select the name of the Web service reference.
    c. SOAP Style - Choose the style of the SOAP response. The style can be determined from the style attribute of the soap:binding element from the WSDL for this operation.
    d. Message Format - Choose the message format. The encoding can be determined from the use attribute of the soap:body element of the output message in the WSDL for this operation.

e. Result Node Path (XPath)(Value Required)- Enter the XPath expression to the node of the response message that you are creating the report on. The node is a child of the soap:Body element.

   For example:

   ```
   /result/myNode
   ```

f. Message Namespace - Enter the namespace for the response message. This can typically be determined by looking for the targetNamespace attribute of the WSDL and click **Next**.

g. Enter the name of the collection where the response message is stored.

h. Enter the names of the parameters that you want to be included in the report.

i. Click **Next**.

11. Click **Create**.

26.6.11 How to Invoke a Legacy Web Service as a Process

Learn how to invoke a legacy Web service as a process.

You can invoke a Web service as a process on the page. The process is created for you if you run one of the Create Form wizards for Web services. Running the process submits the request to the service provider. You can then display the request results in the report.

- Invoking a Web Service as a Process
- Editing a Web Service Process

26.6.11.1 Invoking a Web Service as a Process

To invoke a Web service as a process:

1. Create a Web Service Reference.
2. Run the Create Page Wizard. On Create a Page:
   a. For User Interface, if applicable select a user interface for the page.

      This attribute only displays for applications using older themes for which Desktop and Mobile User Interfaces have been defined.

   b. For Page Type, select **Component** and then **Blank Page**.
3. For Page Attributes:
   a. Page Number - Select a page to contain the new form, or enter a new page number.
   b. Page Name - Specify a name for the page.
   c. Page Mode - Identify the page mode. To learn more, see field-level Help.
   d. Page Group - Identify the name of the page group you would like to associate with this page. Page groups help developers manage the pages within an application. To create a page group, enter the name. To use an existing page group, select the name from the list of values.
   e. Breadcrumb - Enter a title for the region in which the form will appear.
   f. Click **Next**.
4. For Navigation Preference:
   a. Select **Do not associate this page with a navigation menu entry**.

   b. Click **Next**.

5. Click **Finish**.

   Page Designer appears.

6. Under Page Rendering or Page Processing, locate the Processes section, right-click and select **Create Process**.

   The New process appears.

7. Edit the process attributes in the Property Editor.

   Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

8. Edit **Identification** attributes:

   a. Find the **Identification** group.

   b. Name - Enter the name of the page process for easy identification by developers.

   c. Type - Select **Web Service**.

9. Edit **Settings**:

   a. Find the **Settings** group.

   b. Web Service Operation - Select a Web Service Operation.

   c. Store Result In - Select **Item(s)** or **Collection**.

10. Edit **Execution Options** Options:

    a. Find the **Execution Options** group.

    b. Sequence - Specify the sequence for this component. The sequence determines the order of execution.

    c. Point - Select the processing point.

11. Edit **Success Message**:

    a. Find the **Success Message** group.

    b. Success Message - Enter the success message for this process. If the process runs and does not generate an error, then this process success message displays in the notification section of the resulting page displayed.

       To learn more, click the **Help** tab.

12. Edit **Error Message**:

    a. Find the **Error Message** group.

    b. Error Message - Enter the error message for this process. This message displays if an unhandled exception is raised. After any error processing stops, a rollback is issued and an error message displays..

       To learn more, click the **Help** tab.

13. Edit **Server-side Condition**:

a. Find the **Server-side Condition** group.

b. When Button Pressed - If you want this page processing component to execute only when the specified button is clicked, select a button from the list. You can incorporate this button condition with other conditions to further refine when this page processing component executes.

c. Condition Type - Select a condition type that must be met in order for this component to be processed. Depending upon your selection, additional attributes may appear.

14. Click **Save**.

> **See Also:**
>
> - Creating Legacy Web Service References Based on a WSDL
> - Creating Web Service References Manually
> - Creating a RESTful Web Service Reference
> - Managing Pages in an Application

26.6.11.2 Editing a Web Service Process

After you create a process of type Web service on a Web service reference created from a WSDL or a RESTful style Web reference, you can change the attributes of the input and output parameters to the service.

To edit a Web service process:

1. Create a Web service process.
2. Navigate to the page containing the Web service process.

   Page Designer appears.
3. In the left pane, find the process name and expand the tree.

   In Parameters and **Out Parameters** display under the process.
4. Expand **In Parameters** and **Out Parameters**. Defined parameters apear.
5. Select a parameter and edit the attributes in the Property Editor.

   Attributes are organized in groups. To find a group or attribute, enter keywords in the Filter Properties field. The Property Editor displays the group or attributes. Or, you can click **Go to Group** and select the group.

   > **Tip:**
   >
   > To view help for an attribute, select the attribute in the Property Editor and click the **Help** tab in the central pane.

6. Click **Save**.

> **See Also:**
>
> How to Invoke a Legacy Web Service as a Process

26.6.12 Viewing a Legacy Web Service Reference History

View a history of Web service reference changes.

The Web Services History displays changes to Web service references for the current application by application ID, Web service references name, developer, and date.

To view a history of Web service reference changes:

1. Navigate to the Shared Components page:
   a. On the Workspace home page, click **App Builder**.
   b. Select an application.
   c. On the Application home page, click **Shared Components**.

      The Shared Components page appears.
2. Under Data Sources, click **Legacy Web Service References**.

   The Web Service References page appears.
3. Click **History**.

> **Note:**
>
> The History button only appears on the Web Service Reference page if at least one reference exists.

A

About Item Types

Review descriptions of the item types available in App Builder and important attributes in Page Designer that control item behavior.

An item can be part of an HTML form or used in an interactive grid column. An item can be a text field, text area, password, select list, checkbox, and so on. Developers can control how a page item works by editing the item attributes in Page Designer. To learn more about specific item attributes, select the attribute in Page Designer and select the Help tab.

> **Tip:**
>
> Your application may also include plug-in items. Plug-ins display as <*My Plug-in*> [*Plug-in*]. See Implementing Plug-ins.

- Checkbox
  Displays a single checkbox, enabling the user to check or uncheck it.
- Checkbox Group
  Displays multiple values as checkboxes, enabling the end user to select multiple values.
- Color Picker
  Displays a control for choosing or entering a color depending on the Display As setting. End users can enter a color value or select one from an inline or popup color spectrum.
- Combobox
  Displays a text item with a list of values icon.
- Date Picker
  Displays a text field with a calendar icon. End users can either enter the date directly into the text item, or click on the calendar icon to select a date, and optionally a time, from the calendar popup.
- Date Picker (jQuery) (Desupported)
  Displays a text field with a calendar icon. End users can either enter the date directly into the text item, or click on the calendar icon to select a date, and optionally a time, from the calendar popup.
- Display Image
  Displays an image stored in a database `BLOB` columns, or based on an image URL.
- Display Map
  Displays a map based on coordinates in GeoJSON point format.
- Display Only
  Displays a non-enterable text item.
- File Upload
  Displays a text field with a Browse button. The Browse button enables the user to locate and upload a file from a local file system.
- Geocoded Address
  Provides geocoding functionality that turns a postal address to a coordinate.

- Image Upload
  Displays an Image Upload item which enables users to upload one or multiple images from a local file system or device. Images can be optionally cropped or resized.
- Hidden
  Hidden items are included within the page source but are not rendered.
- Markdown Editor
  Displays a simple text editor that enables you to use Markdown shorthand syntax to add formatting to text.
- List Manager
  Displays a text item with a popup list of values icon, Add and Remove buttons, and a list of selected values.
- Number Field
  Displays as a number field and supports automatic formatting.
- Password
  Displays an HTML password form element. As the end user enters text a black dot is displayed for that character, instead of the actual character entered.
- Percent Graph
  Displays the value as a percentage graph. The value retrieved must be between 0 and 100.
- Popup LOV
  Displays a text item with a popup list of values icon. When the end user clicks the icon, a popup window appears with a search field, and a list of supported values.
- QR Code
  Encodes the selected source value into a QR code and displays it.
- Radio Group
  Displays multiple values as radio group options, enabling the end user to select a single value.
- Rich Text Editor
  Displays a text area with comprehensive text formatting options.
- Select List
  Displays an item with a built-in list of values selector. When the end user clicks the item, the list of supported values displays directly inline with the current item.
- Select Many
  Displays a drop-down list that supports multi-select and search filtering.
- Select One
  Displays as an item with a list of values icon which supports the selection of one value and search filtering.
- Shuttle
  Displays as a multiple select list that includes two boxes containing lists. The left list displays the source list of values that have not been selected and the right list shows the currently selected values.
- Textarea
  Displays a multiple-row text area.
- Star Rating
  Displays a star rating (maximum of 10) that end users can click to set a numeric value.
- Switch
  Displays as a flip toggle switch.

- Star Rating
  Displays a star rating (maximum of 10) that end users can click to set a numeric value.
- Text Field
  Displays the item as a text field.
- Text Field with Autocomplete
  Displays a text field that displays a list of possible values based on the text already entered by the end user, inline with the text item. The list is further refined as the end user types in more text.
- Textarea
  Displays a multiple-row text area.

> **See Also:**
>
> Managing Page Items

A.1 Checkbox

Displays a single checkbox, enabling the user to check or uncheck it.

A checkbox item always has one of two values, the checked value or unchecked value.

Page Designer attributes that control behavior:

- **Settings**, **Use Defaults** - If set to **On**, the checkbox uses the application-level component settings defined in Shared Components, Component Settings, checkbox.

  See Also: Configuring Checkbox

  If **Use Defaults** is set to **Off**, the following custom settings appear:

  – **Checked Value** - Enter the value stored in the page item if the end user checks the checkbox. The value can be up to 255 characters long.

    On a Faceted Search page, if the facet filters a query column *based on a List Of Values*, this value must match *both* LOV display and return value. If display and return values differ, then you cannot use this facet type for this column - use the **Checkbox Group** facet instead.

  – **Unchecked Value** - Enter the value stored in the page item if the user unchecks the checkbox. This attribute can also be left blank if you want to store NULL when checked. The value can be up to 255 characters long.

    Note: This value is used if the page item is populated with a value which does not match either the Checked Value or Unchecked Value.

A.2 Checkbox Group

Displays multiple values as checkboxes, enabling the end user to select multiple values.

> **Note:**
>
> For Yes/No checkboxes (that is, a checkbox with only one value), Oracle recommends using a checkbox or a Switch item type.

A list of values is required for items displayed as checkboxes. The values corresponding to the checked boxes are stored in a single colon-delimited string.

The following example displays employee names (`ename`), but returns employee numbers (`empno`):

```
SELECT ename, empno FROM emp;
```

> **See Also:**
>
> APEX_UTIL in *Oracle APEX API Reference* for information about breaking up returned values

Page Designer attributes that control behavior:

- **Settings**, **Number Columns** - Enter the number of layout columns used to display the checkboxes. For example, a value of 2 would display two columns of checkboxes. By default, all values display in one column.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help or Configuring Multiple Value Attributes.

> **See Also:**
>
> Creating a Cascading List of Values

A.3 Color Picker

Displays a control for choosing or entering a color depending on the Display As setting. End users can enter a color value or select one from an inline or popup color spectrum.

Page Designer attributes that control behavior:

- **Settings**, **Use Defaults** - If set to **On**, the Color Picker uses the application-level component settings defined in Shared Components, Component Settings, Color Picker.

  > **See Also:**
  >
  > Configuring Color Picker Display

  If **Use Defaults** is set to **Off**, the following settings appear:

  - **Display As** - Select how the color picker displays. Options include:
    * **Popup** - Displays as a text field with icon to open the color spectrum picker.
    * **Inline** - Displays as an inline color spectrum.
    * **Color-only Popup** - Displays as a button which opens the color spectrum popup.
    * **Native Color Picker** - Displays as a native browser color input (the UI is dependent on the browser).
  - **Mode** - Select the amount of information shown in the color picker. Options include **Simple** or **Full**.
  - **Color Preset** - Determines if the color picker should display a set of predefined colors and whether they are from the theme or custom. Options include **None**, **Theme Style Colors**, or **Custom Colors**.
  - **Contrast With** - Compare the color of this item with a color from another item or a static value and display the contrast ratio between them in the color picker UI. Enabling this option can help the user choose colors with enough contrast to be accessible to everyone. Options include: **None**, **Page Item**, or **Static Color**.
  - **Return Value As** - Specify the format of the color value returned by this item. Even if the color is entered in a different valid format it will be presented, returned, and sent to the server in this format. Options include: **Hex**, **rgb**, **rgb/rgba**, **hsl**, **hsl/hsla**, or **Any CSS Value**. See Help for more details.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.4 Combobox

Displays a text item with a list of values icon.

Oracle recommends using a combobox item for small lists where end users can add their own text. When the end user clicks the icon, a popup window appears with a suggestion dropdown of values. A combobox item supports filtering and free text input. If the **Multiple Values** attribute is enabled, the combobox displays selected options as chips.

> **Tip:**
>
> For selections without free text input, consider using a Select List, Select One, or Select Many. For large result sets, consider a Popup LOV. See Select List, Select One, Select Many, and Popup LOV.

Page Designer attributes that control behavior:

- **Settings**, **Multi Selection** - Controls if the suggestion drop-down should remain opened after selection. If enabled it will add a checkbox next to each option to indicate the selected status. This option is only applicable for multi-value items.
- **Settings**, **Maximum Values in List** - Enter the maximum number of options to display in the select box. If not specified, up to 250 options display. This value can help improve performance, by avoiding returning too many rows, and also prevent a very large return set when only a few letters have been entered.
- **Settings**, **Fetch on Search** - Specify whether matching values are retrieved from the database as the user types on the item. **Fetch on Search** retrieves matching records from

the database each time the user types a character. Otherwise, all of the list values are loaded on page load.

Oracle recommends using **Fetch on Search** when you have a large select list, as loading a large number of records may impact the time to initially load the page. See Help to learn more.

- **Settings**, **Manual Entries Item** - The specified item holds manual entered entries that are not found on the LOV data-set. Use this configuration if you are planning to further process the Combobox value, and you need to know if a value exists on the data-set or not. See Help to learn more.
- **Settings**, **Value HTML Expression** - Define how the choices are rendered. This attribute supports HTML and template directives. See Using Template Directives.

  For example:

```
<div class="your-container">
    <span class="your-item">Name: &NAME.</span>
    {if ?CAPITAL/}<span class="your-item">Capital: &CAPITAL.</span>{endif/}
    {if ?NATIONALITY/}<span class="your-item">Nationality: &NATIONALITY.</
span>{endif/}
    {if ?POPULATION/}<span class="your-item">Population: &POPULATION.</
span>{endif/}
</div>
```

- **Search** , **Match Type** - Select how the search against the entered value is performed. When using **Fetch On Search**, this setting can have an impact on query performance. Options include:
  - **Contains** (prevents the use of column indexes)
  - **Starts With**
- **Search** , **Case-Sensitive** - Controls if the search is case-sensitive.
- **Search** , **Minimum Characters** - Specify the minimum number of characters that must be entered before a search is executed.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help or Configuring Multiple Value Attributes.
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.

    When creating a manual entries item, Oracle recommends selecting **Per Request (Memory Only)**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.
  - **Per User (Persistent)** - Maintain for a user by storing the value in the database, to access it across sessions.

> **Tip:**
>
> To create this item type, you must define a list of values. See Creating Lists of Values at the Application-Level, Working with Multiple Select List Item, and Creating a Cascading List of Values.

A.5 Date Picker

Displays a text field with a calendar icon. End users can either enter the date directly into the text item, or click on the calendar icon to select a date, and optionally a time, from the calendar popup.

Time is only displayed in the calendar popup if the Format Mask for this item includes time components.

Page Designer attributes that control behavior:

- **Settings**, **Show Time** - Enable this option to include time information.
- **Settings**, **Display As** - Select the display mode for the date picker. Options include:
  - **Popup** - Displays in a popup when the item receives focus, or the calendar icon is clicked.
  - **Inline** - The calendar always displays inline on the page. No text field is displayed and the current date selection is shown highlighted in the calendar.
  - **Native HTML** - Renders using the native HTML input date types, if supported by the browser. When 'Show Time' is off, this renders a 'date' type, when 'Show Time' is on, this renders a 'datetime-local' type. Note: The exact display of these date types is dependent on the browser. Native HTML may be the preferred option when the app is being used on a mobile device.

    Note regarding using **Native HTML**. Because the native date types are very specific about the date format they require, the item's Format Mask will be automatically set by APEX, and you will not be able to change it. However this will not be the format displayed to the end user. The end user will see the date in the format appropriate to their current locale.
- **Settings**, **Minimum Date** - Specify if this item has a minimum date. Options include:
  - **None** - No minimum date defined.
  - **Item** - Minimum date is derived dynamically from another item's value.
  - **Static** - Minimum date is a static value, defined as either an absolute value, or relative value to today's date.

  Additional attributes display conditionally based on the selected value.
- **Settings**, **Maximum Date** - Enter the maximum date that can be input or selected. Additional attributes display conditionally based on the selected value.
- **Settings**, **Multiple Months** - Determines if the calendar displays multiple months and how many. Select **No** to disable this option, or select the number of month.
- **Settings**, **Use Defaults** - If set to **On**, the Date Picker uses the application-level component settings defined in Shared Components, Component Settings, Date Picker. See Configuring Date Picker

  If **Use Defaults** is set to **Off**, the following attributes appear:

- **Appearance & Behavior** - Control the appearance of the calendar used to select dates. This attribute only applies when **Display As** is either **Popup** or **Inline**. Options include:
  - **Show Week** - Determines whether the week of the year displays.
  - **Month Picker** - Controls whether the month renders as a button which enables users to make another selection, or just as text.
  - **Year Picker**- Controls whether the year renders as a button which enables users to make another selection, or just as text.
  - **Paginate by Multiple Months** - Determines whether pagination size when displaying several months is only one month or the number of months.
  - **Show Today Button** - Determines whether a today button displays to select the current date.
  - **Show Clear Button** - Controls whether a clear button displays to clear the value of the date picker. This attribute only applies if **Display As** is set to **Inline**.
- **Days Outside Month** - Dictates the behavior of days outside the current viewing month. This attribute only applies if **Display As** is set to either **Popup** or **Inline**. Options include:
  - **Hidden** - Days outside the current viewing month are hidden.
  - **Selectable** - Days outside the current viewing month are visible and selectable.
  - **Visible** - Days outside the current viewing month are visible.
- **Time Increment** - Time increment in minutes used by the time selection control. This attribute is used for selecting the time of increments that can still be manually entered and saved.

  This attribute is only used when **Display As** is set to either **Popup** or **Inline** and **Show Time** is enabled.
- **Show On** - Controls when the calendar popup displays. This option is only used when **Display As** is set to either **Popup** or **Inline**. Options include:
  - **Item Focus** - When the item receives focus or when the calendar icon is clicked.
  - **Icon Click** - Only when the calendar icon is clicked.

- **Appearance**, **Format Mask** - Enter a date format mask to apply to this component.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.6 Date Picker (jQuery) (Desupported)

Displays a text field with a calendar icon. End users can either enter the date directly into the text item, or click on the calendar icon to select a date, and optionally a time, from the calendar popup.

> **Note:**
>
> The Date Picker (jQuery) item type is desupported and should no longer be used. Existing apps do not automatically switch to using the new date picker to preserve possible customizations. You can switch all the date pickers in your app to use the new date pickers by using Upgrade Application in the Utilities section of your application in App Builder. See Running Application Upgrade

Time is only displayed in the calendar popup if the Format Mask for this item includes time components.

Page Designer attributes that control behavior:

- **Settings**, **Show Time** - Enable this option to include time information.
- **Settings**, **Display As** - Select the display mode for the date picker. Options include:
  - **Popup Calendar** - Displays in a popup when the item receives focus, or the calendar icon is clicked.
  - **Inline** - The calendar always displays inline on the page. No text field is displayed and the current date selection is shown highlighted in the calendar.
  - **Native HTML** - Renders using the native HTML input date types, if supported by the browser. When 'Show Time' is off, this renders a 'date' type, when 'Show Time' is on, this renders a 'datetime-local' type. Note: The exact display of these date types is dependent on the browser. Native HTML may be the preferred option when the app is being used on a mobile device.

    Note when using **Native HTML**. Because the native date types are very specific about the date format they require, the item's Format Mask will be automatically set by APEX, and you will not be able to change it. However this will not be the format displayed to the end user. The end user will see the date in the format appropriate to their current locale.
- **Settings**, **Minimum Date** - Specify if this item has a minimum date. Options include:
  - **None** - No minimum date defined.
  - **Item** - Minimum date is derived dynamically from another item's value.
  - **Static** - Minimum date is a static value, defined as either an absolute value, or relative value to today's date.

  Additional attributes display conditionally based on the selected value.
- **Settings**, **Maximum Date** - Enter the maximum date that can be input or selected. Additional attributes display conditionally based on the selected value.
- **Settings**, **Multiple Months** - Determines if the calendar displays multiple months. If set to **On**, the **Number of Months** attribute appears. Enter the number of months to show at once.

- **Settings**, **Use Defaults** - If set to **On**, the Date Picker uses the application-level component settings defined in Shared Components, Component Settings, Date Picker.

  See Also: Configuring Date Picker

  If **Use Defaults** is set to **Off**, the following attributes appear:

  – **Appearance** - Control the appearance of the calendar used to select dates. Options include:

    * **Show Week**
    * **Month Picker**
    * **Year Picker**

    This attribute is only used if **Display As** is set to either **Popup** or **Inline**.

  – **Days Outside Month** - Dictates the behavior of days outside the current viewing month. Options include:

    * **Hidden** - Days outside the current viewing month will be hidden.
    * **Selectable** - Days outside the current viewing month will be visible and selectable.
    * **Visible** - Days outside the current viewing month will be visible.

    This attribute is only used if **Display As** is set to either **Popup** or **Inline**.

  – **Time Increment** - Time increment in minutes used by the time selection control. This attribute is used for selecting the time of increments that can still be manually entered and saved.

    This attribute is only used when **Display As** is set to either **Popup** or **Inline** and **Show Time** is enabled.

  – **Show On** - Controls when the calendar popup is displayed. Options include:

    * **Item Focus** - When the item receives focus or when the calendar icon is clicked.
    * **Icon Click** - Only when the calendar icon is clicked.

    This option is only used when **Display As** is set to either **Popup** or **Inline**.

- **Appearance**, **Format Mask** - Enter a date format mask to apply to this component.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.7 Display Image

Displays an image stored in a database `BLOB` columns, or based on an image URL.

> **Note:**
>
> This item type is not supported for interactive grid columns.

Page Designer attributes that control behavior:

- **Settings**, **Based On** - Determines where the item is retrieved from. To learn more about an attribute, see field-level Help.

- **Settings**, **Alternative Text Column** - Enter the uppercase page or application item name which holds the alternative text for this image. This text is read by screen readers when an end user navigates to this item.
- **Settings**, **Filename Column** - Enter the uppercase page or application item name that holds the filename column value associated with the `BLOB` column.
- **Settings**, **MIME Type Column** - Enter the uppercase page or application item name used to store the mime type of the uploaded file. The mime type is used by the browser to identify which application is used to display the content of the BLOB when the file is downloaded.
- **Settings**, **BLOB Last Updated Column** - Enter the uppercase page or application item name that holds the date column value when the associated `BLOB` was last updated. This value is used for browser caching and to identify if the image has been changed since the last request from the browser. If the image is not updated then the image is retrieved from the browser cache, if available.

A.8 Display Map

Displays a map based on coordinates in GeoJSON point format.

Page Designer attributes that control behavior:

- **Settings**, **Background** - Select an out-of-the-box map background or choose a custom map background defined in Shared Components.

  > **Note:**
  >
  > To learn more about contributors and licensing for out-of-the-box map backgrounds or to view them on Gitbub, see Page Designer Help.

  Options include:

  - **Default** - Use the default map background. (either **OpenStreetMap Bright** or **OpenStreetMap Dark** for dark themes).
  - **OpenStreetMap Bright** - General purpose basemap based on OpenStreetMap data. This basemap style provides detailed location context for your data.
  - **OpenStreetMap Dark** - Non-obtrusive dark basemap based on OpenStreetMap data. This basemap style accentuates visualizations of your data.
  - **OpenStreetMap Positron** - Non-obtrusive light basemap based on OpenStreetMap data. This basemap style accentuates visualizations of your data.
  - **Oracle World Map** - General purpose basemap based on HERE Map data. This base map style provides location context for your data and accentuates visualization of your data.
  - **Shared Component** - Use custom map backgrounds defined in Shared Components.
- **Settings**, **Custom Background** - Only displays if **Background** is set to **Shared Component**. Select a background map to use.
- **Settings**, **Zoom Level** - Specify the zoom level for this layer to become visible. Minimum value is 1. Maximum value is 18.
- **Settings**, **Show Marker** - Specify whether a marker should be displayed..

- **Settings**, **Marker Color** - Specify the marker fill color. See Help for more details.
- **Settings**, **Tooltip** - Specify the tooltip text for a displayed marker. The tooltip attribute supports HTML and APEX template directives.
- **Settings**, **Show Map Controls** - Specify whether map controls should be displayed. This adds zoom and rotation controls to the map.
- **Settings**, **Interactive Map** - Specify whether the map should be interactive. If disabled, it creates a static map with no interactivity.
- **Settings**, **Height** - Specify the height (in pixels) of the displayed map. Map's default height is calculated based on it's width with an 3:2 aspect ratio.

A.9 Display Only

Displays a non-enterable text item.

Page Designer attributes that control behavior:

- **Settings**, **Format** - Options include **Plain Text**, **HTML**, or **Markdown**.
- **Settings**, **Based On** - Specifies what gets displayed in the field. Options include:
  - **Page Item Value** - The current value of the page item displays.
  - **Display Value of List of Values** - The current value of the page item maps to a different display value using a list of values.
- **Settings**, **Show Line Breaks** - Select **On** to convert line breaks in the display value to an HTML line break and display them in HTML output.
- **Settings**, **Send On Page Submit** - Specify whether the current item value should be sent when the page is submitted.
- **Session State**, **Data Type** - Select the data type of the page item (`VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

A.10 File Upload

Displays a text field with a Browse button. The Browse button enables the user to locate and upload a file from a local file system.

> **Note:**
>
> Not supported for interactive grid columns.

Page Designer attributes that control behavior:

- **Display**, **Display As** - Select a display style for how the file browse item should display. Options include:

  - **Inline File Browse** - Display the file browse input field similar to other text fields and items on the page.
  - **Inline Dropzone** - Display the file browse input field as an inline dropzone can features a title and description.
  - **Block Dropzone** - Display the file browse input field as a wizard-style dropzone can features a title and description, and a button to select files.
  - **Native File Browse** - Use the browser's native formatting to display the file browse item.

  The **Display As** style you select determines what other attributes appear.

- **Display**, **Dropzone Title** - Only appears if **Display As** is set to **Inline File Browse**, **Inline Dropzone**, or **Block Dropzone**.

  Specify the title for the Dropzone. If this field is left blank, the Text Messages `APEX.ITEM.FILE.DROP_FILE` or `APEX.ITEM.FILE.DROP_FILES` will be used by default.

- **Display**, **Dropzone Description** - Only appears if **Display As** is set to **Block Dropzone** or **Inline Dropzone**.

  Enter a description for the Dropzone. If this field is left blank, the following Text Messages will be used by default, depending on the selected **Display As** attribute.

  - If **Display As** is set to **Block Dropzone**:
    * `APEX.ITEM.FILE.DROP_OR_CHOOSE_FILE`
    * `APEX.ITEM.FILE.DROP_OR_CHOOSE_FILES`
  - If **Display As** is set to **Inline Dropzone**:
    * `APEX.ITEM.FILE.DROP_OR_SELECT_FILE`
    * `APEX.ITEM.FILE.DROP_OR_SELECT_FILES`

- **Display**, **Capture Using** - Sets the camera default. Use this attribute to enable the item to open the device's camera when the app is used from a mobile device.

  Note: If you configure the **Capture Using** attribute, users on mobile devices will not be able to select files. However, they will be able to take a photo, switch cameras, or use photos already existing on their device.

  On a desktop, the item will always function the same regardless of the value chosen.

- **Storage**, **Type** - Select the database table where the uploaded file is stored. Options include:

  - **BLOB column specified in Item Source attribute** - Stores the uploaded file in the table used by the Automatic Row Processing (DML) process and the column specified in the item source attribute. The column has to be of data type `BLOB`. If the file gets downloaded, the table name of the Automatic Row Fetch process is used.
  - **Table APEX_APPLICATION_TEMP_FILES** - Stores the uploaded file(s) in a temporary location that you can access with the view `APEX_APPLICATION_TEMP_FILES`. Oracle APEX will automatically delete the file(s) at the end of the session or at the end of the upload request, depending on what you choose for **Purge File At**.

  If **Storage Type** is **Table APEX_APPLICATION_TEMP_FILES**:

  - **Purge File at** - Define when Oracle APEX should remove the temporary file.
  - **Allow Multiple Files** - Enable or disable this attribute to specify whether multiple files can be uploaded at once. Multiple file names will be stored as a colon delimited list in the page item.

To access a file if **Allow Multiple Files** is disabled:

```
select *
  from apex_application_temp_files
 where name = :page item name;
```

To access a file if **Allow Multiple Files** is enabled:

```
declare
    l_file_names apex_t_varchar2;
    l_file       apex_application_temp_files%rowtype;
begin
    l_file_names := apex_string.split (
                        p_str => :page item name,
                        p_sep => ':' );
    for i in 1 .. l_file_names.count loop
        select *
          into l_file
          from apex_application_temp_files
         where name = l_file_names(i);
        -- add code here to store the uploaded file in your own table
    end loop;
end;
```

If **Storage Type** is **BLOB column specified in Item Source**:

- **MIME Type Column** - Enter the uppercase page or application item name used to store the mime type of the uploaded file. The mime type is used by the browser to identify which application is used to display the content of the BLOB when the file is downloaded.
- **Filename Column** - Enter the page or application item used to store the filename of the uploaded file. .
- **Character Set Column** - Enter the uppercase page or application item name used to store the character set of the uploaded file. The character set of the `BLOB` is not automatically set on upload. End users should set the character set for the document they are uploading.
- **BLOB Last Updated Column** - Enter the uppercase page or application item name used to store the current timestamp when a file is uploaded. This date value is used for browser caching.

- **Storage**, **File Types** - Specify a comma delimited list of file types which might be used by the browser to restrict which files can be uploaded. See Help for examples.
- **Storage**, **Maximum File Size** - Enter the maximum file size (in KB) that should be accepted for files uploaded via this item.

A.11 Geocoded Address

Provides geocoding functionality that turns a postal address to a coordinate.

Geocoding is performed by the browser using a REST request to the Oracle Elocation Geocoding Service (https://elocation.oracle.com/elocation/home.html). In order for geocoding to work, the user's browser must be connected to the internet.

Geocoding input data comes from other page items which are mapped to parts of an address such as Street, House Number, Postal Code, or City. The Geocoder displays a popup window with a list of possible matches. After choosing a match, the item's session state value will be the address coordinate in GeoJSON format.

The **Structured Address** attribute determines whether the address parts are provided to the geocoding service in a *structured* or *unstructured* manner. *Structured* means that explicit page items are mapped to address parts such as Street, Postal Code, City or Region. A structured address has more accurate geocoding results, and also allows for address correction to be sent back to the page item. *Unstructured* means that only one item is used for the whole address and the address parts are separated by comma. *Unstructured* mode does not support address corrections.

A Geocoded Address item needs at least one other input item to work and for an exact address requires a maximum of eight input items.

If **Structured Address** is **Off**, the item maps to only one text field or text area where end users enter address parts, separated by comma. The following item attributes display under **Settings** in the Property Editor:

- **Settings**, **Country Type** - Select whether to use a static country for geocoding, or whether to derive the country from an item. Options include: **Static** or **Item**.
- **Settings**, **Country** - Select the country to use for geocoding.
- **Settings**, **Structured Address** - Select **Off**. Determines whether address parts are provided to the geocoding service in a structured or unstructured manner.
- **Settings**, **Address Item** - Select the item containing unstructured address parts, separated by a comma.
- **Settings**, **Match Vector Item** - Select a page item for the Geocoder to return the Match Vector. The match vector is a 17-digit string with detailed information about how each address attribute has been matched against the data used for geocoding. See item Help for more details.
- **Settings**, **Trigger Geocoding** - Determines when to trigger geocoding. This includes the browser performing the REST request to the geocoding service and showing the popup window with possible matches. Options include: **Automatic**, **Dynamic Action**, and **jQuery Selector**. See item Help for more details.

If **Structured Address** is **On**, the following item attributes display under **Settings** in the Property Editor:

- **Settings**, **Country Type** - Select whether to use a static country for geocoding, or whether to derive the country from an item. Options include: **Static** or **Item**.
- **Settings**, **Country** - Select the country to use for geocoding.
- **Settings**, **Structured Address** - Select **On**. Determines whether address parts are provided to the geocoding service in a *structured* or *unstructured* manner.
- **Settings**, **Sanitize Address** - Enable this switch to feed address data from the geocoding service back to the mapped items.
- **Settings**, **Street Item** - Select the item containing the Street part of the address.
- **Settings**, **House Number Item** - Select the item containing the *House Number* part of the address.
- **Settings**, **Postal Code Item** - Select the item containing the *Postal Code* part of the address.
- **Settings**, **City Item** - Select the item containing the *City* part of the address.

- **Settings**, **City Sub Area Item** - Select the item containing the City Sub Area part of the address.
- **Match Vector Item** - Select a page item for the Geocoder to return the Match Vector into. The match vector is a 17-digit string with detailed information about how each address attribute has been matched against the data used for geocoding. See item Help for more information.
- **Trigger Geocoding** - Determines when to trigger geocoding. This includes the browser performing the REST request to the geocoding service and showing the popup window with possible matches. Options include: **Automatic**, **Dynamic Action**, and **jQuery Selector**. See item Help for more information.

> **Note:**
>
> The Geocoded Address and Display Map items both work with the Oracle Elocation Map Cloud service (https://maps.oracle.com/elocation/ajax/index.jsp). This service is accessed by the browser, not by the APEX database on the server. Be aware this may result in data protection consequences for users. The geocoding items as and display map items store a coordinate in GeoJSON format (https://de.wikipedia.org/wiki/GeoJSON) Note that only the "geometry" portion is used. Currently only "Point" geometries are supported, for example:
>
> ```
> {
>  "type": "Point",
>  "coordinates": [
>  13.74081,
>  51.05363
>  ]
> }
> ```

A.12 Image Upload

Displays an Image Upload item which enables users to upload one or multiple images from a local file system or device. Images can be optionally cropped or resized.

The Image Upload item type includes attributes specific to previewing, uploading, resizing, and cropping an image.

From the user's perspective, this item type displays an upload input or dropzone which enables users to select the image (or images). If cropping is enabled, a dialog appears that enables the user to zoom in or out, or move the image around before cropping. Once cropping is complete or immediately after uploading, if cropping is not enabled, an image preview appears. Image resizing is handled on the client side but in the background.

Page Designer attributes that control behavior:

- **Display**, **Display As** - Select a display style for the Image Upload item. Options include:
  - **Inline File Browse** - Displays an image upload input field similar to other text fields and items on the page.
  - **Icon Dropzone** - Displays an icon-only dropzone which is replaced with a preview of the image or a multi-image icon when there is a value.

  - **Inline Dropzone** - Displays an image upload input field as an inline dropzone and features a title and description. When the inline dropzone has a value, the icon is replaced with a preview of the selected image or an icon for multiple images.
  - **Block Dropzone** - Displays an image upload input field as a wizard-style dropzone that features a title and description, and a button to select files. When the dropzone has a value, the icon is replaced with a preview of the selected image or an icon for multiple images.
  - **Native File Browse** - Use the browser's native formatting to display the file browse item.

  The **Display As** style you select determines what other attributes appear.

- **Display**, **Dropzone Title** - Only appears if **Display As** is set to **Inline File Browse**, **Inline Dropzone**, or **Block Dropzone**.

  Specify the title for the Dropzone. If this field is left blank, the Text Messages `APEX.ITEM.FILE.DROP_FILE` or `APEX.ITEM.FILE.DROP_FILES` will be used by default.

- **Display**, **Dropzone Description** - Only appears if **Display As** is set to **Inline Dropzone** or **Block Dropzone**.

  Enter a description for the dropzone. If this field is left blank, the following Text Messages will be used by default, depending on the selected **Display As** attribute.

  - If **Display As** is set to **Inline Dropzone**:
    * `APEX.ITEM.FILE.DROP_OR_SELECT_FILE`
    * `APEX.ITEM.FILE.DROP_OR_SELECT_FILES`
  - If **Display As** is set to **Block Dropzone**:
    * `APEX.ITEM.FILE.DROP_OR_CHOOSE_FILE`
    * `APEX.ITEM.FILE.DROP_OR_CHOOSE_FILES`

- **Display**, **Preview Size** - Set the size of the displayed preview. Options include:
  - **Auto** - Default selection. Uses the image's size.
  - **Extra Small**
  - **Small**
  - **Medium**
  - **Large**
  - **Extra Large**

  Define additional sizes in the theme using the class `apex-item-image--{size}`.

- **Display**, **Capture Using** - Sets the camera default. Use this attribute to enable the item to open the device's camera when the app is used from a mobile device.

  Note: If you configure the **Capture Using** attribute, users on mobile devices will not be able to select files. However, they will be able to take a photo, switch cameras, or use photos already existing on their device.

  On a desktop, the item will always function the same regardless of the value chosen.

- **Storage**, **Type** - Select the database table where the uploaded file is stored. Options include:
  - **BLOB column specified in Item Source attribute** - Stores the uploaded file in the table used by the Automatic Row Processing (DML) process and the column specified

in the item source attribute. The column has to be of data type `BLOB`. If the file gets downloaded, the table name of the Automatic Row Fetch process is used.

– **Table APEX_APPLICATION_TEMP_FILES** - Stores the uploaded file(s) in a temporary location that you can access with the view `APEX_APPLICATION_TEMP_FILES`. Oracle APEX will automatically delete the file(s) at the end of the session or at the end of the upload request, depending on what you choose for **Purge File At**.

If **Storage Type** is **Table APEX_APPLICATION_TEMP_FILES**:

– **Purge File at** - Define when Oracle APEX should remove the temporary file. Options include:

  * **End of Session** - Delete the file at the end of the current Oracle APEX session.
  * **End of Request** - Delete the file at the end of the current HTTP request.

– **Allow Multiple Files** - Enable or disable this attribute to specify whether multiple files can be uploaded at once. Multiple file names are stored as a colon delimited list in the page item.

  To access a file if **Allow Multiple Files** is disabled:

```
select *
  from apex_application_temp_files
 where name = :page item name;
```

  To access a file if **Allow Multiple Files** is enabled:

```
declare
    l_file_names apex_t_varchar2;
    l_file       apex_application_temp_files%rowtype;
begin
    l_file_names := apex_string.split (
                        p_str => :page item name,
                        p_sep => ':' );
    for i in 1 .. l_file_names.count loop
        select *
          into l_file
          from apex_application_temp_files
         where name = l_file_names(i);
        -- add code here to store the uploaded file in your own table
    end loop;
end;
```

If **Storage Type** is **BLOB column specified in Item Source**:

– **MIME Type Column** - Enter the uppercase page or application item name used to store the mime type of the uploaded file. The mime type is used by the browser to identify which application is used to display the content of the BLOB when the file is downloaded.

– **Filename Column** - Enter the page or application item used to store the filename of the uploaded file. .

– **BLOB Last Updated Column** - Enter the uppercase page or application item name used to store the current timestamp when a file is uploaded. This date value is used for browser caching.

- **Cropping**, **Allow Cropping** - Enable or disable to determine if users can crop uploaded images. By default, the aspect ratio will match the file being uploaded unless another option is defined.

  Note: Cropping is not currently supported when **Allow Multiple Files** is enabled.
- **Cropping**, **Aspect Ratio** - Only displays in **Allow Cropping** is enabled. Specify the aspect ratio for the cropper to use.

  Options include:

  – **Original** - Uses the aspect ration of the uploaded image.

  – **1:1 (square)**

  – **16:9 (widescreen)**

  – **4:3 (standard)**

  – **2:1 (univisium)**

  – **Custom** - Displays the Custom Aspect Ratio attribute.
- **Cropping**, **Custom Aspect Ratio** - Only displays if **Cropping**, **Aspect Ratio** is set to **Custom**.

  Set a custom aspect ratio for the crop window to use. An aspect ratio is a proportional relationship between an image's width and height. Enter a ratio either as a fraction, decimal, or using a `:` representation (for example, `3:2`).
- **Resize To**, **Maximum File Size** - Maximum file size for images set in KB. If a value is entered, the image will be scaled down to fit.
- **Resize To**, **Maximum File Width** - Maximum width for images set in pixels. If a value is entered the image will be scaled down to fit.
- **Resize To**, **Maximum Height** - Maximum height for images set in pixels. If a value is entered the image will be scaled down to fit..

A.13 Hidden

Hidden items are included within the page source but are not rendered.

Hidden item values are saved in session state. They are generally used to store values required by page processing or other page items, but should not be displayed to the end user.

Page Designer attributes that control behavior:

- **Settings**, **Value Protected** - Specify whether the item is protected, preventing the value from being manipulated when the page is posted. Enabling this option prevents hidden values from being manipulated when a page is posted.
- **Session State**, **Data Type** - Select the data type of the page item. Options include: `VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:

  – **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.

  – **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

A.14 Markdown Editor

Displays a simple text editor that enables you to use Markdown shorthand syntax to add formatting to text.

> **Note:**
>
> Markdown Editor is not supported for interactive grid columns.

Page Designer attributes that control behavior:

- **Settings**, **Toolbar** - Provides several options to configure the toolbar for the Markdown Editor. Options include:
  - **Simple** - Display basic formatting controls in the toolbar.
  - **Full** - Display full set of formatting controls in the toolbar.
  - **None** - Hide the toolbar.
- **Settings**, **Syntax Highlighting**- Determines whether to apply syntax highlighting for specific Markdown elements such as bold, italics, code blocks, and so on. Highlighting will only take place for content under 4000 characters in length.
- **Validation**, **Value Required**- If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Session State**, **Data Type** - Select the data type of the page item (`VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

> **See Also:**
>
> Markdown Guide

A.15 List Manager

Displays a text item with a popup list of values icon, Add and Remove buttons, and a list of selected values.

> **Note:**
>
> Not supported for interactive grid columns.

You can type in the value or pick from the list of available items. You can then utilize the buttons to manage the values selected. The selected values are stored in a single colon-delimited string.

Page Designer attributes that control behavior:

- **Settings**, **Fetch** - Select how data is displayed when the Popup LOV is opened.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help or Configuring Multiple Value Attributes.

> **See Also:**
>
> Creating a Cascading List of Values

A.16 Number Field

Displays as a number field and supports automatic formatting.

Numbers are formatted on the client after data entry according to the format mask. This item type automatically checks if the entered value is a number. No extra validations are required. Validation is done on both client and server.

> **Note:**
>
> Number Fields are limited to a precision of about 15 to 17 decimal digits due to JavaScript representing numbers in IEEE 754 format. This limit affects applying a format mask and validating that the input is a valid number. This is typically enough precision for most use cases. The Oracle Database can handle numbers with larger precision. It is often primary key columns that use very large numbers and for these, or other large numbers Oracle recommends using a Text Field.

Page Designer attributes that control behavior:

- **Appearance**, **Format Mask** - Specifies how the number is to be formatted for display. The locale determines the decimal separator used for data entry.
- **Settings**, **Minimum Value** - Enter the minimum value permitted. The value can be a static value or a dynamic value using substitution syntax.
- **Settings**, **Maximum Value** - Enter the maximum value permitted. The value can be a static value or a dynamic value using substitution syntax.
- **Settings**, **Number Alignment** - Select how to align the number that displays. This setting does not apply when the item is rendered as Read Only or where HTML Form Element Attributes are specified.

- **Settings**, **Virtual Keyboard** - Select which virtual keyboard to display when using the number field on a touchscreen. Options include:
  - **Decimal** - Numeric only keyboard, with an added decimal key.
  - **Numeric** - Numeric only keyboard. This option is useful for monetary values, PIN, ZIP Codes, CC numbers, and so on.
  - **Text** - Regular alphanumeric keyboard.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted.

  You can replace the predefined error text with an application specific error message, by defining a message called `APEX.PAGE_ITEM_IS_REQUIRED` in Shared Components, Text Messages. Specify the message display location by updating the application-level setting, **Default Error Display Location**.

A.17 Password

Displays an HTML password form element. As the end user enters text a black dot is displayed for that character, instead of the actual character entered.

Page Designer attributes that control behavior:

- **Settings**, **Submit when Enter pressed** - If set to **On**, the page is submitted when the user presses `ENTER`.
- **Appearance**, **Template Options**, **Hide Password Visibility** - By default, the password field has a **Show Password** button that users can click to show their password. If you select **Hide Password Visibility**, the **Show Password** button is disabled.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. If set to **Off**, a NULL value is accepted.
- **Session State**, **Storage** - If set to **Per Request (Memory Only)**, the value is not written to the database and is only available for the current request. For highly sensitive values, this is the preferred setting.

  > **Caution:**
  >
  > If you set **Storage** to **Per Session (Persistent)**, be sure to set **Store value encrypted in session state** to **Yes**.

- **Security**, **Authorization Scheme** - Optionally select an authorization scheme which must evaluate to TRUE in order for this component to be rendered or otherwise processed.
- **Security**, **Session State Protection** - Select the level of session state protection:
  - **Unrestricted** - The item can be set by passing the item in a URL or in a form. No checksum is required in the URL.
  - **Checksum Required - Application Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace and application.
  - **Checksum Required - User Level** - The item can be set by passing the item in a URL that includes a checksum specific to the workspace, application, and user.
  - **Checksum Required - Session Level** - The item can be set by passing the item in a URL that includes a checksum specific to the session.

  - **Restricted - May not be set from browser** - The item can not be altered using the URL or POSTDATA. Select this option to restrict what can set the item value to internal processes, computations, and so on. This attribute only applies to items that are not used as data entry items and is always observed, even if Session State Protection is disabled.
- **Security**, **Store value encrypted in session state** - To encrypt an item when stored in session state, set this attribute to **On**. To learn more, see About Session State and Security.

> **See Also:**
>
> About Session State and Security

A.18 Percent Graph

Displays the value as a percentage graph. The value retrieved must be between 0 and 100.

A.19 Popup LOV

Displays a text item with a popup list of values icon. When the end user clicks the icon, a popup window appears with a search field, and a list of supported values.

Popup LOV is best suited for large lists since end users can enter search criteria to reduce the available values displayed. For relatively small lists Select List is often better suited.

```
SELECT ename name, empno id FROM emp
```

If one of the columns is an expression, remember to use an alias. For example:

```
SELECT ename||' '||job display_value, empno FROM emp
```

Page Designer attributes that control behavior:

- **Settings**, **Display As** - Select how the popup list of values is displayed.
- **Settings**, **Initial Fetch** - Select how data is initially fetched for the list of values. This applies only the first time the popup is opened after the page loads or the item is refreshed such as due to a cascade. See also attribute Minimum Characters.
- **Settings**, **Manual Entry** - Select if manual entry is allowed. If **Yes**, the user can select existing values from the popup list of values or enter a value manually (similar to a Combobox). If **No**, the user can only select values from the popup list of values. (similar to a Select List). When manual entry is allowed, it is the return value of the corresponding List of Values that must be entered.
- **Settings**, **Fetch on Search** - Specify whether matching values are retrieved from the database as the user types on the item. **Fetch on Search** retrieves matching records from the database each time the user types a character.

  When **Yes**, the list of values is filtered as you type in the search field. When **No**, the list of values is not filtered until the user presses **Enter** or the **Search icon** button.
- **Settings**, **Title** - The popup dialog title. If blank, the default title will be used.
- **Search**, **Match Type** - Select how the search against the entered value is performed. Options include: **Contains** or **Starts With**.

> **Note:**
>
> This setting can have an impact on query performance. Additionally, selecting **Contains** prevents the use of column indexes.

- **Search**, **Case-Sensitive** - Controls if the search is case-sensitive.
- **Search**, **Minimum Characters** - Specify the minimum number of characters required in the search term before the list of values will be fetched. This can be useful to reduce the amount of searching done when the LOV query is expensive. All of the display values in the list should be longer than this minimum otherwise they cannot be found.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded. Options include: No, Delimited List, JSON Array.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other. Note that the `"` (double-quote) and `\` (backslash) characters are not supported as a Value Separator, also not in the `\xxxx` format.

  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help orConfiguring Multiple Value Attributes.
- **Validation**, **Value Required**- If set to **On** and the page item is visible, Oracle APEX automatically performs a `NOT NULL` validation when the page is submitted. See Help for more details.

> **See Also:**
>
> Creating a Cascading List of Values

A.20 QR Code

Encodes the selected source value into a QR code and displays it.

A QR code is a barcode that contains a machine-readable optical image. A QR code can be scanned using a smartphone camera and a QR code reader app and can be used to store or share information.

Page Designer attributes that control behavior:

- **Settings**, **Value Protected** - Specifies if the encoded value of the QR code item is protected. Enabling **Value Protected** switch prevents the value from being manipulated when the page is posted.
- **QR Code**, **Data Type** - Select the type of the content that the QR code contains and a corresponding prefix value is inserted in front of the selected source value to affect how the user QR Code scanner will interact with the content. Options include:
  - **Plain Text**
  - **Phone** - Open the phone app with the input number. The phone number can contain special characters `'('` , `')'`, `'+'`, `'-'`, `'.'`, and spaces. The exact format of the phone number delegates to the phone's locale settings.
  - **Email** - Compose a new email to an email address. The expected format of the email address is a valid email address without leading or trailing spaces.

  - **SMS** - Compose a text message to a number. The phone number can contain special characters '(' , ')', '+', '-', '.', and spaces. The exact format of the phone number delegates to the phone's locale settings.
  - **Geo location** - Open a map application to specific coordinates. The expected format of the input coordinates is: `lattitude,longitude` with no leading or trailing spaces.
- **QR Code**, **Size** - Sets the size of the QR Code in pixels. The actual pixel size of the QR Code item differs in different themes. Options include:
  - **Default**
  - **Small**
  - **Medium**
  - **Large**

A.21 Radio Group

Displays multiple values as radio group options, enabling the end user to select a single value.

The following example displays employee names (`ename`), but returns employee numbers (`empno`):

```
SELECT ename, empno FROM emp;
```

Page Designer attributes that control behavior:

- **Settings**, **Number of Columns** - Enter the number of radio group columns to display. For example, a value of `2` would display two columns. If there were five values then it would display over three rows.
- **Settings**, **Page Action on Selection** - Select what action is taken when a radio group value is selected. See Help for more details.
- **Settings**, **Execute Validations** - This attribute only appears if **Page Action when Value Changed** is set to **Submit Page**. Controls the execution of validations when a page is submitted. Set this attribute to **Off** to prevent the execution of validations and associated error messages. Set this attribute to **On** to execute all page and built-in validations and associated error messages when the page is submitted.

> **See Also:**
>
> About Determining When Validations Execute and Creating a Cascading List of Values

- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.22 Rich Text Editor

Displays a text area with comprehensive text formatting options.

End users can enhance the content displayed in a similar fashion to using a word processor, such as Microsoft Word. End users can also cut and paste documents that are in rich text format (RTF) file format into this item.

Page Designer attributes that control behavior:

- **Settings**, **Library** - Specify the JavaScript library on which the Rich Text Editor should be based. Select **ORTL**.
- **Settings**, **Format** - Specify the output format of the content entered in the rich text editor. Options include:
  - **HTML** - Use HTML for the source and output of this rich text editor. HTML is more flexible and offers more formatting options than Markdown, but is less secure and must be trusted or sanitized when displayed.
  - **Markdown** - Use Markdown format for the source and output of this rich text editor. Markdown is more restrictive than HTML in terms of features, but is more secure as it must first pass through a parser before being displayed.
- If **Settings**, **Format** is **HTML**, the following additional attributes appear:
  - **Settings**, **Toolbar** - Select the number of text formatting options available within the editor. Available options include: **Basic**, **Intermediate**, or **Full**.
  - **Settings**, **Toolbar Style** - Choose whether the toolbar items should only take up one line and overflow into a popup menu if wider than the editor's width, or whether to split them into multiple lines. Options include: **Overflow Menu** or **Multiline**.
- **Settings**, **Minimum Height** - Set the minimum height in pixels of the editable area.
- **Settings**, **Maximum Height** - Set the maximum height in pixels of the editable area. Leaving this attribute empty will let the editor grow infinitely as you type.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Session State**, **Data Type** - Select the data type of the page item (`VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

A.23 Select List

Displays an item with a built-in list of values selector. When the end user clicks the item, the list of supported values displays directly inline with the current item.

Select List is best suited for relatively small, discrete lists. End users can very quickly select a value from the list without needing to change focus to a popup dialog. For large lists, consider using a Popup LOV.

```
select [displayValue],
       [returnValue]
  from ...
 where ...
 order by ...
```

Page Designer attributes that control behavior:

- **Settings**, **Page Action on Selection** - Select what action is taken when a list value is selected. See help for options.
- **Settings**, **Allow Multi Selection** - Specify whether multiple values or only a single value can be selected from the list of values. Multiple values are returned as a comma-separated list.
- **Settings**, **Execute Validations** - Displays if **Page Action on Selection** is set to **Submit Page**. Controls the execution of validations when a page is submitted. Set this attribute to **Off** to prevent the execution of validations and associated error messages. Set this attribute to **On** to execute all page and built-in validations and associated error messages when the page is submitted. Use this attribute in conjunction with the validation attribute, **Always Execute**.

> **See Also:**
>
> About Determining When Validations Execute

- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help orConfiguring Multiple Value Attributes.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

> **Note:**
>
> To create this item type, you must define a list of values. See Creating Lists of Values at the Application-Level, Working with Multiple Select List Item, and Creating a Cascading List of Values.

A.24 Select Many

Displays a drop-down list that supports multi-select and search filtering.

A Select Many item displays a LOV-based item that shows suggestions that allow multi-values. When the end user clicks the field, a popup window appears with a list of suggested values. A Select Many item provides a more user-friendly select item compared to a Popup LOV and supports filtering, groups, and template directives.

Page Designer attributes that control behavior:

- **Settings**, **Maximum Values in List** - Enter the maximum number of options to display in the select box. If not specified, up to 250 options will display. This value can help improve performance, by avoiding returning too many rows, and also prevent a very large return set when only a few letters have been entered.
- **Settings**, **Fetch on Search** - Specify whether matching values are retrieved from the database as the user types on the item. **Fetch on Search** retrieves matching records from the database each time the user types a character. Otherwise, all of the list values are loaded on page load.

Oracle recommends using **Fetch on Search** when you have a large select list, as loading a large number of records may impact the time to initially load the page.

- **Settings**, **Use Defaults** - Specify whether to use the defaults defined for this component in Shared Components, Component Settings or custom settings.
- **Settings**, **Display Values As** - Only displays if **Use Defaults** is disabled. Specify how to display values when the item is not focused. Options include:
  - **Chips**
  - **Comma-separated list**
- **Settings**, **Value HTML Expression** - Define how the choices are rendered. This attribute supports HTML and template directives. See Using Template Directives.

  For example:

```
<div class="your-container">
    <span class="your-item">Name: &NAME.</span>
    {if ?CAPITAL/}<span class="your-item">Capital: &CAPITAL.</span>{endif/}
    {if ?NATIONALITY/}<span class="your-item">Nationality: &NATIONALITY.</
span>{endif/}
    {if ?POPULATION/}<span class="your-item">Population: &POPULATION.</
span>{endif/}
</div>
```

- **Search** , **Match Type** - Select how the search against the entered value is performed. When using **Fetch On Search**, this setting can have an impact on query performance. Options include:
  - **Contains** (prevents the use of column indexes)
  - **Starts With**
- **Search** , **Case-Sensitive** - Controls if the search is case-sensitive.
- **Search** , **Minimum Characters** - Specify the minimum number of characters that must be entered before a search is executed.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help orConfiguring Multiple Value Attributes.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

  By defining a message called `APEX.PAGE_ITEM_IS_REQUIRED` in Shared Components, Text Messages, the predefined error text can be replaced by an application specific error message. The display location of the message is defined by the application level setting Default error display location.
- **Validation**, **Maximum Length** - Enter the maximum number of characters users can type into this item.
- **Session State**, **Storage** - Select how session state is maintained. Options include:

- **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.

  When creating a manual entries item, Oracle recommends selecting **Per Request (Memory Only)**.
- **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.
- **Per User (Persistent)** - Maintain for a user by storing the value in the database, to access it across sessions.

> **Note:**
>
> To create this item type, you must define a list of values. See Creating Lists of Values at the Application-Level, Working with Multiple Select List Item, and Creating a Cascading List of Values.

A.25 Select One

Displays as an item with a list of values icon which supports the selection of one value and search filtering.

When the end user clicks the field, a popup window appears with a list of suggested values. A Select One item provides a more user-friendly option compared to a Select List item and supports filtering, groups, and template directives.

Page Designer attributes that control behavior:

- **Settings**, **Maximum Values in List** - Enter the maximum number of options to display in the select box. If not specified, up to 250 options will display. This value can help improve performance, by avoiding returning too many rows, and also prevent a very large return set when only a few letters have been entered.
- **Settings**, **Fetch on Search** - Specify whether matching values are retrieved from the database as the user types on the item. **Fetch on Search** retrieves matching records from the database each time the user types a character. Otherwise, all of the list values are loaded on page load.

  Oracle recommends using **Fetch on Search** when you have a large select list, as loading a large number of records may impact the time to initially load the page.
- **Settings**, **Value HTML Expression** - Define how the choices are rendered. This attribute supports HTML and template directives. See Using Template Directives.

  For example:

```
<div class="your-container">
    <span class="your-item">Name: &NAME.</span>
    {if ?CAPITAL/}<span class="your-item">Capital: &CAPITAL.</span>{endif/}
    {if ?NATIONALITY/}<span class="your-item">Nationality: &NATIONALITY.</
span>{endif/}
    {if ?POPULATION/}<span class="your-item">Population: &POPULATION.</
span>{endif/}
</div>
```

- **Search** , **Match Type** - Select how the search against the entered value is performed. When using **Fetch On Search**, this setting can have an impact on query performance. Options include:
  - **Contains** (prevents the use of column indexes)
  - **Starts With**
- **Search** , **Case-Sensitive** - Controls if the search is case-sensitive.
- **Search** , **Minimum Characters** - Specify the minimum number of characters that must be entered before a search is executed.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

  By defining a message called `APEX.PAGE_ITEM_IS_REQUIRED` in Shared Components, Text Messages, the predefined error text can be replaced by an application specific error message. The display location of the message is defined by the application level setting Default error display location.
- **Validation**, **Maximum Length** - Enter the maximum number of characters users can type into this item.
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.

    When creating a manual entries item, Oracle recommends selecting **Per Request (Memory Only)**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.
  - **Per User (Persistent)** - Maintain for a user by storing the value in the database, to access it across sessions.

> **Note:**
>
> To create this item type, you must define a list of values. See Creating Lists of Values at the Application-Level, Working with Multiple Select List Item, and Creating a Cascading List of Values.

A.26 Shuttle

Displays as a multiple select list that includes two boxes containing lists. The left list displays the source list of values that have not been selected and the right list shows the currently selected values.

End users can select one or more values in a list then use the shuttle controls to move the selected values or all values. The current values are stored in a single colon-delimited string.

Page Designer attributes that control behavior:

- **Settings**, **Show Controls** - Select what data movement controls displays in the middle of the shuttle. Options include: None, Moving Only, Ordering Only, or All.

- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Multiple Values**, **Type** - Determines whether the Source Column contains multiple values and how these are encoded.
- **Multiple Values**, **Separator** - Identifies the character which separates values from each other.
  To learn more about **Multiple Values** attributes and view examples, see Page Designer Help orConfiguring Multiple Value Attributes.

> **Note:**
>
> To create this item type, you must define a list of values. See Creating Lists of Values at the Application-Level, Working with Multiple Select List Item, and Creating a Cascading List of Values.

A.27 Textarea

Displays a multiple-row text area.

Page Designer attributes that control behavior:

- **Settings**, **Resizable** - Specify whether the text area resizing using the mouse is enabled.
- **Settings**, **Auto-Height** - Specify whether the height of the text area varies based on the amount of text displayed. Use this option to scale the text area to the amount of data.
- **Settings**, **Character Counter** - If set to **On** include a counter that displays the number of characters entered in the field.
- **Settings**, **Trim Spaces** - Select how the item value is trimmed. This setting trims spaces, tabs, and new lines from the text entered. Options include: **Leading**, **Trailing**, **Leading and Trailing**, or **None**.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.
- **Session State**, **Data Type** - Select the data type of the page item (`VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

A.28 Star Rating

Displays a star rating (maximum of 10) that end users can click to set a numeric value.

Page Designer attributes that control behavior:

- **Settings**, **Number of Stars** - Set the total number of stars (one to ten) for end users to choose (default five).
- **Settings**, **Use Defaults** - If set to **On**, the Star Rating uses the application-level component settings defined in Shared Components, Component Settings, Star Rating.

  See Also: Configuring Star Rating

  If **Use Defaults** is set to **Off**, the following custom settings appear:

  – **Icon** - The icon class to use (such as `fa-thumb-up`). Click the drop-down list to pick an icon.

  – **Active Color** - Default color for active stars.

  – **Inactive Color** - Default color for inactive stars.

  – **Tooltip** - The template for the tooltip when end users hover the mouse over the item. This rating value replaces the `#VALUE#` substitution (for example, the movie rating is `#VALUE#`).

  – **Show Clear Button** - Choose whether to display the Clear button (`x`) next to the stars.

  – **Show Value** - Choose whether to display a value next to the stars.

  – **Read Only Display** - Choose what to display if the item is read only. Options include **Stars**, **Value**, **Stars and Value**, and **Single Star and Value**.

A.29 Switch

Displays as a flip toggle switch.

Page Designer attributes that control behavior:

- **Settings**, **Use Defaults** - If set to **On**, the Color Picker uses the application-level component settings defined in Shared Components, Component Settings, Color Picker.

  > **Note:**
  >
  > Switch items can display as a flip toggle switch (the default), Pill Button, or Select List. See Configuring Switch Defaults.

  If **Use Defaults** is set to **Off**, the following custom settings appear:

  – **On Value** - Enter the value stored in the page item if the user picks the On option. The value can be up to 255 characters long.

  – **On Label** - Enter the text displayed for the On option.

  – **Off Value** Enter the value stored in the page item if the user picks the Off option. This attribute can also be left blank if you want to store NULL when the Off option is selected. The value can be up to 255 characters long.

  Note: This value is used if the page item is populated with a value which does not match either the On Value or Off Value.

  – **Off Label** - Enter the text displayed for the Off option.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.30 Star Rating

Displays a star rating (maximum of 10) that end users can click to set a numeric value.

Page Designer attributes that control behavior:

- **Settings**, **Number of Stars** - Set the total number of stars (one to ten) for end users to choose (default five).
- **Settings**, **Use Defaults** - If set to **On**, the Star Rating uses the application-level component settings defined in Shared Components, Component Settings, Star Rating.

  See Also: Configuring Star Rating

  If **Use Defaults** is set to **Off**, the following custom settings appear:

  – **Icon** - The icon class to use (such as `fa-thumb-up`). Click the drop-down list to pick an icon.

  – **Active Color** - Default color for active stars.

  – **Inactive Color** - Default color for inactive stars.

  – **Tooltip** - The template for the tooltip when end users hover the mouse over the item. This rating value replaces the `#VALUE#` substitution (for example, the movie rating is `#VALUE#`).

  – **Show Clear Button** - Choose whether to display the Clear button (`x`) next to the stars.

  – **Show Value** - Choose whether to display a value next to the stars.

  – **Read Only Display** - Choose what to display if the item is read only. Options include **Stars**, **Value**, **Stars and Value**, and **Single Star and Value**.

A.31 Text Field

Displays the item as a text field.

Page Designer attributes that control behavior:

- **Settings**, **Subtype** - Select the HTML5 text subtype. This enables devices with on-screen keyboards to show an optimized keyboard layout specific to the subtype, for easier data input. The subtype selection is also used to render an appropriate link with the value of the page item, if it is rendered read only. This attribute does not change the text item, or data entry, when using a physical keyboard.

  > **Note:**
  >
  > This HTML5 feature only works in modern browsers. Older, non HTML5-compliant, browsers ignore this attribute and render the page item as a normal text field.

- **Settings**, **Trim Spaces** - Select how the item value is trimmed. This setting trims spaces, tabs, and new lines from the text entered. Options include: **Leading**, **Trailing**, **Leading and Trailing**, or **None**.
- **Settings**, **Text Case** - Select how the item value text case is transformed. Options include: **No Change**, **Lower**, and **Upper**.

- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.32 Text Field with Autocomplete

Displays a text field that displays a list of possible values based on the text already entered by the end user, inline with the text item. The list is further refined as the end user types in more text.

Consider the following example:

```
select [column]
  from ...
 where ...
 order by ...
```

Page Designer attributes that control behavior:

- **Settings**, **Search** - Select how the search against the entered value is performed. When using Lazy Loading this setting can have an impact on query performance. Selecting **Contains & Case Sensitive** or **Contains & Ignore Case** prevents the use of column indexes.
- **Settings**, **Fetch On Type** - Specify whether matching values are retrieved from the database as the user types on the autocomplete item. **Fetch On Type** retrieves matching records from the database each time the user types a character. Otherwise, all of the list values are loaded on page load.

  Oracle recommends using **Fetch On Type** when you have a large select list since loading a large number of records may impact the time to initially load the page.
- **Settings**, **Maximum Values in List** - Enter the maximum number of items to display in the select box. This value can help avoid performance problems by returning too many rows, and also prevent a very large return set when only a few letters have been entered.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

A.33 Textarea

Displays a multiple-row text area.

Page Designer attributes that control behavior:

- **Settings**, **Resizable** - Specify whether the text area resizing using the mouse is enabled.
- **Settings**, **Auto-Height** - Specify whether the height of the text area varies based on the amount of text displayed. Use this option to scale the text area to the amount of data.
- **Settings**, **Character Counter** - If set to **On** include a counter that displays the number of characters entered in the field.
- **Settings**, **Trim Spaces** - Select how the item value is trimmed. This setting trims spaces, tabs, and new lines from the text entered. Options include: **Leading**, **Trailing**, **Leading and Trailing**, or **None**.
- **Validation**, **Value Required** - If set to **On** and the page item is visible, Oracle APEX automatically performs a NOT NULL validation when the page is submitted. See Help for more details.

- **Session State**, **Data Type** - Select the data type of the page item (`VARCHAR2` or `CLOB`).
- **Session State**, **Storage** - Select how session state is maintained. Options include:
  - **Per Request (Memory Only)** - Do not save state in the database. State is only available when processing the current request. When AJAX requests need to use an item, make sure to pass the item name using **Page Items To Submit**.
  - **Per Session (Persistent)** - Maintain for each session by storing the value in the database, to access it across requests.

B

Oracle APEX Limits

Learn about limits associated with Oracle APEX components.

- Component Limits
  Learn about limits associated with Oracle APEX components, such as interactive reports, classic reports, forms, and tabular forms.

Related Topics

- Splitting Export Files into Separate Scripts
  Learn about splitting application, page, and component export files into separate SQL scripts.

B.1 Component Limits

Learn about limits associated with Oracle APEX components, such as interactive reports, classic reports, forms, and tabular forms.

Component	Limits	To Learn More...
Data loading	When loading data, the maximum number of columns is 45.	Managing Applications with Legacy Data Loading Capability
Export of interactive report and classic report using external print server	If using an external print server, there is a 32k limitation on the generated report template header sub-template. Also, depending on the column alias lengths, approximately 26 columns (or less) are supported.	Exporting Workspaces, Applications, and App Components
Forms	32767 bytes for a text area or rich text editor item. Two columns for primary key (when using built in DML processes). For users with more than two primary key columns defined, you can also specify a ROWID.	Developing Forms
Interactive reports	999 rows per column heading filter (if no custom LOV is specified in the column attributes). A chart within an interactive report can render up to 10000 data points.	Developing Reports
Item names	Item names should not exceed 30 characters. Items longer than 30 characters cannot be referenced using bind variable syntax.	Referencing Session State Using Bind Variable Syntax About Item Naming Conventions

Component	Limits	To Learn More...
Native PDF and Excell support in reports	Oracle APEX includes native support for printing basic reports to PDF and Microsoft Excel. If you want full control and advanced layouts for your reports, Oracle recommends using a report server. Limitations for native PDF and Microsoft Excel: • A report row in PDF may span multiple lines and may have a single page break. The text is truncated if a row exceeds this limit. • The maximum text size of a cell in Microsoft Excel is 32k. The text is truncated If a cell exceeds this limit.	n/a
Tabular Forms	One wizard-generated tabular form per page (using built-in DML). 50 editable tabular form columns (`apex_application.g_f01-apex_application.g_f50`), generated with `apex_item` or the built-in tabular form display types.	Managing Legacy Tabular Forms
Validations	Text entered for validations may not exceed 3,950 characters.	Understanding Validations

C

Directory Structure When Splitting Export Files into Separate Scripts

Exporting an application, page, or component as separate files creates a directory structure organized by component type.

This directory structure may extend multiple levels. This directory structure is the same whether you are using the application-level flag, **Split into Multiple Files**, or the page or component-level flag, **Export as Zip**.

- Export Directory Structure
  Learn about the directory structure of applications, pages, or components exported as separate files.

Related Topics

- Splitting Export Files into Separate Scripts
  Learn about splitting application, page, and component export files into separate SQL scripts.

C.1 Export Directory Structure

Learn about the directory structure of applications, pages, or components exported as separate files.

Component	Directory Structure
Supporting Objects	`application/deployment`
Page	`application/pages`
Page Group	`application/pages/page_groups`
Plug-in Settings	`application/plugin_settings`
Data Loading	`application/shared_components/data_loading/tables`
Data Profile	`application/shared_components/data_profiles`
Email Template	`application/shared_components/email/templates`
Application Static File	`application/shared_components/files`
Text Message	`application/shared_components/globalization/messages`
Application Computation	`application/shared_components/logic/application_computations`
Application Item	`application/shared_components/logic/application_items`
Application Process	`application/shared_components/logic/application_processes`
Application Setting	`application/shared_components/logic/application_settings`
Build Option	`application/shared_components/logic/build_options`
Build Option Status	`application/shared_components/logic/build_options/status`
Web Service	`application/shared_components/logic/webservices`

Component	Directory Structure
Breadcrumb Entry	`application/shared_components/navigation/breadcrumbentry`
Breadcrumb	`application/shared_components/navigation/breadcrumbs`
List	`application/shared_components/navigation/lists`
Navigation Bar Entry	`application/shared_components/navigation/navigation_bar`
Parent Tab	`application/shared_components/navigation/tabs/parent`
Tab	`application/shared_components/navigation/tabs/standard`
Tree	`application/shared_components/navigation/trees`
Plug-in	`application/shared_components/plugins`
Progressive Web App Screenshot	`application/shared_components/pwa/screenshots`
Progressive Web App Shortcut	`application/shared_components/pwa/shortcuts`
Report Layout	`application/shared_components/reports/report_layouts`
Report Query	`application/shared_components/reports/report_queries`
Application Access Control Role	`application/shared_components/security/app_access_control`
Authentication Scheme	`application/shared_components/security/authentications`
Authorization Scheme	`application/shared_components/security/authorizations`
List of Values	`application/shared_components/user_interface/lovs`
Shortcut	`application/shared_components/user_interface/shortcuts`
Breadcrumb Template	`application/shared_components/user_interface/templates/breadcrumb`
Button Template	`application/shared_components/user_interface/templates/button`
Calendar Template	`application/shared_components/user_interface/templates/calendar`
Label Template	`application/shared_components/user_interface/templates/label`
List Template	`application/shared_components/user_interface/templates/list`
Page Template	`application/shared_components/user_interface/templates/page`
Popup List of Values Template	`application/shared_components/user_interface/templates/popuplov`
Region Template	`application/shared_components/user_interface/templates/region`
Report Template	`application/shared_components/user_interface/templates/report`
REST Data Source (Web Source)	`application/shared_components/web_sources`
Concatenated File	`application/user_interfaces/combined_files`
Credential	`workspace/credentials`
Workspace Static File	`workspace/files`
Remote Server	`workspace/remote_servers`

D

Correlating APEX Sessions to Database Sessions

Learn about best practices that enable administrators and developers to correlate the database session with a specific APEX application in order to troubleshoot application performance issues.

- About APEX Sessions and Database Sessions
  Learn about APEX sessions and database sessions.
- Session Attributes
  Learn about session attributes visible to system administrators and developers.
- APEX Processes
  Learn how APEX sets session attributes for an application or background mail session.
- SQL Workshop
  Learn how APEX sets session attributes in SQL Commands and SQL Scripts.
- Data Workshop
  Learn how APEX sets session attributes while importing data using Data Workshop.
- Automations
  Learn how APEX sets session attributes while running Automations.
- Installer
  Learn how APEX installation scripts set session attributes while APEX is being installed or upgraded.

> **See Also:**
>
> SET_IDENTIFIER Procedure and DBMS_APPLICATION_INFO in *Oracle Database PL/SQL Packages and Types Reference*

D.1 About APEX Sessions and Database Sessions

Learn about APEX sessions and database sessions.

Oracle APEX includes a database account called `APEX_PUBLIC_USER`. All database sessions in the database session pool of Oracle REST Data Services (ORDS) use a single database user such as `ORDS_PUBLIC_USER`, `APEX_PUBLIC_USER`, or some other user depending how the instance was configured. There is no permanent, fixed correlation between an APEX session and a database session. Because the assignment of a session in the session pool to service an APEX request is essentially unpredictable and not constant, this correlation can only be done for active database sessions, which are actively servicing APEX requests.

D.2 Session Attributes

Learn about session attributes visible to system administrators and developers.

While one or more APEX database processes are active, they may set the following session attributes which are visible to system administrators from `v$session`, `v$sqlarea`, `v$session_longops`, and from various audit and SQL tracing tools. Developers can set these session attributes using PL/SQL.

client_id

`client_id` is a 64 byte string that is generally used to identify the user/client session. Since it is possible for the processing for a single APEX session to execute in different database sessions, you can correlate the workload across all database sessions to each APEX session. `client_id` is also often used in conjunction with Virtual Private Database (VPD) using an application context. In PL/SQL, you can set `client_id` for a session by calling `dbms_session.set_identifier` and retrieve it with `sys_context('USERENV','CLIENT_IDENTIFIER')`. You can view `client_id` for active sessions in the `client_identifier` column in the `v$session` view.

Within an APEX session, setting this attribute is not necessary and generally should be left unchanged. If custom PL/SQL code executed within an APEX session changes `client_id`, this change will not cause problems for Oracle APEX. However, it may affect the ability for some monitoring tools to effectively identify the APEX session.

client_info

`client_info` is a 64 byte string that is usually used by APEX to identify the workspace and user. In PL/SQL, you can set `client_info` for the session by calling `dbms_application_info.set_client_info` and retrieve it using `dbms_application_info.read_client_info` or with `sys_context('USERENV','CLIENT_INFO')`. You can view `client_info` for active sessions in the `client_info` column in the `v$session` view.

Within an APEX session, setting this attribute is not necessary and generally should be left unchanged. If custom PL/SQL code executed within an APEX session changes `client_info`, this change will not cause problems for Oracle APEX. However, it may affect the ability for some monitoring tools to effectively identify the APEX session.

module

`module` is a 48 byte string that is usually used to identify the currently running module or process. In PL/SQL, you can set `module` by calling `dbms_application_info.set_module` and retrieve it using `dbms_application_info.read_module` or with `sys_context('USERENV','MODULE')`. You can view `module` for active sessions in the `module` column in the `v$session` view.

Within an APEX session, Oracle recommends setting this attribute (along with `action`) as a way to allow monitoring tools to identify what is currently being processed.

action

`action` is a 32 byte string that is usually used to identify the current step being performed within a module or process. In PL/SQL, you can set `action` for the session by calling `dbms_application_info.set_module` or `set_action` and retrieve it using

`dbms_application_info.read_module` or `sys_context('USERENV','ACTION')`. You can view `action` for active sessions in the `action` column in the `v$session` view.

Within an APEX session, Oracle recommends setting this attribute (along with `module`) as a way to allow monitoring tools to identify what is currently being processed.

Session long ops

`Session longops` (long operations) is a record structure that describes the number of units processed so far versus the total number of units expected to be processed. You can use `session longops` to estimate completion time for long-running processes. In PL/SQL, you can set `session longops` for the session by calling `dbms_application_info.set_session_longops`. You can monitor long operations by querying the `v$session_longops` view.

Within an APEX session, Oracle recommends setting this attribute for processes that are expected to take a long time, if the workload can be reasonably broken up into discrete units of work (for example, "number of records processed").

> **Note:**
>
> Oracle APEX calls the **SYS** version of `dbms_application_info`; even if a wrapper for the package exists, it will not be used.

D.3 APEX Processes

Learn how APEX sets session attributes for an application or background mail session.

> **Tip:**
>
> Most APEX processes reset the session values to NULL when they finish. They do not restore prior values.

The following table details how APEX sets session attributes for an application session.

Session Attribute	Expression	Example.	Notes More...
`client_id`	*app_user* `||` `':'` `||` *session-id*	`jane.doe@example.com:12161645673208`	If app_user is too long, it will be truncated to fit (for example, the first 50 or 49 bytes). Before authentication is completed, will be: `'nobody:' || session-id`
`client_info`	*workspace-id* `||` `':'` `||` *app_user*	`3574091691765823934:jane.doe@example.com`	n/a

Session Attribute	Expression	Example.	Notes More...
`module`	`schema || '/APEX:APP ' || app-id || ':' || page-id`	`MY_DB/APEX:APP 17251:4`	n/a

`action`	A range of different values may be seen, such as:
	
	`'start_page_request'`
	`'PAGE ' || page-id`
	`'computation and processes'`
	`'fetch meta data'`
	`'Processes - point: ' || process-point`
	`'Validations'`
	`'branching'`
	`'import app ' || app-id`
		n/a	n/a

Mail Background Session

The following table details how APEX sets session attributes for a mail background session.

Session Attribute	Expression	Notes
`client_id`	`mail-from || ':' || message-guid`	Long email addresses are truncated to fit.
`client_info`	`workspace-id || ':' || mail-from`	n/a
`module`	(unchanged)	n/a
`action`	(unchanged)	n/a

D.4 SQL Workshop

Learn how APEX sets session attributes in SQL Commands and SQL Scripts.

> **Note:**
>
> The **Autocommit** check box displays in SQL Commands if **Enable Transactional SQL Commands** is enabled in Administration Services, Feature Configuration. To learn more, see Configuring SQL Workshop in *Oracle APEX Administration Guide*

SQL Commands (Autocommit = ON)

The following table details how APEX sets session attributes in SQL Commands when **Autocommit** is enabled.

Session Attribute	Expression
client_id	*app_user* || ':' || *session-id*
client_info	*workspace-id* || ':' || *app_user*
module	'APEX:SQL Workshop'
action	*sql-statement*

SQL Commands (Autocommit = OFF)

The following table details how APEX sets session attributes in SQL Commands when **Autocommit** is disabled. When **Autocommit** is disabled, SQL Commands are executed by a separate database job. While awaiting a command from the SQL Commands window, the job is in an "idle" state.

Session Attribute	Expression
client_id	*app_user* || ':' || *session-id*
client_info	*workspace-id* || ':' || *app_user*
module	'APEX:SQL Workshop-No Autocommit'
action	Working *sql-statement* Idle, awaiting work Stopping

> **See Also:**
>
> Using SQL Commands in *Oracle APEX SQL Workshop Guide*

SQL Scripts

The following table details how APEX sets session attributes while running SQL Scripts.

Session Attribute	Expression
`client_id`	*app_user* `||` `':'` `||` *session-id*
`client_info`	*workspace-id* `||` `':'` `||` *app_user*
`module`	`'APEX:SQL Workshop:`*Script*`'`
`action`	*script-name*
`longops`	`SQL statement` *N* `of` *M*

> **See Also:**
>
> Using SQL Scripts in *Oracle APEX SQL Workshop Guide*

D.5 Data Workshop

Learn how APEX sets session attributes while importing data using Data Workshop.

The following table details how APEX sets session attributes while importing data using Data Workshop.

Session Attribute	Expression	Notes
`client_id`	(unchanged)	n/a
`client_info`	*workspace-id* `||` `':'` `||` *app-user*	n/a

Session Attribute	Expression	Notes
module	schema || '/APEX:APP' || app-id || ':DATA_LOADING' or schema || '/APEX:APP' || app-id || ':DATA_LOADING_SODA'	If the schema name is long, it may be truncated so that the module string fits within 48 bytes.
action	Data Loading Job	n/a

D.6 Automations

Learn how APEX sets session attributes while running Automations.

The following table details how APEX sets session attributes while running Automations.

Session Attribute	Expression	Notes
client_id	(unchanged)	n/a
client_info	*workspace-id* || ':'	n/a
module	*automation-module*	n/a
action	*automation-static-id* || ':I'	Executing initialisation procedure. In all cases, automation-static-ID is truncated to 16 characters.
action	*automation-static-id*	Starting / fetching data.
action	*automation-static-id* || ':R' || *row-num* || ':B'	Before action.
action	*automation-static-id* || ':R' || *row-num* || ':A' || action-sequence	Executing action.
action	*automation-static-id* || ':R' || *row-num*	Finished action.

Session Attribute	Expression	Notes
`action`	*`automation-static-id`* `|| ':T'`	Cleanup.

> **See Also:**
>
> Managing Automations

D.7 Installer

Learn how APEX installation scripts set session attributes while APEX is being installed or upgraded.

The following table details how APEX installation scripts set session attributes while installing or upgrading APEX.

Session Attribute	Expression	Notes
`client_id`	(unchanged)	n/a
`client_info`	(unchanged)	n/a
`module`	`'APEX '` `||` *`install-type`* `|| ' PHASE '` `||` `phase`	*`install-type`* is something like `'FULLINS'`, `'RUNTIME'`, `'PATCH'`, and so on.

`action`	A range of different values may be seen, such as:
	
	`'upgrade'`
	`'recompiling'`
		n/a

E
Available Conditions

Learn about conditions available in App Builder.

- What Are Conditions?
  Learn about conditions.
- Conditions Available in App Builder
  Review descriptions of conditions available in Oracle APEX.

Related Topics

- Understanding Conditional Rendering and Processing
  Use conditions in an application to control how pages and page components display and when processes, computations, and validations execute.

E.1 What Are Conditions?

Learn about conditions.

A condition is a small unit of logic that helps developers control the display of regions, items, buttons, and tabs as well execute processes, computations and validations. When you apply a condition to a control or component, the condition is evaluated. Whether a condition passes or fails determines whether a control or component displays, or page processing executes.

You can specify conditions by selecting a condition type when you create the control (region, button, or item) or component (tab, list, or navigation bar), or by making a selection under the condition attribute.

E.2 Conditions Available in App Builder

Review descriptions of conditions available in Oracle APEX.

> **Tip:**
>
> To view a complete listing of all available conditions for a given control or component, expand the condition type list.

Condition	Description
Rows returned	The SQL query returns one or more rows.
No Rows returned	The SQL Query returns no rows
SQL Expression	The SQL expression evaluates to TRUE.
PLSQL Expression	The SQL expression evaluates to TRUE.
PL/SQL Function Body	The PL/SQL function body returns TRUE.
Request = Value	The page request is equal to the text you enter into the Value attribute.
Request != Value	The page request is not equal to the text you enter into the Value attribute.

Condition	Description
Request is contained in Value	The page request is contained in the text you enter into the Value attribute.
Request is NOT contained in Value	The page request is not contained in the text you enter into the Value attribute.
Item = Value	The value of the selected item is equal to the text you enter into the Value attribute.
Item != Value	The value of the selected item is not equal to the text you enter into the Value attribute.
Item is NULL	The value of the selected item is empty.
Item is NOT NULL	The value of the selected item is not empty.
Item is zero	The value of the selected item is the number zero.
Item is NOT zero	The value of the selected item is not the number zero.
Item is NULL or zero	The value of the selected item is empty or the number zero.
Item is NOT NULL and NOT zero	The value of the selected item is not empty and not the number zero.
Item contains no spaces	The value of the selected item has no spaces.
Item is numeric	The value of the selected Item is numeric.
Item is NOT numeric	The value of the selected item is not numeric.
Item is alphanumeric	The value of the selected item is alphanumeric, containing only letters or numbers and no special characters.
Item is in colon delimited list	The value of the selected item is completely contained in the text you enter into the Value attribute.
Item is NOT in colon delimited list	The value of the selected Item is not completely contained in the text you entered into the Value attribute.
User Preference = Value	The value of the Preference entered is equal to the text you enter into the Value attribute.
User Preference != Value	The value of the Preference entered is not equal to the text you enter into the Value attribute.
Current Page = Page	The current page is equal to the value you enter into Page.
Current Page != Page	The current page is not equal to the value you enter into Page.
Current Page is in comma delimited list	The current page is in the comma separated list you enter into Pages.
Current Page is NOT in comma delimited list	The current page is not in the comma separated list you enter into Pages.
Current page is in Printer Friendly mode	The current page has been toggled to 'Printer Friendly Mode' by the end user.
Current page is NOT in Printer Friendly mode	The current page has not been toggled to 'Printer Friendly Mode' by the end user.
Page/Region is Read Only	Either the page or parent region, if appropriate, 'Read Only' condition evaluates to TRUE.
Page/Region is NOT Read Only	The page and parent region, if appropriate, are displayed normally.
User is authenticated (not public)	The end user is logged into the application. **See Also:** Establishing User Identity Through Authentication for information about authentication
User is the public user (user has not authenticated)	The end user is not logged into the application.
Inline Validation Errors Displayed	An inline validation error is displayed on the page, following a page submission.
Inline Validation Errors NOT displayed	An inline validation error is not displayed on the page.

Condition	Description
Current Language = Value	The end user is currently running the application in the language you enter into the Value attribute.
Current Language != Value	The end user is not currently running the application in the language you enter into the Value attribute.
Current Language is contained in Value	The end user is currently running the application in one of the languages you enter into the Value attribute.
Current Language is NOT contained in Value	The end user is not currently running the application in one of the languages you enter into the Value attribute.
Never	The component is never rendered. **Note:** Changing a condition to Never will remove any existing condition. For temporarily disabling a component, Oracle recommends using the Build Option attribute.

F

About Granted Privileges

Learn about grant privileges in Oracle APEX.

- About Granting Privileges to Database Users
  Review granted privileges Oracle recommends for database users.
- Understanding Privileges Granted to PUBLIC
  Learn about privileges granted to PUBLIC.

F.1 About Granting Privileges to Database Users

Review granted privileges Oracle recommends for database users.

In APEX, no privileges are granted to database users when creating a workspace on an existing schema. Unless direct grants exist, database users in APEX will not be able to create tables or other database objects. To allow the creation of typical database objects, Oracle recommends the following list of directly granted privileges

```
grant CREATE SESSION to [database username]

grant CREATE CLUSTER to [database username]

grant CREATE DIMENSION to [database username]

grant CREATE INDEXTYPE to [database username]

grant CREATE JOB to [database username]

grant CREATE MATERIALIZED VIEW to [database username]

grant CREATE OPERATOR to [database username]

grant CREATE PROCEDURE to [database username]

grant CREATE SEQUENCE to [database username]

grant CREATE SNAPSHOT to [database username]

grant CREATE SYNONYM to [database username]

grant CREATE TABLE to [database username]

grant CREATE TRIGGER to [database username]

grant CREATE TYPE to [database username]

grant CREATE VIEW to [database username]
```

F.2 Understanding Privileges Granted to PUBLIC

Learn about privileges granted to PUBLIC.

This section describes public synonyms that exist and for which the execute privilege is granted to `PUBLIC` for the packages, procedures, functions, tables, and views owned by the APEX product schema, (for example, `APEX_240200`).

> **Tip:**
>
> To avoid conflicts with operation of the APEX engine, Oracle recommends not using the following names for database objects in schemas associated with Oracle APEX workspaces.

- Packages
- Procedures
- Functions
- Tables
- Views
- Sequences
- Types

F.2.1 Packages

Public synonyms exist and execute privilege is granted to `PUBLIC` for the following packages owned by the APEX product schema, (for example, `APEX_240200`):

```
APEXWS

APEX_MIG_PROJECTS_UPDATE

APEX_UI_DEFAULT_UPDATE

HTMLDB_CUSTOM_AUTH

HTMLDB_ITEM

HTMLDB_LANG

HTMLDB_SITE_ADMIN_PRIVS

WWV_FLOW_APPLICATION_INSTALL

HTMLDB_UTIL

WWV_FLOW

WWV_FLOW_API

WWV_FLOW_AUDIT

WWV_FLOW_CSS_API

WWV_FLOW_APPLICATION_INSTALL

WWV_FLOW_AUDIT

WWV_FLOW_COLLECTION

WWV_FLOW_CSS
```

```
WWV_FLOW_CSS_API
WWV_FLOW_CUSTOMIZE
WWV_FLOW_CUSTOM_AUTH
WWV_FLOW_CUSTOM_AUTH_LDAP
WWV_FLOW_CUSTOM_AUTH_STD
WWV_FLOW_DEBUG_MESSAGE
WWV_FLOW_EPG_INCLUDE_MODULES
WWV_FLOW_FILE_MGR
WWV_FLOW_FND_USER_API
WWV_FLOW_GLOBAL
WWV_FLOW_HINT
WWV_FLOW_HTML_API
WWV_FLOW_ID
WWV_FLOW_IMAGE_API
WWV_FLOW_ITEM
WWV_FLOW_ITEM_HELP
WWV_FLOW_JAVASCRIPT
WWV_FLOW_LANG
WWV_FLOW_LDAP
WWV_FLOW_LOOKUP_TABLES
WWV_FLOW_MAIL
WWV_FLOW_PLSQL_JOB
WWV_FLOW_PLUGIN
WWV_FLOW_PLUGIN_UTIL
WWV_FLOW_PREFERENCES
WWV_FLOW_RANDOM
WWV_FLOW_RENDER_SHORTCUT
WWV_FLOW_REST
WWV_FLOW_SVG
WWV_FLOW_USER_API
WWV_FLOW_UTILITIES
WWV_FLOW_WEBSERVICES_API
WWV_FLOW_WS_IMPORT_API
WWV_MIG_ACC_LOAD
```

WWV_RENDER_CALENDAR2

WWV_RENDER_CHART2

Execute privilege is granted to PUBLIC for the following packages owned by the APEX product schema, (for example, APEX_240200):

WWV_CALCULATOR

WWV_FLOW_FLASH_CHART

WWV_FLOW_GENERATE_DDL

WWV_FLOW_IMAGE_GENERATOR

WWV_FLOW_TREE_GLOBAL_VARS

F.2.2 Procedures

Public synonyms exist and execute privilege is granted to PUBLIC for the following procedures owned by the Oracle APEX product schema, (for example, APEX_240200):

APEX

APEX_ADMIN

DEVELOPMENT_SERVICE_HOME

DEVELOPMENT_SERVICE_HOME_LOGIN

DEVELOPMENT_SERVICE_SIGNUP

F

HTMLDB

HTMLDB_ADMIN

HTMLDB_LOGIN

P

WS

WWV_FLOW_INIT_HTP_BUFFER

Z

F.2.3 Functions

Public synonyms exist and execute privilege is granted to PUBLIC for the following functions owned by the Oracle APEX product schema, (for example, APEX_240200):

NV

V

Execute privilege is granted to PUBLIC for the following function owned by the APEX product schema (for example, APEX_240200):

WWV_POPUP_FILTER

F.2.4 Tables

Public synonyms exist and select privilege is granted to `PUBLIC` for the following tables owned by the Oracle APEX product schema, (for example, `APEX_240200`):

```
WWV_FLOW_DUAL100

WWV_FLOW_TEMP_TABLE

WWV_FLOW_LOV_TEMP
```

F.2.5 Views

Public synonyms exist and select privilege is granted to `PUBLIC` for the following views owned by the Oracle APEX product schema (for example, `APEX_240200`):

```
APEX_APPLICATIONS

APEX_APPLICATION_ALL_AUTH

APEX_APPLICATION_AUTH

APEX_APPLICATION_AUTHORIZATION

APEX_APPLICATION_BC_ENTRIES

APEX_APPLICATION_BREADCRUMBS

APEX_APPLICATION_BUILD_OPTIONS

APEX_APPLICATION_CACHING

APEX_APPLICATION_COMPUTATIONS

APEX_APPLICATION_GROUPS

APEX_APPLICATION_ITEMS

APEX_APPLICATION_LISTS

APEX_APPLICATION_LIST_ENTRIES

APEX_APPLICATION_LOVS

APEX_APPLICATION_LOV_ENTRIES

APEX_APPLICATION_NAV_BAR

APEX_APPLICATION_PAGES

APEX_APPLICATION_PAGE_BRANCHES

APEX_APPLICATION_PAGE_BUTTONS

APEX_APPLICATION_PAGE_CHARTS

APEX_APPLICATION_PAGE_CHART_S

APEX_APPLICATION_PAGE_CHART_A

APEX_APPLICATION_PAGE_COMP

APEX_APPLICATION_PAGE_DA
```

```
APEX_APPLICATION_PAGE_DA_ACTS
APEX_APPLICATION_PAGE_DB_ITEMS
APEX_APPLICATION_PAGE_FLASH5
APEX_APPLICATION_PAGE_FLASH5_S
APEX_APPLICATION_PAGE_GROUPS
APEX_APPLICATION_PAGE_IR
APEX_APPLICATION_PAGE_IR_CAT
APEX_APPLICATION_PAGE_IR_CGRPS
APEX_APPLICATION_PAGE_IR_COL
APEX_APPLICATION_PAGE_IR_COMP
APEX_APPLICATION_PAGE_IR_COND
APEX_APPLICATION_PAGE_IR_GRPBY
APEX_APPLICATION_PAGE_IR_PIVOT
APEX_APPLICATION_PAGE_IR_PVAGG
APEX_APPLICATION_PAGE_IR_PVSRT
APEX_APPLICATION_PAGE_IR_RPT
APEX_APPLICATION_PAGE_IR_SUB
APEX_APPLICATION_PAGE_ITEMS
APEX_APPLICATION_PAGE_MAP
APEX_APPLICATION_PAGE_PROC
APEX_APPLICATION_PAGE_REGIONS
APEX_APPLICATION_PAGE_RPT
APEX_APPLICATION_PAGE_RPT_COLS
APEX_APPLICATION_PAGE_TREES
APEX_APPLICATION_PAGE_VAL
APEX_APPLICATION_PARENT_TABS
APEX_APPLICATION_PROCESSES
APEX_APPLICATION_RPT_LAYOUTS
APEX_APPLICATION_RPT_QRY_STMTS
APEX_APPLICATION_RPT_QUERIES
APEX_APPLICATION_SETTINGS
APEX_APPLICATION_SHORTCUTS
APEX_APPLICATION_SUBSTITUTIONS
APEX_APPLICATION_SUPP_OBJECTS
```

```
APEX_APPLICATION_SUPP_OBJ_BOPT
APEX_APPLICATION_SUPP_OBJ_CHCK
APEX_APPLICATION_SUPP_OBJ_SCR
APEX_APPLICATION_TABS
APEX_APPLICATION_TEMPLATES
APEX_APPLICATION_TEMP_BC
APEX_APPLICATION_TEMP_BUTTON
APEX_APPLICATION_TEMP_CALENDAR
APEX_APPLICATION_TEMP_LABEL
APEX_APPLICATION_TEMP_LIST
APEX_APPLICATION_TEMP_PAGE
APEX_APPLICATION_TEMP_POPUPLOV
APEX_APPLICATION_TEMP_REGION
APEX_APPLICATION_TEMP_REPORT
APEX_APPLICATION_THEMES
APEX_APPLICATION_TRANSLATIONS
APEX_APPLICATION_TRANS_MAP
APEX_APPLICATION_TREES
APEX_APPLICATION_WEB_SERVICES
APEX_APPL_ACL_ROLES
APEX_APPL_ACL_USER_ROLES
APEX_APPL_ACL_USERS
APEX_APPL_PAGE_IG_COLUMNS
APEX_APPL_PAGE_IG_COL_GROUPS
APEX_APPL_PAGE_IGS
APEX_APPL_PAGE_IG_RPTS
APEX_APPL_PAGE_IG_RPT_VIEWS
APEX_APPL_PAGE_IG_RPT_COMPUTES
APEX_APPL_PAGE_IG_RPT_FILTERS
APEX_APPL_PAGE_IG_RPT_COLUMNS
APEX_APPL_PAGE_IG_RPT_HIGHLTS
APEX_APPL_PAGE_IG_RPT_AGGS
APEX_APPL_PAGE_IG_RPT_CHT_COLS
APEX_APPL_PLUGINS
```

```
APEX_APPL_PLUGIN_ATTRIBUTES
APEX_APPL_PLUGIN_ATTR_VALUES
APEX_APPL_PLUGIN_EVENTS
APEX_APPL_PLUGIN_FILES
APEX_DEBUG_MESSAGES
APEX_DEVELOPER_ACTIVITY_LOG
APEX_DICTIONARY
APEX_DG_BLUEPRINT_SAMPLES
APEX_DG_META_TAB_COL_EXAMPLES
APEX_PATCHES
APEX_PKG_APP_ACTIVITY
APEX_PKG_APP_AUTHENTICATIONS
APEX_PKG_APPS
APEX_RELEASE
APEX_STANDARD_CONDITIONS
APEX_TEAM_BUGS
APEX_TEAM_FEEDBACK
APEX_TEAM_MILESTONES
APEX_TEAM_TODOS
APEX_THEMES
APEX_UI_DEFAULTS_ATTR_DICT
APEX_UI_DEFAULTS_COLUMNS
APEX_UI_DEFAULTS_GROUPS
APEX_UI_DEFAULTS_LOV_DATA
APEX_UI_DEFAULTS_TABLES
APEX_WORKSPACES
APEX_WORKSPACE_ACCESS_LOG
APEX_WORKSPACE_ACTIVITY_LOG
APEX_WORKSPACE_APEX_USERS
APEX_WORKSPACE_CLICKS
APEX_WORKSPACE_DEVELOPERS
APEX_WORKSPACE_FILES
APEX_WORKSPACE_LOG_SUMMARY
APEX_WORKSPACE_LOG_SUMMARY_USR
```

```
APEX_WORKSPACE_SCHEMAS
APEX_WORKSPACE_SESSIONS
APEX_WORKSPACE_SQL_SCRIPTS
APEX_WS_APPLICATIONS
APEX_WS_APP_PAGES
APEX_WS_DATA_GRID
APEX_WS_DATA_GRID_COL
WWV_FLOW_CLICKTHRU_LOG
WWV_FLOW_COLLECTIONS
WWV_FLOW_FEEDBACK_TYPES
WWV_FLOW_GROUP_USERS
WWV_FLOW_LISTS_OF_VALUES$_V
WWV_FLOW_LIST_OF_VALUES_DATA_V
WWV_FLOW_MONTHS_MON
WWV_FLOW_MONTHS_MONTH
WWV_FLOW_PLSQL_JOBS
WWV_FLOW_USERS
WWV_FLOW_USER_ACCESS_LOG
WWV_FLOW_USER_ACTIVITY_LOG
WWV_FLOW_USER_MAIL_LOG
WWV_FLOW_YEARS
```

Additionally, insert is granted on the following:

```
WWV_FLOW_FILES
```

Additionally, the following views have select privilege granted to `PUBLIC`, but no synonym:

```
WWV_FLOW_HOURS_12
WWV_FLOW_HOURS_24
WWV_FLOW_MINUTES
WWV_FLOW_MINUTES_5
```

F.2.6 Sequences

The following sequences have public synonyms and select privilege granted to public:

```
WWV_FLOW_SESSION_SEQ
WWV_SEQ
```

F.2.7 Types

The following types have execute granted to public but no synonyms:

```
WWV_FLOW_TREE_ENTRY

WWV_FLOW_TREE_SUBS
```

Index

Symbols

#APEX_FILES# Path attribute, *5-97*

A

Accept Page, *3-7*
access control, *21-40*
    about exporting, *21-36*
    authorization schemes, *21-34*
    configuring, *21-35*
    creating, *21-33*
    creating with Create Application Wizard, *5-15*
    creating with Create Page Wizard, *6-12*
access control list, *21-32*
    adding multiple users, *21-38*
    adding users, *21-38*
    adding with Create Page Wizard, *21-37*
    configuring, *21-39*
    deleting users, *21-38*
    editing users, *21-38*
access control, authorization schemes, *21-39*
Accessible Read-Only Items attribute, *5-64*
Account menu, *2-18*
activity log, *23-8*
Activity Reporting
    creating with Create Application Wizard, *5-15*
    creating with Create Page Wizard, *6-12*
activity reports, creating custom, *23-8*
Administration Menu
    Administration, *2-15*
    Change My Password, *2-15*
    Dashboards, *2-15*
    Manage Service, *2-15*
    Manage Users and Groups, *2-15*
    Monitory Activity, *2-15*
Advisor
    running on application, *24-18*
AI
    Consent Message, *5-104*
    creating application, *5-20*
    Generative AI, *5-104*
    managing, *18-122*
    Service, *5-104*
AI Assistant
    adding to app, *18-143*
AI Assistant *(continued)*
    example, *18-149*
AI attributes
    accessing, *5-104*
    configuring for runtime app, *5-103*
AI Configuration
    about, *18-137*
    copying, *18-127*, *18-141*
    creating, *18-138*
    deleting, *18-140*
    editing, *18-140*
    History report, *18-142*
    managing, *18-136*
    RAG Sources, *18-136*
    subscribing, *18-127*, *18-141*
    Utilization report, *18-142*
AI service
    creating, *18-122*
    deleting, *18-126*
    understanding, *18-121*
Allow Feedback attribute, *5-63*
Allows URLS Created After attribute, *5-81*
APEX
    user roles, *21-2*
APEX Advisor
    about, *24-18*
    running on a page, *24-19*
APEX Assistant
    creating an app, *5-20*
APEX Engine, *2-3*
APEX menu bar
    about, *2-9*
    Account menu, *2-18*
    Administration Menu, *2-15*
    Global Search, *2-10*
    Help menu, *2-17*
APEX sessions
    about, *D-1*
    correlating to database sessions, *D-1*
APEX Views, *5-134*
APEX_ACTIVITY_LOG view, *23-8*
APEX_DEBUG
    DISABLE, *24-14*
    ENABLE, *24-14*
    SET_DEBUG, *24-14*
APEX_MAIL, *19-3*

apex.util.applyTemplate, 3-13
App Builder
    accessing, 2-20, 2-25
    home page, 2-21
App Builder home page, 2-20, 2-21, 2-25
    Create button, 2-24
    navigation bar, 2-22
    Recent, 2-24
App Builder Tasks
    Recent list, 2-24
app components
    export, 25-17
App Description attribute, 5-101
App Gallery
    using, 2-39
app icons
    managing, 5-108
app properties
    reviewing, 17-4
app Session Management
    Deep Linking, 5-78
    Maximum Session Idle Time in Seconds, 5-79
    Rejoin Sessions, 5-78
    Session Idle Timeout URL, 5-80
    Session State Commits, 5-79
    Session Timeout URL, 5-79
APP_AJAX_X01, 3-48
APP_AJAX_X10, 3-48
Appearance switch
    Automatic, 2-32
    configuring, 2-32
    Dark Mode, 2-32
    Light Mode, 2-32
application
    adding a page, 6-1
    adding content from Web sites, 12-18
    adding Data Load page, 18-1
    adding legacy Data Load page, 26-9
    attaching an authorization scheme, 21-74
    changing authentication scheme, 21-65, 21-67
    controlling navigation using branches, 14-2
    copy and paste, 5-22
    copying, 5-37
    copying from Application home page, 5-38
    copying with Create Application Wizard, 5-37
    creating from SQL queries, 5-6
    creating full, 5-4
    creating fusion integration, 5-31
    creating groups, 5-121
    creating on tables, 5-6
    creating simple, 5-4
    creating with generative AI, 5-20
    debugging, 23-1, 24-1
    defining primary language, 22-6
    deleting, 5-59
application *(continued)*
    deploying, 25-3, 25-10
    editing AI attributes, 18-136
    exporting, 25-10
    from a file, 5-22
    from CSV file, 5-22, 5-27
    from JSON, 5-22
    from JSON file, 5-22, 5-27
    from spreadsheet, 5-22
    from XLSX file, 5-22, 5-27
    from XML file, 5-22, 5-27
    hiding features, 25-55
    importing export file, 25-24
    including AI Assistant, 18-133
    including Generative AI, 18-133
    language preference, 22-2
    loading application blueprint, 5-35
    remotely deploying, 25-57
    resource use, 24-17
    running, 3-3
    translatable components, 22-3
    translating, 22-2
    translating multibye languages, 22-7
    translation rendering, 22-3
    using blueprint, 5-34
    viewing app blueprint, 5-34
    viewing application blueprint, 5-35
    viewing object dependencies, 5-127
application access control
    about, 21-41
    adding user role assignments, 21-44
    copying role from another app, 21-42
    copying role from current app, 21-42
    creating roles, 21-42
    deleting roles, 21-44
    deleting user role assignments, 21-45
    editing user role assignments, 21-45
    refresh subscription, 21-43
    subscribing role, 21-42
    subscribing to, 21-43
    unsubscribe from subscription, 21-43
Application Alias attribute, 5-62
application attributes
    editing, 5-60
application backups
    about, 5-135
    viewing all, 5-136
    viewing for app, 5-137
application blueprint
    about, 5-34
    loading, 5-35
    using, 5-34
    viewing, 5-35
application checksum, 5-141
    viewing, 5-141

application computations
about, 15-32
copying from another app, 15-33
copying from current app, 15-33
creating, 15-32
executing On New Instance, 15-32
history report, 15-35
refresh subscription, 15-34
subscribing, 15-33
subscribing to, 15-34
unsubscribe from subscription, 15-34
Application Dashboard
Application Components, 5-126
Application Overview, 5-126
Native Region Component Utilization, 5-126
Page by type, 5-126
Page Components, 5-126
Plugin Region Utilization, 5-126
Security, 5-126
Templates, 5-126
Application Date Format attribute, 5-87
Application Date Time Format attribute, 5-87
Application Definition
Application Alias, 5-62
Application File Storage, 5-73
Application Groups, 5-63
Build Options, 5-72
Build Status, 5-69
Copyright Banner, 5-74
Default Error Display Location, 5-70
Default Error Display Location attribute, 5-70
Error Handling, 5-70
Error Handling Function, 5-71
Error Handling Function attribute, 5-71
Global Notification, 5-71
Message for unavailable application, 5-69
Name attribute, 5-62
Report Printing, 5-73
restrict access, 5-70
Status, 5-69
Substitutions, 5-72
Version, 5-62
Workflow Settings, 5-72
application Definition page
accessing, 5-61
enabling Friendly URLs, 3-22
application definition, editing, 5-60
Application Errors report, 5-119
Application File Storage attribute, 5-73
application groups, 5-119
assigning applications, 5-121
creating, 5-121
deleting, 5-122
managing, 5-119
removing, 5-122
using, 5-120
application groups *(continued)*
viewing, 5-120
Application Groups attribute, 5-63
application Help, creating, 10-61
Application home page
Add Feature Page, 6-14
adding page, 6-1
Export/Import icon, 2-28
Run Application button, 2-28
Shared Components icon, 2-28
Supporting Objects icon, 2-28
Utilities icon, 2-28
Application home Task list
Add Feature Page, 2-31
Browse by Facets, 2-31
Copy Application, 2-31
Delete Application, 2-31
Manage Backups, 2-31
Workflow Designer, 2-31
application item
about, 13-2, 16-19
copying from another app, 16-22
copying from current app, 16-22
creating, 16-20
creating a global variable, 16-25
editing attributes, 16-22
History report, 16-24
subscribing, 16-22
subscribing to, 16-23
Utilization report, 16-24
application language, 5-85
Application Language Derived From attribute, 5-87, 22-6
application layout
print preview mode, 12-2
shortcuts, 16-124
application page
adding, 6-1
assigning to groups, 6-22
copying, 6-4
copying in Page Designer, 7-33
creating, 6-1
creating from Application home, 6-14
deleting, 7-46
deleting in Page Designer, 7-46
deleting multiple, 7-47
deleting range of, 7-47
determining lock status, 6-24
grouping, 6-21
locking, 6-24
locking from Application Home page, 6-25
locking from Page Designer, 6-25
locking from Page Locks page, 6-25
managing, 6-1
preventing conflicts, 6-24
removing from groups, 6-23

application page *(continued)*
running, *3-3*
unlock as a Workspace Administrator, *6-27*
unlocking, *6-24*, *6-26*
unlocking from Application Home page, *6-26*
unlocking from Page Designer, *6-26*
unlocking from Page Locks page, *6-27*
application page groups, *6-22*
assigning pages, *6-22*
creating, *6-22*
editing group definition, *6-22*
removing, *6-23*
removing pages, *6-23*
viewing, *6-21*
Application Primary Language attribute, *5-87*, *22-6*
application processes
copying from another app, *15-27*
copying from current app, *15-27*
creating, *15-26*
creating error message, *15-29*
editing attributes, *15-27*
example, *15-26*
executing On New Instance, *15-24*
On Demand, *15-24*
refresh subscription, *15-28*
reports, *15-30*
subscribing, *15-27*
subscribing to, *15-28*
understanding, *15-23*
unsubscribe from subscription, *15-28*
application search
about, *16-81*
adding, *16-79*
creating list search, *16-96*
example, *16-80*
Oracle TEXT, *16-91*
Standard, *16-82*
ubiquitous, *16-101*
application settings, *16-16*
copying, *16-17*
creating, *16-16*
deleting, *16-17*
editing, *16-17*
refresh subscription, *16-18*
subscribing to, *16-17*, *16-18*
unsubscribe from subscription, *16-18*
Application Timestamp Format, *5-88*
Application Timestamp Time Zone Format, *5-88*
Application Upgrade wizard
running, *17-4*
application utilities
accessing, *5-124*
Advisor, *5-124*
Application Checksum, *5-124*
Application Dashboard, *5-124*
application utilities *(continued)*
Attribute Dictionary, *5-124*
Change History, *5-124*
Database Object Dependencies, *5-124*
Debug Messages, *5-124*
Embedded Code Utility, *5-124*
Export Repository, *5-124*
Font APEX, *5-124*
Manage Backups, *5-124*
Oracle APEX Views, *5-124*
Page Checksum, *5-124*
Recently Updated Pages, *5-124*
Upgrade Application, *5-124*
using, *5-123*
application, errors, *5-119*
applications
about, *3-2*
approvals
See tasks
apps
custom, *2-39*
refreshing Universal Theme, *11-4*, *17-2*
running Application Upgrade wizard, *17-4*
sample, *2-39*
starter, *2-39*
updating Compatibility Mode, *17-3*
upgrading, *17-1*
architecture, *2-3*
Attribute Dictionary, *15-36*
about, *15-36*
accessing, *15-36*
attributes
viewing field-level help, *2-36*
authentication, *3-9*, *21-46*
Authentication Schemes Repository, *21-66*
changing, *21-65*, *21-67*
configuring existing schemes, *21-66*
copying from another app, *21-67*
copying from current app, *21-67*
creating an authentication scheme, *21-65*
preconfigured authentication schemes, *21-48*
refresh subscription, *21-68*
reports, *21-71*
security attributes, *5-75*
subscribing to from Create / Edit page, *21-68*
understanding, *21-46*, *21-47*
unsubscribe from subscription, *21-68*
viewing current scheme, *21-65*
authentication report
Authentication Scheme History, *21-71*
Authentication Scheme Subscription, *21-71*
Authentication Scheme attribute, *5-76*
Authentication Scheme History report, *21-71*
Authentication Scheme Subscription report, *21-71*
authentication schemes
Builder Extension Sign-in, *21-50*

authentication schemes *(continued)*
cookie user accounts, *21-58*
Custom authentication, *21-51*
DAD Credentials, *21-57*
Database Account Credentials, *21-53*
HTTP Header Variable, *21-54*
LDAP Directory, *21-55*
Open Door Credentials, *21-49*
Oracle APEX Accounts, *21-58*
SAML, *21-61*
Single Sign-On verification, *21-59*
Social Sign-In, *21-62*
authorization, *3-9*
Authorization attributes, *5-76*
authorization scheme
attaching to application, *21-74*
attaching to component, *21-75*
attaching to control, *21-75*
attaching to page, *21-74*
By Component report, *21-79*
History report, *21-79*
reports, *21-79*
specifying, *5-76*
Subscription report, *21-79*
Utilization report, *21-79*
Authorization Scheme attribute, *5-76*
Authorization Scheme History report, *21-79*
Authorization Scheme Subscription report, *21-79*
Authorization Scheme Utilization report, *21-79*
authorization schemes, *21-72*
attaching to application, *21-74*
attaching to component, *21-75*
attaching to control, *21-75*
attaching to page, *21-74*
copying from another app, *21-78*
copying from current app, *21-78*
creating, *21-75*
for controlling access, *21-39*
subscribing to from Edit page, *21-77*
Automatic CSV Encoding attribute, *5-89*
Automatic mode
change appearance, *2-32*
Automatic Time Zone, *5-89*
Automation Execution Log
managing, *19-37*
purging, *19-39*
viewing, *19-37*
Automation History report
viewing, *19-40*
automations
about, *19-27*
creating, *19-26*, *19-29*
editing, *19-34*
importing and exporting, *19-28*
managing Automation Execution Log, *19-37*
purging Automation Execution Log, *19-39*
automations *(continued)*
viewing Automation Execution Log, *19-37*
viewing History report, *19-40*

B

background process, *15-19*
monitoring, *15-22*
reimporting, *15-19*, *25-29*
terminating, *15-22*
backups
about, *5-135*
managing, *5-137*
viewing all, *5-136*
viewing for app, *5-137*
basic templates
about, *3-10*
bind variables
syntax, *3-35*
using in regions, *3-36*
using in regions based on PL/SQL, *3-36*
blank page
creating with Create App Wizard, *5-10*
BLOB support in forms
about, *9-23*
displaying the BLOB, *9-24*
downloading files, *9-24*
removing the image reference, *9-25*
uploading files, *9-24*
BLOBs
APEX_UTIL.GET_BLOB_FILE_SRC, *8-210*
displaying in report, *8-209*
download support, *9-23*
including download link, *8-208*
populating, *9-23*
support, *8-207*
support in forms, *8-207*, *9-23*
support in reports, *8-207*
working with procedurally, *8-210*
blueprint
using, *5-34*
Bookmark Hash Function attribute, *5-81*
BOSS REST Data Source
about, *18-50*
about building Query By Example use cases, *18-58*
cascading list use case, *18-58*
configuring BOSS REST Service Base URL, *18-59*
creating, *18-51*
Data Profile, *18-53*
Data Profile Column Annotations, *18-53*
definition, *18-53*
DML Operations on Objects with ETags, *18-64*
features, *18-55*

BOSS REST Data Source *(continued)*
invoking business objects with nested collections, *18-69*
Operations, *18-53*
parent and child data, *18-60*
Query by Example page use case, *18-58*
reading business objects with nested collections, *18-64*
Root Resource Name Setting, *18-53*
using in PL/SQL, *18-61*
working with, *18-49*
writing business objects with nested collections, *18-68*
BOSS REST Data Source features
bulk DML, *18-55*, *18-58*
data filtering, *18-55*, *18-56*
data ordering, *18-55*, *18-57*
ETag support, *18-55*, *18-57*
lost update protection, *18-55*, *18-57*
pagination, *18-55*
validation errors display, *18-55*, *18-58*
branch
about, *15-31*
creating, *14-2*
managing in Page Designer, *7-31*
page submission, *7-32*
pre-rendering, *7-31*
When Button Pressed attribute, *3-60*
BRANCH_TO_PAGE_ACCEPT, *3-61*
branching
making conditional, *14-4*
using buttons, *13-37*
breadcrumb
Grid Edit report, *14-18*
History report, *14-18*
Utilization report, *14-18*
breadcrumb region
adding from Gallery, *14-12*
adding with Wizard, *14-11*
breadcrumb reports, *14-18*
breadcrumb template, *11-59*
Definition, *11-61*
Name, *11-61*
Substitution Strings, *11-61*
breadcrumb template substitution strings
#LINK#, *11-62*
#LONG_NAME#, *11-62*
#NAME#, *11-62*
NAME_ESC_SC#, *11-62*
breadcrumbs
about, *14-5*
about creating manually, *14-6*
adding entries, *14-15*
adding to a page, *14-12*
creating, *14-4*, *14-7*, *14-9*
creating dynamic, *14-13*
breadcrumbs *(continued)*
creating entries, *14-7*, *14-9*
deleting unused, *14-17*
editing, *14-14*
editing Attributes, *14-18*
editing Region attributes, *14-18*
navigations, *11-60*
reparenting entries, *14-17*
Browse button, *A-12*
build options, *25-47*
about, *25-50*
applying, *25-52*
changing status, *25-52*
Commented Out, *7-35*, *24-16*
copying from another app, *25-48*
copying from current app, *25-48*
creating, *25-51*
deleting, *25-53*, *25-54*
editing, *25-51*
excluding, *25-52*
exporting, *25-55*
exporting status, *25-55*
history, *25-55*
including, *25-52*
refresh subscription, *25-49*
removing associated components, *25-53*
reports, *25-54*
selecting, *25-52*
subscribing, *25-48*
subscribing to, *25-49*
unsubscribe from subscription, *25-49*
Build Options attribute, *5-72*
Build Status attribute, *5-69*
Builder Extension authentication
setting up, *21-51*
Builder Extension Sign-in
about, *21-50*
built-in substitution strings, *3-43*
button
branching, *13-37*
confirmation message, *13-42*
creating, *13-38*
displaying conditionally, *13-40*
editing, *13-39*, *13-41*, *13-42*
managing, *13-36*
names, *13-39*
preventing double-clicking, *13-41*
requiring confirmation, *13-42*
show processing, *13-41*
template, *11-62*
Warn on Unsaved Changes, *13-41*
button template
Definition, *11-63*
Name, *11-62*
Substitution Strings, *11-63*

button template substitution strings
##ICON_CSS_CLASSES##, 11-64
#BUTTON_ATTRIBUTES#, 11-63, 11-64
#BUTTON_CSS_CLASSES#, 11-63, 11-64
#BUTTON_ID#, 11-63, 11-64
#BUTTON_NAME#, 11-63, 11-64
#LABEL#, 11-63, 11-64
#LINK#, 11-63, 11-64
#ONCLICK#, 11-63, 11-64
buttons
copying in Page Designer, 7-27
creating in Page Designer, 7-27

C

Cache attribute, 5-82
calendar
accessing attributes, 10-11
add initialization JavaScript code, 10-13
creating, 10-3
creating in Page Designer, 10-9
creating legacy calendar, 26-7
creating with Create App Wizard, 5-10
creating with Create Application Wizard, 10-6
creating with Create Page Wizard, 6-7, 10-7
creating with REST Enabled SQL, 18-102
editing attributes, 10-10
enabling drag and drop, 10-15
including add and edit functions, 10-11
supported substitution strings, 26-8
viewing sample, 10-5
calendar Attributes
accessing, 10-11
calendar, legacy
accessing attributes, 26-8
cards
about, 8-19
creating actions, 8-25
creating with Create App Wizard, 5-10
creating with Create Application Wizard, 8-20
creating with Create Page Wizard, 6-7, 8-21
editing cards Attributes, 8-24
editing Region attributes, 8-22
example, 8-19
linking from, 8-25
use cases, 8-3
cascading style sheets
about, 11-42
referencing in page template, 11-92
uploading, 11-92
using custom, 11-92
Catalog Groups
adding, 18-74
deleting, 18-74
editing, 18-74
Change History report, 5-127
Character Value Comparison, 5-88
Character Value Comparison Behavior, 5-88
chart page
creating with Create Application Wizard, 10-20
chart synchronous update
creating refresh dynamic action, 10-28
chart type
switching in page designer, 10-27
charts
adding to a page, 10-24
altering display, 10-26
automatic refresh, 10-27
automatic refresh attribute
enabling, 10-28
creating, 10-17
creating with Create App Wizard, 5-11
creating with Create Page Wizard, 6-7, 10-23
creating with REST Enabled SQL, 18-101
custom JavaScript, 10-29
dynamic action, 10-27
editing attributes, 10-26
managing in Page Designer, 10-26
viewing sample, 10-19
check box
Component Settings, 16-27
configure default value stored in page item, 16-27
Checkbox Group item type, A-3
Multiple Value attributes, 13-20
Checkbox item type, A-3
classic calendar template
comments, 11-65
Daily Calendar, 11-65
Monthly Calendar, 11-65
Name, 11-65
Template Subscription, 11-65
Weekly Calendar, 11-65
classic navigation bar, 14-26
about, 14-27
copying entry, 14-30
creating entry, 14-27
creating for feedback, 14-29
editing entry, 14-30
editing multiple entries, 14-31
History report, 14-31
reports, 14-31
Subscription report, 14-31
classic report
creating with Create App Wizard, 5-14
editing attributes, 8-186
support for file export, 8-189
use cases, 8-3
classic report regions
printing methods, 8-205

classic reports
about, 8-180
adding download link, 8-196
controlling column breaks, 8-190
controlling column display, 8-195
creating a column link, 8-193
creating updatable columns, 8-194
creating with Create Application Wizard, 8-182
creating with Create Page Wizard, 6-8, 8-183
defining a column as a list of values, 8-194
editing Appearance attributes, 8-187
editing column layout, 8-191
editing pagination, 8-188
editing templates, 8-187
managing Attributes, 8-185
managing column attributes, 8-190
sorting columns, 8-192
supported column types, 8-181
Cleanup PL/SQL Code, 5-83
collections, 18-13
color picker
Component Settings, 16-26, 16-28
configure application-level component settings, 16-28
configure display style, 16-26
Color Picker item type, A-4
column toggle report (legacy)
supported column types, 26-2
Column Toggle report (legacy)
creating, 26-3
columns templates, using JavaScript, 11-91
Combobox item type, A-5
Multiple Value attributes, 13-20
commit points, 3-7
Compatibility Mode
updating when upgrading, 17-3
Compatibility Mode attribute, 5-64
component group
refreshing, 16-133
component groups, 16-130
about, 16-131
copying, 16-132
creating, 16-131
deleting, 16-136
publishing, 16-134
subscribing to, 16-132
unsubscribing from, 16-135
Component menus
about, 2-10
Component Settings
checkbox, 16-27
Color Picker, 16-26, 16-28
Date Picker, 16-28
Geocoded Address, 16-30
interactive report, 16-31
Component Settings *(continued)*
managing, 16-25
Map, 16-32
Oracle Cloud Applications (BOSS) REST Service, 16-33
Oracle Cloud Applications (SaaS) REST Service, 16-33
Region Display Selector, 16-34
Select Many, 16-35
Server Side Geocoding, 16-35
Show AI Assistant, 16-36
Star Rating, 16-37
Switch, 16-38
viewing History report, 16-38
components
attaching an authorization scheme, 21-75
controlling access to, 3-9
displaying on all pages, 12-3
displaying on every page, 12-2
managing availability, 21-13
translating messages, 22-17, 22-18
conditional
processing, 3-7
rendering, 3-7
conditions
list of, E-1
understanding, 3-7
using, 3-7
using with buttons, 13-40
configuration
controlling, 25-47
Configuration Options
creating with Create Application Wizard, 5-16
creating with Create Page Wizard, 6-13
Configuration Procedure attribute, 5-76, 21-69
configuring, 18-85, 18-107
Consent Message attribute, 5-104
controls, controlling access to, 3-9
Copy Page Wizard
copying page, 6-4
Copyright Banner attribute, 5-74
Create App Wizard, Advanced Settings
Description, 5-18
Security, 5-18
Settings, 5-18
User Interface Attributes, 5-18
Create Application Wizard
about, 5-2
About Page
creating with Create Application Wizard, 5-15
access control, 5-15
Activity Monitoring, 5-15
adding blank page, 5-10
adding calendar, 5-10
adding cards, 5-10

Create Application Wizard *(continued)*
adding cards page, *8-12*
adding chart, *5-11*
adding classic report, *5-14*, *8-13*
adding dashboard, *5-11*
adding faceted search, *5-11*, *8-12*
adding form, *5-11*
adding interactive grid, *5-12*, *8-12*
adding interactive report, *5-13*, *8-12*
adding map, *5-13*
adding master detail, *5-14*
adding multiple reports, *5-14*
adding smart filters, *5-11*, *8-12*
adding timeline, *5-14*
adding wizard, *5-14*
Configuration Options, *5-16*
copy and paste from File, *5-27*
create fusion integration, *5-31*
Create Fusion Integration, *5-32*
creating calendar, *10-6*
creating chart page, *10-20*
creating dashboard chart page, *10-21*
creating Feedback, *5-16*
creating forms, *9-5*
creating master detail forms, *9-14*
creating Progressive Web Application (PWA), *5-54*
creating reports, *8-11*, *8-96*
creation methods, *5-2*
creation options, *5-2*
From a File, *5-22*
Install Progressive Web App
creating with Create Application Wizard, *5-15*
overview, *5-5*
page types, *5-10*
report options, *8-11*
running, *5-6*
Settings, *5-16*
Theme Style Selection, *5-16*
upload from File, *5-22*
using APEX Assistant, *5-20*
using full wizard, *5-4*
using generative AI, *5-20*
using simple wizard, *5-4*
Create Application Wizard, Settings
Application ID, *5-16*
Authentication, *5-17*
Schema, *5-17*
UI Defaults, *5-19*
Create Branch Wizard, *14-2*
Create Navigation Bar Entry Wizard, *10-65*, *14-29*
Create Page Computation Wizard, *15-2*
Create Page Wizard
About Page, *6-12*

Create Page Wizard *(continued)*
About this Application
creating with Create Page Wizard, *6-12*
Access Control, *6-12*
Activity Reporting, *6-12*
adding page, *6-1*
blank page
creating with Create Page Wizard, *6-7*
Comments
creating with Create Page Wizard, *6-8*
Component page types, *6-7*
Configuration Options, *6-13*
Content Row
creating with Create Page Wizard, *6-8*
copying page, *6-4*
creating blank page, *6-7*
creating calendar, *6-7*, *10-7*
creating cards, *6-7*, *8-14*
creating chart, *6-7*
creating charts, *10-23*
creating classic report, *6-8*, *8-15*
creating Comments, *6-8*
creating Content Row, *6-8*
creating Data Loading page, *6-8*
creating faceted search, *8-14*
creating Faceted Search, *6-8*
creating form, *6-9*
creating forms, *9-7*
creating interactive grid, *6-9*, *8-15*
creating interactive report, *6-10*, *8-15*
creating Login Page, *6-13*
creating map, *6-10*
creating maps, *10-37*
creating master detail, *6-11*
creating plug-ins, *6-12*
creating reports, *8-13*, *8-16*
creating search page, *6-11*
creating smart filters, *8-14*
creating Smart Filters, *6-11*
creating Timeline, *6-11*
creating tree, *6-11*
creating Unified Task List, *6-11*
creating wizard, *6-12*
Email Reporting, *6-13*
Faceted Search
creating with Create Page Wizard, *6-8*
Feature page types, *6-12*
Feedback, *6-13*
Legacy page types, *6-13*
Media List
creating with Create Page Wizard, *6-11*
Push Notifications
creating with Create Page Wizard, *6-12*
report options, *8-13*
Smart Filters
creating with Create Page Wizard, *6-11*

Create Page Wizard *(continued)*
Theme Style Selection, *6-13*
Timeline
creating with Create Page Wizard, *6-11*
Workflow Console, *6-12*
Create Page Wizard (Legacy)
Form on Local Procedure, *6-13*
Legacy Data Loading, *6-6*, *6-13*
Summary Page, *6-13*
Create Region Wizard
creating breadcrumb, *14-11*
credentials
authentication types, *18-114*
Database Credential support, *18-115*
defining valid urls, *18-114*
deleting, *18-119*
exporting, *18-115*
importing, *18-115*
managing, *18-113*
protecting, *18-114*
storing for external REST services, *18-113*
storing for REST Enabled SQL services, *18-113*
understanding, *18-113*
Cross Application Reports
accessing, *5-154*
cross page utilities
accessing, *5-147*
Delete Multiple Pages, *5-147*
Page Attributes, *5-147*
Page Locks, *5-147*
cross site-scripting
about, *21-18*
about safe item types, *21-19*
dynamic output, *21-20*
escaping values, *21-20*
HTML region, *21-19*
protecting dynamic output, *21-20*
protecting form items, *21-22*
protecting report regions, *21-21*
report regions, *21-21*
restricting characters on items, *21-23*
static display areas, *21-19*
using safe item types, *21-20*
CSS Finder, *5-115*
custom application
about, *25-38*
about Supporting Objects page, *25-40*
build options, *25-41*
creating supporting objects, *25-39*
creating with Supporting Objects utility, *25-38*
defining an upgrade script, *25-44*
deinstallation scripts, *25-42*
deinstalling, *25-45*
deleting installation options, *25-43*
deleting messages, *25-43*
custom application *(continued)*
deleting supporting object scripts, *25-43*
installation scripts, *25-41*
installing supporting objects, *25-43*
messages, *25-41*, *25-42*
prerequisites, *25-41*
substitutions strings, *25-41*
testing installation scripts, *25-43*
upgrade messages, *25-42*
upgrade scripts, *25-42*
upgrading, *25-44*
validations, *25-41*
viewing an install summary, *25-46*
Custom authentication
about, *21-51*
setting up, *21-52*
Custom Manifest attribute, *5-102*
custom map backgrounds
copying, *10-57*
creating, *10-54*
deleting, *10-57*
editing, *10-56*
history, *10-58*
icons, *2-4*
menu bar, *2-4*
supported types, *10-54*
usage, *10-58*
using, *10-58*

D

DAD Credentials Verification
setting up, *21-58*
DAD Credentials Verification authentication
about, *21-57*
Dark Mode
change appearance, *2-32*
Dashboard
creating with Create App Wizard, *5-11*
dashboard chart page
creating with Create Application Wizard, *10-21*
Dashboard report, *23-5*
data
accessing with database links, *18-155*
accessing with REST Enabled SQL Reference, *18-81*
Data Load Definition
about creating, *18-2*
copying copying from another app, *18-10*
copying from current app, *18-10*
creating, *18-3*
editing, *18-8*
subscribing to, *18-10*
Data Load page
about creating, *18-2*

Data Load page *(continued)*
adding to application, *18-1*
creating, *18-7*
Data Load Wizard
about creating, *18-2*
creating Data Load page, *18-7*
data loading
about, *26-9*
adding, *18-1*
Data Loading
creating with Create Page Wizard, *6-8*
data synchronization
about, *18-75*
exporting, *18-76*
importing, *18-76*
Database Account Credentials authentication
about, *21-53*
setting up, *21-53*
Database Credential
support, *18-115*
database link, *18-155*
about, *18-156*
creating, *18-156*
viewing existing database links, *18-156*
Database Object Dependencies
about, *5-128*
clear dependency records, *5-133*
generating, *5-129*
rescan application, *5-132*
understanding, *5-130*
using tree, *5-130*
viewing, *5-127*
viewing reports, *5-130*
database sessions
about, *D-1*
database users
granted privileges, *F-1*
granting privileges, *F-1*
Date Picker
Component Settings, *16-28*
configure application-level component settings, *16-28*
Date Picker (jQuery)(Desupported), *A-9*
Date Picker item type, *A-7*
DBMS_APPLICATION_INFO, *24-17*
Debug Level
configuring, *24-4*
debug levels
about, *24-3*
Configuring programatically, *24-5*
debug messages
adding to APEX Debug Log, *24-8*
APEX_DEBUG API, *24-14*
generated by an app, *24-7*
purging, *24-7*
querying, *24-8*
debug messages *(continued)*
using Find icon, *24-7*
viewing, *24-6*
viewing from app, *24-7*
viewing from Utilities, *24-7*
Debug Messages page, *24-7*
Debug mode
accessing with f?p syntax, *24-5*
accessing with Friendly URL syntax, *24-5*
configuring for current session, *24-4*
controlling from Runtime Developer toolbar, *24-4*
debug levels, *24-3*, *24-5*
disabling, *24-4*
enabling, *24-4*
enabling and disabling, *24-2*
using URL parameters, *24-5*
utilizing, *24-2*
debug scripts
activity.sql, *24-9*
d0.sql, *24-9*
d1.sql, *24-9*
d2.sql, *24-9*
ds.sql, *24-9*
debugging, *23-1*, *24-1*
an APEX session, *24-11*
checking for errors, *24-18*
client-side logging, *24-15*
debug utility scripts, *24-9*
disabling, *24-14*
disabling for a session, *24-14*
enabling, *24-14*
enabling and disabling, *24-2*
enabling for a session, *24-14*
removing controls and components, *24-15*
sever-side logging, *24-15*
sharing debug output, *24-13*
SQL queries, *23-3*
SQL tracing, *24-17*
using debug mode, *24-2*
Debugging attribute, *5-64*
deep linking, *21-47*
Define Authorization Scheme, *5-76*
defining, *26-21*
deleting export files, *25-37*
deployment
creating supporting objects, *25-39*
exporting a page, *25-13*
exporting application definition, *25-10*
exporting feedback, *25-22*
exporting script files, *25-24*
exporting static files, *25-9*
exporting themes, *25-19*
exporting UI Defaults, *25-20*
exporting workspaces, *25-9*
importing export files, *25-24*

deployment *(continued)*
importing files, *25-24*
installing application export, *25-35*, *25-36*
installing files, *25-37*
of an application, *25-3*
options, *25-4*
publishing the URL, *25-46*
to remote instance, *25-57*
understanding the process, *25-3*
detail view
interactive grids, *8-114*
developer comments
adding in Page Designer, *5-111*
adding to an application, *5-112*
adding to application, *5-112*
deleting, *5-113*
editing, *5-113*
viewing, *5-113*
viewing reports, *5-114*
Developer Comments calendar, *5-114*
Developer Navigation tools
Account menu, *2-18*
Administration menu, *2-15*
Help menu, *2-17*
dialog
configuring attributes in Page Designer, *7-40*
dialog page template, *6-16*
dialog page template attributes, *11-76*
dialog pages, *6-15*
About branching, *6-20*
about supported, *6-16*
creating a dialog page template, *6-16*
defining, *6-17*
generating a Cancel dynamic action, *6-19*
generating a close dialog page process, *6-19*
handling refresh actions, *6-20*
Modal Dialog, *6-16*
non-modal dialog, *6-16*
required template attributes, *11-76*
Display attribute, *5-100*
Display Image item type, *A-10*
Display Map item
applying map background, *10-60*
Display Map item type, *A-11*
Display Only item type, *A-12*
doc library
downloading, *2-36*
Document Direction attribute, *5-87*
documentation
about, *2-34*
accessing, *2-34*
documents
downloading, *2-36*
downloading
doc library, *2-36*
Drill Down master detail
creating with Create Page Wizard, *9-19*
Duality View
about, *18-14*
copying, *18-17*
creating, *18-15*
Data Profile, *18-16*
deleting, *18-16*
editing, *18-16*
managing, *18-14*
subscribing, *18-17*
viewing history, *18-18*
viewing utilization, *18-18*
dynamic action
calendars, *10-16*
displaying AI Assistant, *18-143*
Generate Text with AI True Action, *18-146*
generating a text response, *18-146*
Show AI Assistant True Action, *18-143*
Dynamic Actions
creating, *13-29*
debugging, *13-35*
deleting, *13-34*
editing, *13-32*
frequency and scope, *13-33*
managing, *13-25*
understanding, *13-26*
dynamic list
about creating, *16-60*
creating from scratch, *16-64*
editing, *16-67*
syntax, *16-61*
dynamic list of values, *16-43*
dynamic text
email templates, *16-140*
dynamic translation, *22-72*

E

Edit App Icon
using, *5-111*
Edit Application Definition
Accessible Read-Only Items, *5-64*
Allow Feedback, *5-63*
Compatibility Mode, *5-64*
Debugging, *5-64*
Email from Address, *5-65*
Friendly URLs, *5-63*
Logging, *5-63*
Maximum Background Page Process Jobs, *5-68*
Proxy Server, *5-65*
Text Query Function, *5-66*
Tokenize Row Search, *5-68*
Edit Application Definition page
about, *5-61*

Edit Globalization Attributes, 5-85
Application Date Format, 5-87
Application Date Time Format, 5-87
Application Language Derived From, 5-87
Application Primary Language, 5-87
Application Timestamp Format, 5-88
Application Timestamp Time Zone Format, 5-88
Automatic CSV Encoding, 5-89
Automatic Time Zone, 5-89
Character Value Comparison, 5-88
Character Value Comparison Behavior, 5-88
Document Direction, 5-87
Edit Item Help report
accessing, 10-64
Edit Logo, 11-8
Edit Profile
change password, 2-31
email address, 2-31
first and last name, 2-31
profile photo, 2-31
Edit Security Attributes, 5-74
accessing, 5-74
Advanced, 5-84
Authentication, 5-75
Authorization, 5-76
Browser Security, 5-81
Database Session, 5-83
Session Management, 5-77
Session State Protection, 5-80
editable interactive grid, 9-2
email
configuring for instance, 19-2
email log, 19-1
email queue, 19-1
enabling, 21-11
managing, 19-1
sending with APEX_MAIL package, 19-3
Email from Address attribute, 5-65
Email Reporting
creating with Create Page Wizard, 6-13
email templates, 16-136
copying, 16-142
creating, 16-137
deleting, 16-142
editing, 16-142
History report, 16-144
refresh subscription, 16-143, 16-166
subscribing to, 16-142, 16-143, 16-166
unsubscribe from subscription, 16-143, 16-166
Utilization report, 16-144
viewing samples, 16-139
Embed in Frames attribute, 5-82
Embedded Code Utility, 5-139
Endpoint URL
populating with callback procedure, 18-85, 18-107
engine, 2-3
error message
creating for a page process, 15-18
creating for application processes, 15-29
errors, handling, 5-70
Execution Chain page process
about, 15-19
creating, 15-20
Expire Bookmarks attribute, 5-81
export
app components, 25-6
app related files, 25-6
application, 25-6, 25-10
application attributes, 25-17
breadcrumb, 25-17
build option, 25-17
feedback, 25-22
list, 25-17
page, 25-13
page from Export Page, 25-14
page from Page Designer, 25-15
plug-in, 25-19
REST Source Catalog, 25-23
script file, 25-24
static files, 25-9
templates, 25-17
text for translations, 22-9
themes, 25-19
UI Defaults, 25-20
workspace, 25-6
workspaces, 25-9
Export As Zip flag
directory structure, C-1
Export Repository, 25-35–25-37
managing export files, 25-35
using, 25-35
exported application
importing, 25-24
Extensions menu
about, 2-15

F

f?p syntax, 3-26
f?p URL Syntax
understanding, 3-25
faceted group
creating, 8-84
faceted search
about creating, 8-54
about Data Dictionary Cache, 8-54
auto-discovered facets, 8-54
configuring chart display, 8-87

faceted search *(continued)*
creating manually, *8-58*
creating with Create Application Wizard, *8-56*
creating with Create Page Wizard, *8-57*
editing, *8-65*, *8-66*
editing range facet, *8-88*
example, *8-52*
facet types, *8-55*
messages requiring translation, *22-65*
understanding, *8-68*
use cases, *8-3*
faceted search messages
requiring translation, *22-65*
faceted search region
editing region Attributes, *8-70*
faceted search region Attributes
Batch Facet Changes, *8-70*
Show Charts, *8-70*, *8-87*
Show Total Row Count, *8-70*
features
adding, *6-14*
managing, *6-14*
removing, *5-19*, *6-14*, *6-15*, *21-40*
feedback
about Feedback, *4-33*
adding with Create Application Wizard, *4-34*
adding with Create Page Wizard, *4-35*
configuring feedback, *4-37*
configuring support for attachments, *4-37*
Feedback page, *4-36*
reviewing, *4-36*
submitting, *4-36*
tracking, *4-37*
viewing By Filing User, *4-39*
viewing Calendar page, *4-39*
viewing Dashboard, *4-37*
viewing report, *4-38*
Feedback
creating with Create Application Wizard, *5-16*
creating with Create Page Wizard, *6-13*
field-level Help, *2-36*
viewing, *2-36*
viewing in Page Designer, *2-38*
File Upload item type, *A-12*
file upload table
security risks, *21-31*
Find icon, *5-115*
Font APEX
about, *5-143*
browsing icons from Page Designer, *5-145*
viewing, *5-143*
footer, substitution strings, *12-13*
form
creating with Create App Wizard, *5-11*
creating with Create Page Wizard, *6-9*
form on a table, *9-3*
form with report, *9-4*
forms
BLOB support, *8-207*
creating a tabular form, *26-16*
creating master detail, *9-10*
creating using Create Page Wizard, *9-7*
creating with Create Application Wizard, *9-5*
editable interactive grid, *9-2*
form on a table, *9-3*
Lost Update Detection, *9-8*
Lost Update Type region Attribute, *9-8*
master detail, *9-10*
understanding types, *9-2*
validating input, *9-22*
with report, *9-4*
Friendly URL
customizing, *3-23*
Friendly URL Syntax
understanding, *3-20*
Friendly URLs
enabling, *3-22*
Friendly URLs attribute, *5-63*
full development environment, *2-3*
functions, searching for, *5-118*
Fusion application
integrating with, *5-31*
Fusion Applications end point
using selectively required attributes, *18-47*
Fusion Applications integration
Create Application Wizard, *5-32*
Database Tools Fusion Integration Wizard, *5-32*
requirements, *5-31*
starter application, *5-32*
fusion integration
with APEX, *5-31*

G

gallery
installing apps from, *2-41*
uninstalling apps from, *2-43*
Gallery
about, *7-15*
creating region, *12-6*
using, *7-26*, *7-28*, *7-29*
viewing legacy components, *26-7*
Gallery menu
Show Legacy Components, *26-7*
General attribute, *5-99*
Generative AI
dynamic action to display AI Assistant, *18-143*
dynamic action to generate text with AI, *18-146*
enabling for application, *18-136*
enabling for runtime app, *5-103*

Generative AI *(continued)*
including in application, *18-133*
including in runtime app, *18-133*
managing, *18-120*, *18-122*
Generative AI service
creating, *18-122*, *18-129*
editing, *18-126*
History report, *18-128*
understanding, *18-121*
Utilization report, *18-128*
Geocoded Address
Component Settings, *16-30*
Geocoded Address item type, *A-14*
Geocoded Address items
applying map background, *10-60*
github, *2-40*
Global Notifications attribute, *5-71*
Global page, *12-3*
creating, *12-4*
editing, *12-4*
viewing, *12-4*
Global Search, *2-10*
global template option group, *11-48*
creating, *11-48*
editing, *11-49*, *11-53*
global template options
creating, *11-50*
editing, *11-51*
managing, *11-45*
Global Templates, *11-28*
global variable, creating, *16-25*
globalization
attributes, *22-6*
determining application language, *5-85*
understanding, *22-2*
Globalization attributes
accessing, *5-86*
graphics
managing, *12-19*
grid messages requiring translation
interactive grid, *22-23*
groups
of application pages, *6-21*

H

Help
accessing, *2-34*
creating, *10-61*
creating navigation bar entry, *10-65*
searching, *2-34*
using, *2-34*
Help menu, *2-17*
About, *2-17*
configuring, *2-36*
Discussion Forum, *2-17*
Help menu *(continued)*
Documentation, *2-17*
Learn More about Oracle APEX, *2-17*
Help text
seeding, *10-65*
Hidden item type, *A-19*
HTML
generated using custom PL/SQL, *12-19*
not handled by APEX, *12-19*
HTML Escaping Mode attribute, *5-82*
HTML password form element, *A-22*
HTTP Header Variable authentication
about, *21-54*
setting up, *21-54*
HTTP Response Headers attribute, *5-82*

I

icon view
interactive grids, *8-112*
icons
about, *5-108*
adding, *5-110*
managing in User Interface attributes, *5-91*, *5-110*
updating, *5-110*
Image Repository, *12-19*
Image Upload item type, *A-16*
image, displaying from URL, *A-10*, *A-11*
images
managing, *12-19*
implementing
plug-ins, *19-4*
import
feedback, *25-33*
REST Source Catalog, *25-34*
importing
application, *25-28*
application with background processes, *25-29*
export files, *25-24*
page, *25-25*, *25-27*, *25-28*
plug-in, *25-32*
themes, *25-31*
UI Defaults, *25-33*
Initialization PL/SQL Code, *5-83*
injection attacks, restricting characters on items, *21-23*
Installable attribute, *5-99*
installation
enabling network services, *21-15*
installing
application files, *25-35*, *25-36*
export files, *25-37*
installing application export, *25-35*, *25-36*
installing export files, *25-37*
instance proxy, specifying, *21-5*

instance settings
configuring, 21-10
configuring email, 19-2
interactive grid
creating with Create App Wizard, 5-12
creating with Create Application Wizard, 8-96
creating with Create Page Wizard, 6-9, 8-98
managing, 8-91
message requiring translation, 22-23
translating, 22-16
use cases, 8-4
interactive grid messages, translating, 22-23
interactive grids
about, 8-92, 8-94
controlling pagination, 8-108
creating a column link, 8-119
creating master detail form, 8-99
customizing in runtime, 8-121
customizing toolbar, 8-108
defining a column as list of values, 8-118
deleting a report, 8-125
displaying conditionally, 8-105
editing Attributes, 8-106
editing column attributes, 8-116
enabling Define Chart View, 8-110, 8-111
enabling Detail view, 8-114
enabling Flashback, 8-110, 8-111
enabling icon view, 8-112
enabling saving public reports, 8-110
invalid linking examples, 8-129
linking examples primary report, 8-131
linking examples saved reports, 8-131
linking restrictions, 8-129
linking to, 8-126
linking with clear cache syntax, 8-128
linking with name and value syntax, 8-127
linking with Request syntax, 8-127
making editable, 8-107
managing Attributes, 8-105
managing column attributes, 8-116
managing region attributes, 8-103
renaming saved report, 8-125
resetting a report, 8-125
saving, 8-121
saving and authentication, 8-122
saving configuration dependencies, 8-122
saving Primary report, 8-124
saving private report, 8-123
saving public report, 8-123
user save options, 8-122
valid linking examples, 8-129
interactive report
Component Settings, 16-31
configuring Actions menu structure, 16-31
creating with Create App Wizard, 5-13
interactive report *(continued)*
creating with Create Page Wizard, 6-10, 8-136
translating, 22-16
use cases, 8-4
interactive report filters, 8-166
editing, 8-166
enabling and disabling, 8-166
removing, 8-166
showing and hiding details, 8-166
interactive report messages, translating, 22-40
interactive reports
about, 8-132
about link columns, 8-141
about Search bar, 8-164
Actions menu, 8-165
adding download option, 8-148
advanced attributes, 8-156
Attributes, 8-141
changing pagination, 8-153
column attributes, 8-159
creating a column link, 8-160
creating with Create Application Wizard, 8-135
creating with REST Enabled SQL, 18-99
Customizing Actions menu, 8-147
customizing in runtime, 8-163
customizing search bar, 8-144
defining a column as a list of values, 8-161
deleting private report, 8-173
deleting public report, 8-173
displaying conditionally, 8-139
Email From Address, 8-157
enabling Icon view, 8-154
exporting, 8-148
filtering on conditional links, 8-162
formatting columns, 8-161
invalid linking examples, 8-178
linking restrictions, 8-176
linking to, 8-173
linking with clear cache syntax, 8-176
linking with name and value syntax, 8-175
linking with Request syntax, 8-174
managing Attributes, 8-140
managing column attributes, 8-158
managing region attributes, 8-138
managing saved reports, 5-148
managing subscriptions, 5-148
message requiring translation, 22-40
Region Alias, 8-157
renaming private report, 8-172
renaming public report, 8-172
Report ID Item, 8-157
saving, 8-167
saving alternative, 8-171
saving Alternative report, 8-124

interactive reports *(continued)*
saving as private, *8-169*
saving as public, *8-169*
saving primary default, *8-171*
supported column types, *8-134*
user save options, *8-168*
using filters, *8-166*
valid linking examples, *8-177*
internal messages, requiring translation, *22-19*
iOS Status Bar Style attribute, *5-101*
IP address, restricting access, *21-5*
isolating problems
Comment Out/Uncomment, *24-16*
removing controls and components, *24-15*
Server-side Condition, *24-16*
isolating workspaces, about, *21-7*
item
naming conventions, *13-3*
referencing values, *13-3*
searching for, *5-117*
Item Finder, *5-115*, *5-117*
item help
editing, *10-64*
item Help
viewing, *2-36*
item type
Checkbox, *A-3*
checkbox Group, *A-3*
Color Picker, *A-4*
Combobox, *A-5*
Date Picker, *A-7*
Date Picker (jQuery)(Desupported), *A-9*
Display Image, *A-10*
Display Map, *A-11*
Display Only, *A-12*
File Upload, *A-12*
Geocoded Address, *A-14*
Hidden, *A-19*
Image Upload, *A-16*
List Manager, *A-20*
Markdown Editor, *A-20*
Number Field, *A-21*
Password, *A-22*
Percent Graph, *A-23*
Popup LOV, *A-23*
QR Code, *A-24*
Radio, *A-25*
Rich Text Editor, *A-25*
Select list, *A-26*
Select Many, *A-27*
Select One, *A-29*
Shuttle, *A-30*
Star Rating, *A-31*, *A-33*
Switch, *A-32*
Text Area, *A-31*, *A-34*
Text Field, *A-33*
item type *(continued)*
Text Field with autocomplete, *A-34*

J

JavaScript
adding using Page Designer, *7-37*
in row templates, *11-91*
including in .js file, *11-43*
including in page template, *11-43*
incorporating, *13-48*
referencing items, *13-4*
using a dynamic action, *13-35*
JavaScript Finder, *5-115*
jQuery
support, *19-26*
JSON Sources
about, *18-19*
copying, *18-22*
creating, *18-20*
Data Profile, *18-21*
deleting, *18-21*
editing, *18-21*
managing, *18-14*, *18-19*
subscribing, *18-22*
viewing history, *18-23*
viewing utilization, *18-23*

K

keyboard shortcuts
workflows, *20-57*

L

label template
#CURRENT_ITEM_HELP_TEXT#, *11-67*
creating page item Help, *11-67*
Definition, *11-66*
editing, *11-65*
Error Display, *11-66*
Field Container, *11-67*
Name, *11-66*
Substitution Strings, *11-67*
language
defining for application, *22-6*
multibyte, *22-7*
preference, *22-2*
layout
controlling, *11-1*
LDAP Directory Verification
about, *21-55*
setting up, *21-56*
legacy calendar, *26-7*
legacy calendar template, *11-64*

legacy components
viewing in Gallery, 26-7
legacy Data Load page
adding to application, 26-9
re-creating, 26-13
Legacy Data Load page
creating, 26-11
legacy Data Load Wizard
re-creating page, 26-13
Legacy Data Load Wizard
about creating, 26-10
creating Data Load page, 26-11
creating with Create Page Wizard, 26-10
legacy data loading
about, 26-9
adding, 26-9
Light Mode
change appearance, 2-32
limits
exporting classic reports, B-1
exporting interactive reports, B-1
forms, B-1
interactive reports, B-1
item names, B-1
native Excell, B-2
native PDF, B-2
tabular forms, B-2
validations, B-1, B-2
Link Column
about, 8-141
excluding, 8-144
linking a single row view, 8-142
linking to a page, 8-143
linking to URL, 8-144
linking
deep, 21-47
list
about creating, 16-52, 16-53
about dynamic, 16-60
about navigation menus, 16-53
about static, 16-54
adding to a page, 16-68
Conditional Entries report, 16-72
copying from another app, 16-70
copying from current app, 16-70
creating hierarchical, 11-68
deleting, 16-69
editing, 16-65
editing attributes, 16-65
editing dynamic, 16-67
editing dynamic list, 16-67
editing static, 16-65
editing static list, 16-65
History report, 16-72
Lists page, 16-65
list *(continued)*
navigation menu
about creating, 16-53
refresh subscription, 16-71
reports, 16-72
subscribing to, 16-70, 16-71
Subscription report, 16-72
tab
switching to navigation menu, 16-53
types, 16-53
unsubscribe from subscription, 16-71
Unused report, 16-72
Utilization report, 16-72
list details
Grid Edit List Entries, 16-72
List Manager item type, A-20
List Manager type
Multiple Value attributes, 13-20
list of values
about, 16-40
copying from another app, 16-50
copying from current app, 16-50
creating, 16-39
creating at application-level, 16-40, 16-43
dynamic, 16-40
editing, 16-46
named, dynamic, 16-43
named, static, 16-40
referencing session state, 16-51
refresh subscription, 16-49, 16-50
shared, 16-40, 16-43
static, 16-40
static bulk edit, 16-48
subscribing to, 16-49, 16-50
translating, 22-72
unsubscribe from subscription, 16-49, 16-50
list reports
Conditional Entries, 16-72
History, 16-72
Subscription, 16-72
Unused, 16-72
Utilization, 16-72
list search
create search configuration, 16-97
creating, 16-96
creating a list, 16-96
creating search page, 16-98
list template
After List Entry, 11-70
After Sub List Entry, 11-70
Before List Entry, 11-69
Before Sub List Entry, 11-70
editing, 11-68
Name, 11-69
Sub List Entry, 11-70
Substitution Strings, 11-71, 11-75

list template *(continued)*
Template Definition, *11-69*
Template Definitions for First Entry, *11-70*
list view report (legacy)
creating, *26-4*
lists of values
Conditional Entries report, *16-52*
Edit Null Text report, *16-52*
Grid Edit report, *16-52*
History report, *16-52*
Locally Defined report, *16-52*
reports, *16-52*
Subscription report, *16-52*
Utilization report, *16-52*
Lists page, *16-65*, *16-67*
Lists Wizard, *16-60*
live template options
using, *11-54*
locks
determining lock status, *6-24*
for application page, *6-24*
Logging attribute, *5-63*
login controls, creating, *21-4*
login page
creating, *21-69*
Login Page
creating with Create Page Wizard, *6-13*
logo
adding, *5-105*
adding to navigation bar, *5-105*
editing, *5-106*
managing, *5-105*
updating, *5-105*
logs
managing, *23-5*
performance, *23-4*

M

Main application
refreshing from Working Copy, *5-49*
Manage Session State Protection attribute, *5-81*
Map
Component Settings, *16-32*
map background
applying to Display Map item, *10-60*
configuring for Geocoded Address items, *10-60*
map backgrounds
creating custom, *10-54*
maps
about, *10-31*
adding layer to existing map, *10-42*
connecting map and report, *10-47*
controlling layer visibility, *10-45*
creating, *10-30*
maps *(continued)*
creating custom map backgrounds, *10-54*
creating in Page Designer, *10-35*
creating with Create App Wizard, *5-13*
creating with Create Application Wizard, *10-34*
creating with Create Page Wizard, *6-10*, *10-37*
editing map region, *10-42*
enable Vector Tile Layers, *16-32*
one layer, *10-32*
samples, *10-32*
two layers, *10-32*
viewing in Page Designer, *10-40*, *10-59*
Markdown Editor item type, *A-20*
master application, creating, *11-33*
master detail
creating, *9-10*
creating Drill Down, *9-19*
creating Side by Side master detail, *9-18*
creating Stacked master detail, *9-16*
creating with Create App Wizard, *5-14*
creating with Create Page Wizard, *6-11*
master detail form
creating with Create Application Wizard, *9-14*
master detail form, about, *9-10*
master detail forms
about, *9-4*
master detail, single page
about, *9-4*
master detail, two page
about, *9-4*
master shared components
publishing, *16-15*
Maximum Background Page Process Jobs, *5-68*
menu, *2-18*
Message for unavailable application attribute, *5-69*
messages
about translating, *22-17*
copying from another app, *22-71*
copying from current app, *22-71*
creating for translation, *22-67*
deleting, *22-68*
editing, *22-68*
in PL/SQL procedures, *22-18*
in reports
translating, *22-18*
managing, *22-66*
refresh subscription, *22-71*
subscribing, *22-71*
subscribing to, *22-71*
translating, *22-17*, *22-18*, *22-66*
translating internal, *22-18*
translation workflow, *22-66*
unsubscribe from subscription, *22-71*

messages in reports, 22-18
messages requiring translation
interactive reports, 22-40
internal, 22-19
mobile applications
Column Toggle report (legacy), 26-2
creating, 5-51
legacy forms, 26-1, 26-2
legacy reports, 26-2
List View (legacy), 26-2
Reflow report (legacy), 26-2
monitor workspace
Application Errors, 23-4
Developer Activity, 23-4
Environment, 23-4
Login Attempts, 23-4
Page View Analysis, 23-4
Page Views, 23-4
Sessions, 23-4
Workspace Schema Reports, 23-4
multiple reports
creating with Create App Wizard, 5-14
multiple select list item, 13-5
handling return values, 13-5
using apex_string.split, 13-5
MySQL
APEXunupported features, 18-83
creating REST Enabled SQL service reference, 18-92
supported APEX components, 18-83

N

navigation
branch, 14-2
breadcrumb, 14-4
list, 16-52
navigation menu, 16-72
parent tabs, 14-19
standard tabs, 14-19
tab, 14-19
navigation bar
creating from scratch, 14-27
edit attributes, 16-78
navigation bar list
about, 16-77
editing, 16-78
managing, 16-77
navigation menu
about, 16-73
changing display, 16-75
edit attributes, 16-75
editing, 16-75
managing, 16-72
Mega Menu Navigation, 16-73
Side Navigation Menu, 16-73
navigation menu *(continued)*
switching from tabs, 16-76
Top Navigation Menu, 16-73
network services
enabling, 21-15
Number Field item type, A-21

O

object dependencies
viewing for application, 5-127
Object Storage
synchronizing with database, 16-153
Object Storage Files
downloading, 16-152
uploadiing, 16-150
viewing, 16-152
one-click remote deployment, 25-57
about, 25-57
online Help, 2-34
searching, 2-34
Open Door Credentials authentication
setting up, 21-50
OpenAPI
creating REST Data Source Catalog, 18-71
Oracle
creating REST Enabled SQL service reference, 18-90
Oracle APEX
about, 2-2
accessing in Oracle Cloud, 2-1
architecture, 2-3
component limits, B-1
signing in, 2-1
Oracle APEX Accounts authentication, 21-58
about, 21-58
setting up, 21-59
Oracle APEX administrator, 21-2
Oracle APEX Advisor
checking application integrity, 24-18
Oracle APEX Application Development
accessing APEX, 2-1
Oracle Autonomous Database
accessing Oracle APEX, 2-1
Oracle BI Publisher, integrating with, 21-14
Oracle Cloud
accessing Oracle APEX, 2-1
Oracle Cloud Applications (BOSS) REST Service Component Settings, 16-33
Oracle Cloud Applications (SaaS) REST Service Component Settings, 16-33
configuring sandbox name, 16-33
Oracle Cloud Apps BOSS REST Service
about, 18-50
creating, 18-51
REST Data Sources definitions, 18-53

Oracle Cloud Apps BOSS REST Service *(continued)*
working with, *18-49*
Oracle Cloud SaaS app
about REST Data Source support, *18-39*
configuring a sandbox, *18-45*
creating REST Data Source, *18-40*
Oracle Cloud SaaS apps
REST Data Sources definitions, *18-41*
using REST Data Sources, *18-39*
Oracle Help Center
searching, *2-34*
Oracle JET
about, *10-1*
integrated with Oracle APEX, *10-2*
integration, *10-1*
Oracle Text
search configuration, *16-92*
Oracle TEXT
creating Oracle TEXT index, *16-91*
creating search page, *16-94*
Oracle TEXT index
creating, *16-91*
Oracle TEXT search
creating, *16-91*
creating Oracle TEXT index, *16-91*
Oracle Vector Search configuration
adding a VECTOR column, *16-107*
creating, *16-107*, *16-108*
retrieving embeddings, *16-108*
Order By Item
configuring, *8-197*
example, *8-196*
output escaping
substitution strings, *3-11*, *3-41*, *11-43*

P

packages, searching for, *5-118*
page
copying, *6-4*
creating, *6-1*
page attributes
editing in Page Designer, *7-36*
viewing in Page Designer, *7-36*
Warn on Unsaved Changes, *7-41*
page checksum, *5-141*
viewing, *5-142*
page components, controlling access, *21-39*
page computation
computation point, *15-4*
creating, *15-2*
defining computation point, *15-3*
defining computation source, *15-3*
editing attributes, *15-3*
managing in Page Designer, *7-31*
page submission, *7-32*
page computation *(continued)*
pre-rendering, *7-31*
syntax, *15-4*
Page Designer
accessing, *7-23*
accessing the Property Editor, *7-23*
adding a report, dragging and dropping, *8-17*
adding buttons, *7-26*, *7-28*, *7-29*
adding items, *7-26*, *7-28*, *7-29*
adding regions, *7-26*, *7-28*, *7-29*
browsing Font APEX icons, *5-145*
Code Editor, *7-17*
Comment Out context menu, *7-35*, *24-16*
configuring dialog attributes, *7-40*
copying buttons, *7-30*
copying items, *7-30*
copying page, *7-33*
copying regions, *7-30*
creating region, *12-5*
deleting application page, *7-46*
dragging and dropping tabs, *7-24*
editing page attributes, *7-36*
editing pages, *7-1*
Gallery, *7-15*
Help, *7-14*
hiding and showing components, *7-35*, *24-16*
Layout, *7-11*
left pane, *7-6*
managing page attributes, *7-36*
marking region titles as translatable, *22-5*
Page Search tab, *7-13*
Property Editor, *7-16*
Reset Layout, *7-24*, *7-25*
Right Pane, *7-16*
running a page, *7-42*
Runtime Developer toolbar, *7-43*
searching for pages, *7-22*
supported region types, *12-8*
Three Pane mode, *7-25*
Tree pane, *7-6*
Two Pane mode, *7-25*
Uncomment context menu, *7-35*, *24-16*
using, *7-1*, *7-21*
viewing field-level help, *2-38*
working with, *7-21*
Page Designer left pane, *7-6*
about, *7-6*
Dynamic Actions tab, *7-8*
Page Shared Components tab, *7-10*
Processing tab, *7-9*
Rendering tab, *7-7*
Page Designer toolbar, *7-3*
Create menu, *7-3*
Create, Breadcrumb Region, *7-3*
Create, Copy Page, *7-3*
Create, Developer Comment, *7-3*

Page Designer toolbar *(continued)*
Create, Issue, *7-3*
Create, Page, *7-3*
Create, Page Group, *7-3*
Create, Shared Component, *7-3*
Go To Page, *7-3*
Page Lock, *7-3*
Page Unlock, *7-3*
Redo, *7-3*
Save and Run Page button, *7-3*
Save button, *7-3*
Shared Components, *7-3*
Undo, *7-3*
Utilities, *7-3*
Utilities, Advisor, *7-3*
Utilities, Caching, *7-3*
Utilities, Checksum, *7-3*
Utilities, Delete Page, *7-3*
Utilities, Export, *7-3*
Utilities, History, *7-3*
Utilities, Layout, *7-6*
Utilities, Layout, Reset Layout, *7-3*
Utilities, Layout, Three Pane Mode, *7-3*
Utilities, Layout, Two Pane Mode, *7-3*
Utilities, Show, Layout View, *7-3*
Utilities, Show, Text Messages Picker, *7-3*, *22-69*
Utilities, Show, Tooltips, *7-3*
Page Designer, page attributes
Dialog: Attributes, *7-40*
Dialog: Chained, *7-40*
Dialog: CSS Classes, *7-40*
Dialog: Height, *7-40*
Dialog: Maximum Width, *7-40*
Dialog: Width, *7-40*
Page Finder, *5-115*
page item
about, *13-2*
applying format mask., *13-23*
Authorization Scheme, *13-18*
changing to quick picks, *13-20*
configuring security, *13-18*
controlling how items write session state, *13-24*
copying in Page Designer, *7-27*
creating, *13-2*, *13-7*
creating a shuttle, *13-14*
creating cascading list of values, *13-13*
creating from Gallery, *13-8*
creating from Rendering tab, *13-7*
creating in Page Designer, *7-27*
defining default values, *13-18*
displaying as conditional, *13-22*
displaying as read only, *13-22*
editing in Property Editor, *13-17*
ession State Protection, *13-18*
page item *(continued)*
Maintain Session State attribute, *13-24*
multiple select list item, *13-5*
Multiple Value attributes, *13-20*
naming conventions, *13-3*
positioning, *13-17*
referencing using JavaScript, *13-4*
referencing values, *13-3*
Restricted Characters, *13-18*
static list of values, *13-9*
Store value encrypted in session state, *13-18*
types, *A-1*
viewing, *13-6*
viewing utilities, *13-25*
Warn on Unsaved Changes, *13-23*
page item reports
All Page Items, *13-25*
Conditional Items, *13-25*
Grid Edit of all Item Help Text, *13-25*
Grid Edit of all Item Labels, *13-25*
Item Help Subscriptions, *13-25*
page layout, about, *12-1*
page processes
background, *15-19*
changing processing points, *15-17*
changing source, *15-17*
creating, *15-14*
creating error message, *15-18*
creating success message, *15-18*
editing attributes, *15-16*
making conditional, *15-17*
managing in Page Designer, *7-31*
page submission, *7-32*
pre-rendering, *7-31*
When Button Pressed attribute, *3-60*
page processing, *3-7*
understanding, *3-6*
page rendering, *3-7*
understanding, *3-6*
Page Selector, *7-22*
page specific utilities, *5-146*
Branch Utilities, *5-146*
Button Utilities, *5-146*
Computations Utilities, *5-146*
Cross Page Utilities, *5-146*
Dynamic Action Utilities, *5-146*
Item Utilities, *5-146*
Page Groups, *5-146*
Process Utilities, *5-146*
Region Utilities, *5-146*
Validation Utilities, *5-146*
page template
Body, *11-72*
Cascading Style Sheet, *11-74*
Definition, *11-72*
Dialog, *11-75*

page template *(continued)*
editing, *11-71*
Error Page Template Control, *11-75*
Footer, *11-72*
Header, *11-72*
Image Based Tab Attributes, *11-74*
JavaScript, *11-73*
JavaScript, Dialog Cancel Code, *11-73*
JavaScript, Dialog Closure Code, *11-73*
JavaScript, Dialog Initialization Code, *11-73*
JavaScript, Execute when Page Loads, *11-73*
JavaScript, File URLs, *11-73*
JavaScript, Function and Global Variable Declaration, *11-73*
Layout, *11-74*
Name, *11-72*
Positions, *11-75*
samples, *11-71*
selecting, *11-43*
specifying defaults, *11-44*
substitution strings, *11-71*
Subtemplate, *11-74*
types, *11-71*
page template substitution strings, *11-71*
page zero, *12-2*
page-level template, *11-44*
page, attaching authorization scheme, *21-74*
pages
about Global page, *12-3*
calling from a button, *3-28*
calling with an alias, *3-27*
controlling access to, *21-39*
enabling user customization, *12-14*
exporting, *25-13*
exporting from Export Page, *25-14*, *25-15*
linking with f?p syntax, *3-26*
optimizing for printing, *12-2*
resource use, *24-17*
searching for, *5-117*
specifying page template, *11-44*
Pages Finder, *5-117*
pagination
classic report, *8-188*
interactive reports, *8-153*
reports, *11-85*, *11-90*
Pagination Subtemplate
Next Page Template, *11-91*
Next Set Template, *11-91*
Pagination Template, *11-91*
Previous Page Template, *11-91*
Previous Set Template, *11-91*
parent tabs, creating, *14-19*
Parsing Schema, *5-83*
password
best practices, *21-16*
identifying risks, *21-17*
password *(continued)*
restricting reuse, *21-8*
Security Profiles Report, *21-17*
Password item type, *A-22*
passwords
Password Items Report, *21-18*
patch, creating, *25-17*
Percent Graph item type, *A-23*
performance, *23-1*
best practices, *23-1*
database parameters, *23-6*, *23-7*
debugging problematic SQL queries, *23-3*
diagnosing issues, *23-3*
identifying issues, *23-2*
limiting resources, *23-7*
slow queries, *23-3*
slow running applications, *23-2*
uploading static files to Web server, *23-8*
using logs and report, *23-4*
PL/SQL code, searching for, *5-118*
PL/SQL Finder, *5-115*
plug-in
copying from another app, *19-23*
copying from current app, *19-23*
subscribing to, *19-23*
plug-ins
about, *19-5*
adding events, *19-22*
apex.world examples, *19-8*
attribute groups, *19-21*
Attribute Types, *19-17*
copying, *19-12*
creating, *19-8*, *19-13*
creating with Create Page Wizard, *6-12*
custom attributes, *19-15*
deleting, *19-24*
editing, *19-13*
examples, *19-8*
GitHub repository examples, *19-8*
implementing, *19-4*
importing, *25-32*
Item, Standard Attributes, *19-10*
refresh subscription, *19-14*
resetting, *19-25*
subscribing to, *19-14*
Template Component, *3-15*, *19-5*
understanding, *19-4*
unsubscribe from subscription, *19-14*
utilization, *19-25*
viewing history, *19-25*
viewing Plug-in Repository, *19-24*
Popup LOV item type, *A-23*
Multiple Value attributes, *13-20*
Popup LOV template
Buttons, *11-77*
editing, *11-76*

Popup LOV template *(continued)*
Page Attributes, *11-78*
Pagination, *11-77*
Result Set, *11-78*
Search Field, *11-77*
Window, *11-77*
preconfigured authentication schemes, *21-48*
Printer Friendly template, *12-2*
printing reports, *8-201*
privileges
about granting to database users, *F-1*
granted to database users, *F-1*
Procedure Finder, *5-118*
procedures, searching for, *5-118*
process
Cache Management, *15-14*
Data Manipulation, *15-14*
Execution Chain, *15-19*
Form Pagination, *15-14*
On Demand, *15-14*
PL/SQL, *15-14*
Report Pagination, *15-14*
Web Services, *15-14*
programming techniques
collections, *18-13*
implementing Web services, *26-19*
Progressive Web App
about support, *5-52*
App Description, *5-101*
configuring, *5-55*
creating, *5-52*, *5-54*
Custom Manifest, *5-102*
Display, *5-100*
General, *5-99*
Installability, *5-100*
Installable, *5-99*
iOS Status Bar Style, *5-101*
Push Notifications, *5-102*
Screen Orientation, *5-100*, *5-101*
Screenshots, *5-102*
Service Workers Configuration, *5-103*
Shortcuts, *5-102*
Progressive Web App attributes, *5-98*
accessing, *5-98*
configuring, *5-98*
Progressive Web Application
message requiring translation, *22-57*
Progressive Web Application, translating, *22-57*
proxy server, *26-21*
Proxy Server attribute, *5-65*
public file upload, enabling, *21-5*
public synonyms, *F-1*
Public User attribute, *5-76*
Push Notifications attribute, *5-102*
PWA messages requiring translation
Progressive Web Application (PWA), *22-57*

Q

QR Code item type, *A-24*
Query Finder, *5-115*, *5-117*
query, searching for, *5-117*

R

radio group, *A-25*
Radio item type, *A-25*
RAG Source
creating, *18-138*
deleting, *18-140*
editing, *18-140*
RAG Sources
about, *18-137*
range facet
editing static list of values, *8-88*
Recently Updated Pages report, *5-127*
reflow report (legacy)
supported column types, *26-2*
reflow report(legacy)
creating, *26-5*
region
adding an image, *12-15*
based on a URL, *12-18*
Component Settings, *16-34*
controlling positioning, *12-13*
copying, *12-17*
copying in Page Designer, *7-27*
creating for a list, *16-68*
creating from Gallery, *12-6*
creating from Rendering Tab, *12-6*
creating in Page Designer, *7-27*, *12-5*, *12-6*
creating region display selector, *12-16*
deleting, *12-18*
editing attributes, *12-11*
editing region attributes, *12-11*
enabling customization, *12-14*
enabling users to customize a page, *12-14*
Name, *12-12*
specifying a static region ID, *12-15*
specifying header and footer, *12-13*
supported types, *12-8*
Title, *12-12*
understanding, *12-5*
region attributes
Static ID, *12-15*
region display selector, creating, *12-16*
region template
Cascading Style Sheet, *11-82*
Definition, *11-80*
editing, *11-79*
Form Table Attributes, *11-81*
Image Attributes, *11-81*
JavaScript, *11-82*

region template *(continued)*
JavaScript, File URLs, *11-82*
JavaScript, Function and Global Variable Declaration, *11-82*
Name, *11-79*
Sub Regions Attributes, *11-81*
Substitution Strings, *11-83*
region type
Calendar, *12-8*
region Type
Breadcrumb, *12-8*
Cards, *12-8*
Chart, *12-8*
Classic Report, *12-8*
Column Toggle Report, *12-8*
Dynamic Content, *12-8*
Faceted Search, *12-8*
Form, *12-8*
Help Text, *12-8*
Interactive Grid, *12-9*
Interactive Report, *12-9*
List, *12-9*
List View, *12-9*
Map, *12-9*
Reflow Report, *12-9*
Region Display Selector, *12-9*
Search, *12-9*
Smart Filters, *12-10*
Static Content, *12-10*
Trees, *12-10*
URL, *12-10*
Workflow Diagram, *12-10*
Rejoin Sessions
about security risks, *21-6*
configuring at page-level, *7-38*
Release Notes
reviewing, *17-1*
reviewing Changed Behavior, *17-1*
reviewing Deprecated Features, *17-1*
reviewing Deprecated in Previous Releases, *17-1*
reviewing Desupported Features, *17-1*
remote database
accessing from existing components, *18-104*
remote deployment, *25-57*
about, *25-57*
using REST Enabled SQL references, *25-57*
Remote Server
about, *18-107*
creating, *18-110*
deleting, *18-111*
editing, *18-111*
exporting, *18-109*
importing, *18-109*
managing, *18-106*
understanding, *18-106*
Remote Server *(continued)*
viewing history, *18-113*
viewing utilization, *18-112*
Rendering tab
creating region, *12-6*
report
BLOB support, *8-207*
Oracle database, *23-6*
report column templates
colorizing cell headings, *11-85*
omitting headings, *11-85*
report column templates (generic)
After Rows, *11-87*
Before Each Row, *11-86*, *11-87*
Before Rows, *11-85*
Column Headings, *11-85*
Column Templates, *11-86*
displaying HTML after all columns, *11-87*
displaying HTML after last row, *11-87*
Next Page Template, *11-88*
Next Set Template, *11-88*
Pagination Subtemplate, *11-87*, *11-88*
Previous Page Template, *11-88*
Previous Set Template, *11-88*
Report Template, *11-85*
Row Highlighting, *11-87*
report column templates (named)
Before Rows attribute, *11-90*
colorizing column header cell, *11-90*
Column Headings, *11-90*
Pagination Subtemplate, *11-90*
Report Template Identification, *11-89*
Row Templates, *11-89*
Subscription, *11-89*
report layouts
about, *16-162*
copying, *16-165*
creating, *16-164*
editing, *16-165*
options, *16-163*
Subscribing to, *16-165*
report page comparison
cards, *8-5*
classic report, *8-6*
faceted search, *8-7*
interactive grid, *8-8*
interactive report, *8-10*
smart filters, *8-11*
report page types
comparison, *8-5*
understanding, *8-2*
use cases, *8-2*
Report Printing attribute, *5-73*
report queries
about, *16-158*
copying, *16-161*

report queries *(continued)*
creating, *16-159*
editing, *16-161*
report query
deleting unused, *16-162*
viewing History, *16-162*
viewing Utilization, *16-162*
report regions
about configuration options, *8-202*
configuring report printing, *8-203*
Print Server Type attribute, *8-203*
printing, *8-201*
printing to PDF, *8-203*
report templates, *11-83*
editing, *11-83*
generic column templates, *11-84*
named column templates, *11-84*
using JavaScript, *11-91*
reports, *11-55*
about, *8-1*
Active Session History (ASH) reports, *23-6*
adding by dragging and dropping, *8-17*
Application Dashboard, *5-126*
Automatic Workload Repository (AWR) reports, *23-6*
Change History, *5-127*
class references, *11-40*
creating on local database, *8-16*
creating with Create Application Wizard, *8-11*, *8-96*
creating with Create Page Wizard, *8-13*, *8-16*
creating with external data source, *8-16*
CSV export, *11-86*
Dashboard, *23-5*
exporting to CSV format, *11-85*, *11-90*
including pagination above, *11-85*, *11-90*
managing interactive grids, *8-91*
monitor activity, *23-6*
performance, *23-4*
printing, *8-201*
supported substitution strings, *11-40*
template subscription, *11-55*
template utilization, *11-55*
templates in a theme, *11-38*
theme file references, *11-39*
theme template counts, *11-39*
themes, *11-38*
workspace activity, *23-5*
REQUEST
about scope and value, *3-60*
about When Button Pressed attribute, *3-60*
button names, *13-39*
referencing using declarative conditions, *3-60*
referencing value of, *3-59*
substitution string, *3-59*
using for Show processing, *3-61*
resource use, monitoring, *24-17*
REST Data Source Catalog
creating, *18-70*
creating from OpenAPI, *18-71*
editing, *18-73*
managing, *18-69*
REST Data Sources
about, *18-25*
about authentication, *18-28*
about credentials, *18-28*
about Oracle Cloud Apps BOSS REST Service, *18-50*
copying, *18-38*
copying from another app, *18-38*
creating, *18-29*
creating for Oracle Cloud SaaS app, *18-40*
creating Oracle Cloud Apps BOSS REST Service, *18-51*
Data Profile, *18-26*
data synchronization, *18-75*
deleting, *18-33*, *18-37*
deleting unused, *18-81*
differences from Legacy Web Service References, *18-28*
editing, *18-33*, *18-34*
executing a SQL query, *18-32*
for Oracle Cloud SaaS app, *18-39*
managing, *18-24*
Oracle Cloud Apps BOSS REST Service, *18-49*
REST Data Source Catalogs, *18-69*
subscribing to, *18-38*
understanding, *18-24*
viewing history, *18-81*, *18-98*
viewing utilization, *18-81*
REST Data Sources (SaaS)
configuring Base URL, *18-46*
creating filter parameters, *18-44*
Data Profile, Additional Info attribute, *18-47*
parameter for REST framework version, *18-46*
runtime features, *18-42*
viewing definitions, *18-41*
working with, *18-39*
REST Data synchronization settings, *18-79*
clearing, *18-79*
REST Enabled SQL
creating calendars, *18-102*
creating charts, *18-101*
creating reports, *18-99*
editing existing components, *18-104*
REST Enabled SQL Reference, *18-81*
in remote deployment, *25-57*
REST Enabled SQL service
creating on Oracle, *18-90*
creating reference, *18-88*

REST Enabled SQL service *(continued)*
creating reference on MySQL, *18-92*
deleting, *18-97*
editing, *18-95*
requirements, *18-87*
viewing utilization, *18-98*
REST Enabled SQL Services
about, *18-83*
differences with database links, *18-84*
exporting, *18-87*
importing, *18-87*
MySQL requirements, *18-83*
Oracle Database requirements, *18-83*
requirements, *18-83*
REST Source Catalog
exporting, *18-74*
exporting and importing, *18-74*
importing, *18-74*
REST synchronization
configuring, *18-77*
executing, *18-77*
managing, *18-75*
viewing existing in app, *18-79*
viewing report, *18-80*
REST Synchronization log
about, *18-80*
log, *18-80*
RESTful access, enabling, *21-6*
restrict access, to application attribute, *5-70*
Rich Text Editor item type, *A-25*
Run Application button, *2-28*
Run on Public Pages attribute, *5-77*
Runtime API Usage, *5-83*
Runtime Developer toolbar, *7-43*
App, *7-43*
Customize, *7-43*
Debug, *7-43*
Developer Toolbar Options, *7-43*
Edit App Icon, *5-111*
Edit Logo, *5-106*
Info, *7-43*
Oracle APEX Home, *7-43*
Page, *7-43*
Quick Edit, *7-43*, *11-54*
Session, *7-43*
runtime environment, *2-3*
advantages, *21-14*

S

safe item types, *21-19*
SAML Sign In
setting up, *21-61*
SAML Sign In authentication
about, *21-61*
SAML Sign-In
about, *21-61*
sample apps
installing from gallery, *2-41*
uninstalling from gallery, *2-43*
sandbox
Oracle Cloud Applications (SaaS) REST Service, *16-33*
Save Public Report Authorization, *8-110*
Screen Orientation attribute, *5-100*, *5-101*
Screenshot attribute, *5-102*
SDLC
rapid application development, *25-3*
Spiral, *25-3*
search
about, *16-81*
adding to application, *16-79*
creating List search page, *16-98*
creating Oracle TEXT search page, *16-94*
creating Standard search page, *16-88*
creating ubiquitous search page, *16-104*
creating vector search page, *16-110*
example, *16-80*
History report, *16-124*
Utilization report, *16-124*
search configuration
about Oracle Vector Search, *16-107*
copying from another app, *16-116*
copying from current app, *16-116*
creating Oracle Vector Search, *16-106*
editing, *16-86*, *16-114*
editing Link attributes, *16-86*
Oracle Text, *16-92*
Oracle Vector Search, *16-108*
refresh subscription, *16-117*
subscribing to, *16-116*, *16-117*
ubiquitous database, *16-102*
unsubscribe from subscription, *16-117*
search facet
adding, *8-78*
Depending On, *8-72*
editing attributes, *8-72*
on column storing multiple values, *8-81*
types, *8-55*
search facet attribute
Client-Side Filtering, *8-72*
Show Chart, *8-87*
Show Chart Initially, *8-87*
search page
components, *16-118*
creating, *16-88*, *16-94*, *16-98*, *16-104*, *16-110*
creating with Create Page Wizard, *6-11*
editing, *16-118*
Search Results
Attributes, *16-120*
editing Attributes, *16-120*

Search Results *(continued)*
editing Region attributes, *16-119*
Region attributes, *16-119*
Search Source attributes, *16-122*
Search Sources
editing attributes, *16-122*
search type
Oracle TEXT, *16-91*
standard, *16-82*
ubiquitous database search, *16-101*
secure sockets layer, best practices, *21-6*
security attributes, *5-74*
Allows URLS Created After, *5-81*
authentication, *5-75*
Authentication Scheme, *5-76*
Authorization Scheme, *5-76*
Bookmark Hash Function, *5-81*
Cache, *5-82*
Configuration Procedure, *5-76*, *21-69*
Database Session, *5-83*
Define Authorization Schemes, *5-76*
Embed in Frames, *5-82*
Expire Bookmarks, *5-81*
HTML Escaping Mode, *5-82*
HTTP Response Headers, *5-82*
Manage Session State Protection, *5-81*
Pass ECID, *5-85*
Public User, *5-76*
Run on Public Pages, *5-77*
Runtime API Usage, *5-84*
Session State Protection, *5-81*
Source for Role or Group Schemes, *5-77*
Security Attributes
Advanced, *5-81*, *5-84*
Authorization Scheme, *5-76*
Database Session, *5-83*
Session Management, *5-77*
Session State Protection, *5-80*
security best practices
automating workspace provisioning, *21-10*
configuring instance settings, *21-10*
configuring storage options, *21-11*
configuring workspace settings, *21-12*
creating login controls, *21-4*
creating wallets, *21-12*
cross site-scripting protection, *21-18*
enabling email, *21-11*
enabling public file upload, *21-5*
enabling RESTful access, *21-6*
enabling RESTful services, *21-9*
for administrators, *21-1*
for developers, *21-15*
identifying risk passwords, *21-17*
isolating workspaces, *21-7*
items of type password, *21-16*
managing component availability, *21-13*
security best practices *(continued)*
managing instance security, *21-3*
managing users, *21-13*
managing workspace to schema assignments, *21-13*
managing workspaces, *21-12*
Oracle BI Publisher, *21-14*
restricting access by IP address, *21-5*
restricting password reuse, *21-8*
runtime environment, *21-14*
secure sockets layer, *21-6*
self service provisioning, *21-10*
session joins, *21-6*
session state, *21-23*
session timeout, *21-8*
specifying instance proxy, *21-5*
Security page, *5-75*
security risks, file upload table, *21-31*
seeding, *22-9*
Select list item type, *A-26*
Multiple Value attributes, *13-20*
Select Many
Component Settings, *16-35*
Select Many item type, *A-27*
Select One item type, *A-29*
Server Side Geocoding
Component Settings, *16-35*
Server-side Condition
Never, *24-16*
Service attribute, *5-104*
Service Worker attribute, *5-103*
session attributes, *D-2*
action, *D-2*
client_id, *D-2*
client_info, *D-2*
module, *D-2*
Session longops, *D-2*
session cloning
enabling, *3-37*
session ID, *3-16*
session joins
preventing browser attacks, *21-6*
security best practices, *21-6*
Session Management
specifying, *5-77*
session state
clearing, *3-32*
clearing application cache, *3-34*
clearing cache by item, *3-33*
clearing cache by page, *3-33*
clearing cache for current user session, *3-35*
clearing report regions, *3-34*
management, *3-16*, *3-17*
referencing, *3-30*
resetting an application completely, *3-35*
security best practices, *21-23*

session state *(continued)*
- session state
  - setting with bind variable, *3-31*
  - setting with computation, *3-31*
  - setting with f?p syntax, *3-31*
- setting with form submission, *3-31*
- sharing across applications, *16-25*

session state protection
- configuring all attributes, *21-27*
- configuring security attributes, *21-25*
- enabling from Session State Protection, *21-25*
- enabling using a wizard, *21-25*
- preventing URL tampering, *21-24*
- reports, *21-26*
- understanding, *21-24*
- viewing existing settings, *21-26*
- viewing summaries, *21-26*

Session State Protection attribute, *5-81*

session state values, managing, *3-29*

session timeout, utilizing, *21-8*

shared component subscriptions
- about, *16-10*
- refreshing, *16-14*
- subscription status, *16-13*

Shared Component subscriptions
- using, *16-9*

shared components
- about, *16-2*
- about automations, *19-27*
- accessing, *16-2*
- automatic copy of dependents, *16-11*
- automations, *19-26*, *19-29*
- component groups, *16-130*
- creating, *16-2*
- dependent, *16-11*
- export app components, *25-17*
- publishing master, *16-15*
- subscribing to, *16-12*
- subscriptions, *16-10*
- task definitions, *20-76*
- tasks, *20-76*
- unsubscribing to, *16-15*

Shared Components page
- AI Attributes, *16-8*
- AI Configurations, *16-8*
- AI Services, *16-8*
- Application Access Control, *16-4*
- Application Computations, *16-3*
- Application Definition, *16-3*
- Application Items, *16-3*
- Application Processes, *15-23*, *16-3*
- Application Settings, *16-3*
- Application Translations, *16-8*
- Authentication Schemes, *16-4*
- Authorization Schemes, *16-4*
- Automations, *16-7*

Shared Components page *(continued)*
- Breadcrumbs, *16-5*
- Build Options, *16-4*
- Component Groups, *16-5*
- Component Settings, *16-4*
- Credentials, *16-9*
- Data Load Definitions, *16-5*
- Duality Views, *16-7*
- Edit Application Comments, *16-9*
- Email Templates, *16-6*
- Export Application Components, *16-9*
- Globalization Attributes, *16-8*
- JSON Sources, *16-7*
- Legacy Web Service References (SOAP), *16-7*
- List of Values, *16-4*
- Lists, *16-5*
- Manage Supporting Objects, *16-9*
- Map Backgrounds, *16-6*
- Navigation Bar List, *16-5*
- Navigation Menu, *16-5*
- Plug-ins, *16-4*
- Progressive Web App, *16-6*
- Remote Servers, *16-9*
- Report Layouts, *16-7*
- Report Queries, *16-6*
- REST Data Source, *16-7*
- REST Enabled SQL, *16-9*
- REST Synchronizations, *16-9*
- Review Developer Comments, *16-9*
- Search Configurations, *16-6*
- Security Attributes, *16-4*
- Session State Protection, *16-4*
- Shortcuts, *16-5*
- Static Application Files, *16-6*
- Static Workspace Files, *16-6*
- Tabs, *16-5*
- Task Definitions, *16-7*
- Templates, *16-6*
- Text Messages, *16-8*
- Themes, *16-6*
- User Interface Attributes, *16-6*
- Workflows, *16-7*

shared interactive grids, linking to, *8-130*

shared interactive reports, linking to, *8-178*

shortcuts, *16-124*
- about, *16-124*
- copying an existing shortcut, *16-127*
- copying from another app, *16-128*
- copying from current app, *16-128*
- defining, *16-126*, *16-127*
- defining from scratch, *16-126*
- editing attributes, *16-127*
- History report, *16-130*
- HTML Text, *16-125*, *16-126*

shortcuts *(continued)*
HTML Text with Escaped Special Characters, *16-125*, *16-126*
Image, *16-125*, *16-126*
Message, *16-125*, *16-126*
Message with JavaScript Escaped Special Quotes, *16-125*, *16-126*
PL/SQL Function Body, *16-125*, *16-126*
referencing, *16-124*
refresh subscription, *16-129*
reports, *16-130*
subscribing to, *16-128*, *16-129*
Subscription report, *16-130*
supporting translatable messages, *22-4*
syntax, *16-124*
Text with JavaScript Escaped Single Quotes, *16-125*, *16-126*
unsubscribe from subscription, *16-129*
Show AI Assistant
Component Settings, *16-36*
Show Page, *3-7*
Shuttle item type, *A-30*
Multiple Value attributes, *13-20*
shuttle multiple select list, *A-30*
Side by Side master detail
creating with Create Page Wizard, *9-18*
sign in page, *2-1*
Single Sign-On
setting up, *21-60*
Single Sign-On authentication
about, *21-60*
smart filter
messages requiring translation, *22-65*
smart filter messages
requiring translation, *22-65*
smart filters
about creating, *8-30*, *8-31*
about Data Dictionary Cache, *8-31*
adding, *8-45*
auto-discovered smart filters, *8-31*
creating manually, *8-36*
creating with Create Application Wizard, *8-33*
creating with Create Page Wizard, *8-34*
editing, *8-41*
editing attributes, *8-48*
example, *8-28*
types, *8-32*
understanding, *8-43*
use cases, *8-5*
SOAP, *26-19*
Social Sign-In authentication
about, *21-62*
setting up, *21-63*
Source for Role or Group Schemes attribute, *5-77*
Split into Multiple Files flag
directory structure, *C-1*
Spotlight Search, *2-10*
spreadsheet
creating app from, *5-22*
SQL tracing, enabling, *24-17*
Stacked master detail
creating with Create Page Wizard, *9-16*
standard search
create search configuration, *16-83*
creating, *16-82*
creating search page, *16-88*
editing search configuration, *16-86*
standard tabs, creating, *14-19*
Star Rating
Component Settings, *16-37*
Star Rating item type, *A-31*, *A-33*
starter apps
installing from gallery, *2-41*
uninstalling from gallery, *2-43*
static application files
creating, *16-146*
deleting, *16-148*
deleting all, *16-149*
deleting multiple, *16-149*
downloading, *16-147*
downloading in Zip file, *16-148*
managing, *16-144*
managing in database, *16-144*, *16-146*
managing in Object Storage, *16-149*
managing in remote file storage, *16-144*
managing in remote storage, *16-149*
referencing using #APP_FILES#, *16-145*
synchronizing with Object Storage, *16-153*
uploading to Object Storage, *16-150*
uploading to remote storage, *16-150*
viewing in database, *16-147*
viewing Static Application Files report, *16-147*
Static Application Files page
viewing, *16-147*
static files
exporting, *25-9*
importing into another instance, *25-9*
static list
about, *16-54*
about creating, *16-54*
adding entries, *16-56*
adding list entries, *16-55*
adding sublists, *16-55*, *16-56*
copying entries, *16-58*
creating from scratch, *16-55*
editing, *16-65*
managing orphaned list entries, *16-59*
reparenting entries, *16-59*
resequencing entries, *16-60*
using Create List Wizard, *16-55*
static list of values, *16-40*
static substitution string, *5-72*

static workspace files
creating, 16-155
deleting, 16-157
deleting all, 16-158
downloading, 16-156
downloading all in Zip file, 16-157
managing, 16-154
referencing using #WORKSPACE_FILES#, 16-155
viewing report, 16-156
Status attribute, 5-69
storage options, configuring, 21-11
style
compatibility, 11-16
style sheet, 11-42
subscription status
viewing, 16-13
substitution strings, 3-48
#CSV_LINK#, 11-85, 11-90
#GLOBAL_NOTIFICATION#, 5-71
about built-in, 3-43
APEX SCHEMA OWNER, 3-61
APEX_CSP_DISPLAY_NONE, 3-45
APEX_FILES, 3-46
APEX$ROW_NUM, 3-46
APEX$ROW_SELECTOR, 3-47
APEX$ROW_STATUS, 3-47
APP_ALIAS, 3-48
APP_BUILDER_SESSION, 3-49
APP_DATE_TIME_FORMAT, 3-49
APP_FILES, 3-49
APP_ID, 3-47
APP_NLS_DATE_FORMAT, 3-50
APP_NLS_TIMESTAMP_FORMAT, 3-50
APP_NLS_TIMESTAMP_TZ_FORMAT, 3-50
APP_PAGE_ALIAS, 3-51
APP_PAGE_ID, 3-51
APP_REGION_ID, 3-51
APP_REGION_STATIC_ID, 3-51
APP_REQUEST_DATA_HASH, 3-52
APP_SESSION, 3-52
APP_SESSION_VISIBLE, 3-53
APP_TEXT$Message_Name, APP_TEXT$Message_Name$Lang, 3-58
APP_TITLE, 3-53
APP_UNIQUE_PAGE_ID, 3-53
APP_USER, 3-54
APP_VERSION, 3-55
AUTHENTICATED_URL_PREFIX, 3-55
best practices, 3-38
BROWSER_LANGUAGE, 3-55
calendar, 26-8
CURRENT_PARENT_TAB_TEXT, 3-55
DEBUG, 3-56
for application items, 3-39
substitution strings *(continued)*
for cards, 3-39
for case sensitivity, 3-40
for interactive grids, 3-39
for map columns, 3-39
for other report columns, 3-39
for page items, 3-39
for special characters, 3-40
in page templates, 11-71
JET_BASE_DIRECTORY, 3-56
JET_CSS_DIRECTORY, 3-57
JET_JS_DIRECTORY, 3-57
LOGOUT_URL, 3-57, 3-58
output escaping, 3-11, 3-41, 11-43
PRINTER_FRIENDLY, 3-58
PROXY SERVER, 3-59
PUBLIC_URL_PREFIX, 3-59
report of supported, 11-40
REQUEST, 3-59
SQLERRM, 3-61
static, 5-72
supported in region footer, 12-13
SYSDATE_YYYYMMDD, 3-62
tasks, 20-95
text messages (24.2 or later), 3-40
THEME_DB_FILES, 3-62
THEME_FILES, 3-62
understanding, 3-37
using double quotation marks, 3-40
viewing for template, 3-43
within templates, 3-38
workflows, 20-55
WORKSPACE_FILES, 3-63
WORKSPACE_ID, 3-63
Substitution Strings report
templates, 3-43
success message, creating for a page process, 15-18
Supported Charts
Oracle JavaScript Extension Toolkit (Oracle JET), 10-17
supporting objects
creating, 25-39
deinstalling, 25-45
Switch
Component Settings, 16-38
Switch item type, A-32
Synchronize Usage report, 18-80
System message, 2-6

T

tab
about Tabs page, 14-20
adding, 14-19
Conditional Display report, 14-25

tab *(continued)*
create Standard tab, *14-25*
create Standard tab set, *14-25*
creating, *14-20*, *14-22*
editing, *14-24*
editing multiple, *14-24*
History report, *14-25*
managing, *14-20*
move standard to different parent, *14-25*
Parent, *14-20*
rename standard, *14-25*
reports, *14-25*
resequence, *14-25*
Standard, *14-20*
Standard Tab Task, *14-25*
template support, *14-21*
Utilization report, *14-25*
tab reports
Conditional Display, *14-25*
History, *14-25*
Table Finder, *5-115*
Tables Finder, *5-118*
tables, searching for, *5-118*
Tabs page, *14-20*
tabular form
creating, *26-16*
task definitions
about, *20-77*
actions, *20-86*
copying, *20-82*
creating, *20-78*
history report, *20-88*
parameters, *20-85*
participants, *20-77*, *20-83*, *20-84*
utilization report, *20-87*
task details page
about, *20-88*
editing, *20-89*
tasks, *20-76*, *20-90*
about, *20-66*
action tasks, *20-66*
actions, *20-86*
APIs, *20-94*
approval tasks, *20-66*
bind variables, *20-95*
creating programmatically, *20-94*
creating with page process, *20-92*, *20-93*
history, *20-89*
lifecycle, *20-67*
participants, *20-83*, *20-84*
process plug-ins, *20-94*
purging, *20-94*, *20-95*
reports
Archive of Purged Tasks, *20-94*, *20-95*
Task Definition History, *20-88*
Task Definition Utilization, *20-87*
tasks *(continued)*
retention, *20-94*, *20-95*
sample application, *20-72*
states, *20-67*
substitution strings, *20-95*
task definitions
about, *20-77*
copying, *20-82*
creating, *20-78*
participants, *20-77*
task details page
about, *20-88*
editing, *20-89*
transitions, *20-67*
troubleshooting, *20-97*
Unified Task List
about, *20-90*
creating, *20-90*
editing, *20-91*
views, *20-96*
Team Development
managing feedback, *4-33*
Template Component plug-ins
about, *3-15*, *19-5*
copying, *11-78*
viewing, *11-78*
template directives, *3-64*
about, *3-64*
built-in substitutions, *3-68*
case directive, *3-66*
if directive, *3-65*
loop directive, *3-67*
with and apply directives, *3-67*
Template directives
about, *3-15*
template edit history, *11-55*
template option group
creating, *11-52*
template option presets, *11-18*
template options
about, *11-18*
default, *11-18*
groups, *11-18*
presets, *11-18*
using, *11-17*
template options groups, *11-18*
templates
about, *3-10*
apex.util.applyTemplate, *3-13*
basic, *3-10*
breadcrumbs, *11-59*
built-in substitution strings, *3-12*
button, *11-62*
changing defaults, *11-24*
columns, *11-86*, *11-91*
creating, *11-45*

templates *(continued)*
customizing, *11-41*
labels, *11-65*
legacy calendar, *11-64*
lists, *11-68*
managing, *11-56*
page, *11-71*
popup LOV, *11-76*
publishing, *11-59*
regions, *11-79*
replacing, *11-57*
replacing all, *11-58*
reports, *11-83*
selecting for a page, *11-44*
selecting theme default, *11-44*
substitution strings, *3-11*
Substitution Strings report, *3-43*
substitution syntax, *3-11*
supported substitution strings, *3-43*, *11-40*
unsubscribing to, *11-59*
using JavaScript, *3-13*
viewing for specific page, *11-57*
viewing History report, *11-55*
viewing on Templates page, *11-56*
viewing region position utilization, *11-58*
viewing reports, *11-55*
viewing Subscription report, *11-55*
viewing Utilization report, *11-55*
Text Area item type, *A-31*, *A-34*
Text Field item type, *A-33*
Text Field with autocomplete item type, *A-34*
text messages
substitutions (legacy), *3-12*
Text Messages Picker, *22-69*
Text Query Function, *5-66*
text strings, *22-18*
managing, *22-66*
translating, *22-17*, *22-18*, *22-66*
theme attributes, *11-24*
Breadcrumb, *11-27*
Button, *11-27*
Charts, *11-27*
Classic Report, *11-27*
CSS File URLs, *11-26*
Custom Classes, *11-28*
Custom Library File URLs, *11-28*
Custom Prefix Class, *11-28*
Date Picker Icon Attributes, *11-29*
Date Picker Icon Name, *11-28*
Default Label, *11-27*
Dialog Button Region, *11-28*
Dialog Content Region, *11-28*
Dialog Page, *11-28*
Error Page, *11-26*
Files, *11-29*
Footer Toolbar, *11-27*
theme attributes *(continued)*
Forms, *11-27*
Global Template Options, *11-28*
Header Toolbar, *11-27*
Interactive Reports, *11-27*
JavaScript File URLs, *11-26*
Legacy Calendar, *11-27*
Library, *11-28*
List, *11-27*
Lists, *11-27*
Login Page, *11-26*
Name, Application, *11-25*
Name, Description, *11-25*
Name, Identifier, *11-25*
Name, Name, *11-25*
Name, Navigation Bar Implementation, *11-25*
Name, Navigation Type, *11-25*
Name, Theme Number, *11-25*
Navigation Bar List, *11-26*
Navigation Menu List (Top), *11-26*
Navigation Menu List Position, *11-26*
Optional Label, *11-27*
Page, *11-26*
Printer Friendly Page, *11-27*
Region, *11-27*
Report, *11-27*
Required Label, *11-27*
Styles, *11-29*
Theme Subscription, *11-25*
Wizards, *11-27*
theme reports
Application Templates, *11-38*
class references, *11-40*
Class References, *11-40*
File References, *11-39*
files references, *11-39*
supported substitution strings, *11-40*
template counts, *11-39*
Template Substitution Strings, *11-40*
templates in a theme, *11-38*
Theme Template Counts, *11-39*
Theme Roller, *11-8*
about, *11-8*
using, *11-9*, *11-14*
theme style
selecting on User Interface page, *11-7*
Theme Style Selection
creating with Create Application Wizard, *5-16*
creating with Create Page Wizard, *6-13*
theme styles, *11-5*
about, *11-5*
creating, *11-6*
deleting, *11-7*
editing, *11-7*
using, *11-1*
using Theme Roller, *11-9*, *11-14*

theme subscriptions
about, 11-20
managing, 11-19
modifying, 11-20
re-subscribing, 11-22
refreshing, 11-21
unsubscribing, 11-22
verifying refresh, 11-21
theme templates
about, 3-13
themes
about, 11-2
about subscriptions, 11-19
about Universal Theme, 11-3
changing ID, 11-32
changing identification number (ID), 11-32
copying, 11-32
creating, 11-23
creating master application, 11-33
default templates, 11-24
deleting, 11-33
editing, 11-24
editing theme attributes, 11-24
exporting, 11-38, 25-19
exporting and importing, 11-38
exporting with different ID, 11-32
importing, 11-38, 25-31
live template options, 11-54
managing, 11-1
reports, 11-38
selecting page-level template, 11-44
switching, 11-30
switching to Universal Theme, 11-3
theme styles, 11-5
using, 11-1
timeline
creating with Create App Wizard, 5-14
Tokenize Row Search, 5-68
Trace Mode
configuring, 24-4
translating, 22-18
messages in PL/SQL procedures, 22-18
translation, 22-2
applications, 22-2
default reports, 22-16
dynamic, 22-72
dynamic text strings, 22-4
dynamic translations, 22-5
editing manually, 22-15
exporting text, 22-9
globalization support, 22-2
language identification, 22-2
mapping primary application ID, 22-8
mapping target application ID, 22-8
messages, 22-4
rules, 22-3
translation *(continued)*
seeding, 22-9
shortcuts, 22-4
steps, 22-7
templates, 22-5
translation file, 22-10
understanding, 22-7
understanding application rendering, 22-3
XLIFF, 22-10
XLIFF Target Elements, 22-12
translation file, 22-10
publishing, 22-12
uploading, 22-12
tree
creating with Create Page Wizard, 6-11
trees
about, 13-43
creating, 13-45
creating on new page, 13-44
editing attributes, 13-47
managing, 13-43

U

ubiquitous database
creating ubiquitous search index, 16-102
ubiquitous database search
creating, 16-101
creating index, 16-102
ubiquitous search
creating search page, 16-104
ubiquitous search configuration
creating, 16-102
ubiquitous search index
creating, 16-102
UI Defaults
exporting, 25-20
importing, 25-33
Unified Task List, 20-90
creating with Create Page Wizard, 6-11
See also tasks
Universal Theme
advantages, 5-51
Migration Guide, 11-3
Mobile Theme - 51, 11-3
refreshing, 11-4, 17-2
responsive design, about, 11-3
switching to, 11-3
Universal Theme - 42, 11-3
Universal Theme Reference app, 3-14, 5-51
accessing, 3-14
unsaved changes
warning users, 7-41, 13-23, 13-41
upgrading
apps, 17-1

upgrading apps, 17-1
Application Upgrade wizard, 17-4
refreshing Universal Theme, 11-4, 17-2
reviewing app properties, 17-4
reviewing Release Notes, 17-1
updating Compatibility Mode, 17-3
URL
f?p URL Syntax, 3-25
Friendly URL Syntax, 3-20
hiding session ID, 3-28
publishing, 25-46
SSL-enabled, 12-18
understanding syntax, 3-20
URL syntax
f?p, 3-25
friendly, 3-20
understanding, 3-20
URL tampering
preventing, 21-24
user identity
establishing, 21-46
verifying, 3-9
user interface
about, 2-2
controlling, 11-1
User Interface attributes
#APEX_FILES# Path, 5-97
#APP_FILES# Path attribute, 5-97
accessing, 5-90
Add "Built with APEX" to Footer, 5-92
Advanced, 5-97
Attributes, 5-92
Auto Dismiss Success Messages, 5-93
configuring, 5-89
CSS, 5-96
Display Navigation, 5-93
Enable End Users to choose Theme Style, 5-92
Global Page, 5-93
Home URL, 5-93
icons, 5-110
Icons, 5-91
JavaScript, 5-95
List Template, 5-94
Login URL, 5-93
Logo, 5-92, 5-105
Logo Type, 5-92
Media Type, 5-98
Navigation Bar, 5-94
Navigation Menu, 5-93
Navigation Menu List, 5-93
Position, 5-94
Template Options, 5-94
Theme, 5-93
Theme Style, 5-93
User Interface Attributes
File URLs, 5-95
Implementation, 5-94
Include deprecated JavaScript, 5-95
Include desupported JavaScript, 5-95
List Template, 5-94
Navigation Bar List, 5-94
Template Options, 5-94
User Interface Details
Concatenated Files, 5-96
Include j Query Migrate, 5-96
OpenTelemetry, 5-97
user preferences
Default Date Format, 2-32
Default Schema, 2-32
Run application new window, 2-32
Unicode CSV Downloads, 2-32
Use single window, 2-32
user roles
developer, 21-2
Oracle APEX administrator, 21-2
Workspace administrator, 21-2
users
about managing, 21-13

V

V function, 3-30
validations
about, 15-6
associating with item, 9-22
creating, 15-7, 15-9
creating with error messages, 9-22
defining error messages, 15-10
Go to Error link, 15-11
making conditional, 15-11
page submission, 7-32
When Button Pressed attribute, 3-60
Vector Providers
about, 5-151
creating, 5-152
deleting, 5-153
editing, 5-153
History report, 5-154
managing, 5-151
recent modifications, 5-154
Vector Search
creating search page, 16-110
Vector Tile Layers
Component Settings, 16-32
Version attribute, 5-62

W

wallets, creating, 21-12
Warn on Unsaved Changes, 7-41, 13-23

Web credentials
creating, 18-116
deleting unused, 18-119
editing, 18-119
managing, 18-113
viewing History, 18-120
viewing Utilization, 18-119
Web service process
editing a process, 26-36
mapping input parameters to static values, 26-36
Web service reference, 26-20
creating, 26-22
creating a report, 26-32
creating forms, 26-30
creating forms and reports, 26-29
creating manually, 26-23, 26-25
specifying WSDL, 26-22
viewing a history, 26-37
Web Service Reference page, 26-21
Web services, 26-19
creating a reference from a WSDL, 26-23
creating RESTful Web services, 26-26
editing a Web service process, 26-36
invoking as a process, 26-34
specifying proxy server address, 26-21
SSL enabled, 26-21
testing a RESTful Web service reference, 26-27
testing Web service reference manually, 26-25
using, 26-20
viewing Web service reference history, 26-37
Web service reference repository, 26-28
Web Services References page, 26-21
wizard
creating with Create App Wizard, 5-14
creating with Create Page Wizard, 6-12
wizards
creating a calendar, 10-3
creating a chart, 10-17
creating a named list of values, 16-40, 16-43
creating a page computation, 15-2
creating navigation bar entry, 10-65, 14-29
workflow activities
about, 20-4
creating, 20-29
custom process type plug-in, 20-39
end, 20-32
human task, 20-38
invoke workflow, 20-41
start, 20-31
states, 20-22
switch, 20-33
case, 20-36
check workflow variable, 20-35
workflow activities *(continued)*
switch *(continued)*
if elsif else, 20-36
true false, 20-34
transitions, 20-22
variables, 20-42
wait, 20-37
workflow branches
See workflow connections
workflow connections, 20-5
about, 20-5
adding, 20-43
Workflow Console
about, 20-15
creating, 20-45
creating with Create Page Wizard, 6-12
workflow debug messages
viewing, 20-53
workflow definitions, 20-3
Workflow Designer, 20-6
workflow parameters
about, 20-4
creating, 20-28
workflow participants
about, 20-6
adding, 20-44
Workflow Settings attribute, 5-72
workflow transitions
See workflow connections
workflow variables
about, 20-4
creating, 20-29
workflow versions
about, 20-3
activating, 20-47
copying, 20-48
creating, 20-46
deactivating, 20-48
editing, 20-47
running the active version, 20-16
saving, 20-47
workflows
about, 20-2
activities
about, 20-4
creating, 20-29
custom process type plug-in, 20-39
end, 20-32
human task, 20-38
invoke workflow, 20-41
start, 20-31
states, 20-22
switch, 20-33
case, 20-36
check workflow variable, 20-35

workflows *(continued)*
activities *(continued)*
switch *(continued)*
if elsif else, *20-36*
true false, *20-34*
transitions, *20-22*
variables, *20-42*
wait, *20-37*
connections
about, *20-5*
adding, *20-43*
copying, *20-44*
creating, *20-27*
example, *20-24*
execution order, *20-16*
frequently asked questions, *20-58*
importing, *20-49*
keyboard shortcuts, *20-57*
parameters
about, *20-4*
creating, *20-28*
participants
about, *20-6*
adding, *20-44*
resuming, *20-51*
retrying, *20-52*
starting, *20-49*
states, *20-17*
substitution strings, *20-55*
suspending, *20-51*
terminating, *20-50*
transitions, *20-17*
variables
about, *20-4*
creating, *20-29*
versions
about, *20-3*
activating, *20-47*
copying, *20-48*
creating, *20-46*
deactivating, *20-48*
editing, *20-47*
running the active version, *20-16*
saving, *20-47*
views, *20-55*
metadata, *20-56*
runtime, *20-56*
workflow console
about, *20-15*
creating, *20-45*
workflow definitions, *20-3*
Workflow Designer, *20-6*
workflow details
about, *20-15*
creating, *20-45*
workflow history report, *20-54*
workflows *(continued)*
workflow instance retention, *20-54*
workflow runtime engine, *20-16*
workflow utilization report, *20-53*
Working Copy
about, *5-38*, *5-39*
comparing between copies, *5-46*
comparing to Main, *5-44*
creating, *5-40*
deleting, *5-50*
editing, *5-43*
editing description, *5-50*
editing name, *5-50*
limitations, *5-39*
menu, *5-41*
merging into Main, *5-47*
refreshing from Main, *5-49*
tasks, *5-41*
unsupported component types, *5-39*
viewing, *5-43*
viewing details, *5-50*
Working Copy menu, *5-41*
Workspace administrator, *21-2*
Workspace home page
Administration menu, *2-15*
Available Updates region, *2-7*
Resources region, *2-7*
Summary region, *2-6*
System Message region, *2-6*
Top Apps region, *2-6*
Top Users region, *2-6*
understanding, *2-4*
Workspace Message region, *2-6*
Workspace message, *2-6*
workspace provisioning
enabling self service, *21-10*
workspace settings, configuring, *21-12*
workspace themes
adding, *11-35*
deleting, *11-36*
exporting, *11-37*
managing, *11-35*, *11-37*
modifying, *11-37*
viewing existing, *11-36*
workspace to schema assignments, managing, *21-13*
Workspace Utilities
about, *5-150*
accessing, *5-149*
APEX Views, *5-150*
Credentials, *5-150*
Cross Application Reports, *5-150*, *5-154*
Data Dictionary Cache, *5-151*
Export, *5-150*
Generative AI services, *5-150*
Manage Backups, *5-125*, *5-150*

Workspace Utilities *(continued)*
OpenTelemetry, *5-150*
Remote Servers, *5-150*
REST Enabled SQL Services, *5-150*
REST Source Catalogs, *5-150*
Vector Providers, *5-150*
Workspace Themes, *5-150*
workspaces, managing, *21-12*
WSDL document, *26-20*

X

XLIFF, *22-10*
applying, *22-13*
deleting an uploaded document, *22-14*
Target Elements, *22-12*
uploading, *22-12*
uploading and publishing, *22-12*

Z

Zip file
downloading static application files, *16-148*
downloading static workspace files, *16-157*