

Oracle® Fusion Middleware

Upgrade Guide for Oracle SOA Suite, WebCenter, and ADF

11g Release 1 (11.1.1)

E10127-01

May 2009

Oracle Fusion Middleware Upgrade Guide for Oracle SOA Suite, WebCenter, and ADF, 11g Release 1 (11.1.1)

E10127-01

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Peter LaQuerre

Contributing Authors: Liz Lynch, Savita Thakur, Thomas Van Raalte

Contributors: Tal Broda, Paul Dickson, Robert Donat, Bruno Henriques, Jay Kasi, Yogeshwar Kuntawar, HT Ma, Sankar Mani, Michael Meiner, William Norcott, Sundari Revanur, Michael Rubino, Reza Shafii, Vishal Sharma, Sitaraman Swaminathan, Arun Thomas

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 software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

Contents

| | |
|---|------|
| Preface | xiii |
| Audience | xiii |
| Documentation Accessibility | xiii |
| Related Documents | xiv |
| Conventions | xiv |
| | |
| 1 Summary of the Oracle SOA Suite, WebCenter, and ADF Upgrade Process | |
| 1.1 Flow Chart of the Oracle SOA Suite, WebCenter, and ADF Upgrade Process | 1-1 |
| 1.2 Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process.. | 1-2 |
| | |
| Part I Preparing to Upgrade Oracle SOA Suite, WebCenter, and ADF | |
| | |
| 2 Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF Upgrade | |
| 2.1 Summary of Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF..... | 2-1 |
| 2.2 Information for Oracle SOA Suite 10g Release 2 (10.1.2) Users | 2-1 |
| | |
| 3 Oracle SOA Suite, WebCenter, and ADF Information for 10g Users | |
| 3.1 Information for Oracle ADF Users | 3-1 |
| 3.2 Information for Oracle SOA Suite Users | 3-2 |
| 3.2.1 Comparing Oracle SOA Suite on OC4J and Oracle WebLogic Server | 3-2 |
| 3.2.2 Comparing the Oracle SOA Suite Directory Structure on OC4J and Oracle WebLogic Server | 3-3 |
| 3.2.3 Upgrading an Oracle SOA Suite Enterprise Deployment | 3-3 |
| 3.2.4 About Oracle BAM and Oracle B2B in Oracle Application Server 10g and Oracle Fusion Middleware 11g..... | 3-4 |
| 3.3 Information for Oracle WebCenter Users | 3-4 |
| 3.3.1 Comparing Oracle WebCenter on OC4J and Oracle WebLogic Server | 3-4 |
| 3.3.2 Comparing the Oracle WebCenter Directory Structure on OC4J and Oracle WebLogic Server | 3-5 |
| | |
| 4 Special Considerations for Oracle BPEL Long-Running Processes | |
| 4.1 What Is an Oracle BPEL Process Manager Long-Running Process? | 4-1 |
| 4.2 Two Approaches to Upgrading Long-Running Processes | 4-1 |

| | | |
|-------|---|-----|
| 4.2.1 | Using a Side-by-Side Approach to Long-Running Processes During Upgrade..... | 4-2 |
| 4.2.2 | Using a Grace Period and a Hard Stop of Long-Running Processes | 4-3 |

Part II Upgrading the Oracle SOA Suite, WebCenter, and ADF Environment

5 Upgrading an Oracle ADF Environment

| | | |
|---------|---|-----|
| 5.1 | Task 1: Decide Upon an Oracle Fusion Middleware 11g Topology for Your ADF Applications | 5-1 |
| 5.2 | Task 2: Install and Configure a Oracle WebLogic Server Domain For Your ADF Applications | 5-1 |
| 5.2.1 | Installing and Configuring a Development Domain with Oracle JDeveloper | 5-1 |
| 5.2.2 | Installing and Configuring a Domain without JDeveloper | 5-2 |
| 5.2.3 | Using the Java Required Files (JRF) Domain Template | 5-2 |
| 5.2.3.1 | Creating a New Domain With the JRF Template..... | 5-2 |
| 5.2.3.2 | Extending an Existing Domain With the JRF Template..... | 5-2 |
| 5.2.3.3 | Upgrading Existing Oracle WebLogic Server Instances to the 11.1.1.1.0 ADF Runtime | 5-3 |

6 Upgrading an Oracle SOA Suite Environment

| | | |
|---------|---|------|
| 6.1 | Task 1: Decide Upon an Oracle SOA Topology | 6-1 |
| 6.2 | Task 2: Verify that Any Required External Partners are Up and Running | 6-2 |
| 6.3 | Task 3: If Necessary, Upgrade the Oracle Database That Hosts the Oracle SOA Suite Schemas | 6-2 |
| 6.4 | Task 4: Use the Repository Creation Utility to Install the 11g SOA Schemas | 6-2 |
| 6.4.1 | Running the Repository Creation Utility in Preparation for an Oracle SOA Suite Upgrade..... | 6-3 |
| 6.4.2 | Selecting the Schemas Required for a Oracle SOA Suite Upgrade | 6-3 |
| 6.5 | Task 5: Install and Configure the Oracle SOA Middle Tiers | 6-4 |
| 6.6 | Task 6: If Necessary, Upgrade the SOA Component Schemas | 6-5 |
| 6.6.1 | Task 6a: Start the Upgrade Assistant for an Oracle SOA Schema Upgrade | 6-5 |
| 6.6.2 | Task 6b: If Necessary, Upgrade the B2B Schema | 6-7 |
| 6.6.2.1 | General Information About the B2B Schema Upgrade..... | 6-7 |
| 6.6.2.2 | Applying the Required Patch Set to the Oracle B2B 10g Middle Tier | 6-7 |
| 6.6.2.3 | Setting the Maximum Message Size Attribute In the Oracle WebLogic Server Administration Console | 6-7 |
| 6.6.2.4 | Performing the B2B Schema Upgrade | 6-8 |
| 6.6.3 | Task 6c: If Necessary, Upgrade the BAM Schema | 6-10 |
| 6.6.3.1 | Understanding the Oracle BAM Schema Upgrade | 6-10 |
| 6.6.3.2 | About Upgrading From Previous Versions of Oracle BAM | 6-10 |
| 6.6.3.3 | Applying the Required Patch Set to the Oracle Database That Contains the Oracle BAM Schema | 6-10 |
| 6.6.3.4 | Performing the Oracle BAM Schema Upgrade | 6-10 |
| 6.7 | Task 7: Configure Any Required Java EE Features to Support Your SOA Applications | 6-11 |
| 6.8 | Task 8: Perform Any Required Post-Upgrade Tasks for the SOA Components | 6-12 |
| 6.8.1 | Configuring Oracle SOA Suite 11g to Use Identity Management Components | 6-12 |

| | | |
|-------|---|------|
| 6.8.2 | About Case Sensitivity in Oracle Human Workflow | 6-12 |
| 6.8.3 | Oracle B2B Post-Upgrade Information and Tasks | 6-13 |

7 Upgrading an Oracle WebCenter Environment

| | | |
|-------|---|-----|
| 7.1 | Task 1: Decide Upon a Oracle WebCenter Topology | 7-1 |
| 7.2 | Task 2: If Necessary, Upgrade the Oracle Database That Hosts the Oracle WebCenter Schemas | 7-1 |
| 7.3 | Task 3: Use the Repository Creation Utility to Install the 11g Oracle WebCenter Schemas | 7-2 |
| 7.3.1 | Running the Repository Creation Utility in Preparation for an Oracle WebCenter Upgrade 7-2 | |
| 7.3.2 | Selecting the Schemas Required for a Oracle WebCenter Upgrade | 7-2 |
| 7.4 | Task 4: Install and Configure the Oracle WebCenter Middle Tiers | 7-3 |
| 7.5 | Task 5: Configure Any Required Java EE Features to Support Your Oracle WebCenter Applications..... | 7-4 |

Part III Upgrading the Oracle SOA Suite and Oracle WebCenter Applications

8 Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications

| | | |
|-----|--|-----|
| 8.1 | Task 1: Review Sample Application Upgrade Procedures | 8-1 |
| 8.2 | Task 2: Verify that the Applications Are Up and Running Successfully on Oracle Application Server 10g | 8-2 |
| 8.3 | Task 3: Verify That You Have Upgraded Your 10g Environment to 11g | 8-2 |
| 8.4 | Task 4: Install and Start Oracle JDeveloper 11g | 8-3 |
| 8.5 | Task 5: Open and Upgrade the Application in Oracle JDeveloper | 8-3 |
| 8.6 | Task 6: Locate Specific Upgrade Information for the Components of Your Application | 8-4 |
| 8.7 | Task 7: Review and Recompile the Application..... | 8-4 |
| 8.8 | Task 8: Redeploy the Application..... | 8-5 |

9 Considerations When Upgrading Oracle SOA Applications

| | | |
|-----------|--|-----|
| 9.1 | About Using the Oracle JDeveloper 11g Migration Wizard for Oracle SOA Suite Applications..... | 9-1 |
| 9.2 | Upgrade Tasks Associated with All Java Applications..... | 9-2 |
| 9.3 | Upgrade Tasks Associated with All Oracle SOA Suite Applications | 9-2 |
| 9.3.1 | Understanding Oracle SOA Suite API Changes for Oracle Fusion Middleware 11g | 9-3 |
| 9.3.1.1 | Upgrading to the Oracle Fusion Middleware Java API for Oracle Business Rules | 9-3 |
| 9.3.1.1.1 | Overview of the Oracle Business Rules SDK and API Changes for 11g | 9-3 |
| 9.3.1.1.2 | Accessing a Dictionary in the Development Environment..... | 9-4 |
| 9.3.1.1.3 | Accessing a Repository in a Production Environment | 9-5 |
| 9.3.1.1.4 | Generating RL Code..... | 9-6 |
| 9.3.1.2 | Upgrading to the Oracle Fusion Middleware Infrastructure Management Java API for Oracle SOA Suite | 9-6 |
| 9.3.2 | Reviewing Your Projects for Dependent JAR Files..... | 9-6 |
| 9.3.3 | Upgrading Applications That Require Proxy Settings for Web Services | 9-6 |
| 9.3.4 | Recreating build.xml and build.properties Files Not Upgraded by the | |

| | | |
|---------|---|------|
| | Migration Wizard | 9-7 |
| 9.3.5 | Upgrading Projects That Use UDDI-Registered Resources..... | 9-7 |
| 9.3.5.1 | Verifying that serviceKey Endpoints Are Available Before Upgrade | 9-7 |
| 9.3.5.2 | Changing to the orauddi Protocol..... | 9-7 |
| 9.3.6 | Using the Oracle SOA Suite Command-Line Upgrade Tool..... | 9-8 |
| 9.3.6.1 | Benefits of Using the Oracle SOA Suite Command-Line Upgrade Tool..... | 9-8 |
| 9.3.6.2 | Using the Oracle SOA Suite Command-Line Upgrade Tool with Oracle JDeveloper 11g | 9-8 |
| 9.3.6.3 | Limitations When Upgrading Human Workflow Applications with the Oracle SOA Suite Command-Line Upgrade Tool..... | 9-9 |
| 9.3.6.4 | Upgrading BPEL Projects with the Oracle SOA Suite Command-Line Upgrade Tool | 9-9 |
| 9.3.6.5 | Combining Multiple BPEL Projects Into a Single Composite with the Oracle SOA Suite Command-Line Upgrade Tool | 9-10 |
| 9.3.6.6 | Upgrading Oracle Enterprise Service Bus (ESB) Projects with the Oracle SOA Suite Command-Line Upgrade Tool | 9-11 |
| 9.3.6.7 | Upgrading Domain Value Maps (DVMs) and Cross References with the Oracle SOA Suite Command-Line Upgrade Tool..... | 9-11 |
| 9.4 | Upgrade Tasks Associated with Oracle BPEL Process Manager Applications | 9-13 |
| 9.4.1 | Manual Upgrade Steps After Migrating Oracle BPEL Process Manager Applications..... | 9-13 |
| 9.4.1.1 | Providing Missing Binding Information for External References | 9-13 |
| 9.4.1.2 | Correcting Problems With Oracle BPEL Process Manager Test Suites | 9-14 |
| 9.4.1.3 | Using Oracle BPEL Process Manager Deployment Plans After Upgrade..... | 9-14 |
| 9.4.1.4 | Upgrading Fault Policies in an Oracle BPEL Process Manager Project..... | 9-14 |
| 9.4.1.5 | Upgrading a 10g Project With No BPEL Folder | 9-15 |
| 9.4.2 | Additional Considerations for Oracle BPEL Process Manager Applications | 9-15 |
| 9.4.2.1 | Verifying New and Deprecated Properties in the bpel.xml Deployment Descriptor | 9-15 |
| 9.4.2.2 | Upgrading User-Defined (Custom) XPath Functions in an Oracle BPEL Process Manager Project..... | 9-16 |
| 9.4.2.3 | Change in Support for Multiple BPEL Implementations | 9-16 |
| 9.4.2.4 | Verifying the Properties File Name When Upgrading Projects That Use Correlation Sets in an Asynchronous Service..... | 9-16 |
| 9.4.2.5 | Upgrading Projects With the transaction=participate Property..... | 9-16 |
| 9.4.2.6 | Specifying Domain Descriptor Properties in Oracle BPEL Process Manager 11g | 9-17 |
| 9.5 | Upgrade Tasks Associated with Oracle Enterprise Service Bus | 9-21 |
| 9.5.1 | Upgrading Oracle Enterprise Service Bus Routing Rules That Are Not Exposed as Services | 9-21 |
| 9.5.2 | Upgrading Oracle Enterprise Service Bus Domain Value Maps (DVMs) and Cross References | 9-21 |
| 9.5.3 | Upgrading Oracle Enterprise Service Bus Projects with SOAP Headers to Oracle Mediator 11g | 9-21 |
| 9.5.4 | Upgrading Projects with Multiple Routing Services That Use the Same Namespace..... | 9-22 |
| 9.5.5 | Upgrading Routing Services to Use Asynchronous Routing..... | 9-22 |

| | | |
|----------|--|------|
| 9.5.6 | Upgrading Filtering or Setting Message Headers in Oracle Enterprise Service Bus..... | 9-22 |
| 9.6 | Upgrade Tasks Associated with Oracle Business Activity Monitoring..... | 9-23 |
| 9.6.1 | Migrating Oracle BAM Users and Other LDAP Information to 11g..... | 9-23 |
| 9.6.2 | Verifying Oracle BAM Action Form Templates After Upgrade..... | 9-24 |
| 9.6.3 | Upgrading Oracle BAM External Data Sources..... | 9-24 |
| 9.6.4 | Missing Images in Oracle BAM Reports..... | 9-25 |
| 9.6.5 | Using Oracle BAM Web Services in 11g..... | 9-25 |
| 9.6.5.1 | Using Web Services Clients with Oracle BAM 11g..... | 9-25 |
| 9.6.5.2 | Invoking Web Services from Oracle BAM Alerts..... | 9-26 |
| 9.6.5.3 | Using the ManualRuleFire Web Service in Oracle BAM 11g..... | 9-26 |
| 9.6.5.4 | New Support for Web Services in Oracle BAM 11g..... | 9-26 |
| 9.6.6 | Upgrading Oracle BAM Sensor Actions..... | 9-26 |
| 9.6.7 | Upgrading Enterprise Link Plans to Oracle BAM 11g..... | 9-26 |
| 9.7 | Upgrade Tasks Associated with Technology Adapters..... | 9-27 |
| 9.7.1 | Verifying Adapters with the Oracle JDeveloper 11g Adapter Configuration Wizard..... | 9-27 |
| 9.7.2 | Upgrading Adapter Header Variables..... | 9-27 |
| 9.7.3 | Upgrading Non-Managed Connection Information in the Adapter JCA File..... | 9-28 |
| 9.7.4 | Description Lost After Upgrade of Technology Adapter..... | 9-28 |
| 9.7.5 | Upgrading an Inbound Synchronous Request-Reply Use Case..... | 9-28 |
| 9.7.6 | Upgrading Projects That Use Outbound Dequeue..... | 9-29 |
| 9.7.7 | Upgrading AQ Adapter Headers..... | 9-29 |
| 9.7.8 | Upgrading an AQ Outbound Request-Reply Scenario..... | 9-30 |
| 9.7.9 | Upgrading Database Adapters Used for Stored Procedures..... | 9-30 |
| 9.8 | Upgrade Tasks Associated with Human Workflow Tasks..... | 9-31 |
| 9.8.1 | Upgrading Task Details for an Associated Human Workflow Task..... | 9-31 |
| 9.8.2 | Upgrading Multiple Projects That Share a Common Human Workflow Task..... | 9-31 |
| 9.9 | Upgrade Tasks Associated with Oracle Business Rules..... | 9-32 |
| 9.10 | Upgrading Oracle SOA Suite Client Applications..... | 9-32 |
| 9.11 | Upgrading Oracle Web Services Manager (WSM) Policies..... | 9-36 |
| 9.11.1 | Before You Upgrade..... | 9-36 |
| 9.11.1.1 | A Note About Oracle WSM 10g Gateways..... | 9-36 |
| 9.11.1.2 | A Note About Third-party Software..... | 9-37 |
| 9.11.2 | Upgrading Oracle WSM 10g Predefined Policies..... | 9-37 |
| 9.11.3 | Upgrading Oracle WSM Custom Policies..... | 9-40 |
| 9.12 | Upgrading Oracle Containers for J2EE (OC4J) Security Environments..... | 9-40 |
| 9.12.1 | Before You Upgrade..... | 9-41 |
| 9.12.2 | Anonymous Authentication with Message Protection (WS-Security 1.0)..... | 9-41 |
| 9.12.2.1 | OC4J 10g..... | 9-41 |
| 9.12.2.2 | Oracle WSM 11g..... | 9-42 |
| 9.12.3 | Anonymous Authentication with Message Integrity (WS-Security 1.0)..... | 9-42 |
| 9.12.3.1 | OC4J 10g..... | 9-42 |
| 9.12.3.2 | Oracle WSM 11g..... | 9-43 |
| 9.12.4 | Anonymous Authentication with Message Confidentiality (WS-Security 1.0)..... | 9-43 |
| 9.12.4.1 | OC4J 10g..... | 9-44 |
| 9.12.4.2 | Oracle WSM 11g..... | 9-44 |

| | | |
|-----------|---|------|
| 9.12.5 | Username Token with Message Protection (WS-Security 1.0) | 9-44 |
| 9.12.5.1 | OC4J 10g..... | 9-44 |
| 9.12.5.2 | Oracle WSM 11g..... | 9-46 |
| 9.12.6 | ID Propagation Using SAML Token (Sender Vouches) with Message Protection (WS-Security 1.0)..... | 9-46 |
| 9.12.6.1 | OC4J 10g..... | 9-46 |
| 9.12.6.2 | Oracle WSM 11g..... | 9-47 |
| 9.12.7 | ID Propagation Using SAML Token (Holder of Key) with Message Protection (WS-Security 1.0)..... | 9-47 |
| 9.12.7.1 | OC4J 10g..... | 9-48 |
| 9.12.7.2 | Oracle WSM 11g..... | 9-49 |
| 9.12.8 | Mutual Authentication with Message Protection (WS-Security 1.0) | 9-49 |
| 9.12.8.1 | OC4J 10g..... | 9-49 |
| 9.12.8.2 | Oracle WSM 11g..... | 9-50 |
| 9.12.9 | Username token over SSL..... | 9-50 |
| 9.12.9.1 | OC4J 10g..... | 9-51 |
| 9.12.9.2 | Oracle WSM 11g..... | 9-51 |
| 9.12.10 | ID Propagation with SAML Token (Sender Vouches) over SSL (WS-Security 1.0) | 9-51 |
| 9.12.10.1 | OC4J 10g..... | 9-51 |
| 9.12.10.2 | Oracle WSM 11g..... | 9-52 |
| 9.12.11 | Log Information | 9-52 |
| 9.12.11.1 | OC4J 10g..... | 9-52 |
| 9.12.11.2 | Oracle WSM 11g..... | 9-52 |

10 Considerations When Upgrading Oracle WebCenter Applications

| | | |
|------------|--|-------|
| 10.1 | About Migration of WebCenter Applications | 10-1 |
| 10.1.1 | About WebCenter Application Templates..... | 10-1 |
| 10.1.2 | About Migration Tasks | 10-2 |
| 10.2 | Prerequisite Tasks for Migrating a WebCenter Application | 10-2 |
| 10.3 | How To Migrate WebCenter Applications | 10-4 |
| 10.3.1 | Preparing Your Applications for Migration | 10-5 |
| 10.3.1.1 | Backing Up Your WebCenter Applications..... | 10-5 |
| 10.3.1.2 | Preparing Applications That Rely on Oracle Content DB..... | 10-5 |
| 10.3.1.2.1 | Upgrading Your Application in JDeveloper 10.1.3.4 | 10-6 |
| 10.3.1.2.2 | Migrating Oracle Content DB Data | 10-6 |
| 10.3.1.3 | Preparing Applications That Contain Portlets..... | 10-7 |
| 10.3.2 | Migrating Your WebCenter Application | 10-7 |
| 10.3.3 | Performing Post Migration Tasks..... | 10-11 |
| 10.3.3.1 | Migrating Oracle Portal Connections | 10-11 |
| 10.3.3.2 | Configuring ADF Security | 10-11 |
| 10.3.3.3 | Downloading the Content Integration Suite API | 10-12 |
| 10.3.3.4 | Migrating Producer Registrations of Preconfigured Portlet Producers | 10-13 |
| 10.3.3.5 | Redeploying Your Applications..... | 10-13 |
| 10.4 | How to Migrate Portlet Producers | 10-13 |
| 10.4.1 | Determining WebCenter Consumer Application and Portlet Producer Compatibility..... | 10-14 |
| 10.4.2 | Migrating Your Portlet Producer Applications..... | 10-15 |

| | | |
|------------|---|-------|
| 10.4.2.1 | Migrating Portlet Producers Created in JDeveloper | 10-15 |
| 10.4.2.2 | Migrating Portlet Producers Created Outside of JDeveloper | 10-16 |
| 10.4.2.2.1 | Creating a JDeveloper Application from an EAR File | 10-16 |
| 10.4.2.2.2 | Creating a JDeveloper Application from a WAR File..... | 10-17 |
| 10.4.2.3 | Migrating Preconfigured Portlet Producers | 10-18 |
| 10.4.2.3.1 | Migrating Customizations from the Default Location | 10-18 |
| 10.4.2.3.2 | Migrating Customizations from a Nondefault Location..... | 10-18 |
| 10.4.3 | Performing Post Migration Tasks for Portlet Producers..... | 10-19 |
| 10.4.3.1 | Migrating Customizations | 10-19 |
| 10.4.3.2 | Redeploying a Portlet Producer Application | 10-19 |
| 10.5 | How to Migrate Wiki Data | 10-19 |
| 10.5.1 | Migrating Content to a File System-Based Repository | 10-20 |
| 10.5.2 | Migrating Content to a Database Repository | 10-20 |

A Example: Upgrading and Redeploying SOA Order Booking

| | | |
|---------|---|------|
| A.1 | About the SOA Order Booking Application..... | A-1 |
| A.2 | Task 1: Verify That the SOA Order Booking and Its External Partners are Up and Running..... | A-2 |
| A.2.1 | Task 1a: Verify That the SOA Order Booking Is Successfully Deployed on Oracle Application Server 10g Release 3 (10.1.3.4) | A-2 |
| A.2.2 | Task 1b: Verify That SOA Order Booking Is Currently Running on Oracle Application Server 10g Release 3 (10.1.3.4)..... | A-3 |
| A.2.3 | Task 1c: Verify That All External Partners Are Available | A-4 |
| A.3 | Task 2: Install and Upgrade the Oracle Fusion Middleware SOA Suite..... | A-6 |
| A.4 | Task 3: Use the Oracle WebLogic Suite Administration Console to Configure Required Resources | A-6 |
| A.4.1 | Task 3a: Create the SOADEMO Data Source..... | A-6 |
| A.4.2 | Task 3b: Configure the DbAdapter Application | A-7 |
| A.4.2.1 | Creating a New Outbound Connection Pool for the DbAdapter Application... | A-7 |
| A.4.2.2 | Configuring the New Connection Pool for the DbAdapter Application | A-8 |
| A.4.3 | Task 3c: Create the SOA Order Booking JMS Resources | A-8 |
| A.4.3.1 | Creating the JMS Queue for the SOA Order Booking Demo | A-9 |
| A.4.3.2 | Creating a JMS Connection Factory for the SOA Order Booking Demo..... | A-9 |
| A.4.3.3 | Creating an Outbound Connection Pool for the JmsAdapter Application | A-9 |
| A.4.3.4 | Configuring the JmsAdapter Connection Pool | A-10 |
| A.4.3.5 | Redeploying the Modified JmsAdapter Application | A-10 |
| A.5 | Task 4: Install Oracle JDeveloper 11g..... | A-11 |
| A.6 | Task 5: Set Up Connections in Oracle JDeveloper Release 11 | A-11 |
| A.6.1 | Task 5a: Start Oracle JDeveloper 11g | A-11 |
| A.6.2 | Task 5b: Verify That You Have the Required Oracle JDeveloper Extensions | A-11 |
| A.6.3 | Task 5c: Create a Connection to the SOADEMO Schema..... | A-12 |
| A.6.4 | Task 5d: Create a Connection to Oracle WebLogic Server | A-13 |
| A.7 | Task 6: Open the SOA Order Booking Application in Oracle JDeveloper 11g and Migrate the Application Projects | A-14 |
| A.7.1 | Opening and Migrating the Application..... | A-14 |
| A.7.2 | Viewing the Migration Log Files..... | A-16 |

| | | |
|---------|---|------|
| A.8 | Task 7: Compile and Deploy Each of the Application Projects on Oracle Fusion Middleware 11g | A-17 |
| A.8.1 | About the CreditService, RapidService, and CustomerService Projects | A-17 |
| A.8.2 | SelectManufacturer..... | A-17 |
| A.8.3 | DHLShipment | A-18 |
| A.8.4 | FulfillmentESB..... | A-18 |
| A.8.4.1 | Modifying the FulfillmentESB Project in Oracle JDeveloper 11g | A-18 |
| A.8.4.2 | Compile and Deploy the FulfillmentESB Project..... | A-20 |
| A.8.5 | CreditService | A-20 |
| A.8.5.1 | Upgrading the CreditService Web Services..... | A-20 |
| A.8.5.2 | Compiling and Deploying the CreditService Project | A-21 |
| A.8.6 | RapidService..... | A-21 |
| A.8.6.1 | Upgrade the RapidService Web Service from OC4J JAX-RPC to Oracle WebLogic Server JAX-WS | A-22 |
| A.8.6.2 | Compiling and Deploying the RapidService Project..... | A-24 |
| A.8.7 | CustomerService | A-24 |
| A.8.7.1 | Modifying Specific CustomerService Project Class Files..... | A-24 |
| A.8.7.2 | Modifying the persistence.xml File in the CustomerService Project | A-27 |
| A.8.7.3 | Compiling and Deploying the CustomerService Project..... | A-27 |
| A.8.8 | SOAOrderBooking | A-27 |
| A.8.8.1 | Modifying the SOAOrderBooking Project..... | A-27 |
| A.8.8.2 | Optionally, Changing SOAOrderBooking to Use Upgraded Web Services | A-28 |
| A.8.8.3 | Compiling and Deploying the SOAOrderBooking Project: | A-29 |
| A.8.9 | OrderBookingESB..... | A-29 |
| A.8.9.1 | Modifying the OrderBookingESB Project | A-30 |
| A.8.9.2 | Compiling and Deploying the OrderBookingESB Project:..... | A-30 |
| A.8.10 | ApproveOrderTaskForm..... | A-30 |
| A.9 | Task 8: Test the SOA Order Booking Application in Fusion Middleware Control..... | A-30 |

B Additional Oracle WebCenter Migration Details

| | | |
|---------|---|------|
| B.1 | Automated Changes in Migrated WebCenter Applications | B-1 |
| B.1.1 | Customizable Component Changes..... | B-1 |
| B.1.1.1 | General Changes..... | B-2 |
| B.1.1.2 | Panel Customizable Changes | B-2 |
| B.1.1.3 | Show Detail Frame | B-4 |
| B.1.2 | External Application Changes | B-5 |
| B.1.2.1 | External Applications Changes in Oracle Application Server 10g and Oracle Fusion Middleware 11g | B-5 |
| B.1.2.2 | External Application Migration Changes | B-5 |
| B.1.3 | Portlet Component Changes | B-6 |
| B.1.3.1 | Portlet Tag Changes | B-6 |
| B.1.3.2 | Attribute Changes | B-6 |
| B.2 | Content Integration Connections: Migration of Credentials to External Applications... | B-7 |
| B.3 | Preconfigured Portlet Producer: Customization Store Location | B-10 |
| B.3.1 | WSRP Portlet Producers | B-10 |
| B.3.2 | Web Clipping Portlet Producer | B-10 |
| B.3.3 | OmniPortlet and other PDK-Java Portlet Producers..... | B-10 |

C Upgrading Oracle Business Rules Dictionaries and Projects

| | | |
|---------|---|------|
| C.1 | Using Oracle JDeveloper to Migrate an Oracle Business Rules Dictionary | C-1 |
| C.2 | Using Rule Migrator Tool to Migrate an Oracle Business Rules Dictionary..... | C-2 |
| C.3 | Using MigrateRuleRepository with Oracle Business Rules SDK to Migrate a Dictionary..... | C-4 |
| C.3.1 | How to Migrate a Dictionary with Oracle Business Rules SDK..... | C-4 |
| C.3.2 | What You May Need to Know About the MigrateRuleRepository API | C-11 |
| C.3.2.1 | Input Properties | C-12 |
| C.3.2.2 | Output Properties..... | C-12 |
| C.4 | Oracle Business Rules Manual Migration Tasks | C-13 |
| C.4.1 | How to Migrate JAXB 1.0 to JAXB 2.0 | C-13 |
| C.4.2 | How to Migrate RL Functions | C-14 |
| C.4.3 | What You May Need to Know About Manual Migration..... | C-15 |

Index

Preface

This preface contains the following sections:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This guide is intended for Oracle Fusion Middleware system administrators who are responsible for installing, maintaining, and upgrading Oracle Fusion Middleware instances. It is assumed that readers of this manual have knowledge of the following:

- Oracle Fusion Middleware system administration and configuration
- The configuration and expected behavior of the system being upgraded

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at

<http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Related Documents

For more information, see the following related documentation available in the Oracle Fusion Middleware 11g documentation library:

- *Oracle Fusion Middleware Upgrade Planning Guide*
- *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Administrator's Guide*

Note: For more detailed information about the Oracle Fusion Middleware upgrade documentation, see "Documentation Roadmap for Upgrading Your Middleware Environment" in the *Oracle Fusion Middleware Upgrade Planning 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. |
| <i>italic</i> | 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. |

Summary of the Oracle SOA Suite, WebCenter, and ADF Upgrade Process

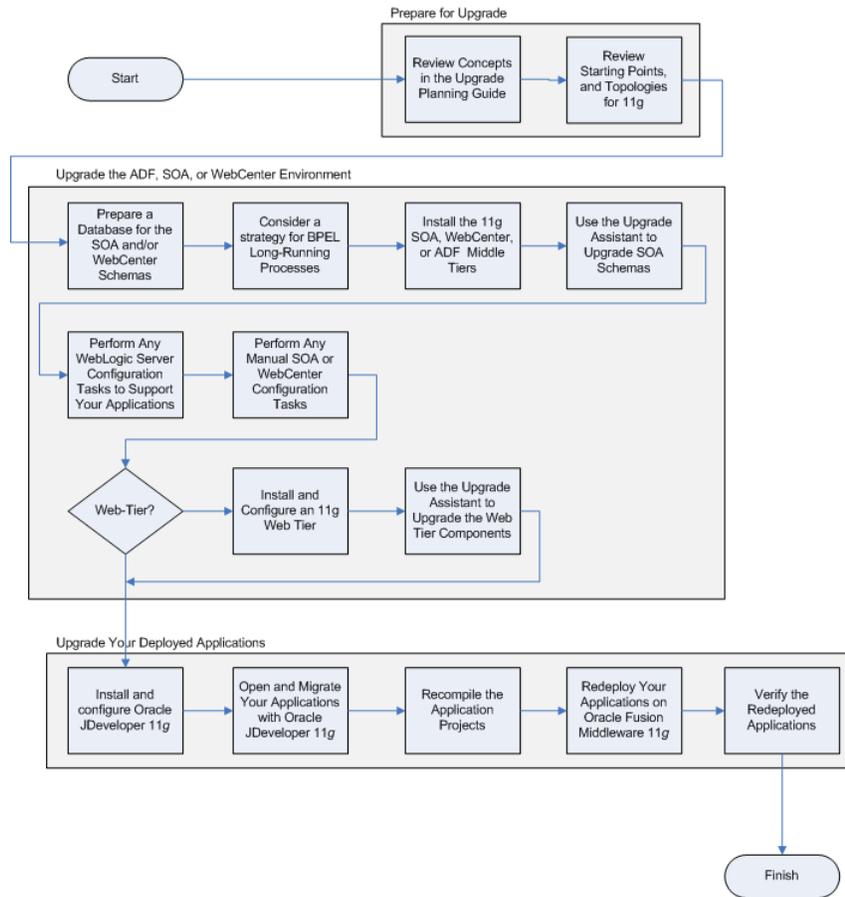
This chapter provides a high-level overview of the Oracle SOA Suite, WebCenter, and ADF upgrade process. Refer to the following sections for more information:

- [Flow Chart of the Oracle SOA Suite, WebCenter, and ADF Upgrade Process](#)
- [Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process](#)

1.1 Flow Chart of the Oracle SOA Suite, WebCenter, and ADF Upgrade Process

[Figure 1–1](#) provides a flow chart of the Oracle SOA, WebCenter, and ADF upgrade process. Review this chart to get familiar with the tasks you will be required to perform, based on your existing version of Oracle Application Server.

Figure 1–1 Flow Chart of the Oracle SOA Suite and WebCenter Upgrade Process



1.2 Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process

Table 1–1 describes each of the steps in the upgrade process flow chart, which is shown in Figure 1–1. The table also provides information on where to get more information on each step in the process.

Table 1–1 Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process

| Step | Description | More Information |
|--|--|--|
| Review Concepts in the Oracle Fusion Middleware Upgrade Planning Guide | <p>The <i>Oracle Fusion Middleware Upgrade Planning Guide</i> provides a high-level overview of how to upgrade your entire existing Oracle Application Server 10g environment to Oracle Fusion Middleware.</p> <p>It also includes compatibility information and instructions for upgrading any databases that support your middleware components.</p> | <i>Oracle Fusion Middleware Upgrade Planning Guide</i> |
| Review the Starting Points and Topologies for 11g | <p>Before starting your upgrade, you should be familiar with the new features, supported started points, and recommended topologies for Oracle Fusion Middleware 11g.</p> <p>Oracle Fusion Middleware 11g introduces a new architecture, based on Oracle WebLogic Server. You should get familiar with the new 11g features before starting your upgrade.</p> | <p>Part I, "Preparing to Upgrade Oracle SOA Suite, WebCenter, and ADF"</p> <p>Chapter 3, "Oracle SOA Suite, WebCenter, and ADF Information for 10g Users"</p> |
| Prepare a Database for the SOA and WebCenter Schemas | <p>The Oracle SOA Suite and WebCenter components require schemas that must be installed in a database.</p> <p>As a result, before you begin the upgrade of a SOA or WebCenter environment, make sure your database is a supported version, and then use the Oracle Fusion Middleware Metadata Repository Creation Utility to install the new, 11g schemas.</p> | <p>SOA Environments:</p> <ul style="list-style-type: none"> ▪ Section 6.3, "Task 3: If Necessary, Upgrade the Oracle Database That Hosts the Oracle SOA Suite Schemas" ▪ Section 6.4, "Task 4: Use the Repository Creation Utility to Install the 11g SOA Schemas" <p>WebCenter Environments:</p> <ul style="list-style-type: none"> ▪ Section 7.2, "Task 2: If Necessary, Upgrade the Oracle Database That Hosts the Oracle WebCenter Schemas" ▪ Section 7.3, "Task 3: Use the Repository Creation Utility to Install the 11g Oracle WebCenter Schemas" |
| Consider a strategy for BPEL Long-Running Processes | <p>If you are supporting any Oracle BPEL Process Manager long-running processes, you should consider the some different options when upgrading your Oracle SOA Suite environment.</p> | Chapter 4, "Special Considerations for Oracle BPEL Long-Running Processes" |

Table 1–1 (Cont.) Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process

| Step | Description | More Information |
|--|--|--|
| Install the 11g SOA, WebCenter, or ADF Middle Tiers | Use the appropriate software installer and configuration wizard to install and configure an Oracle Fusion Middleware farm to support your Oracle SOA Suite, WebCenter, and ADF applications. | SOA Environments: <ul style="list-style-type: none"> Section 6.5, "Task 5: Install and Configure the Oracle SOA Middle Tiers" WebCenter environments: <ul style="list-style-type: none"> Section 7.4, "Task 4: Install and Configure the Oracle WebCenter Middle Tiers" ADF environments: <ul style="list-style-type: none"> Section 5.2, "Task 2: Install and Configure a Oracle WebLogic Server Domain For Your ADF Applications" |
| Use the Upgrade Assistant to Upgrade the SOA Schemas | If you are using Oracle Business Activity Monitoring or Oracle B2B, use the Upgrade Assistant to upgrade the BAM and B2B schemas. | Section 6.6, "Task 6: If Necessary, Upgrade the SOA Component Schemas" |
| Perform Any WebLogic Server Configuration Tasks to Support Your Applications | Manually modify any Oracle WebLogic Server properties or configuration settings; create data sources or other Java EE resources required by your applications. | <i>Oracle Fusion Middleware Upgrade Guide for Java EE</i> |
| Perform Any Manual SOA or WebCenter Configuration Tasks | Often, there are configuration tasks that you need to perform on the new Oracle WebLogic Server environment to support your Oracle SOA or WebCenter applications. In most cases, no manual configuration tasks are required to support ADF applications, as long as you have installed and configured your ADF environment as documented. | Chapter 9, "Considerations When Upgrading Oracle SOA Applications" Chapter 10, "Considerations When Upgrading Oracle WebCenter Applications" |
| Install and Configure an 11g Web Tier | Use the appropriate software installer and configuration wizard to install and configure the Oracle HTTP Server and Oracle Web Cache Web Tier components. | "Task 1: Understand the Differences Between Using a Web Tier with OC4J and Oracle WebLogic Server" in the <i>Oracle Fusion Middleware Upgrade Guide for Java EE</i> |
| Use the Upgrade Assistant to Upgrade the Web Tier Components | The Oracle Fusion Middleware Upgrade Assistant copies configuration data from your existing Oracle Application Server Web tier components to the newly installed Oracle Fusion Middleware Web tier components. | "Upgrading a Web Server and Java EE Environment" in the <i>Oracle Fusion Middleware Upgrade Guide for Java EE</i> |
| Install and Configure Oracle JDeveloper 11g | You can download Oracle JDeveloper 11g Release 1 (11.1.1) from the Oracle Technology Network (OTN), or copy the ZIP file from the Oracle Fusion Middleware Companion CD-ROM. | Section 8.4, "Task 4: Install and Start Oracle JDeveloper 11g" |

Table 1–1 (Cont.) Table Describing the Steps in the Oracle SOA Suite and WebCenter Upgrade Process

| Step | Description | More Information |
|---|---|--|
| Open and Migrate Your Applications with Oracle JDeveloper Release 11g | Oracle JDeveloper 11 offers a migration feature that helps you convert your Oracle Application Server 10g applications so they can be recompiled for Oracle Fusion Middleware 11g Release 1 (11.1.1). | Section 8.5, "Task 5: Open and Upgrade the Application in Oracle JDeveloper" |
| Recompile the Application Projects | The application sources should be recompiled with JDeveloper 11g; check for any compilation errors. | Section 8.7, "Task 7: Review and Recompile the Application" |
| Implement any other code changes required for the application to run successfully on Oracle Fusion Middleware 11g | Identify any particular technologies or services required by your application and review the considerations | |
| Redeploy Your Applications on Oracle Fusion Middleware 11g | You can use JDeveloper to deploy your applications, or you can create and EAR file and deploy the application using the <code>asctl</code> command line or using Oracle Enterprise Manager Fusion Middleware Control. | Section 8.8, "Task 8: Redeploy the Application" |
| Verify the Redeployed Applications | Make sure that the redeployed applications are working properly and no errors occurred during the redeployment. | N/A |

Part I

Preparing to Upgrade Oracle SOA Suite, WebCenter, and ADF

Part I contains the following chapters:

- [Chapter 2, "Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF Upgrade"](#)
- [Chapter 3, "Oracle SOA Suite, WebCenter, and ADF Information for 10g Users"](#)

Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF Upgrade

Refer to the following sections for information about the supported starting points for upgrading to Oracle SOA Suite, WebCenter, and ADF 11g:

- [Summary of Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF](#)
- [Information for Oracle SOA Suite 10g Release 2 \(10.1.2\) Users](#)

2.1 Summary of Supported Starting Points for Oracle SOA Suite, WebCenter, and ADF

This guide provides instructions for upgrading from the Oracle Application Server releases and installation types described in [Table 2-1](#).

Table 2-1 Releases and Installation Types Supported By This Guide

| Release | Installation Type | Description or Notes | Latest Patch Sets |
|----------------------------|------------------------|--|--------------------------|
| 10g Release 3 (10.1.3.1.0) | All Installation Types | This release included the components of the Oracle SOA Suite | 10g Release 3 (10.1.3.4) |
| 10g Release 3 (10.1.3.2.0) | All Installation Types | This release included the Oracle WebCenter components | 10g Release 3 (10.1.3.4) |

Note: The patch sets listed in [Table 2-1](#) were the latest patch sets available at the time this guide was published.

For a list of the latest patch sets available for your installation, refer to My Oracle Support (formerly *OracleMetaLink*):

<http://metalink.oracle.com/>

2.2 Information for Oracle SOA Suite 10g Release 2 (10.1.2) Users

If you are using Oracle BPEL Process Manager or Oracle BAM 10g Release 2 (10.1.2), then you must first migrate your projects to 10g Release 3 (10.1.3) before upgrading to Oracle Fusion Middleware 11g. For more information on migrating from Oracle BPEL Process Manager 10g Release 2 (10.1.2) to 10g Release 3 (10.1.3), see:

- "Oracle BPEL Process Manager Migration" in the 10g Release 3 (10.1.3.1) *Oracle BPEL Process Manager Installation Guide*

- "Migrating Workflow Definitions from 10.1.2 to 10.1.3.1" in the 10g Release 3 (10.1.3.1) *Oracle BPEL Process Manager Developer's Guide*

Oracle SOA Suite, WebCenter, and ADF Information for 10g Users

This chapter provides an overview of the Oracle SOA Suite, WebCenter, and ADF software available in Oracle Fusion Middleware 11g. The following sections compare the architecture and directory structure of these technologies in 10g Release 3 (10.1.3) and in Oracle Fusion Middleware 11g:

- [Information for Oracle ADF Users](#)
- [Information for Oracle SOA Suite Users](#)
- [Information for Oracle WebCenter Users](#)

3.1 Information for Oracle ADF Users

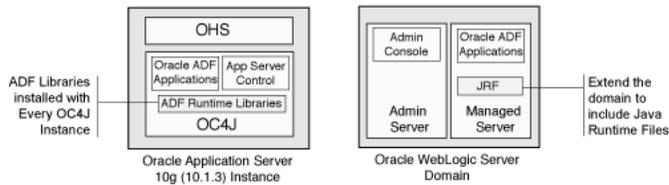
In Oracle Application Server 10g Release 3 (10.1.3), Oracle provided the Oracle Application Development Framework (ADF), an end-to-end application framework that builds on Java EE standards and open-source technologies to simplify and accelerate implementing service-oriented applications.

The development and runtime environment required to deploy and manage ADF applications is similar in many ways to the environment required for other Java EE applications. For information about the types of Java EE environments available in 10g and instructions for upgrading those environments to Oracle Fusion Middleware 11g, refer to the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

The difference between a typical Java EE environment and an environment that supports ADF applications is the availability of the ADF runtime libraries:

- In Oracle Application Server 10g, each instance of OC4J automatically provided the ADF runtime libraries required to support ADF applications.
- In Oracle Fusion Middleware 11g, an Oracle WebLogic Server domain, by default, does not contain the ADF runtime libraries. However, you can optionally configure or extend your domain to include the Java Runtime Files (JRF). The Oracle ADF runtime libraries are included as part of the JRF component.

[Figure 3–1](#) compares how Oracle ADF applications are supported in OC4J and in Oracle WebLogic Server. Note that the Oracle WebLogic Server domain can be extended with the Java Runtime Files (JRF) domain template, which includes the required Oracle ADF libraries, as well as other important Oracle-specific technologies.

Figure 3–1 Support for Oracle ADF Applications in OC4J and Oracle WebLogic Server

For more information about extending your domain to support Oracle ADF, see Chapter 17, "Scaling Your Environment," in the *Oracle Fusion Middleware Administrator's Guide*.

Note that unlike Oracle Application Server 10g, Oracle HTTP Server is not automatically installed and configured to route requests to Oracle WebLogic Server. Instead, you install Oracle HTTP Server and Oracle Web Cache in a separate Oracle home using the WebTier and Utilities CD-ROM. For more information, see "Upgrading a Web Server and Java EE Environment" in the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

3.2 Information for Oracle SOA Suite Users

Oracle Application Server 10g Release 3 (10.1.3) provided installation options that allowed you to install and configure the Oracle Service Oriented Architecture (SOA) system components (except for Oracle BAM) as part of the Oracle Application Server installation. You could then use Oracle JDeveloper to develop and deploy SOA applications on the OC4J instances where the SOA components were installed.

Similarly, for Oracle Fusion Middleware, you can install and configure an Oracle WebLogic Server domain that supports the Oracle SOA infrastructure and system components, including Oracle BAM.

For more information, refer to the following sections:

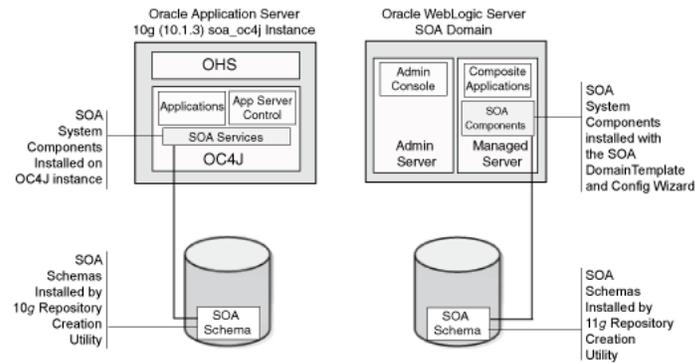
- [Comparing Oracle SOA Suite on OC4J and Oracle WebLogic Server](#)
- [Comparing the Oracle SOA Suite Directory Structure on OC4J and Oracle WebLogic Server](#)
- [Upgrading an Oracle SOA Suite Enterprise Deployment](#)
- [About Oracle BAM and Oracle B2B in Oracle Application Server 10g and Oracle Fusion Middleware 11g](#)

3.2.1 Comparing Oracle SOA Suite on OC4J and Oracle WebLogic Server

[Figure 3–2](#) shows how the SOA system components are installed as part of an OC4J 10g instance and compares that to how the SOA components are installed on a managed server when you install and configure the Oracle Fusion Middleware Oracle SOA Suite 11g.

Note that the SOA system components require a set of SOA schemas, which you install using the Repository Creation Utility (RCU). For more information about the schemas installed by RCU, refer to the *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

Figure 3–2 Oracle SOA Suite Installed on an OC4J Instance and an Oracle WebLogic Server Domain



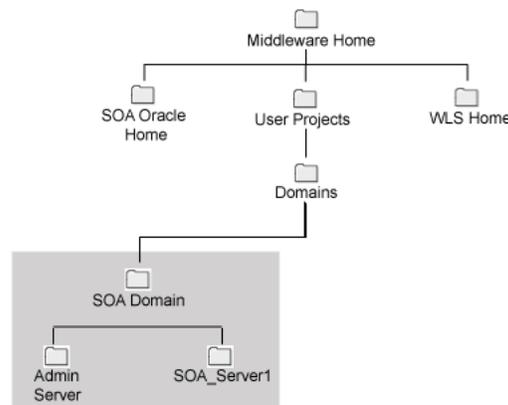
3.2.2 Comparing the Oracle SOA Suite Directory Structure on OC4J and Oracle WebLogic Server

When you install and configure the Oracle SOA Suite, you create a Oracle WebLogic Server domain that has been extended to support the SOA system components. The resulting files on disk consist of the SOA domain directory structure, as well as the SOA Oracle home, which contains the SOA component binary and support files. The SOA Oracle home must be installed inside the Middleware home.

This is in contrast to the single Oracle home that was created during an Oracle Application Server 10g Release 3 (10.1.3) installation.

Figure 3–3 shows the directory structure that results from installing and configuring the Oracle SOA Suite domain.

Figure 3–3 The Directory Structure of an Oracle Fusion Middleware SOA Suite Installation



3.2.3 Upgrading an Oracle SOA Suite Enterprise Deployment

The instructions in this guide provide the steps for upgrading a basic Oracle SOA Suite environment. After reviewing the content of this guide, you should be able to transfer this knowledge to a more complex production environment.

For specific information about setting up a production environment that takes advantage of clustering and high availability, refer to the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.

For an overview of the high-level tasks you must perform when upgrading an enterprise deployment, see "Planning an Enterprise Deployment" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

3.2.4 About Oracle BAM and Oracle B2B in Oracle Application Server 10g and Oracle Fusion Middleware 11g

Unlike the other Oracle SOA Suite components, Oracle BAM 10g and Oracle B2B were installed and configured separately.

Oracle BAM was supported on the Microsoft platform only, and Oracle B2B was installed on a separate server from the other Oracle SOA Suite 10g components.

For Oracle Fusion Middleware 11g, Oracle BAM 11g and Oracle B2B 11g are integrated with the other Oracle SOA Suite components. Oracle BAM can be deployed as part of the same Oracle WebLogic Server domain, and Oracle B2B is installed and configured automatically with the other Oracle SOA Suite components.

3.3 Information for Oracle WebCenter Users

Oracle Application Server 10g Release 3 (10.1.3) provided installation options that allowed you to install and configure the WebCenter system components as part of an Oracle Application Server install. You could then use Oracle JDeveloper to develop and deploy WebCenter applications on the OC4J instances where the WebCenter system components were installed.

Similarly, for Oracle Fusion Middleware, you can install and configure a Oracle WebLogic Server domain that supports the WebCenter system components.

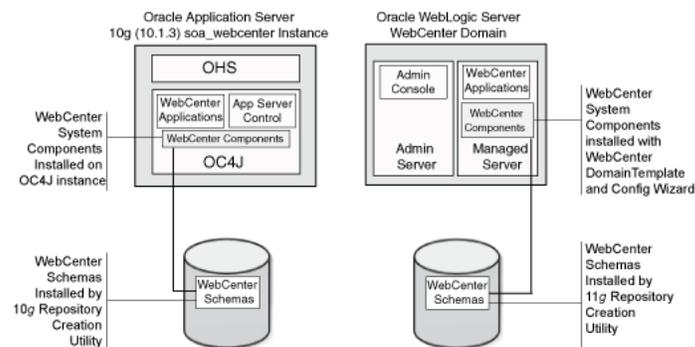
Refer to the following sections for more information:

- [Comparing Oracle WebCenter on OC4J and Oracle WebLogic Server](#)
- [Comparing the Oracle WebCenter Directory Structure on OC4J and Oracle WebLogic Server](#)

3.3.1 Comparing Oracle WebCenter on OC4J and Oracle WebLogic Server

[Figure 3–4](#) shows how the Oracle WebCenter system components are installed as part of an OC4J 10g instance and compares that to how the Oracle WebCenter components are installed on a managed server when you install and configure the Oracle Fusion Middleware 11g.

Note that the WebCenter system components require a set of WebCenter schemas, which you install using the Repository Creation Utility (RCU). For more information about the schemas installed by RCU, refer to the *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

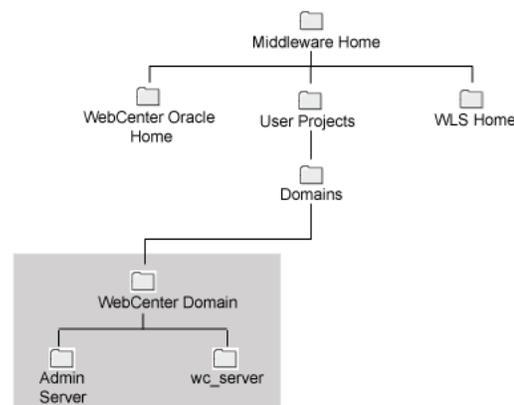
Figure 3–4 Oracle WebCenter Installed on OC4J and on Oracle WebLogic Server

3.3.2 Comparing the Oracle WebCenter Directory Structure on OC4J and Oracle WebLogic Server

When you install and configure Oracle WebCenter, you create an Oracle WebLogic Server domain that has been extended to support the WebCenter system components. The resulting files on disk consist of the WebCenter domain directory structure, as well as the WebCenter Oracle home, which contains the WebCenter binary and support files. The WebCenter Oracle home, like the SOA Oracle home, must reside inside the Middleware Oracle home.

This is contrast to the single Oracle home that was created during an Oracle Application Server 10g Release 3 (10.1.3) installation.

Figure 3–5 shows the directory structure that results from installing and configuring the Oracle SOA Suite domain.

Figure 3–5 The Directory Structure of an Oracle Fusion Middleware WebCenter Installation

Special Considerations for Oracle BPEL Long-Running Processes

Before you begin the upgrade of your Oracle SOA Suite 11g environment, use the information in this chapter to determine whether or not you need to consider the impact of the upgrade on Oracle BPEL Process Manager long-running processes:

- [What Is an Oracle BPEL Process Manager Long-Running Process?](#)
- [Two Approaches to Upgrading Long-Running Processes](#)

4.1 What Is an Oracle BPEL Process Manager Long-Running Process?

Like any other BPEL process, a long-running process involves accessing a service and returning a value or result to Oracle BPEL Process Manager. However, the difference is that a long-running process may not return with a value for a relatively long period of time.

For example, suppose you had a process that is designed to invoke an external task to process a customer order. The process invokes an asynchronous backend application to start processing the order. The backend application interacts with a human for approval of the order. After the order is approved, the backend application performs a callback to the process which then performs a callback to the client application. This type of complex process could take several days to complete and is an example of a long-running process.

Long-running processes can affect your upgrade planning because you must consider the timing of shutting down your existing Oracle BPEL Process Manager 10g environment and starting up your Oracle BPEL Process Manager 11g environment. Ideally, you want all your long-running processes to finish successfully before you shut down and decommission your existing Oracle BPEL Process Manager 10g environment and start using your 11g environment.

4.2 Two Approaches to Upgrading Long-Running Processes

Oracle SOA Suite supports a wide variety of organizations and companies. As a result, it is difficult to determine the best upgrade strategy for all environments. However, in the area of long-running processes, there are two possible solutions you can consider:

Note, that both the solutions discussed here may not be suitable to all customers. If you are unclear about the best possible strategy for your particular organization, contact Oracle Support for more information about other possible alternatives to these possible solutions:

- [Using a Side-by-Side Approach to Long-Running Processes During Upgrade](#)

- [Using a Grace Period and a Hard Stop of Long-Running Processes](#)

4.2.1 Using a Side-by-Side Approach to Long-Running Processes During Upgrade

The side-by-side approach to long-running processes involves upgrading to an Oracle Fusion Middleware 11g environment while keeping the Oracle Application Server 10g environment up and running until the long-running processes finish up their tasks.

[Table 4–1](#) shows a summary of the upgrade process you would use if you were considering the side-by-side approach to supporting Oracle BPEL Process Manager long running processes during upgrade.

Table 4–1 Steps for Using the Side-by-Side Approach to Long-Running Processes

| Step Number | Task | More Information |
|-------------|---|---|
| 1 | Install and configure a new Oracle Fusion Middleware 11g environment. | Chapter 6, "Upgrading an Oracle SOA Suite Environment" |
| 2 | Upgrade the Application in Oracle JDeveloper 11g. | Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications" |
| 3 | Deploy the application to Oracle Fusion Middleware 11g. | Section 8.8, "Task 8: Redeploy the Application" |
| 4 | Make a copy of any client applications, upgrade them, and test them with the application you deployed on Oracle Fusion Middleware 11g. | Section 9.10, "Upgrading Oracle SOA Suite Client Applications" |
| 5 | Retire all processes on Oracle BPEL Process Manager 10g; this will stop any new processes from starting, but allow existing 10g processes to finish. | "How to Manage a Process Life Cycle" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle SOA Suite Developer's Guide</i> , available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation |
| 6 | With the 10g system still running, decommission the 10g client applications, and start the upgraded application clients, which are directing requests to the new 11g Oracle BPEL Process Manager. | "Starting and Stopping Oracle Fusion Middleware" in the <i>Oracle Fusion Middleware Administrator's Guide</i> |
| 7 | Monitor the 10g long-running processes until they all complete. | "Using the Instances Tab to Monitor Process Instances" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle SOA Suite Developer's Guide</i> , available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation |
| 8 | Shut down and decommission the 10g Oracle BPEL Process Manager 10g. | <ul style="list-style-type: none"> ■ "Starting and Stopping Oracle BPEL Process Manager Components" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle BPEL Process Manager Quick Start Guide</i> ■ "Starting and Stopping Application Server Instances" in the in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle Application Server Administrator's Guide</i>. <p>The Oracle Application Server 10g Release 3 (10.1.3) documentation is available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation</p> |

4.2.2 Using a Grace Period and a Hard Stop of Long-Running Processes

As an alternative to the side-by-side approach to upgrading Oracle BPEL Process Manager long-running processes, you might consider the approach described in this section if you do not have many long-running processes, your long-running processes are less critical to your business model, or if you have more flexibility in terms of the timing or duration of the upgrade.

Table 4–2 provides a summary of the upgrade process you would use if you were considering an alternative approach to the side-by-side approach to long-running processes.

Table 4–2 Steps for Using the Grace Period and a Hard Stop Approach to Long-Running Processes

| Step Number | Task | More Information |
|-------------|--|---|
| 1 | Install and configure a new Oracle Fusion Middleware 11g environment. | Chapter 6, "Upgrading an Oracle SOA Suite Environment" |
| 2 | Upgrade the Application in Oracle JDeveloper 11g. | Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications" |
| 3 | Deploy the application to Oracle Fusion Middleware 11g. | Section 8.8, "Task 8: Redeploy the Application" |
| 4 | Make a copy of any client applications, upgrade them, and test them with the application you deployed on Oracle Fusion Middleware 11g. | Section 9.10, "Upgrading Oracle SOA Suite Client Applications" |
| 5 | Retire all processes on Oracle BPEL Process Manager 10g; this will stop any new processes from starting, but allow existing 10g processes to finish. | "How to Manage a Process Life Cycle" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle SOA Suite Developer's Guide</i> , available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation |
| 6 | Define a grace period and set a deadline for stopping the Oracle Application Server 10g environment and cutover to the Oracle Fusion Middleware 11g environment. | Use the Oracle Application Server 10g Oracle BPEL Control to monitor your long-running processes; use this information to define a reasonable time for the upgrade to Oracle Fusion Middleware 11g. |

Table 4–2 (Cont.) Steps for Using the Grace Period and a Hard Stop Approach to Long-Running Processes

| Step Number | Task | More Information |
|-------------|--|--|
| 7 | At the predefined deadline, abort all existing Oracle BPEL Process Manager process instances. | <p>You can abort a process in one of two ways:</p> <ul style="list-style-type: none"> ■ By deleting the instance from the BPEL Control. For more information, see "Using the Instances Tab to Monitor Process Instances" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle SOA Suite Developer's Guide</i>, available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation ■ By creating a Java program that locates and aborts all active instances. For this you can use the <code>abort()</code> method in the method in the interface <code>com.oracle.bpel.client.IInstanceHandle</code>. For more information, see the 10g Release 3 (10.1.3) <i>Oracle BPEL Process Manager Client Java API Reference</i>. |
| 8 | Shut down and decommission the Oracle Application Server 10g environment. | <ul style="list-style-type: none"> ■ "Starting and Stopping Oracle BPEL Process Manager Components" in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle BPEL Process Manager Quick Start Guide</i> ■ "Starting and Stopping Application Server Instances" in the in the Oracle Application Server 10g Release 3 (10.1.3.1.0) <i>Oracle Application Server Administrator's Guide</i>. <p>The Oracle Application Server 10g Release 3 (10.1.3) documentation is available on the Oracle Technology Network (OTN): http://www.oracle.com/technology/documentation</p> |
| 9 | Start and roll out the new upgraded client applications, which send requests to the new Oracle BPEL Process Manager 11g. | "Starting and Stopping Oracle Fusion Middleware" in the <i>Oracle Fusion Middleware Administrator's Guide</i> |

Part II

Upgrading the Oracle SOA Suite, WebCenter, and ADF Environment

Part II contains the following chapters:

- [Chapter 5, "Upgrading an Oracle ADF Environment"](#)
- [Chapter 6, "Upgrading an Oracle SOA Suite Environment"](#)
- [Chapter 7, "Upgrading an Oracle WebCenter Environment"](#)

Upgrading an Oracle ADF Environment

This chapter describes how to upgrade your Oracle Application Server 10g Oracle ADF environment to an equivalent environment in Oracle Fusion Middleware 11g.

Refer to the following sections for more information:

- [Task 1: Decide Upon an Oracle Fusion Middleware 11g Topology for Your ADF Applications](#)
- [Task 2: Install and Configure a Oracle WebLogic Server Domain For Your ADF Applications](#)

5.1 Task 1: Decide Upon an Oracle Fusion Middleware 11g Topology for Your ADF Applications

Before you begin the upgrade process, decide upon a topology for your new Oracle Fusion Middleware 11g environment.

For more information, see [Section 3.1, "Information for Oracle ADF Users"](#).

5.2 Task 2: Install and Configure a Oracle WebLogic Server Domain For Your ADF Applications

Before you upgrade to Oracle Fusion Middleware 11g, you must install an Oracle Fusion Middleware environment that is similar to the topology you set up for Oracle Application Server 10g.

Refer to the following sections for more information:

- [Installing and Configuring a Development Domain with Oracle JDeveloper](#)
- [Installing and Configuring a Domain without JDeveloper](#)
- [Using the Java Required Files \(JRF\) Domain Template](#)

5.2.1 Installing and Configuring a Development Domain with Oracle JDeveloper

Oracle WebLogic Server is available as part of Oracle JDeveloper Studio. As a result, if you are using Oracle JDeveloper Studio as your integrated development environment (IDE), you can install Oracle WebLogic Server as part of the Oracle JDeveloper installation.

For complete instructions for installing and configuring Oracle WebLogic Server with Oracle JDeveloper Studio, see "Installing the Oracle JDeveloper Studio Edition" in *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*.

5.2.2 Installing and Configuring a Domain without JDeveloper

If you are not using Oracle JDeveloper for developing your applications, then you can install the Oracle WebLogic Server software and the Oracle Fusion Middleware Application Developer software suite.

The Application Developer installer allows you to install the software you need to configure a domain with the Java Required Files (JRF) and Oracle Enterprise Manager Fusion Middleware Control domain templates. For information about the JRF template, see [Section 5.2.3, "Using the Java Required Files \(JRF\) Domain Template"](#).

For information about installing and configuring the Oracle Fusion Middleware Application Developer software, see the *Installation Guide for Application Developer*.

5.2.3 Using the Java Required Files (JRF) Domain Template

When you configure Oracle WebLogic Server, you configure each domain using domain templates. One of the domain templates available with Oracle Fusion Middleware 11g is the Java Required Files (JRF) template.

The JRF template provides important Oracle libraries and other capabilities that support new versions of APIs that many OC4J applications depend upon.

For information on the types of APIs in the JRF template that are important to upgraded OC4J applications, see "APIs Available With the Java Required Files (JRF) Domain Template" in the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

To create or extend a domain using the JRF template, refer to the following:

- [Creating a New Domain With the JRF Template](#)
- [Extending an Existing Domain With the JRF Template](#)
- [Using the Java Required Files \(JRF\) Domain Template](#)

5.2.3.1 Creating a New Domain With the JRF Template

There are multiple ways to create a new Oracle WebLogic Server domain using the JRF template:

- Install and configure a development domain using the Oracle JDeveloper 11g installer.

The resulting domain is automatically created using the JRF template.

- Install and configure an Application Developer, Oracle SOA Suite, WebCenter, Oracle Identity Management, or Oracle Portal, Forms, Reports, and Discoverer domain.

When you configure any Oracle Fusion Middleware software suite, you have the option of selecting the JRF template while running the configuration tool.

For more information, refer to the appropriate Oracle Fusion Middleware installation guide.

5.2.3.2 Extending an Existing Domain With the JRF Template

To extend an existing domain with the JRF template, use one of the following methods:

- Use the Oracle JDeveloper 11g installer.

In the Oracle JDeveloper installer, select a custom installation and select the ADF Runtime component. This step allows you to install the ADF runtime jar files and domain templates to the server environment.

For more information, see "Creating and Extending WebLogic Domains" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

- Run the Oracle WebLogic Server configuration wizard from an Oracle SOA Suite, WebCenter, Oracle Identity Management, or Oracle Portal, Forms, Reports, and Discoverer Oracle home.

Select the options to extend the domain and select the JRF template when prompted with the list of available templates.

For more information, refer to the appropriate Oracle Fusion Middleware installation guide.

- Use Oracle Enterprise Manager Fusion Middleware Control or the ApplyJRF WebLogic Scripting Tool (WLST) command to apply the JRF template to an existing WebLogic server instance.

For more information, see Section 17.3.1, "Applying Java Required Files to a Managed Server or Cluster," in the *Oracle Fusion Middleware Administrator's Guide*.

5.2.3.3 Upgrading Existing Oracle WebLogic Server Instances to the 11.1.1.1.0 ADF Runtime

A version of Oracle JDeveloper 11g (11.1.1.0.x) was made available to Oracle customers before the release of Oracle Fusion Middleware 11g.

If you have already installed and configured an Oracle WebLogic Server domain using the Oracle JDeveloper 11g (11.1.1.0.x), then you should be aware that Oracle Fusion Middleware 11g and the new Oracle JDeveloper 11g (11.1.1.1.0) provides a newer version of the Oracle ADF runtime software.

To upgrade your existing Oracle WebLogic Server domain with the newer Oracle ADF 11g (11.1.1.1.0) runtime software:

1. Open the uninstaller for of the Oracle WebLogic Server instance you want to upgrade.

You can find the Oracle WebLogic Server uninstaller in the following location:

On UNIX systems:

```
WLS_HOME/wlserver_10.3/uninstall/uninstall.sh
```

On Windows systems:

```
WLS_HOME\wlserver_10.3\uninstall\uninstall.cmd
```

Note that on a headless server (such as a Linux or UNIX-based system), you may have to set the DISPLAY variable to an X Server like VNC or Xming.

2. Select JDeveloper 11g (11.1.1.0.x) Studio Edition 11.1.1.0.0.
3. Deselect everything under WebLogic Server.
4. Click **Next** to complete the uninstall. If you're interested you can click on the Details button to see what was uninstalled.
5. Open the JDeveloper 11.1.1.1.0 installer.
6. Select the Oracle WebLogic Server instance where you want to install the new version of the ADF runtime.
7. Deselect everything, and then select only **Application Development Framework Runtime**.

Note that this will also automatically select the JDeveloper and ADF folders.

- 8.** To avoid class loading issues, rebuild all your EAR files.

Upgrading an Oracle SOA Suite Environment

This chapter describes how to upgrade your Oracle SOA Suite environment from Oracle Application Server 10g to Oracle Fusion Middleware 11g.

Upgrading this environment and your applications includes the following tasks:

- [Task 1: Decide Upon an Oracle SOA Topology](#)
- [Task 2: Verify that Any Required External Partners are Up and Running](#)
- [Task 3: If Necessary, Upgrade the Oracle Database That Hosts the Oracle SOA Suite Schemas](#)
- [Task 4: Use the Repository Creation Utility to Install the 11g SOA Schemas](#)
- [Task 5: Install and Configure the Oracle SOA Middle Tiers](#)
- [Task 6: If Necessary, Upgrade the SOA Component Schemas](#)
- [Task 7: Configure Any Required Java EE Features to Support Your SOA Applications](#)
- [Task 8: Perform Any Required Post-Upgrade Tasks for the SOA Components](#)

6.1 Task 1: Decide Upon an Oracle SOA Topology

Before you begin the upgrade process, decide upon a topology for your new Oracle Fusion Middleware 11g environment.

For more information, see [Section 3.2, "Information for Oracle SOA Suite Users"](#).

Note that the instructions in this guide provide the steps for upgrading a basic Oracle SOA Suite environment. After reviewing the content of this guide, you should be able to transfer this knowledge to a more complex production environment.

For specific information about setting up a production environment that takes advantage of clustering and high availability, refer to the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*.

For an overview of the high-level tasks you must perform when upgrading an enterprise deployment, see "Planning an Enterprise Deployment" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

6.2 Task 2: Verify that Any Required External Partners are Up and Running

Before you upgrade a SOA application to 11g, you must ensure that the external partners (for example, in the form of remote Web services) are up and running and available.

Oracle JDeveloper 11g must be able to access these external partners; otherwise, it can not properly migrate the projects in your application to 11g.

To verify that the required Web services are available, you should review each of the projects in your Oracle SOA Suite applications before you migrate them to 11g. For example, you can review the Oracle BPEL Process Manager configuration files for each project and verify that the Web services URIs referenced in those configuration files are valid and working.

For an example of how to identify and verify the external partners for a specific, demonstration application (the SOA Order Booking demo application), see [Section A.2.3, "Task 1c: Verify That All External Partners Are Available"](#).

6.3 Task 3: If Necessary, Upgrade the Oracle Database That Hosts the Oracle SOA Suite Schemas

If you are upgrading an Oracle SOA Suite environment, then you must ensure that the version of the database that will contain the Oracle SOA Suite schemas is supported by Oracle Fusion Middleware 11g.

You can install a new database, or upgrade your existing database to a supported version.

Note that if your upgrade involves multiple databases, then you might have to upgrade both the source database and the target database before proceeding with the upgrade. For example, the Oracle BAM upgrade can involve separate source and target databases.

For instructions on verifying that your database meets the requirements of Oracle Fusion Middleware 11g, see "Upgrading and Preparing Your Databases" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

6.4 Task 4: Use the Repository Creation Utility to Install the 11g SOA Schemas

If you are upgrading Oracle SOA Suite, use the Repository Creation Utility to install the schemas in the database.

Repository Creation Utility is a graphical and command-line tool used to create and manage Oracle Fusion Middleware database schemas in your database.

For more information, refer to the following sections:

- [Running the Repository Creation Utility in Preparation for an Oracle SOA Suite Upgrade](#)
- [Selecting the Schemas Required for a Oracle SOA Suite Upgrade](#)

6.4.1 Running the Repository Creation Utility in Preparation for an Oracle SOA Suite Upgrade

To start the Repository Creation Utility (RCU), you run the `rcu` command, which is in the root directory of the Repository Creation Utility CD-ROM or (if you have already installed the Oracle SOA Suite software on disk) in the `bin` directory of the Oracle SOA Suite Oracle home.

For more detailed instructions on how to obtain and run RCU, refer to the *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

For more information about which schemas to select when running the Repository Creation Utility, refer to [Section 6.4.2, "Selecting the Schemas Required for a Oracle SOA Suite Upgrade"](#).

For more information about the Oracle SOA Suite schemas that are installed by the Repository Creation Utility, refer to "Metadata Repository Schemas" in the *Oracle Fusion Middleware Administrator's Guide*.

6.4.2 Selecting the Schemas Required for a Oracle SOA Suite Upgrade

You can use the Repository Creation Utility to install the schemas required for all of the Oracle Fusion Middleware software components that require a schema. However, there is no need to install all the schemas unless you plan to install a complete Oracle Fusion Middleware environment and you plan to use the same database for all the Oracle Fusion Middleware component schemas.

For a Oracle SOA Suite upgrade, you must select the following schemas when you prompted by the Repository Creation Utility (as shown in [Figure 6-1](#)):

- Select the **Metadata Services** schema in the **AS Common Schemas** category.

This schema supports Oracle Fusion Middleware Metadata Services (MDS), which is required by some Oracle Fusion Middleware components, as well as for custom applications that support MDS. For more information, see "Understanding the MDS Repository" in the *Oracle Fusion Middleware Administrator's Guide*.

- Select the entire **SOA Infrastructure** category of schemas.

These schemas are required for the Oracle SOA Suite components.

Figure 6–1 Selecting the Required Schemas for a Oracle SOA Suite Upgrade

A Prefix groups the components associated with one deployment.

Select an existing Prefix DMTDEV

Create a new Prefix DEV

Prefix can contain only alpha-numeric characters. Prefix should not start with a number and should not contain any special characters.

| Component | Schema Owner |
|--|--------------|
| <input type="checkbox"/> Oracle AS Repository Components | |
| <input checked="" type="checkbox"/> AS Common Schemas | |
| <input checked="" type="checkbox"/> Metadata Services | DEV_MDS |
| <input type="checkbox"/> Audit Services | IAU |
| <input type="checkbox"/> Enterprise Scheduler Service | ORAESS |
| <input type="checkbox"/> Identity Management | |
| <input type="checkbox"/> ECM_SUITE | |
| <input type="checkbox"/> ODI_REPOSITORIES | |
| <input type="checkbox"/> WebLogic Communication Services | |
| <input checked="" type="checkbox"/> SOA Infrastructure | |
| <input checked="" type="checkbox"/> SOA Infrastructure | DEV_SOAINFRA |
| <input checked="" type="checkbox"/> Business Activity Monitoring | DEV_ORABAM |
| <input checked="" type="checkbox"/> User Messaging Service | DEV_ORASDPM |
| <input type="checkbox"/> Webcenter Suite | |
| <input type="checkbox"/> Portal and BI | |

6.5 Task 5: Install and Configure the Oracle SOA Middle Tiers

Before you upgrade to Oracle Fusion Middleware 11g, you must install an Oracle Fusion Middleware environment that is similar to the topology you set up for Oracle Application Server 10g.

Refer to the following documentation resources for complete instructions on installing and configuring the Oracle SOA Suite middle tiers:

- *The Oracle Fusion Middleware Installation Planning Guide*
- *The Oracle Fusion Middleware Quick Installation Guide for Oracle SOA Suite*
- *The Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*

[Table 6–1](#) provides a high-level summary of the main steps you must perform when installing and configuring the Oracle SOA Suite middle tiers, as well as specific references to the procedures you must follow for each step in the *Oracle Fusion Middleware Quick Installation Guide for Oracle SOA Suite*.

Table 6–1 Main Steps Required to Install and Configure the Oracle SOA Suite Middle Tiers

| Step # | Description | Section in the <i>Oracle Fusion Middleware Quick Installation Guide for Oracle SOA Suite</i> |
|--------|---|--|
| 1 | Install Oracle WebLogic Server and create the Middleware home | "Install Oracle WebLogic Server and Create the Middleware Home" |
| 2 | Install the Oracle SOA Suite software and create the Oracle SOA Suite Oracle home | "Installing the SOA Suite" |
| 3 | Configure the Oracle SOA Suite | "Configuring Oracle SOA Suite" |
| 4 | Configure Oracle BAM | "Configuring Oracle BAM" |
| 5 | Start and verify the Oracle WebLogic Server domain | "Post-Installation Tasks" |

6.6 Task 6: If Necessary, Upgrade the SOA Component Schemas

If you are upgrading the Oracle B2B or Oracle BAM components, then you must run the Oracle Fusion Middleware Upgrade Assistant to upgrade these component schemas to 11g.

The Oracle Fusion Middleware Upgrade Assistant automates the upgrade of many aspects of your Oracle Application Server 10g environment.

The Upgrade Assistant is installed automatically into the `bin` directory of your Oracle Fusion Middleware Oracle home. You run the Upgrade Assistant once for each 10g schema you are upgrading.

Note: If you are upgrading a 10g Release 3 (10.1.3.1.0) Oracle home that was installed using the Basic Installation option, then there is no need to run the Upgrade Assistant. When you select the 10g Release 3 (10.1.3.1.0) Basic Installation option, the installer does not install any components that can be upgraded by the Oracle Fusion Middleware Upgrade Assistant.

Refer to the following sections for more information:

- [Task 6a: Start the Upgrade Assistant for an Oracle SOA Schema Upgrade](#)
- [Task 6b: If Necessary, Upgrade the B2B Schema](#)
- [Task 6c: If Necessary, Upgrade the BAM Schema](#)

6.6.1 Task 6a: Start the Upgrade Assistant for an Oracle SOA Schema Upgrade

To start the Upgrade Assistant using the graphical user interface:

Note: You can also use the Upgrade Assistant command-line interface to upgrade your Oracle Application Server 10g Oracle homes. For more information, see "Using the Upgrade Assistant Command-Line Interface" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

1. Change directory to the `ORACLE_HOME/bin` directory of the Oracle Fusion Middleware installation.
2. Enter the following command to start the Upgrade Assistant.

On UNIX system:

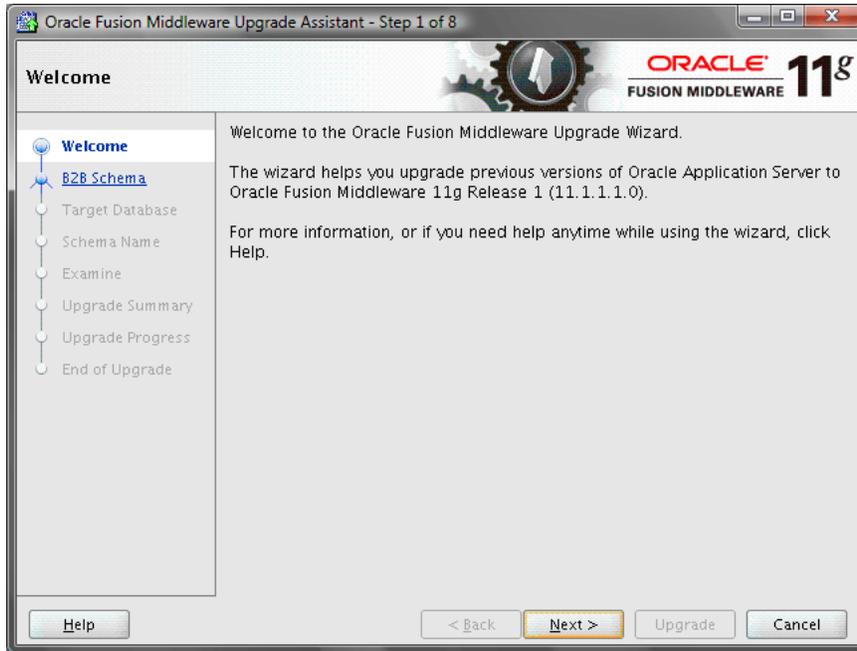
```
./ua
```

On Windows systems:

```
ua.bat
```

The Upgrade Assistant displays the Welcome screen as shown in [Figure 6-2](#).

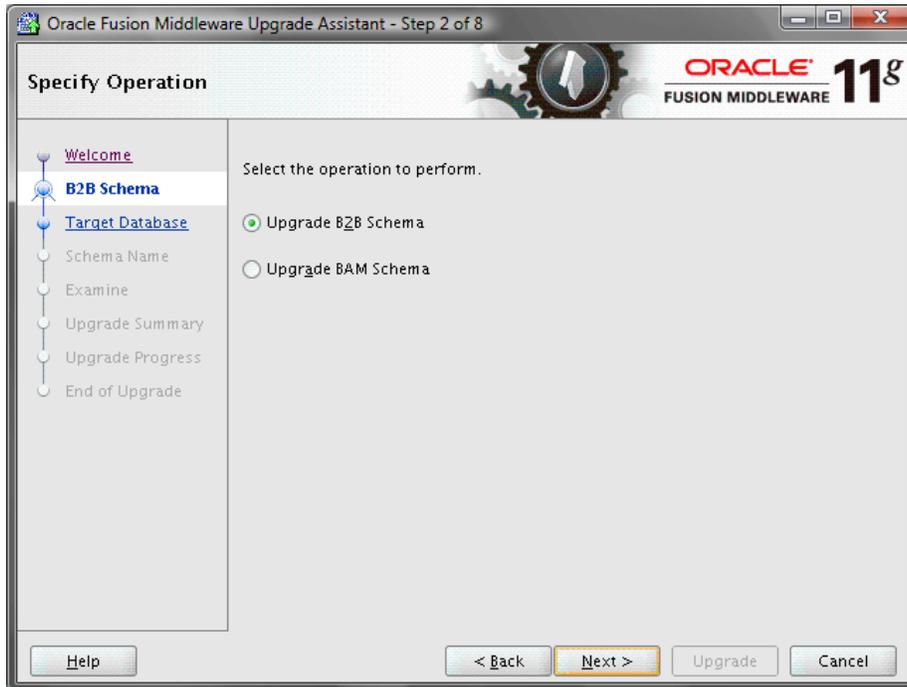
Figure 6–2 Upgrade Assistant Welcome Screen



3. Click **Next** to display the Specify Operation screen (Figure 6–3).

The options available in the Upgrade Assistant are specific to the Oracle home from which it started. When you start Upgrade Assistant from an Oracle SOA Suite and WebCenter Oracle home, the options shown on the Select Operation screen are the valid options for an Oracle SOA Suite and WebCenter Oracle home.

Figure 6–3 Upgrade Assistant Specify Operation Screen for a SOA Upgrade



6.6.2 Task 6b: If Necessary, Upgrade the B2B Schema

Refer to the following sections for information about upgrading the Oracle B2B schema:

- [General Information About the B2B Schema Upgrade](#)
- [Applying the Required Patch Set to the Oracle B2B 10g Middle Tier](#)
- [Setting the Maximum Message Size Attribute In the Oracle WebLogic Server Administration Console](#)
- [Performing the B2B Schema Upgrade](#)

6.6.2.1 General Information About the B2B Schema Upgrade

If you are using Oracle Application Server 10g Release 2 (10.1.2) and you are upgrading Oracle Application Server Integration B2B, then you will need to upgrade the B2B schema in your database.

See Also: *Oracle Fusion Middleware User's Guide for Oracle B2B* in the Oracle Application Server 10g Release 2 (10.1.2) Documentation Library on the Oracle Technology Network:

<http://www.oracle.com/technology/documentation/>

When you upgrade the B2B schema from 10g Release 2 (10.1.2), you must apply a software patch to your 10g Release 2 (10.1.2) installation, and then export the B2B 10g schema from your database. The exported data is saved to a ZIP file on disk. Then, you can import the data into the database that contains your Oracle Fusion Middleware 11g component schemas.

6.6.2.2 Applying the Required Patch Set to the Oracle B2B 10g Middle Tier

Before you can perform the Oracle B2B schema upgrade, you must apply the a required patch set to the Oracle Application Server Integration B2B 10g Release 2 (10.1.2) middle tier.

This patch set is required in order to enable the Java command that you use to export the trading partner agreements, as described in the following step.

For specific information about the patch set requirements for upgrade and interoperability with previous Oracle Application Server releases, see the *Oracle Fusion Middleware Release Notes*.

6.6.2.3 Setting the Maximum Message Size Attribute In the Oracle WebLogic Server Administration Console

The exported customer data from your Oracle B2B 10g schema can result in a very large ZIP file on disk. To be sure that the large file can be imported successfully into the Oracle Fusion Middleware 11g metadata repository database, you should modify the default **Maximum Message Size** general protocol attribute for the WebLogic managed server where the SOA Suite components are deployed.

To perform this task:

1. Log in to the Oracle WebLogic Server Administration Console:

<http://localhost:7001/console>

2. In the left pane of the Console, expand **Environment** and select **Servers**.

3. On the Summary of Servers page, select the server instance you want to configure. In most cases, the managed server that hosts the SOA components is called `soa_server1`.
4. Click **Protocols** to display the general connection settings for network protocols.
5. Scroll down to the **Maximum Message Size** property and set it to 200000000.

6.6.2.4 Performing the B2B Schema Upgrade

To upgrade the B2B schema:

1. Apply the required patch set to the Oracle Application Server Integration B2B 10g Release 2 (10.1.2) middle tier.

For more information, see [Section 6.6.2.2, "Applying the Required Patch Set to the Oracle B2B 10g Middle Tier"](#).

2. Stop the managed server in the Oracle WebLogic Server domain where Oracle B2B 11g is running.
3. Export the 10g Release 2 (10.1.2) trading partner agreements to an external ZIP file by using the following procedure:
 - a. Export the 10g Release 2 (10.1.2) trading partner agreements that you have deployed by running the following command in the Oracle Application Server Integration B2B Oracle home:

```
java -ms1024M -mx1024M oracle.tip.repos.export.Export 11g
```

This Java command creates the following directory, which contains the exported data:

On Windows systems:

```
%TEMP%/export.zip
```

On UNIX systems:

```
/tmp/export.zip
```

- b. Create the following directory in the Oracle Fusion Middleware 11g Oracle home, which can be on the same host or on a different host from the 10g Oracle home:

On Windows systems:

```
ORACLE_HOME\b2b\upgrade
```

On UNIX systems:

```
ORACLE_HOME/b2b/upgrade
```

- c. Copy the `export.zip` file to the directory you just created.
4. Use the Oracle WebLogic Server Administration Console to increase the **Maximum Message Size** property for the SOA component managed server (for example, `soa_server1`).

For more information, see [Section 6.6.2.3, "Setting the Maximum Message Size Attribute In the Oracle WebLogic Server Administration Console"](#).

5. Start the Upgrade Assistant as described in [Task 6a: Start the Upgrade Assistant for an Oracle SOA Schema Upgrade](#).

6. Select **B2B Schema** on the Specify Operation screen (Figure 6–3).
7. Refer to Table 6–2 for a description of the Upgrade Assistant screens that require input from you during a B2B Schema upgrade and the options on each screen.
Note that you are not prompted for the location of the ZIP file that contains the Oracle B2B 10g schema and customer data. Instead, the Upgrade Assistant automatically locates the ZIP file in the location specified in Step 3.
8. After the Specify Target Database screen, the Upgrade Assistant performs the following tasks and provides the progress on each task:
 - Examines the components and schemas to be upgraded and verifies that they can be upgraded successfully.
 - Provides a summary of the components to be upgraded so you can verify that Upgrade Assistant is upgrading the components and schemas you expect.
 - Provides a progress screen so you can see the status of the upgrade as it proceeds.
 - Alerts you of any errors or problems that occur during the upgrade.

See Also: "Troubleshooting Your Upgrade" in the *Oracle Fusion Middleware Upgrade Planning Guide* for specific instructions for troubleshooting problems that occur while running the Upgrade Assistant

 - Displays the End of Upgrade screen, which confirms that the upgrade was complete.
9. When the Oracle B2B schema upgrade is complete, restart the Oracle SOA Suite managed server where Oracle B2B 11g is running.

Table 6–2 Upgrade Assistant Screens That Require Input During a B2B Schema Upgrade

| Upgrade Assistant Screen | Description |
|--------------------------|--|
| Specify Target Database | <p>Use this screen to enter the connection details for the database where you installed the Oracle B2B 11g schema with the Repository Creation Utility.</p> <ul style="list-style-type: none"> ■ Target Database Type - Select Oracle from this drop-down list to upgrade the Oracle B2B schema to an Oracle database. ■ Connect String - Enter the connection string in the following format: host:port:SID For example: host3:1521:soaDB3 ■ User Name - Enter the database user name; if you enter SYS, be sure to enter the string, SYS as SYSDBA. ■ Password - The password for the database user. |
| Specify Schema Name | <p>Use this screen to select the Oracle B2B 11g schema name and to enter the password for the schema.</p> <p>This is the password you defined when you were installing the B2B schema with the 11g Repository Creation Utility.</p> |

6.6.3 Task 6c: If Necessary, Upgrade the BAM Schema

To upgrade the Oracle BAM schema, refer to the following sections:

- [Understanding the Oracle BAM Schema Upgrade](#)
- [About Upgrading From Previous Versions of Oracle BAM](#)
- [Applying the Required Patch Set to the Oracle Database That Contains the Oracle BAM Schema](#)
- [Performing the Oracle BAM Schema Upgrade](#)

6.6.3.1 Understanding the Oracle BAM Schema Upgrade

If you are using Oracle Application Server 10g Release 3 (10.1.3) and you are upgrading Oracle Business Activity Monitoring, then you will need to upgrade the BAM schema in your database.

Note the following about the Oracle BAM schema upgrade:

- This procedure assumes you have already run the Repository Creation Utility and created the new Oracle BAM 11g schema in a supported database. When you create the Oracle BAM 11g schema it does not overwrite the existing Oracle BAM 10g schema.
- To upgrade the Oracle BAM schema, you must identify a database that contains the Oracle BAM 10g schema and a database that contains the Oracle BAM 11g schema.
- Note that the 10g schema and the 11g schema can reside in the same database or in different databases; in either case, you must identify the source and the target database.
- For information about locating and configuring the Oracle BAM upgrade log files, see "Locating and Configuring the Oracle BAM Upgrade Log Files" in the Oracle Fusion Middleware Upgrade Planning Guide.

6.6.3.2 About Upgrading From Previous Versions of Oracle BAM

If you are using Oracle Application Server 10g Release 2 (10.1.2) and you want to upgrade Oracle Business Activity Monitoring to Oracle Fusion Middleware 11g, then you must first upgrade to Oracle Application Server 10g Release 3 (10.1.3).

Refer to Section 3.5, "Upgrading an Installation" in the *Oracle Business Activity Monitoring Installation Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library on the Oracle Technology Network:

<http://www.oracle.com/technology/>

6.6.3.3 Applying the Required Patch Set to the Oracle Database That Contains the Oracle BAM Schema

Before you can upgrade the Oracle BAM schema, you must apply a patch set to the Oracle Database that contains the Oracle BAM 10g schemas.

For specific information about the patch set requirements for upgrade and interoperability with previous Oracle Application Server releases, see the *Oracle Fusion Middleware Release Notes*.

6.6.3.4 Performing the Oracle BAM Schema Upgrade

To upgrade the BAM schema:

1. Stop the Oracle WebLogic Server instance that hosts the BAM application.
For more information, see "Starting and Stopping Oracle Fusion Middleware" in the *Oracle Fusion Middleware Administrator's Guide*.
2. Start the Upgrade Assistant as described in [Task 6a: Start the Upgrade Assistant for an Oracle SOA Schema Upgrade](#).
3. Select **BAM Schema** on the Specify Operation screen ([Figure 6–3](#)).
4. Refer to [Table 6–3](#) for a description of the Upgrade Assistant screens that require input from you during a Oracle BAM schema upgrade and the options on each screen.
5. After the Specify Schema Name screen, the Upgrade Assistant performs the following tasks and provides the progress on each task:
 - Examines the components and schemas to be upgraded and verifies that they can be upgraded successfully.
 - Provides a summary of the components to be upgraded so you can verify that Upgrade Assistant is upgrading the components and schemas you expect.
 - Provides a progress screen so you can see the status of the upgrade as it proceeds.
 - Alerts you of any errors or problems that occur during the upgrade.

See Also: "Troubleshooting Your Upgrade" in the *Oracle Fusion Middleware Upgrade Planning Guide* for specific instructions for troubleshooting problems that occur while running the Upgrade Assistant

 - Displays the End of Upgrade screen, which confirms that the upgrade was complete.

Table 6–3 Upgrade Assistant Screens That Require Input During a BAM Schema Upgrade

| Upgrade Assistant Screen | Description |
|---------------------------------|--|
| Specify Source Database | Use this screen to enter the database connection details for the database that contains the Oracle BAM 10g schema. |
| Specify Target Database Details | Use this screen to enter the database connection details for the database that contains the Oracle BAM 11g schema that you created with the Oracle Fusion Middleware Repository Creation Utility. |
| Specify Schema Name | Use this screen to select the Oracle BAM 11g schema and enter the password for the BAM schema. This is the Oracle BAM schema password that you set in the Repository Creation Utility when you installed the schema in the database. |
| Examining Components | This page shows the status of examination phase, where the upgrade assistant verifies the source and target schema. |

6.7 Task 7: Configure Any Required Java EE Features to Support Your SOA Applications

Besides the SOA system components, which are installed during the Oracle WebLogic Server installation, you might have to configure additional Oracle WebLogic Server features to support the SOA applications you deploy.

For example, you may have to define application-required data sources, JMS resources, or other services. The resources and services provided by the Oracle WebLogic Server domain are common to all Java-based applications, including Oracle SOA Suite applications.

For more information, see the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

6.8 Task 8: Perform Any Required Post-Upgrade Tasks for the SOA Components

Depending upon the components you are upgrading, the following post-upgrade tasks might apply:

- [Configuring Oracle SOA Suite 11g to Use Identity Management Components](#)
- [About Case Sensitivity in Oracle Human Workflow](#)
- [Oracle B2B Post-Upgrade Information and Tasks](#)

6.8.1 Configuring Oracle SOA Suite 11g to Use Identity Management Components

If your Oracle Application Server 10g environment relied on any Oracle Identity Management components, such as Oracle Internet Directory, or if you were using any third-party LDAP servers, then you can associate those components with your new Oracle Fusion Middleware 11g environment.

Specifically, you can use Oracle Enterprise Manager Fusion Middleware Control to associate an LDAP directory with Oracle Fusion Middleware, so you can use the LDAP directory as the credential or policy store for the Oracle WebLogic Server domain.

For more information, see the following:

- "Reassociating Domain Stores with Oracle Enterprise Manager" in the *Oracle Fusion Middleware Security Guide*.
- "Configuring LDAP Authentication Providers" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

6.8.2 About Case Sensitivity in Oracle Human Workflow

Oracle Human Workflow 10g and Oracle Human Workflow 11g support different default behavior in terms of the case sensitivity of user names, group names, and application role names:

- In Oracle Human Workflow 11g:
 - By default, all user names in the Human Workflow component are case insensitive, and all user names are stored in lower case.
 - By default, all group names and application role names are always case sensitive.
- In Oracle Workflow 10g Release 3 (10.1.3), the system was case sensitive to user names, by default, and all user names were stored in same case as seeded.

To keep both environments compatible, you should set the case sensitivity to be the same in 11g as it was in 10g Release 3 (10.1.3).

User name case insensitivity can be changed in `workflow-identity-config.xml`. The `caseSensitive` property identifies whether or not user names are case sensitive. This flag should be upgraded using the MBean browser.

For more information, see "Case Sensitivity" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

6.8.3 Oracle B2B Post-Upgrade Information and Tasks

Table 6–4 summarizes the information you should know about the Oracle B2B upgrade, as well as some specific post-upgrade tasks you might have to perform after upgrading to Oracle B2B 11g.

Table 6–4 Post-Upgrade Information for Oracle B2B Users

| Topic | Description | More Information |
|--|---|---|
| Host and Partner Security Certificates | <p>Oracle B2B host and partner security certificates are not upgraded into the expected format for Oracle B2B 11g.</p> <p>Instead, you must copy the certificates from the root directory of the <code>export.zip</code> file (created in Section 6.6.2.4, "Performing the B2B Schema Upgrade") and import them into the keystore.</p> <p>Note that Oracle B2B now saves certificates in the keystore, rather than an SSL wallet.</p> | <p>"Security Features of Oracle B2B" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i></p> <p>"Configuring Identity and Trust" in the <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i>.</p> |
| Callout library jar files | <p>If you created your own custom callout library jar files in Oracle B2B 10g, then those jar files are not upgraded to Oracle B2B 11g.</p> <p>Instead, you must recreate your custom callout jar files using Oracle B2B 11g.</p> | "Managing Callouts" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Document Protocol Revision and Document Type | <p>These two fields have been modified for Oracle B2B 11g. In Oracle B2B 10g, both these fields included a name and version. Oracle B2B uses only the Document Protocol Version and Document Type Name fields.</p> <p>As a result, when you enqueue a document to the <code>IP_OUT_QUEUE</code>, note that you must provide a different set of attributes than those provided in Oracle B2B 10g.</p> | "MS Enqueue and Dequeue Utilities" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Exchange protocol parameter identifiers | <p>In Oracle B2B 10g, you could specify identifiers for the exchange protocol parameter of a channel. Adding identifiers to the exchange protocol parameter is no longer supported in Oracle B2B 11g, and these identifiers are not upgraded to Oracle B2B 11g.</p> <p>Instead, Oracle B2B 11g, you must recreate the required identifiers as part of a Trading Partner Profile.</p> | "Creating Trading Partner Profiles" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |

Table 6–4 (Cont.) Post-Upgrade Information for Oracle B2B Users

| Topic | Description | More Information |
|--|--|--|
| Exchange protocol parameters for the MLLP channel. | <p>The following exchange protocol parameters in Oracle B2B 10g are now attributes of the MLLP channel:</p> <ul style="list-style-type: none"> ■ ImmediateACK ■ ImmediateACK-Custom-File ■ ImmediateACK-mapAckControlID ■ ImmediateACK-MapImmTriggerEvt ■ DiscardHL7ACK <p>If you set these parameters in Oracle B2B 10g, then you will need to set them as attributes of the MLLP channel.</p> | "Configuring Channels" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Channel passwords | <p>Oracle B2B 11g uses a new method for saving user credentials and password information. As a result, any passwords you entered for channels in 10g are not upgraded to Oracle B2B 11g.</p> <p>Instead, after upgrade, you must enter the passwords as part of the channel parameters.</p> | "Configuring Channels" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Receipt Recipient property of the AS2-1.1 channel | <p>If you set the Receipt Delivery Option property of the AS2-1.1 channel in your Oracle B2B 10g environment, then this property value is not upgraded to Oracle B2B 11g. After you upgrade, you must set this property accordingly.</p> | "Configuring Channels" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Some optional fields are now mandatory in Oracle B2B 11g | <p>Note that the following fields have changed from optional to required; as a result, after upgrade, you might have to provide values for these fields:</p> <ul style="list-style-type: none"> ■ If you were using EDI EDIFACT and EDI X12 documents in Oracle B2B 10g, then note that the Group ID identifier is now a required identifier. ■ If you are using RosettaNet documents in Oracle B2B 10g, then note that the Collaboration Code and Time to perform for Collaboration fields are not required. In Oracle B2B 11g, they are required. | "Creating Types" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |
| Inbound and outbound partner agreements | <p>When you upgrade to Oracle B2B 11g, a partner agreement that existing in Oracle B2B 10g be displayed in Oracle B2B 11g as two agreements: one agreement is for the inbound exchange; the other is for the outbound exchange.</p> <p>For example, a 10g Rosetta 3A4 agreement is split into the following:</p> <ul style="list-style-type: none"> ■ Pip3A4PurchaseOrderConfirmation_Inbound ■ Pip3A4PurchaseOrderRequest_Outbound | "Creating and Deploying Trading Partner Agreements" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i> |

Table 6–4 (Cont.) Post-Upgrade Information for Oracle B2B Users

| Topic | Description | More Information |
|--|---|--|
| RosettaNet outbound messages | <p>The behavior of RosettaNet outbound messages has changed in Oracle B2B 11g.</p> <p>When using RosettaNet with Oracle B2B 11g, you must now set an additional enqueue property (msgType=2) for any outbound message that needs to be treated as a response message (for example, a PurchaseOrderConfirmation message).</p> <p>Otherwise, the outbound message will be treated as a request message. In Oracle B2B 10g, this property was not necessary.</p> | <p>"Utilities for Enqueuing and Dequeuing" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i>.</p> |
| MLLP channels configured for inbound and outbound messages | <p>After upgrade, perform one of the following tasks, depending on how you configured the MLLP channel in Oracle B2B 10g:</p> <ul style="list-style-type: none"> ■ If you used the <code>singleDC</code> property in the <code>tip.properties</code> file to enable one MLLP channel to transfer both inbound and outbound messages, then you must create a new channel and then update the partner agreement to use the new channel. ■ If you did not use the <code>singleDC</code> property, then set the connection mode of the host trading partner to Server. | <p>"Adding Channel Details" in the <i>Oracle Fusion Middleware User's Guide for Oracle B2B</i></p> |

Upgrading an Oracle WebCenter Environment

This chapter describes how to upgrade your Oracle WebCenter environment from Oracle Application Server 10g to Oracle Fusion Middleware 11g.

Upgrading this environment and your applications includes the following tasks:

- [Task 1: Decide Upon a Oracle WebCenter Topology](#)
- [Task 2: If Necessary, Upgrade the Oracle Database That Hosts the Oracle WebCenter Schemas](#)
- [Task 3: Use the Repository Creation Utility to Install the 11g Oracle WebCenter Schemas](#)
- [Task 4: Install and Configure the Oracle WebCenter Middle Tiers](#)
- [Task 5: Configure Any Required Java EE Features to Support Your Oracle WebCenter Applications](#)

7.1 Task 1: Decide Upon a Oracle WebCenter Topology

Before you begin the upgrade process, decide upon a topology for your new Oracle Fusion Middleware 11g environment.

For more information, see [Section 3.3, "Information for Oracle WebCenter Users"](#).

7.2 Task 2: If Necessary, Upgrade the Oracle Database That Hosts the Oracle WebCenter Schemas

If you are upgrading an Oracle WebCenter environment, you must ensure that the version of the database where you plan to install the Oracle WebCenter schemas is a supported by Oracle Fusion Middleware 11g.

You can install a new database, or upgrade your existing database to a supported version.

For instructions on verifying that your database meets the requirements of Oracle Fusion Middleware 11g, see "Upgrading and Preparing Your Databases" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

7.3 Task 3: Use the Repository Creation Utility to Install the 11g Oracle WebCenter Schemas

If you are upgrading to Oracle WebCenter, use the Repository Creation Utility to install the schemas in the database you identified and prepared in [Task 2: If Necessary, Upgrade the Oracle Database That Hosts the Oracle WebCenter Schemas](#).

For more information, refer to the following sections:

- [Running the Repository Creation Utility in Preparation for an Oracle WebCenter Upgrade](#)
- [Selecting the Schemas Required for a Oracle WebCenter Upgrade](#)

7.3.1 Running the Repository Creation Utility in Preparation for an Oracle WebCenter Upgrade

To start the Repository Creation Utility, you run the `rcu` command, which is in the root directory of the Repository Creation Utility CD-ROM or (if you have already installed the Oracle WebCenter software on disk) in the `bin` directory of the Oracle WebCenter Oracle home.

For more detailed instructions on how to obtain and run RCU, refer to the *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

For more information about which schemas to select when running the Repository Creation Utility, refer to [Section 7.3.2, "Selecting the Schemas Required for a Oracle WebCenter Upgrade"](#).

For more information about the Oracle WebCenter schemas that are installed by the Repository Creation Utility, refer to "Metadata Repository Schemas" in the *Oracle Fusion Middleware Administrator's Guide*.

7.3.2 Selecting the Schemas Required for a Oracle WebCenter Upgrade

You can use RCU to install the schemas required for all of the Oracle Fusion Middleware software components that require a schema. However, there is no need to install all the schemas unless you plan to install a complete Oracle Fusion Middleware environment and you plan to use the same database for all the Oracle Fusion Middleware component schemas.

For a Oracle WebCenter upgrade, you must select the following schemas when you prompted by the Repository Creation Utility (as shown in [Figure 7-1](#)):

- Select the **Metadata Services** schema in the **AS Common Schemas** category.
This schema supports Oracle Fusion Middleware Metadata Services (MDS), which is required by the Oracle WebCenter Spaces component.
- Select the entire **Oracle WebCenter** category of schemas.

These are the schemas supported by the various Oracle WebCenter 11g

Figure 7–1 Selecting the Required Schemas for Oracle WebCenter

A Prefix groups the components associated with one deployment.

Select an existing Prefix

Create a new Prefix

Prefix can contain only alpha-numeric characters. Prefix should not start with a number and should not contain any special characters.

| Component | Schema Owner |
|---|-----------------|
| <input type="checkbox"/> Oracle AS Repository Components | |
| <input checked="" type="checkbox"/> AS Common Schemas | |
| <input checked="" type="checkbox"/> Metadata Services | DEV_MDS |
| <input type="checkbox"/> Audit Services | IAU |
| <input type="checkbox"/> Enterprise Scheduler Service | ORAESS |
| <input type="checkbox"/> Identity Management | |
| <input type="checkbox"/> ECM_SUITE | |
| <input type="checkbox"/> ODI_REPOSITORIES | |
| <input type="checkbox"/> WebLogic Communication Services | |
| <input type="checkbox"/> SOA Infrastructure | |
| <input checked="" type="checkbox"/> Webcenter Suite | |
| <input checked="" type="checkbox"/> WebCenter Spaces | DEV_WEBCENTER |
| <input checked="" type="checkbox"/> Portlet Producers | DEV_PORTLET |
| <input checked="" type="checkbox"/> Oracle Content Server 10g | DEV_OCSERVER |
| <input checked="" type="checkbox"/> Discussions | DEV_DISCUSSIONS |
| <input checked="" type="checkbox"/> Wiki and Blogs | DEV_WIKI |
| <input type="checkbox"/> Portal and BI | |

7.4 Task 4: Install and Configure the Oracle WebCenter Middle Tiers

Before you upgrade to Oracle Fusion Middleware 11g, you must install an Oracle Fusion Middleware environment that is similar to the topology you set up for Oracle Application Server 10g.

For complete instructions on installing Oracle Fusion Middleware, refer to the following documentation resources:

- *The Oracle Fusion Middleware Installation Planning Guide*
- *The Oracle Fusion Middleware Quick Installation Guide for Oracle WebCenter*
- *The Oracle Fusion Middleware Installation Guide for Oracle WebCenter*

Table 7–1 provides a high-level summary of the main steps you must perform when installing and configuring the Oracle WebCenter middle tiers, as well as specific references to the procedures you must follow for each step in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.

Table 7–1 Main Steps Required to Install and Configure the Oracle SOA Suite Middle Tiers

| Step # | Description | Section in the <i>Oracle Fusion Middleware Installation Guide for Oracle WebCenter</i> |
|--------|---|--|
| 1 | Install Oracle WebLogic Server and create the Middleware home | "Install Oracle WebLogic Server and Create the Middleware Home" |
| 2 | Install the Oracle SOA Suite software and create the Oracle SOA Suite Oracle home | "Installing Oracle WebCenter" |
| 3 | Configure the Oracle WebCenter | "Configuring Oracle WebCenter" |
| 4 | Start Using Oracle WebCenter | "Working with Oracle WebCenter Components" |

7.5 Task 5: Configure Any Required Java EE Features to Support Your Oracle WebCenter Applications

Besides the Oracle WebCenter system components, which are installed during the Oracle WebLogic Server installation, you might have to configure additional Oracle WebLogic Server features to support the Oracle WebCenter applications you deploy.

For example, you may have to define application-required data sources, JMS resources, or other services. The resources and services provided by the Oracle WebLogic Server domain are common to all Java-based applications, including Oracle WebCenter applications.

For more information, see the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

Part III

Upgrading the Oracle SOA Suite and Oracle WebCenter Applications

Part II contains the following chapters:

- [Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications"](#)
- [Chapter 9, "Considerations When Upgrading Oracle SOA Applications"](#)
- [Chapter 10, "Considerations When Upgrading Oracle WebCenter Applications"](#)

Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications

After you have installed and upgraded your Oracle Fusion Middleware environment, you can then upgrade and redeploy your Oracle Application Server 10g Release 2 (10.1.2) and 10g Release 3 (10.1.3) applications on your new Oracle Fusion Middleware topology.

This chapter describes the recommended process of migrating and redeploying your Oracle SOA Suite, WebCenter, and Java EE applications on Oracle Fusion Middleware 11g.

This chapter has the following sections:

- [Task 1: Review Sample Application Upgrade Procedures](#)
- [Task 2: Verify that the Applications Are Up and Running Successfully on Oracle Application Server 10g](#)
- [Task 3: Verify That You Have Upgraded Your 10g Environment to 11g](#)
- [Task 4: Install and Start Oracle JDeveloper 11g](#)
- [Task 5: Open and Upgrade the Application in Oracle JDeveloper](#)
- [Task 7: Review and Recompile the Application](#)
- [Task 6: Locate Specific Upgrade Information for the Components of Your Application](#)
- [Task 8: Redeploy the Application](#)

8.1 Task 1: Review Sample Application Upgrade Procedures

Before you begin to upgrade your own custom Oracle SOA Suite, Oracle WebCenter, or Oracle ADF applications, consider reviewing the end-to-end procedures for upgrading some typical sample applications:

- If you want some experience upgrading a sample Oracle ADF application before tackling a larger scale upgrade, there is information available on Oracle Technology Network (OTN) that describes in detail how to migrate the SRDemo sample application, which is an ADF 10.1.3 JSF application using ADF Faces components, to Oracle JDeveloper 11g.

For more information, see the article "Migrating the 10.1.3 SRDemo Sample to 11g", which is available on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/products/jdev/tips/muench/1013srdemo/index.htm>
1

- If you want to review the complete end-to-end steps for upgrading the an example Oracle SOA Suite 10g application, refer to [Appendix A, "Example: Upgrading and Redeploying SOA Order Booking"](#)

8.2 Task 2: Verify that the Applications Are Up and Running Successfully on Oracle Application Server 10g

Before you attempt to redeploy your Oracle Application Server 10g applications on Oracle Fusion Middleware 11g, you must be sure that the applications have been deployed successfully on Oracle Application Server 10g and that all components of the application are currently up and running.

In particular, be sure that any external references that your application relies upon are up and running and available. For example, if your application relies on external Web services, those Web services must be up and running when you start the 11g upgrade and redeployment process.

This step is necessary for two reasons:

- To verify that the application has no issues or problems in the 10g environment that might cause the upgrade and redeployment to fail on Oracle Fusion Middleware 11g.
- To reduce the amount of work required to redefine the external references in your application source code.

For example, if your Oracle Enterprise Service Bus projects are calling external BPEL processes, it is important that the 10g Release 3 (10.1.3) BPEL processes are up and running and available when you upgrade the Oracle Enterprise Service Bus application.

Otherwise, you must modify the Oracle Enterprise Service Bus project before the upgrade so the external references point to the new 11g BPEL processes. If the external references are not available, the upgrade to Oracle JDeveloper 11g will fail.

Because of Oracle JDeveloper 11g development features, the process of redefining your external references is much easier and partially automated if you perform this task after the upgrade to Oracle JDeveloper 11g.

8.3 Task 3: Verify That You Have Upgraded Your 10g Environment to 11g

Before you begin upgrading your applications to 11g, you must perform the upgrade tasks for your particular Oracle SOA Suite, WebCenter, and ADF or environment. For more information, refer to [Part II, "Upgrading the Oracle SOA Suite, WebCenter, and ADF Environment"](#).

In particular, as part of the environment upgrade, you should have:

- Verified that the new Oracle Fusion Middleware 11g environment is up and running.
- Applied any OC4J configuration settings to Oracle Fusion Middleware 11g; this includes any data sources, system properties, and other configuration settings that are required by the applications you are about to redeploy on Oracle Fusion Middleware 11g.

8.4 Task 4: Install and Start Oracle JDeveloper 11g

For complete instructions about downloading, installing, and starting Oracle JDeveloper 11g, refer to the *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*.

After you start Oracle JDeveloper 11g, you should verify that you have the proper Oracle JDeveloper extensions for the applications you will be developing. For more information, see:

- [Section 9.1, "About Using the Oracle JDeveloper 11g Migration Wizard for Oracle SOA Suite Applications"](#)
- [Section 10.2, "Prerequisite Tasks for Migrating a WebCenter Application"](#)

Note: If you intend to migrate application settings from JDeveloper 10.1.3.x, then close and remove your WebCenter applications from the IDE in JDeveloper 10.1.3.x before you start JDeveloper 11g for the first time and install the WebCenter Oracle JDeveloper 11g extension.

You can remove applications from the IDE by right-clicking an application name in the Application Navigator and selecting **Delete** from the shortcut menu. Removing an application removes it only from the IDE and not from the disk.

8.5 Task 5: Open and Upgrade the Application in Oracle JDeveloper

For each Oracle Application Server 10g Oracle SOA Suite, WebCenter, and ADF application, you should have an Oracle JDeveloper 10g workspace (.jws) file. When you open a workspace from a previous version of Oracle JDeveloper, the Migration Wizard automatically upgrades the application to Oracle Fusion Middleware 11g.

This important step saves you time and effort in migrating to the newer versions of Java technologies supported by Oracle Fusion Middleware 11g.

Use the following steps to open a Oracle JDeveloper 10g workspace in Oracle JDeveloper 11g:

1. Save a copy of the project that you can manage using Oracle JDeveloper 10g Release 3 (10.1.3.1.0) in a separate folder or directory on disk.

Oracle JDeveloper backs up all your project directories as they are imported and upgraded to Oracle JDeveloper 11g. However, it is still good development practice to back up your work before starting the upgrade process.

Note that Oracle JDeveloper does not back up the workspace (.jws) file, only the project directories. As a result, if you keep a backup copy of the entire application source file directory, you can easily restore your previous pre-upgrade version.

After you open and upgrade the workspace in Oracle JDeveloper 11g, you cannot open the upgraded copy of the workspace in a previous version of Oracle JDeveloper.

2. Select **Open** from the **File** menu and select the Oracle JDeveloper 10g .jws file.

When selecting the workspace file, be sure the application source code is available in a directory location that will always be available. For example, do not attempt to upgrade a project that is stored on a network drive path (such as \\110.34.40.43\share\Samples\mysample.jws).

Oracle JDeveloper displays Step 1, the Welcome page of the Migration Wizard.

3. Follow the instructions on the screen to migrate the project files and selected Java technologies to Oracle JDeveloper Release 11g and Oracle Fusion Middleware 11g.
4. Review the application migration log files, which are located in the following directory inside the Oracle JDeveloper home directory:

On UNIX systems:

```
JDEV_HOME/JDeveloper/upgrade/logs/
```

On Windows systems:

```
JDEV_HOME\JDeveloper\upgrade\logs\
```

8.6 Task 6: Locate Specific Upgrade Information for the Components of Your Application

This chapter provides an overview of the steps required to upgrade and redeploy your Oracle SOA Suite, WebCenter, and ADF applications on Oracle Fusion Middleware 11g.

However, there are important considerations and upgrade tasks you might need to perform in addition to the general tasks described here.

For more information, review the following resources to determine if there are any special considerations that affect your Oracle SOA Suite, WebCenter, and ADF applications.

Table 8–1 Resources Available for Upgrading Specific Types of Application Source Code

| If your application uses... | Review this documentation... |
|-----------------------------|---|
| Java EE components | <i>Oracle Fusion Middleware Upgrade Guide for Java EE</i> |
| Oracle SOA Suite components | Chapter 9, "Considerations When Upgrading Oracle SOA Applications" |
| Oracle WebCenter components | Chapter 10, "Considerations When Upgrading Oracle WebCenter Applications" |

8.7 Task 7: Review and Recompile the Application

After you have opened the application in Oracle JDeveloper 11g, you should then perform the following tasks:

1. From the Oracle JDeveloper 11g **File** menu, select **Save All** to save all the projects in the selected workspace.
2. From the **Build** menu, select **Make All** to recompile all the projects in the workspace, or right-click one project at a time and select **Make *project_name***.
3. Review the output in the Compiler message panel at the bottom of the Oracle JDeveloper window.
4. For each compilation error, double-click the error icon and review the problem code.

For more information, refer to the appropriate developer guide in the Oracle Fusion Middleware 11g documentation library. The Oracle Fusion Middleware 11g documentation library is available at the following location on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/documentation/>

8.8 Task 8: Redeploy the Application

After you have reviewed the available resources and successfully recompiled the projects in the application, you can then deploy the application on Oracle Fusion Middleware 11g.

For more information about deploying applications, refer to the following resources:

- *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*, for information about deploying Java EE applications to Oracle WebLogic Server
- "Deploying Fusion Web Applications" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*, for information about deploying Oracle ADF applications

Considerations When Upgrading Oracle SOA Applications

This chapter provides important supplementary information upgrading Oracle SOA applications to Oracle Fusion Middleware 11g.

Use [Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications"](#) for the tasks required to upgrade any Oracle SOA Suite, WebCenter, and ADF application.

Use the following sections to understand tasks specific to upgrading Oracle SOA applications:

- [Section 9.1, "About Using the Oracle JDeveloper 11g Migration Wizard for Oracle SOA Suite Applications"](#)
- [Section 9.2, "Upgrade Tasks Associated with All Java Applications"](#)
- [Section 9.3, "Upgrade Tasks Associated with All Oracle SOA Suite Applications"](#)
- [Section 9.4, "Upgrade Tasks Associated with Oracle BPEL Process Manager Applications"](#)
- [Section 9.5, "Upgrade Tasks Associated with Oracle Enterprise Service Bus"](#)
- [Section 9.6, "Upgrade Tasks Associated with Oracle Business Activity Monitoring"](#)
- [Section 9.7, "Upgrade Tasks Associated with Technology Adapters"](#)
- [Section 9.8, "Upgrade Tasks Associated with Human Workflow Tasks"](#)
- [Section 9.9, "Upgrade Tasks Associated with Oracle Business Rules"](#)
- [Section 9.10, "Upgrading Oracle SOA Suite Client Applications"](#)
- [Section 9.11, "Upgrading Oracle Web Services Manager \(WSM\) Policies"](#)
- [Section 9.12, "Upgrading Oracle Containers for J2EE \(OC4J\) Security Environments"](#)

9.1 About Using the Oracle JDeveloper 11g Migration Wizard for Oracle SOA Suite Applications

When you open an Oracle Application Server 10g Oracle SOA Suite application in Oracle JDeveloper 11g, the Oracle JDeveloper Migration Wizard attempts to upgrade your application automatically to Oracle Fusion Middleware 11g.

However, there are some limitations to what the Oracle JDeveloper Migration Wizard can perform automatically. Refer to specific sections of this chapter for information

about the types of manual tasks you might have to perform on your Oracle SOA Suite applications before or after using the Migration Wizard.

Applying the Latest Patch Sets

For best results, Oracle recommends that you apply the most recent patch sets to your Oracle SOA Suite environment and that you use the latest 10g Release 3 (10.1.3) Oracle JDeveloper before upgrading to 11g.

Keeping Oracle JDeveloper and Oracle SOA Suite at the Same Version Level

As a general rule, you should always update your Oracle SOA Suite and Oracle JDeveloper installations at the same time and run the same version of both these Oracle products.

Verifying That You Have the Required SOA Composite Editor Oracle JDeveloper Extension

To upgrade your Oracle SOA Suite applications to 11g, you must have the Oracle SOA Composite Editor extension for Oracle JDeveloper 11g.

To verify that the extension is installed, select **About** from the Oracle JDeveloper **Help** menu, and click the **Version** tab. You should see an entry in the list of components called SOA Composite Editor.

If this component does not appear on the Version tab of the About dialog box, close the About dialog and select **Check for Updates** from the **Help** menu. Use the Check for Updates wizard to locate and install the latest version of the SOA Composite Editor extension.

9.2 Upgrade Tasks Associated with All Java Applications

Before you begin upgrading your Oracle SOA Suite applications, be sure to review the *Oracle Fusion Middleware Upgrade Guide for Java EE*, which contains information about upgrading standard Java EE applications to Oracle WebLogic Server.

If your applications contain any custom Java code, you should review the Java code against the procedures and recommendations available in the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

9.3 Upgrade Tasks Associated with All Oracle SOA Suite Applications

The following information should be reviewed when you are upgrading any Oracle SOA Suite application to Oracle Fusion Middleware 11g:

- [Understanding Oracle SOA Suite API Changes for Oracle Fusion Middleware 11g](#)
- [Reviewing Your Projects for Dependent JAR Files](#)
- [Upgrading Applications That Require Proxy Settings for Web Services](#)
- [Recreating build.xml and build.properties Files Not Upgraded by the Migration Wizard](#)
- [Upgrading Projects That Use UDDI-Registered Resources](#)
- [Using the Oracle SOA Suite Command-Line Upgrade Tool](#)

9.3.1 Understanding Oracle SOA Suite API Changes for Oracle Fusion Middleware 11g

Table 9–1 describes the APIs you can use in an Oracle SOA Suite application. For each Oracle Application Server 10g API, it provides a summary of the changes for Oracle Fusion Middleware 11g and where you can get more information about upgrading your applications that use the API.

Table 9–1 Summary of Oracle SOA Suite API Changes for Oracle Fusion Middleware 11g

| 10g API | Action for 11g | 11g JavaDoc |
|---|---|--|
| Oracle Application Server Integration B2B Callouts Java API | Updated for Oracle Fusion Middleware 11g, but the new 11g version is backward compatible with 10g. | <i>Oracle Fusion Middleware Callout Java API Reference</i> |
| Oracle Application Server Integration B2B Instance Message Java API | Updated for Oracle Fusion Middleware 11g, but the new 11g version is backward compatible with 10g. | <i>Oracle Fusion Middleware Instance Data Access Java API Reference</i> |
| Oracle BPEL Process Manager Workflow Services API | Updated for Oracle Fusion Middleware 11g, but the new 11g version is backward compatible with 10g. | <i>Oracle Fusion Middleware Workflow Services Java API Reference for Oracle BPEL Process Manager</i> |
| Oracle Business Rules Java API | See Section 9.3.1.1, "Upgrading to the Oracle Fusion Middleware Java API for Oracle Business Rules" | <i>Oracle Fusion Middleware Java API Reference for Oracle Business Rules</i> |
| Oracle BPEL Process Manager Client Java API | See Section 9.3.1.2, "Upgrading to the Oracle Fusion Middleware Infrastructure Management Java API for Oracle SOA Suite" | See Section 9.3.1.2, "Upgrading to the Oracle Fusion Middleware Infrastructure Management Java API for Oracle SOA Suite" |
| Oracle BPEL Process Manager Sensor API | Updated for Oracle Fusion Middleware 11g, but the new 11g version is backward compatible with 10g and part of the new Infrastructure Management Java API. | "Using Oracle BPEL Process Manager Sensors" in the <i>Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite</i> |

9.3.1.1 Upgrading to the Oracle Fusion Middleware Java API for Oracle Business Rules

The following sections introduce the Oracle Business Rules 11g SDK and API and provide instructions for upgrading to the Oracle Business Rules API:

- [Overview of the Oracle Business Rules SDK and API Changes for 11g](#)
- [Accessing a Dictionary in the Development Environment](#)
- [Accessing a Repository in a Production Environment](#)
- [Generating RL Code](#)

9.3.1.1.1 Overview of the Oracle Business Rules SDK and API Changes for 11g In Oracle Fusion Middleware 11g, the Oracle Business Rules SDK and API has been significantly improved.

In Oracle Application Server 10g, developers were required to manually manage accessing the repository and creating and using `RuleSession` instances. In Oracle Fusion Middleware 11g, the Decision Point API and decision function features provide an interface and implementation that simplifies the definition and execution of rules.

When upgrading to Oracle Business Rules 11g, look first at how you can use these new features, as documented in the *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.

However, if you want to continue to use the Oracle Business Rules SDK in the same way as you did for 11g, then the following sections describe how to directly translate the Oracle Business Rules 10g SDK to the 11g SDK.

All of the classes discussed in this section are in the `rulesdk2.jar` file and under the `oracle.rules.sdk2` package.

9.3.1.1.2 Accessing a Dictionary in the Development Environment In Oracle Business Rules 10g, it was common for an application developer to use a file-based repository while developing the application, and then switch to using a WebDAV repository for production.

In Oracle Business Rules 11g, you can instead access the rule dictionary file directly before it is packaged into a format which can be deployed to MDS.

In the following examples, compare the code required to access a dictionary in development mode in 10g ([Example 9-1](#)) with the code required in 11g ([Example 9-2](#)).

Note that in general, you should use the Decision Point API rather than continuing to access the `RuleRepository` directly.

Example 9-1 Accessing a Dictionary with Oracle Business Rules 10g in a Development Environment

```
String path; // the path to the file repository
Locale locale; // the desired Locale
// The following code assumes that the path and locale have been set appropriately
RepositoryType rt =
RepositoryManager.getRegisteredRepositoryType("oracle.rules.sdk.store.jar");
RuleRepository repos = RepositoryManager.createRuleRepositoryInstance(rt);
RepositoryContext rc = new RepositoryContext();
rc.setLocale(locale);
rc.setProperty("oracle.rules.sdk.store.jar.path", path);
repos.init(rc);
```

Example 9-2 Accessing a Dictionary with Oracle Business Rules 11g in a Development Environment

```
protected static final String DICT_LOCATION = "C:\\scratch\\CarRental.rules";
...
RuleDictionary dict = null;
Reader reader = null;
try {
    reader = new FileReader(new File(DICT_LOCATION));
    dict = RuleDictionary.readDictionary(reader, new
        DecisionPointDictionaryFinder(null));
    List<SDKWarning> warnings = new ArrayList<SDKWarning>();
    dict.update(warnings);
    if (warnings.size() > 0 ) {
        System.err.println("Validation warnings: " + warnings);
    }
} catch (SDKException e){
    System.err.println(e);
} catch (FileNotFoundException e){
    System.err.println(e);
} catch (IOException e){
    System.err.println(e);
```

```

    } finally {
        if (reader != null) {
            try { reader.close(); } catch (IOException ioe)
                {ioe.printStackTrace();}
        }
    }
}

```

9.3.1.1.3 Accessing a Repository in a Production Environment In Oracle Business Rules 10g, WebDAV was the recommended production repository.

In Oracle Business Rules 11g, WebDAV is no longer supported and Metadata Services (MDS) is the recommended repository. Also, the dictionary "name" and "version" have been replaced with a "package" and "name" (similarly to the Java class naming scheme).

In Oracle Business Rules 10g, the version did not provide true versioning. In Oracle Business Rules 11g, the equivalent to specifying a version is to simply change the name. For example, a 10g dictionary with the name `foo.bar.MyDict` and version 2 would in 11g be packaged as `foo.bar` and name `MyDict2`.

In the following examples, compare the code required to access a dictionary in production mode in 10g ([Example 9–3](#)) with the code required in 11g ([Example 9–4](#)).

Example 9–3 Accessing a Dictionary with Oracle Business Rules 10g in a Production Environment

```

String url; // the URL for the WebDAV repository
Locale locale; // the desired Locale
// The following code assumes that the url and locale have been set appropriately
RepositoryType rt =
    RepositoryManager.getRegisteredRepositoryType("oracle.rules.sdk.store.webdav");
RuleRepository repos = RepositoryManager.createRuleRepositoryInstance(rt);
RepositoryContext rc = new RepositoryContext();
rc.setLocale(locale);
rc.setProperty("oracle.rules.sdk.store.webdav.url", url);
repos.init(rc);
RuleDictionary dictionaryWithInitialVersion =
    repos.loadDictionary(dictionaryName);
RuleDictionary dictionarySpecificVersion = repos.loadDictionary(dictionaryName,
    dictionaryVersion);

```

Example 9–4 Accessing a Dictionary with Oracle Business Rules 11g in a Production Environment

```

import static
    oracle.rules.sdk2.repository.RepositoryManager.createRuleRepositoryInstance;
import static
    oracle.rules.sdk2.repository.RepositoryManager.getRegisteredRepositoryType;
import static oracle.rules.sdk2.store.mds.Keys.CONNECTION;
...
private static final String DICT_PKG = "oracle.middleware.rules.demo";
private static final String DICT_NAME = "CarRental";
private static final DictionaryFQN DICT_FQN =
    new DictionaryFQN(DICT_PKG, DICT_NAME);
...
RuleRepository repo =
    createRuleRepositoryInstance(getRegisteredRepositoryType(CONNECTION));
repo.init(new RepositoryContext() {{
    setDictionaryFinder(new DecisionPointDictionaryFinder(null));
}});
RuleDictionary dict = repo.load(DICT_FQN);

```

9.3.1.1.4 Generating RL Code The following example shows the code required generate RL code in Oracle Business Rules 10g. In 11g, this code remains the same.

```
//init a rule session
String rsname = "vehicleRent";
String dmrl = dict.dataModelRL();
String rsrl = dict.ruleSetRL( rsname );
```

9.3.1.2 Upgrading to the Oracle Fusion Middleware Infrastructure Management Java API for Oracle SOA Suite

Oracle SOA Suite 11g introduces a new infrastructure management API that replaces several other APIs previously available in Oracle SOA Suite 10g.

For more information about the new API, refer to the *Oracle Fusion Middleware Release Notes*, which are available as part of the Oracle Fusion Middleware 11g documentation library.

9.3.2 Reviewing Your Projects for Dependent JAR Files

If you are upgrading an Oracle SOA Suite application that depends upon references to custom JAR file libraries, note that these references may not get upgraded automatically when you open and upgrade your application in Oracle JDeveloper 11g.

As a result, you should review your projects for these types of dependencies and, after the upgrade, make sure that you add any missing references, by selecting the **Libraries and Classpath** link in the Oracle JDeveloper 11g Project Properties dialog.

9.3.3 Upgrading Applications That Require Proxy Settings for Web Services

If you are upgrading an application that uses Web services resources outside your company firewall, you must modify a configuration file that will enable the upgrade to access proxy settings and adjust them accordingly during the upgrade of the application.

To configure Oracle JDeveloper 11g to use the proxies during the upgrade process:

1. Locate the following file in the `JDEV_HOME/bin` directory:

```
sca-upgrade.xml
```

2. Edit the file and modify the following settings to identify the proxy server and port required to resolve the Web services addresses for the applications you are upgrading.

```
<jvmarg value="-Dhttp.proxySet=true"/>
<jvmarg value="-Dhttp.proxyHost=PROXY_SERVER"/>
<jvmarg value="-Dhttp.proxyPort=PROXY_PORT"/>
```

Note that there are two locations in the file to modify these settings. One location is used to set the proxy when you are upgrading an ESB project to Oracle Mediator 11g; the other is used when you are upgrading Oracle BPEL Process Manager projects.

3. Stop and start Oracle JDeveloper 11g so your changes can take effect and then open and upgrade your application in Oracle JDeveloper 11g.

9.3.4 Recreating build.xml and build.properties Files Not Upgraded by the Migration Wizard

When you open and upgrade an application in Oracle JDeveloper 11g, the `build.xml` and `build.properties` files associated with your application projects are not upgraded.

Instead, you must recreate these files in Oracle JDeveloper 11g after the upgrade.

9.3.5 Upgrading Projects That Use UDDI-Registered Resources

The following information is important if any of your Oracle BPEL Process Manager or Oracle Enterprise Service Bus 10g projects use remote resources that are registered in a UDDI (Universal Description, Discover and Integration) registry, such as Oracle Service Registry (OSR).

Refer to the following for more information:

- [Verifying that serviceKey Endpoints Are Available Before Upgrade](#)
- [Changing to the orauddi Protocol](#)

9.3.5.1 Verifying that serviceKey Endpoints Are Available Before Upgrade

If you have a 10g Release 3 (10.1.3) project that references an endpoint URL that uses a `serviceKey` from OSR, then you must be sure that the endpoint URL is up and available when you upgrade the application. Otherwise, the upgrade of the application will fail. To prevent such a failure, verify that the endpoint URLs are available, and if necessary, modify the endpoint URLs in `bpel.xml` or the routing service to point to new URL which is accessible.

9.3.5.2 Changing to the orauddi Protocol

In Oracle Application Server 10g Release 3 (10.1.3), Oracle BPEL Process Manager and Oracle Enterprise Service Bus projects used the `uddi` protocol to obtain resource references from OSR. In Oracle Fusion Middleware 11g, Oracle BPEL Process Manager and Oracle Mediator projects use the `orauddi` protocol.

As a result, prior to upgrading your Oracle BPEL Process Manager or Oracle Enterprise Service Bus projects, you must do the following:

1. Use the service registry to modify the registered service so it uses the new bindings supported by Oracle Fusion Middleware 11g.

For example, in OSR, do the following:

- a. Log in to your Oracle Service Registry.
- b. On the Search tab, click **Businesses**.
- c. Click **Add Name** to search for a business by name.
- d. From the search results, click the name of the business you want to modify.
- e. In the left pane of the View Business page, right-click the service you want to modify and select **Add Binding** from the context menu.
- f. From the **Type** drop-down menu on the Add Binding page, select **wsdlDeployment**.
- g. In the **Access Point** field, enter the URL.

For example:

compile, package, and deploy), or you can choose to move to Oracle JDeveloper, or you use both tools, with no functionality loss.

However, it is important to note that the command-line tool upgrades SOA project artifacts only. Other Oracle JDeveloper artifacts (for example, the `.jpr` and `.jws` files) are ignored.

To work around this restriction, note the following:

- The Oracle SOA Suite Command-Line Upgrade Tool copies files from the BPEL suitcase directory (the BPEL subdirectory or the directory hosting the BPEL files) to the specified target directory, specified on the command line.
- The above copying action does not copy the `.jpr` or `.jws` files. After the upgrade, the target directory contains only the upgraded SOA project contents.
- To remedy this problem in Oracle JDeveloper, you can create a new application or new project, and then define the project directory to be the newly upgraded composite directory.

9.3.6.3 Limitations When Upgrading Human Workflow Applications with the Oracle SOA Suite Command-Line Upgrade Tool

If you attempt to use the Oracle SOA Suite Command-Line Upgrade Tool to upgrade an application that contains Oracle Human Workflow projects, note that the tool will create a separate project for each upgraded task form.

This resulting project is an Oracle ADF project and Oracle ADF does not support command line deployment. As a result, after using the Oracle SOA Suite Command-Line Upgrade Tool, you must open the upgraded projects in Oracle JDeveloper 11g and deploy it from JDeveloper.

For information about how to open the upgraded project in Oracle JDeveloper, see [Chapter 9.3.6.2, "Using the Oracle SOA Suite Command-Line Upgrade Tool with Oracle JDeveloper 11g"](#).

9.3.6.4 Upgrading BPEL Projects with the Oracle SOA Suite Command-Line Upgrade Tool

The files required to run the Oracle SOA Suite Command-Line Upgrade Tool are installed automatically when you install Oracle JDeveloper 11g and when you install Oracle SOA Suite 11g:

- When you install the Oracle SOA Suite 11g, the required XML files are installed in the `bin` directory of the Oracle SOA Suite Oracle home.
- When you install Oracle JDeveloper 11g, the required XML files are installed in the `bin` directory of the Oracle JDeveloper home.

You can use these files in this directory, together with Apache Ant, to migrate your 10g Release 3 (10.1.3) SOA projects to 11g.

Note: For the purposes of this procedure, it is assumed you are running the Oracle SOA Suite Command-Line Upgrade Tool from the Oracle home of an Oracle SOA Suite installation.

If you are running it from a Oracle JDeveloper home, replace `ORACLE_HOME` with `JDEV_HOME` in the following procedures.

To use the ANT project:

1. Set the required environment variables, by running the following script:

On UNIX systems:

```
ORACLE_HOME/bin/soaversion.sh
```

On Windows systems:

```
ORACLE_HOME\bin\soaversion.bat
```

2. Run the Ant project, as follows:

```
ant -f ORACLE_HOME/bin/ant-sca-upgrade.xml
    -Dsource sourceDir
    -Dtarget targetDir
    -DappName app_name
```

When you run the Ant command, note the following:

- Your BPEL files from the source directory are migrated to the target directory you specified on the command line. The migration tool creates an application subdirectory using the value you entered as the *appName* value on the command line.
- The *sourceDir* you specified on the command line can be the `bpel` folder, the parent directory that hosts the `bpel` folder, or the Oracle Enterprise Service Bus project folder.

Note that the source directory is backed up before migration; if you must re-run the upgrade, be sure to restore the source files using the backup that was created before the invoking the tool. You can identify the backup directory by the ".backup" suffix in its name.

- The *sourceDir* and *targetDir* parameters must be fully-qualified path names.
- After upgrade, the logs can be found in the following output directory:

```
ORACLE_HOME/upgrade/logs
```

- After upgrade, you must run the SCA-Compiler (`ant-sca-compile.xml`) which will verify the migrated sources. Because the SCA-Upgrade will not generate complete artifacts in all cases, you will see some errors or warnings from SCA-Compiler with pointers on how to fix them. Check the SCA Compiler for reference.
- After getting a clean pass from the SCA-Compiler you can use the `ant-sca-package.xml` tool to package your application. You can then deploy the application using `ant-sca-deploy.xml`. After deployment, your project will be available to the server for testing.

Note, however, that in most cases, you will likely want to open the upgraded project in Oracle JDeveloper 11g. From there, you can easily review, verify, and make any necessary updates to the application projects.

For more information, see [Section 9.3.6.2, "Using the Oracle SOA Suite Command-Line Upgrade Tool with Oracle JDeveloper 11g"](#).

9.3.6.5 Combining Multiple BPEL Projects Into a Single Composite with the Oracle SOA Suite Command-Line Upgrade Tool

Using the Oracle SOA Suite Command-Line Upgrade Tool, you can merge BPEL projects together and create a single composite. This is not possible when you use the Migration Wizard in Oracle JDeveloper.

To combine multiple BPEL projects into a single composite, provide multiple source directories as part of the `-Dsource` property on the command line.

Path separators can be a colon (:) or a semicolon (;). Ant will convert the separator to platform's local conventions. As a guideline, also use double quotes to identify the multiple source directories to prevent Ant from parsing the input in an unexpected manner.

The first source directory specified will be considered as the root of the 11g project and will determine the composite name.

For example:

```
ant -f ORACLE_HOME/bin/ant-sca-upgrade.xml
    -Dsource "sourceDir1:sourceDir2"
    -Dtarget targetDir
    -DappName app_name
```

The first project in the source list is considered the root project and only those services are exposed as Composite Services. Anytime you use the merge feature, it is recommended that the projects be related.

Merging of projects is supported for BPEL projects only. ESB projects cannot be merged with other BPEL or other ESB projects.

9.3.6.6 Upgrading Oracle Enterprise Service Bus (ESB) Projects with the Oracle SOA Suite Command-Line Upgrade Tool

The Oracle SOA Suite Command-Line Upgrade Tool can also be used to upgrade ESB projects to Oracle Mediator 11g.

To upgrade an ESB project, use the instructions in [Section 9.3.6.4, "Upgrading BPEL Projects with the Oracle SOA Suite Command-Line Upgrade Tool"](#), but replace the Ant command with the following:

```
ant -f $ORACLE_HOME/bin/ant-sca-upgrade.xml -Dsource sourceDir -Dtarget targetDir
    -DappName app_name mediator
```

9.3.6.7 Upgrading Domain Value Maps (DVMs) and Cross References with the Oracle SOA Suite Command-Line Upgrade Tool

If you use domain value maps (DVMs) or Cross References in your Oracle BPEL Process Manager 10g or Oracle Enterprise Service Bus 10g projects, then note the following:

- The XPath functions you use to access the domain value maps or cross references are upgraded automatically to Oracle BPEL Process Manager and Oracle Mediator 10g when you open and upgrade your applications in Oracle JDeveloper 11g.
- However, you must perform a manual upgrade task to upgrade the domain value maps and cross references that are saved in the Oracle Enterprise Service Bus repository. The upgrade process moves the domain value maps from the ESB repository to the Oracle Fusion Middleware 11g Metadata Services (MDS) repository.

For more information, see "Managing the MDS Repository" in the *Oracle Fusion Middleware Administrator's Guide*.

To upgrade your 10g domain value maps in the ESB repository perform the following tasks.

Task 1 Export the domain value map metadata to a ZIP file

1. Change directory to the Oracle Enterprise Service Bus Oracle home.
2. Use the export script to export the metadata to a ZIP file.

For example, on UNIX systems:

```
ORACLE_HOME/export.sh metadata10g.zip
```

Task 2 Convert the ZIP file to an Oracle SOA Suite archive file

Use Apache Ant and the `upgrade-xrefdvm` target in the `sca-upgrade.xml` file to use the metadata ZIP file to generate an Oracle SOA Suite archive JAR file:

1. Change directory to the Oracle SOA Suite 11g Oracle home.
2. Use the following command to generate a SOA archive file, which will automatically be called `sca_XrefDvmFiles10g_rev1.0.jar`:

```
ant -f ant-sca-upgrade.xml upgrade-xrefdvm
-Dsource=location_of_the_zip_file
-Dtarget=location_of_the_soa_archive
```

For example:

```
ant -f ant-sca-upgrade.xml upgrade-xrefdvm
-Dsource=$ORACLE_HOME/temp/upgrade/metadata10g.zip
-Dtarget=$ORACLE_HOME/temp/upgrade
```

Task 3 Import the archive file into MDS repository

1. Start Oracle JDeveloper 11g and create a new application.
2. Import the Oracle SOA Suite archive into a new SOA project:
 - a. From the Oracle JDeveloper 11g **File** menu, select **Import**, then **SOA Archive into SOA Project**.
 - b. In the Create SOA Project from SOA Archive Dialog Box, click **Browse** and locate the `sca_XrefDvmFiles10g_rev1.0.jar` file that you created previously in this procedure.
 - c. Make sure that the **Project Name** and **Composite Name** in the Create SOA Project from SOA Archive dialog box have the same name, which must be `XrefDvmFiles10g`.
 - d. Click **OK** to create the new SOA project, which is called `XrefDvmFiles10g`.
The new project consists of an empty composite, along with the upgraded XRef and DVM files.
3. Create a metadata archive (MAR) file for the application, which includes the XRef and DVM metadata, and then deploy the MAR to the Oracle WebLogic Server domain.

For information about creating a MAR, see "Creating a Deployment Profile" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

After you upgrade the DVM and cross reference metadata to 11g, you can then use Oracle JDeveloper 11g to modify the entries in the `XrefDvmFiles10g` project as needed. Each time you make changes, you must then transfer them to the proper MDS repository using the MAR deployment process.

9.4 Upgrade Tasks Associated with Oracle BPEL Process Manager Applications

Use the following sections to learn about the upgrade tasks you will likely need to perform to upgrade your Oracle BPEL Process Manager applications and projects to Oracle Fusion Middleware 11g:

- [Manual Upgrade Steps After Migrating Oracle BPEL Process Manager Applications](#)
- [Additional Considerations for Oracle BPEL Process Manager Applications](#)

9.4.1 Manual Upgrade Steps After Migrating Oracle BPEL Process Manager Applications

The following sections describe some of the common upgrade tasks you will have to perform after you open your Oracle Application Server 10g Oracle SOA Suite application in Oracle JDeveloper 11g.

These tasks represent items that are not upgraded automatically by the Oracle JDeveloper Migration Wizard:

- [Providing Missing Binding Information for External References](#)
- [Correcting Problems With Oracle BPEL Process Manager Test Suites](#)
- [Using Oracle BPEL Process Manager Deployment Plans After Upgrade](#)
- [Upgrading Fault Policies in an Oracle BPEL Process Manager Project](#)
- [Upgrading a 10g Project With No BPEL Folder](#)

9.4.1.1 Providing Missing Binding Information for External References

This section describes how to fix compiler warning messages about missing bindings after you upgrade an Oracle BPEL Process Manager project in Oracle JDeveloper.

Why the Warning Messages Occur When a source project in your application depends on Oracle Application Server 10g Release 3 (10.1.3) Oracle BPEL Process Manager or Oracle Enterprise Service Bus (ESB) services, then as a pre-cursor to migration, the Oracle JDeveloper Migration Wizard creates local copies of their abstract WSDLs.

The abstract WSDLs, by definition, lack service and port endpoint information. During upgrade, when the Migration Wizard creates external composite references for the service dependencies, the references are created as abstract references with no binding information.

These abstract references are flagged with WARNING messages in the upgrade log. After upgrade, when you attempt to compile the projects, Oracle JDeveloper will generate compilation errors alerting you to the presence of abstract references in the composite.

Note that a similar message also appears when you are using Oracle SOA Suite 10g Web services that use HTTP bindings, which are not supported in Oracle SOA Suite 11g.

Using 10g Release 3 (10.1.3) or 11g Dependency Services Based on the upgrade and deployment schedules of all the services in the dependency tree, you can choose to continue to use 10g Release 3 (10.1.3) dependency services or proceed with the upgrade and redeployment of the dependency service.

In general, Oracle recommends that users study their source project and their dependencies before any upgrade work. This analysis will result in a smoother upgrade experience.

How to Correct Binding Errors Discovered During the Recompile To correct binding errors due to the abstract references, right-click on the reference node in the Oracle JDeveloper 11g composite editor and choose the correct concrete WSDL. If you want to continue using a 10g Release 3 (10.1.3) dependency service, you can do so. Some of the dependency service endpoint URLs that are captured in the upgrade log and can be used for this step.

You can also redesign the dependency in Oracle JDeveloper 11g, using the WSIL Browser (Resource Palette), and discover the service from the UI in the composite modeler.

For more information, refer to the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.1.2 Correcting Problems With Oracle BPEL Process Manager Test Suites

During upgrade in Oracle JDeveloper, BPEL test suites are converted to composite test suites. However, only the "instance initiate" action is migrated. This results in a composite test which initiates only a test run. The rest of any BPEL-based test actions are not automatically migrated and must be manually set up after the upgrade.

To set up the tests in your 11g environment, refer to "Testing SOA Composite Applications" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.1.3 Using Oracle BPEL Process Manager Deployment Plans After Upgrade

Deployment and configuration plans that you set up in Oracle Application Server 10g for your Oracle BPEL Process Manager user projects are not upgraded in Oracle JDeveloper 11g.

In Oracle Fusion Middleware 11g, a composite can be accompanied by deployment and configuration plan during deployment. However, you must create the 11g deployment plan manually, after you have migrated the application in Oracle JDeveloper.

For more information about creating configuration plans in 11g, see "Moving SOA Composite Applications to and from Development, Test, and Production Environments" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.1.4 Upgrading Fault Policies in an Oracle BPEL Process Manager Project

When you open your 10g application in Oracle JDeveloper and use the Oracle JDeveloper Migration Wizard to upgrade the application, any fault policies and bindings that you set up in a user project are not automatically upgraded to 11g.

The reasons are as follows:

- In Oracle BPEL Process Manager 10g, fault policies were stored with the server and bindings were either stored with the server or specified in `bpel.xml`. In Oracle BPEL Process Manager 11g, fault policies and fault bindings are stored in the Oracle JDeveloper 11g project.
- In general, fault policies and bindings in Oracle BPEL Process Manager 11g are different than those in previous releases. For example, the file name and syntax is different, fault policies apply both to Oracle Mediator and Oracle BPEL Process Manager, and bindings are at a reference, component, and composite level in 11g.

For these reasons, you must manually recreate the fault policies and bindings in the Oracle JDeveloper 11g project after you upgrade your applications.

Similarly, when you upgrade an Oracle Enterprise Service Bus 10g Release 3 (10.1.3) project to 11g, you must add the retry parameters defined in `esb_config.ini` file to the 11g `fault-policy.xml` file.

For more information, see:

- "Using Fault Handling in a BPEL Process" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- "Using Mediator Error Handling" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.1.5 Upgrading a 10g Project With No BPEL Folder

If the 10g Release 3 (10.1.3) application you are upgrading includes a project where all the project artifacts are in one folder and the BPEL artifacts are not separated into their own folder in the project, then the Oracle JDeveloper Migration Wizard will not be able to upgrade those artifacts.

To avoid this problem, do one of the following:

- Before upgrading the project, use Oracle JDeveloper 10g to create a BPEL folder and move all the artifacts (except the `build.xml` and `build.properties` files) into that folder. Then you can proceed with upgrading the project.
- Use the Oracle SOA Suite command-line upgrade tool. The command-line tool will upgrade the artifacts even if a BPEL folder does not exist in the project. For more information, see [Section 9.3.6, "Using the Oracle SOA Suite Command-Line Upgrade Tool"](#).

9.4.2 Additional Considerations for Oracle BPEL Process Manager Applications

The following sections describe additional considerations you should review after upgrading Oracle BPEL Process Manager applications:

- [Verifying New and Deprecated Properties in the `bpel.xml` Deployment Descriptor](#)
- [Upgrading User-Defined \(Custom\) XPath Functions in an Oracle BPEL Process Manager Project](#)
- [Change in Support for Multiple BPEL Implementations](#)
- [Verifying the Properties File Name When Upgrading Projects That Use Correlation Sets in an Asynchronous Service](#).
- [Upgrading Projects With the `transaction=participate` Property](#)
- [Specifying Domain Descriptor Properties in Oracle BPEL Process Manager 11g](#)

9.4.2.1 Verifying New and Deprecated Properties in the `bpel.xml` Deployment Descriptor

If any of your Oracle Application Server 10g Release 3 (10.1.3) applications reference any properties that were stored in the `bpel.xml` deployment descriptor, note that these properties are now set in the `composite.xml` deployment descriptor file.

For more information about the properties available in the `composite.xml` deployment descriptor, as well as important information about properties that are no longer supported in Oracle Fusion Middleware 11g, refer to "Deployment Descriptor Properties" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.2.2 Upgrading User-Defined (Custom) XPath Functions in an Oracle BPEL Process Manager Project

When you open and upgrade your applications in Oracle JDeveloper 11g, any custom XPath functions in the application projects are not upgraded automatically.

As a result, after you upgrade your application, you must copy any XPath function classes into the server classpath and register the function in the server configuration file. This is a manual step because the Oracle JDeveloper Migration Wizard cannot assume the server information.

For more information, refer to "Creating User-Defined XPath Extension Functions" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.4.2.3 Change in Support for Multiple BPEL Implementations

You cannot upgrade a single BPEL file with multiple implementations in Oracle BPEL Process Manager 11g.

To migrate and merge multiple 10g Release 3 (10.1.3) projects with a single BPEL implementation file, you must rename the BPEL file in each project. In the `bpel.xml` file in each project, the `bpel` filename and deployment ID name must be renamed the same name, as well.

Before you upgrade the application, review the `bpel.xml` file and `*.bpel` file. Make sure that the BPEL file name and the process name are the same; then review the `bpel.xml` file and verify that the BPEL file name is the same as the deployment ID name.

9.4.2.4 Verifying the Properties File Name When Upgrading Projects That Use Correlation Sets in an Asynchronous Service.

If you created an Oracle BPEL Process Manager 10g project that used correlation sets in an asynchronous service, as described in the Oracle BPEL Process Manager 10g Release 3 (10.1.3) Developer's Guide, make sure that the correlation set properties file uses the following proper naming convention:

```
BPEL_FILE_NAME_Properties.wsdl
```

If this file naming convention is not followed in the Oracle BPEL Process Manager 10g project, the properties will not be upgraded correctly. If you need to change the name to meet the file naming conventions, then also update the `wsdl` file that imports the property file name, accordingly.

9.4.2.5 Upgrading Projects With the `transaction=participate` Property

If you open and upgrade an Oracle BPEL Process Manager 10g project that has a configuration property, `transaction=participate`, then that property will not be migrated to 11g when you open the application in Oracle JDeveloper 11g. This configuration property is no longer supported in Oracle Fusion Middleware 11g.

If you open and upgrade an Oracle BPEL Process Manager 10g project has a `partnerLink` binding property, `transaction=participate`, then that property will be dropped on the caller side. The callee project instead should have the property setting `bpel.config.transaction=required`. This caller to callee setting propagation is not performed by upgrade and must be performed manually after you upgrade the application and before you attempt to deploy it on Oracle Fusion Middleware 11g.

9.4.2.6 Specifying Domain Descriptor Properties in Oracle BPEL Process Manager 11g

In Oracle BPEL Process Manager 10g, you specify domain properties in the `domain.xml` descriptor file. In Oracle BPEL Process Manager 11g, you specify Oracle BPEL Process Manager engine properties in `bpel-config.xml` and `soa-infra-config.xml` (both are accessible as MBeans).

There is only one domain in Oracle BPEL Process Manager 11g, which means that these properties will affect the entire server.

Table 9–2 Domain Descriptor Properties in Oracle BPEL Process Manager 10g and 11g

| Oracle BPEL Process Manager 10g in <code>domain.xml</code> file | Oracle BPEL Process Manager 11g MBean Property | Description |
|---|--|--|
| <code>auditDetailThreshold</code> | <code>auditDetailThreshold</code> | The maximum size (in bytes) of an audit trail details string before it is stored separately from the audit trail. If a details string is larger than this threshold, then it will not be immediately loaded when the audit trail is initially retrieved; a link will be displayed with the size of the details string. Typically, the details string will contain the contents of a BPEL variable. In cases where the variable is very large, performance may be severely impacted by logging it to the audit trail. The 10g default value is 50 kilobytes. |
| <code>auditLevel</code> | <code>auditLevel</code> | Controls the amount of audit events logged by a process. The supported logging levels are: <ul style="list-style-type: none"> ▪ off - absolutely no logging performed whatsoever; may result in a slight performance boost for processing instances. ▪ minimal - all events are logged; however, no audit details are logged. ▪ production - all events are logged. The audit details for <i>assign</i> activities are not logged; the details for all other nodes are logged. ▪ development - all events are logged; all audit details for all additives are logged. The 10g default value is "development". |
| <code>bpelcClasspath</code> | <code>bpelcClasspath</code> | Server-side BPEL process compiler classpath. Any user-specific classes or libraries used by a BPEL java exec node (that have not been packaged in the BPEL archive) need to be specified here so that the server-side BPEL process compiler can successfully compile the BPEL process. |
| <code>datasourceJndi</code> | <code>datasourceJndi</code> (Moved to <code>soa-infra-config.xml</code>) | The JNDI name for the domain data source. This data source may refer to any data source (JTA not required). |

Table 9–2 (Cont.) Domain Descriptor Properties in Oracle BPEL Process Manager 10g and 11g

| Oracle BPEL Process Manager 10g in domain.xml file | Oracle BPEL Process Manager 11g MBean Property | Description |
|--|--|--|
| deliveryPersistPolicy | deliveryPersistPolicy | <p>Changes whether the delivery messages are persisted. Delivery messages include invoke message, callback message and subscription message.</p> <ul style="list-style-type: none"> ▪ on - delivery messages are persisted ▪ off - delivery messages are kept only in memory. ▪ off.immediate - applicable for asynchronous processes only; messages are delivered synchronously regardless of the interface. The 10g default value is "on". |
| dspEngineThreads | dspEngineThreads | <p>The total number of threads that will be allocated to process engine dispatcher messages.</p> <p>Engine dispatch messages are generated whenever an activity must be processed asynchronously by the BPEL engine. If the majority of processed deployed on the BPEL server are durable with a large number of dehydration points (mid-process receive, onMessage, onAlarm, wait), greater performance may be achieved by increasing the number of engine threads.</p> <p>Note that higher threads counts may cause greater CPU utilization due to higher context switching costs. The 10g default value is 30 threads. Any value less than 1 thread will be changed to the default.</p> |
| dspInvokeAllocFactor | N/A | <p>The percentage of active threads that should be tasked to process incoming invocation messages. After a thread has finished processing a message it may be retasked to process an engine or invocation message, depending upon the current thread allocation situation.</p> <p>This property was replaced by dspInvokeThreads in 10g Release 3 (10.1.3.4). The 10.1.3.1.0 default value is 0.4 (40 percent).</p> |
| dspInvokeThreads | dspInvokeThreads | <p>The total number of threads that will be allocated to process engine dispatcher messages. Engine dispatch messages are generated whenever an activity must be processed asynchronously by the BPEL engine.</p> <p>If the majority of processed deployed on the BPEL server are durable with a large number of dehydration points (mid-process receive, onMessage, onAlarm, wait), greater performance may be achieved by increasing the number of engine threads. Note that higher threads counts may cause greater cpu utilization due to higher context switching costs. The 10g default value is 30 threads. Any value less than 1 thread will be changed to the default.</p> |
| dspMaxThreads | N/A | <p>The maximum number of active threads that will be processing messages during peak load times. This is the simplest way to throttle the performance of the domain. The maximum value is dependent upon the application server or OS configuration. This property was replaced by dsp(Engine Invoke System)Threads in 10.1.3.4.0. The 10.1.3.1.0 default value is 100 threads.</p> |

Table 9–2 (Cont.) Domain Descriptor Properties in Oracle BPEL Process Manager 10g and 11g

| Oracle BPEL Process Manager 10g in domain.xml file | Oracle BPEL Process Manager 11g MBean Property | Description |
|--|--|--|
| dspMinThreads | N/A | <p>The minimum number of active threads that will be processing messages during peak load times. If the current number of active threads is under this number the load factor is not taken into consideration when determining if a new thread should be allocated or not.</p> <p>This property was replaced by <code>dsp (Engine Invoke System) Threads</code> in 10.1.3.4.0. The 10.1.3.1.0 default value is 5 threads.</p> |
| dspSystemThreads | dspSystemThreads | <p>The total number of threads that will be allocated to process system dispatcher messages. System dispatch messages are general housecleaning tasks that are typically processed quickly by the server (for example, releasing stateful message beans back to the pool).</p> <p>Typically, only a small number of threads are required to handle the number of system dispatch messages generated during runtime. The 10g default value is 2 threads. Any value less than 1 thread will be changed to the default.</p> |
| expirationMaxRetry | expirationMaxRetry | <p>The maximum number of times a failed expiration call (<code>wait/onAlarm</code>) will be retried before failing. If the activity or instance the expiration call is targeting cannot be found, the call will be rescheduled again. The 10g default is 5.</p> |
| expirationRetryDelay | expirationRetryDelay | <p>The amount of time in seconds to wait before rescheduling a failed expiration call. The 10g default value is 120 seconds.</p> |
| instanceKeyBlockSize | instanceKeyBlockSize | <p>The size of the block of instance ids to allocate from the dehydration store during each fetch. Instance ids for instantiated instances are pre-allocated from the dehydration store and kept in-memory.</p> <p>Once all of the in-memory instance ids have been exhausted, the next request will retrieve the next block from the dehydration store. Generally, the higher the block size, the less impact fetching instance ids will have on the overall performance of the engine (as the cost of allocating the block is amortized over a larger number of instances).</p> <p>The only downside to a large block size is the gap between instances in a clustered installation (as each node will have its own in-memory block of instance ids) and wasting instance ids if the engine is shutdown and the block is not utilized fully. The 10g default value is 10000.</p> |
| largeDocumentThreshold | largeDocumentThreshold | <p>The maximum size (in bytes) a BPEL variable can be before it is stored in a separate location from the rest of the instance scope data.</p> <p>Large XML documents will severely impact the performance of the entire BPEL server if they are constantly read-in and written-out whenever processing on an instance must be performed.</p> <p>By writing the variable to separate storage the performance impact is limited to times when the variable is explicitly used in a BPEL activity. The 10g default value is 50 kilobytes.</p> |

Table 9–2 (Cont.) Domain Descriptor Properties in Oracle BPEL Process Manager 10g and 11g

| Oracle BPEL Process Manager 10g in domain.xml file | Oracle BPEL Process Manager 11g MBean Property | Description |
|--|---|---|
| minBPELWait | minBPELWait | The minimum amount of time allowed for a BPEL wait duration. Wait times shorter than this duration will not be respected and will essentially be "skipped" altogether. The 10g default value is 2 seconds. |
| processCheckSecs | N/A | <p>The number of seconds to wait since the last time the process container checked the bpel archive before checking it again.</p> <p>By "checking" we mean checking the last modified time stamp on the bpel archive for a particular process; if the specified number of seconds has passed and the bpel archive file has been modified since the last time we checked, the process will be refreshed from the new archive. If not enough time has passed since the last time the stale check was performed, the currently loaded process classes will be used.</p> <p>To disable process checking use a value of -1; in this case once a process has been loaded the container will never check if a newer version of the same process has been deployed. The 10g default value is 1 second. This property is not relevant in 11g anymore as the SOA infrastructure now handles deployment via HTTP only.</p> |
| statsLastN | statsLastN | <p>The size of the "most recently processed" request list. After each request is finished, statistics for the request are kept in a list.</p> <p>A value less than or equal to 0 will disable statistics gathering. NOTE: engine throughput may be affected slightly if this option is enabled. Please choose a reasonable value for n (1000 is the recommended value).</p> |
| syncMaxWaitTime | syncMaxWaitTime | The maximum time the process result receiver will wait for a result before returning. Results from asynchronous BPEL processes are retrieved synchronously via a receiver that will wait for a result from the container. The 10g default value is 45 seconds. |
| txDatasourceJndi | txDatasourceJndi (moved to soa-infra-config.xml) | The JNDI name for the domain transactional data source. This data source must be configured for JTA support. |
| uddiLocation | oracle.soa.uddi.registry.inquiryUrl (moved to soa-infra-config.xml) | The inquiry URL of the Uddi v3 compliant registry |
| uddiPassword | oracle.soa.uddi.registry.password (moved to soa-infra-config.xml) | Set if UDDI is secured, password |
| uddiUsername | oracle.soa.uddi.registry.username (moved to soa-infra-config.xml) | Set if UDDI is secured, username |
| validateXML | validateXML | If set to "true" the engine will apply schema validation for incoming and outgoing XML documents. The 10g default value "false". |

9.5 Upgrade Tasks Associated with Oracle Enterprise Service Bus

Refer to the following sections for information about upgrading Oracle Enterprise Service Bus 10g to Oracle Mediator 11g:

- [Upgrading Oracle Enterprise Service Bus Routing Rules That Are Not Exposed as Services](#)
- [Upgrading Oracle Enterprise Service Bus Domain Value Maps \(DVMs\) and Cross References](#)
- [Upgrading Oracle Enterprise Service Bus Projects with SOAP Headers to Oracle Mediator 11g](#)
- [Upgrading Projects with Multiple Routing Services That Use the Same Namespace](#)
- [Upgrading Routing Services to Use Asynchronous Routing](#)
- [Upgrading Filtering or Setting Message Headers in Oracle Enterprise Service Bus](#)

9.5.1 Upgrading Oracle Enterprise Service Bus Routing Rules That Are Not Exposed as Services

If you upgrade an Oracle Enterprise Service Bus 10g project that contains an ESB routing service that is not exposed as a Web service, then the routing service and related composites cannot be invoked after upgrade.

To resolve this issue, modify the routing service in Oracle Mediator 11g so it is exposed as a Web service. For more information, see "Creating Mediator Routing Rules" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Alternatively, you can ensure that all routing services in your Oracle Enterprise Service Bus 10g projects are exposed as Web services before you begin the upgrade process.

9.5.2 Upgrading Oracle Enterprise Service Bus Domain Value Maps (DVMs) and Cross References

If you use domain value maps in your and Oracle Enterprise Service Bus 10g projects, then note that while the XPath functions for DVMs and cross-references are upgraded automatically, you must perform a post-upgrade task to upgrade the DVM and cross reference metadata to 11g.

For more information, see [Section 9.3.6.7, "Upgrading Domain Value Maps \(DVMs\) and Cross References with the Oracle SOA Suite Command-Line Upgrade Tool"](#).

9.5.3 Upgrading Oracle Enterprise Service Bus Projects with SOAP Headers to Oracle Mediator 11g

If you used XSLT functions to manipulate SOAP headers in Enterprise Service Bus (ESB) 10g, then after you upgrade your application, those header manipulation will be modeled as assignments in Oracle Mediator 11g.

For complex header manipulations, you should manually verify the SOAP headers before deploying the upgraded Oracle Mediator 11g projects.

For more information, see "Creating Static Routing Rules" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.5.4 Upgrading Projects with Multiple Routing Services That Use the Same Namespace

Oracle Enterprise Service Bus 10g supports multiple routing services that have WSDLs that reference the same `targetNameSpace` within a single ESB project.

For example, at least one of the sample 10g applications available on the Oracle Technology Network (the `samples_102_responseForward` sample application) supported this scenario.

However, Oracle Mediator 11g no longer supports this configuration. During the upgrade of the application in Oracle JDeveloper 11g, whenever a second service is found to have the WSDL pointing to the same `targetNameSpace`, the namespace is modified by suffixing `_service-qname` to the namespace. In addition, a warning message is added to the log file.

For example, in the `Samples_102_ResponseForward` sample application, the `targetNameSpace` of the service `DispatchShipmentDetails` is:

```
@ http://oracle.com/esb/namespaces/Samples_102_ResponseForward
```

After the upgrade of the application, it appears as follows:

```
@ http://oracle.com/esb/namespaces/
   Samples_102_ResponseForward_Samples.102_ResponseForward.DispatchShipmentDetails
```

In this example, `Samples.102_ResponseForward.DispatchShipmentDetails` is the `qname` of service `DispatchShipmentDetails`.

9.5.5 Upgrading Routing Services to Use Asynchronous Routing

In Oracle Enterprise Service Bus 10g, there was no support for asynchronous routing services. Instead, when Oracle Enterprise Service Bus called an Oracle BPEL Process Manager asynchronous process, then a `OneWay` service was created.

If you created a project that uses such a `OneWay` service, then after you open and upgrade the application in Oracle JDeveloper 11g, you can remove the old routing service and recreate the routing service to use an Oracle Mediator 11g asynchronous routing instead of the `OneWay` invocation.

9.5.6 Upgrading Filtering or Setting Message Headers in Oracle Enterprise Service Bus

Oracle Enterprise Service Bus 10g supports filtering or setting message headers for standard and custom header properties for adapters, such as SOAP, JCA, JMS, and AQ adapters. Filtering in Oracle Enterprise Service Bus is done as part of the `<filterExpression>` tag of the routing rule.

When you open your Oracle Enterprise Service Bus projects in Oracle JDeveloper 11g, each routing rule is upgraded into case, and the filter expression is converted to condition. These changes can be found in the `mplan` file of the Routing Service.

For example, the following code snippet shows the filter expression in an Oracle Enterprise Service Bus 10g routing rule:

```
<filterExpression>{ehdr:getRequestHeader
    ('/fhdr:InboundFileHeaderType/fhdr:fileName') = 'Mobile.xml'};{ namespace
    fhdr=http://xmlns.oracle.com/pcbpel/adapter/file/ namespace
    ehdr=http://www.oracle.com/XSL/Transform/java
    /oracle.tip.esb.server.headers.ESBHeaderFunctions }
</filterExpression>
```

During upgrade, this example is converted to the equivalent condition of the case in the mplan file:

```
<condition language="xpath" xmlns:ehdr="http://www.oracle.com/XSL
/Transform/java/oracle.tip.esb.server.headers.ESBHeaderFunctions"
xmlns:fhdr="http://xmlns.oracle.com/pcbpel/adapter/file/">
  <expression>$in.property.jca.file.FileName = 'Mobile.xml'</expression>
</condition>
```

Setting message headers is accomplished as part of the transformation files and these are upgraded to assign statements.

The following code snippet shows an example of setting message headers in Oracle Enterprise Service Bus 10g:

```
<xsl:variable name="inputCountry" select="/impl:CustomerData/Country"/>
<select="ehdr:setOutboundHeader
(' /jhdr:JMSOutboundHeadersAndProperties
/jhdr:JMSOutboundProperties/jhdr:Property[position()=1]/@value', $inputCountry,
'jhdr=http://xmlns.oracle.com/pcbpel/adapter/jms/;')"/>
```

During upgrade, this example is converted to the following equivalent assign element in 11g:

```
<assign>
  <copy expression="$in.CustomerData/CustomerData/Country"
target="$out.CustomerData/jhdr:JMSOutboundHeadersAndProperties
/jhdr:JMSOutboundProperties/jhdr:Property[position()=1]/@value"/>
</assign>
```

Most of the standard header properties are upgraded to their equivalent 11g headers. Customized headers are upgraded using a best effort approach, but it is not possible to upgrade all the variations of customized headers.

As a result, you must manually upgrade any customized headers the equivalent condition or assign element and equivalent mplan artifacts.

For more information, see "Getting Started with Oracle Mediator" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6 Upgrade Tasks Associated with Oracle Business Activity Monitoring

Refer to the following sections for information on upgrade tasks associated with Oracle Business Activity Monitoring:

- [Migrating Oracle BAM Users and Other LDAP Information to 11g](#)
- [Verifying Oracle BAM Action Form Templates After Upgrade](#)
- [Upgrading Oracle BAM External Data Sources](#)
- [Missing Images in Oracle BAM Reports](#)
- [Using Oracle BAM Web Services in 11g](#)
- [Upgrading Oracle BAM Sensor Actions](#)
- [Upgrading Enterprise Link Plans to Oracle BAM 11g](#)

9.6.1 Migrating Oracle BAM Users and Other LDAP Information to 11g

If your Oracle BAM 10g environment takes advantage of an LDAP directory server to store your Oracle BAM user names or other Oracle BAM information, then you should

make sure that the LDAP directory you were using in Oracle Application Server 10g is associated with your new Oracle Fusion Middleware 11g environment.

In general, after upgrade of the Oracle BAM schema, any Oracle BAM 10g users will appear in the Oracle BAM Administrator as inactive if they don't exist in the 11g LDAP directory. However, you can use Oracle BAM Administrator software to remap 10g users to new Oracle BAM 11g users.

For more information, see [Section 6.8.1, "Configuring Oracle SOA Suite 11g to Use Identity Management Components"](#).

9.6.2 Verifying Oracle BAM Action Form Templates After Upgrade

After you upgrade from Oracle BAM 10g to Oracle Fusion Middleware 11g, some of the Action Form templates you created in Oracle BAM 10g may not function correctly.

As a result, after you perform the upgrade, you should review the templates and verify that they are functioning correctly.

Specifically, use the following instructions to search for typical issues that might need to be corrected after the upgrade to Oracle Fusion Middleware 11g:

See Also: "Creating Action Form Templates" in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*

1. Open the Oracle BAM Start Page.

See Also: "Starting the Oracle BAM Web Applications," in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*

2. Start Oracle BAM Architect.
3. Select **Data Objects** from the menu.
4. Expand the **System** folder and open the **Views** folder.
5. Select the **Action Form Templates** data object and click **Contents**.
6. Click **Edit Contents**.
7. Click **Edit** next to the first row.
8. Copy the contents of the **FormInput** column to a text editor and remove any white space following an element. (in other words replace "> " with ">").
9. Search for the mixed case "ReportServer" and change to lower case "reportserver" to fix the links to the style sheets.
10. Copy and paste the updated text back into the **FormInput** column.
11. Repeat these steps for each entry in the **FormInput** column that contains data.

9.6.3 Upgrading Oracle BAM External Data Sources

External data sources defined in your Oracle BAM 10g environment are not upgraded to Oracle BAM 11g.

As a result, if you use external data sources in your Oracle BAM 10g environment, you must manually recreate those data sources in Oracle BAM 11g. This is because Oracle Fusion Middleware 11g supports a new set of database drivers, which are used to connect to an external database.

To configure external data sources for Oracle BAM 11g, refer to "Creating External Data Sources" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6.4 Missing Images in Oracle BAM Reports

If you used any images in your Oracle BAM 10g reports, the images in any existing reports will not be upgraded to Oracle BAM 11g and will be missing from the reports in the upgraded environment.

However, in your new Oracle Fusion Middleware 11g environment, you can add images to any new or existing reports using Oracle BAM Active Studio 11g.

For more information, see "Setting Backgrounds for Reports" in the *Oracle Fusion Middleware User's Guide for Oracle Business Activity Monitoring*.

9.6.5 Using Oracle BAM Web Services in 11g

The following sections describe how Oracle BAM Web services support in Oracle BAM 10g is affected by upgrade to Oracle Fusion Middleware 11g:

- [Using Web Services Clients with Oracle BAM 11g](#)
- [Invoking Web Services from Oracle BAM Alerts](#)
- [Using the ManualRuleFire Web Service in Oracle BAM 11g](#)
- [New Support for Web Services in Oracle BAM 11g](#)

9.6.5.1 Using Web Services Clients with Oracle BAM 11g

If you have developed client applications that communicate via Web services with Oracle BAM 10g servers, then you can continue to use those clients after you upgrade to Oracle BAM 11g, with no changes to the client code.

This is possible because the `DataObjectOperations` Web service available in Oracle BAM 10g is still available in Oracle BAM 11g. The `DataObjectOperations` Web service has been renamed to `DataObjectOperations10131`, but the WSDL end point URL remains the same, so client processes can continue working with Oracle BAM 11g.

However, to ensure that your clients continue to work successfully with Oracle BAM 11g, Oracle recommends that you apply the latest patch set to any Oracle BPEL Process Manager 10g clients that are using Oracle BAM 11g environment.

Note: At the time of this document was published, the latest patch set available for Oracle SOA Suite 10g and Oracle JDeveloper was 10g Release 3 (10.1.3.5).

However, for more information on the latest patch sets required for Oracle Fusion Middleware 11g interoperability and upgrade, refer to the Oracle Fusion Middleware 11g Release Notes available in the Oracle Fusion Middleware documentation library on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/documentation/>

If you are using the latest version of Oracle JDeveloper 10g Release 3 (10.1.3), to create a connection to Oracle BAM 11g, the **Domain** field in the BAM connection wizard is completely ignored and can be left blank.

For more information about the DataObjectOperations10131 Web service in Oracle BAM 11g, refer to "Using Oracle BAM Web Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6.5.2 Invoking Web Services from Oracle BAM Alerts

In Oracle BAM 11g, you could configure alerts to invoke a Web service by invoking an external action. This method of using alerts to invoke a Web service is no longer supported in Oracle BAM 11g.

Instead, in Oracle BAM 11g, you can create an alert rule that calls the "Call a Web Service" action.

For more information, see "Creating Oracle BAM Alerts" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6.5.3 Using the ManualRuleFire Web Service in Oracle BAM 11g

If you used the ManualRuleFire Web service in Oracle BAM 10g, note that the endpoint URL for this Web service has changed in Oracle BAM 11g.

For more information, refer to "Using the ManualRuleFire Web Service" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6.5.4 New Support for Web Services in Oracle BAM 11g

For complete information about the new Web services features in Oracle BAM 11g, refer to "Using Oracle BAM Web Services" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.6.6 Upgrading Oracle BAM Sensor Actions

When you open and upgrade a Oracle BPEL Process Manager 10g Release 3 (10.1.3) project in Oracle JDeveloper 11g, any Oracle BAM sensor actions are migrated automatically to 11g.

Unlike the behavior in 11g, Oracle BAM sensor actions in Oracle Fusion Middleware 11g use the Oracle BAM connection Factory JNDI (as shown in the BAM Sensor Action dialog box) to connect to Oracle BAM server. They do not use the `bpel.xml` file as they did in 10g Release 3 (10.1.3).

The migrated Oracle BAM 11g sensor actions can be identified by the `eis/bam/soap` connection Factory JNDI name. The Oracle BAM connection specified in the 10g `bpel.xml` file is migrated to an Application Resource BAM connection with the name of `Migrated1`.

All connection properties are preserved "as is" except for PORTS, which are set to 9001.

Note that Oracle JDeveloper 11g cannot connect to Oracle BAM 10g Release 3 (10.1.3). When you open an application in Oracle JDeveloper 11g, it is assumed that you have already installed and upgraded to Oracle BAM 11g.

Similarly, the Oracle BPEL Process Manager 11g runtime cannot connect to Oracle BAM 10g Release 3 (10.1.3).

9.6.7 Upgrading Enterprise Link Plans to Oracle BAM 11g

Enterprise Link for Oracle Business Activity Monitoring 10g is no longer available in Oracle BAM 11g. Instead, Oracle BAM 11g supports integration with Oracle Data Integrator (ODI), which replaces Enterprise Link.

When you upgrade to Oracle BAM 11g, you have the following options, depending on how you use Enterprise Link in Oracle BAM 10g:

- If used Enterprise Link primarily for integrating JMS messages into Oracle BAM, then you should review the Direct EMS functionality available in Oracle BAM 11g. Direct EMS allows JMS messages to be sent directly to Oracle BAM without the need for adapters or ODI.

For more information, see "Creating Oracle BAM Enterprise Message Sources" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

- If your Oracle BAM 10g EnterpriseLink plans are more complex, then you must manually recreate them as ODI scenarios. Oracle BAM 11g supports all the Oracle BAM 10g transformations (for example, insert, update, delete, and so on) as ODI Knowledge Modules via the ODI integration with Oracle BAM 11g.

For more information, see "Using Oracle Data Integrator With Oracle BAM" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

9.7 Upgrade Tasks Associated with Technology Adapters

The following sections describe upgrade tasks you might need to perform when upgrading Technology Adapters to Oracle Fusion Middleware 11g:

- [Verifying Adapters with the Oracle JDeveloper 11g Adapter Configuration Wizard](#)
- [Upgrading Adapter Header Variables](#)
- [Upgrading Non-Managed Connection Information in the Adapter JCA File](#)
- [Description Lost After Upgrade of Technology Adapter](#)
- [Upgrading an Inbound Synchronous Request-Reply Use Case](#)
- [Upgrading Projects That Use Outbound Dequeue](#)
- [Upgrading AQ Adapter Headers](#)
- [Upgrading an AQ Outbound Request-Reply Scenario](#)
- [Upgrading Database Adapters Used for Stored Procedures](#)

9.7.1 Verifying Adapters with the Oracle JDeveloper 11g Adapter Configuration Wizard

After upgrading your Oracle SOA Suite 10g projects to Oracle SOA Suite 11g, Oracle recommends that you use the adapter configuration wizard to verify the upgrade of your technology adapters. This will ensure that all required fields in your 11g are populated and that the adapter connections to your new Oracle Fusion Middleware 11g environment are valid.

For more information, see the information the specific adapters you are using in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

9.7.2 Upgrading Adapter Header Variables

After you upgrade an application that uses adapter headers, you will likely receive an error after you open the application in Oracle JDeveloper 11g. The error will indicate that a variable or parameter was not recognized while it was parsing a header function.

This is caused by a change in the way adapter header variables are defined in Oracle Fusion Middleware 11g.

In Oracle Application Server 10g Release 3 (10.1.3), you create adapter header variables in a `wsdl` file that has a name such as the following, or a similar name for the direction and adapter type you are using:

```
fileAdapterOutboundHeader.wsdl
```

In Oracle Fusion Middleware 11g, adapter header variables are now properties that you set on the **Properties** tab of the Invoke dialog box in Oracle JDeveloper.

For example, the properties you can set for the JCA Adapter for Files/FTP are described in the section, "Inbound Headers" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

After you open your project in Oracle JDeveloper 11g, the `wsdl` header files you created for Oracle Application Server 10g will no longer be attached to the project. You must define each of the variables in the `wsdl` files as properties for each adapter type that uses header variables.

For more information, refer to the appropriate chapter for each adapter type in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

9.7.3 Upgrading Non-Managed Connection Information in the Adapter JCA File

In Oracle Application Server 10g, you can use non-managed connections in the JCA file to point to your development resources. For example, data source definitions can be specified using the `non-managed-connection` element in the JCA file.

In Oracle Application Server 10g, using non-managed connections was reserved for development only. In Oracle Fusion Middleware 11g, non-managed connections are not supported.

Ideally, before upgrading your Oracle Application Server 10g projects to Oracle Fusion Middleware 11g, you should already be using JCA files that include only JNDI references to connections that are defined properly in the `datasources.xml` or `oc4j-ra.xml` files.

If you attempt to upgrade applications that use non-managed connections for adapter connections, you must modify the projects after upgrade to use proper connections defined by using supported administration tools, such as the Oracle WebLogic Server Administration Console.

9.7.4 Description Lost After Upgrade of Technology Adapter

If you created an 10g Release 3 (10.1.3) technology adapter using the Adapter Configuration Wizard, and you entered a description for the adapter in the **Description** field of the wizard, the description will be lost when you open and migrate the project in Oracle JDeveloper 11g.

This is because the `documentation` element in removed from the WSDL file during the upgrade.

If the adapter is important, you can use the Oracle JDeveloper 11g Adapter Configuration Wizard to add the description after you have upgraded the project.

9.7.5 Upgrading an Inbound Synchronous Request-Reply Use Case

In Oracle SOA Suite 10g, if you used the MQSeries Adapter inbound synchronous request-reply use case, you will notice that there is only one message type, which is `REQUEST`.

In Oracle SOA Suite 11g, the MQAdapter supports two message types: REQUEST and NORMAL. The REQUEST MQ message type always has `replyToQueue` set on its header so there is no need for `FallbackQueueName` and `FallbackQueueManagerName`. On the other hand, a NORMAL Message may or may not have `FallbackQueueName` and `FallbackQueueManagerName`.

So, when a 10g inbound synchronous request-reply scenario is upgraded to 11g, then the Message Type remains as REQUEST type. Therefore, at runtime, there is no need for `FallbackQueueName` and `FallbackQueueManagerName`, even though these properties would reside in the `.jca` file. If you want to use these properties, you must set the message type to NORMAL.

For more information about the inbound synchronous request-reply use case, see "Oracle MQ Series Adapter Use Cases" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

9.7.6 Upgrading Projects That Use Outbound Dequeue

If you used the MQSeries Adapter outbound dequeue use case, which was a documented use case scenario in 10g Release 3 (10.1.3), then note that these types of projects cannot be upgraded by Oracle JDeveloper 11g.

Instead, you must recreate these types of projects after the upgrade to Oracle JDeveloper 11g. While recreating the project you must ensure that you set the value for `UITransmissionPrimitive` to `SynchronousDequeue`.

For more information, see "Outbound Dequeue Scenario" in the *Oracle Fusion Middleware User's Guide for Technology Adapters*.

9.7.7 Upgrading AQ Adapter Headers

If you are upgrading an application that uses the Oracle Application Server Adapter for Advanced Queuing (AQ adapter), then note that the AQ Adapter header is accessed differently in Oracle Fusion Middleware 11g.

Specifically, you must perform the following post-upgrade steps for AQ adapter headers after you upgrade to Oracle SOA Suite 11g. This procedure is required for both Oracle BPEL Process Manager and for Oracle Enterprise Service Bus projects that are upgraded to Oracle Mediator.

- After upgrading to Oracle Mediator, assign the "jca.aq.HeaderDocument" value from inbound to outbound in the `.mp1an` file of the Oracle Mediator file.
- After upgrading to Oracle BPEL Process Manager 11g:

1. Define a new variable 'vInboundHeaders' of type string.

For example:

```
<variable name="vInboundHeaders" type="xsd:string"/>
```

2. In the receive activity, make sure that the `jca.aq.HeaderDocument` property gets written to the above variable.

For example:

```
<bpelx:property name="jca.aq.HeaderDocument"
  variable="vInboundHeaders"/>
```

3. At the time of invocation, make sure that the "vInboundHeaders" value is copied back to the `jca.aq.HeaderDocument` property:

```

<invoke name="Invoke_1"
  inputVariable="Invoke_1_Enqueue_InputVariable"
  partnerLink="EnqueueClobPayload" portType="ns2:Enqueue_ptt"
  operation="Enqueue">
  <bpelx:inputProperty name="jca.aq.HeaderDocument"
    variable="vInboundHeaders"/>
</invoke>

```

9.7.8 Upgrading an AQ Outbound Request-Reply Scenario

In Oracle Fusion Middleware 11g, the procedure you use to create AQ outbound request-reply scenarios for Oracle BPEL Process Manager projects has changed.

As a result, when you upgrade an Oracle SOA Suite 10g application that uses this scenario, the WSDL file that is created during the upgrade does not contain the required binding information.

To correct this problem, do the following after you open and upgrade the application in Oracle JDeveloper 11g:

1. Add the JCA entry to the WSDL file created by the upgrade, as follows:

```

<?binding.jca CombinedRequestReply_aq.jca?>
<wsdl:definitions name="CombinedRequestReply" ... >
  .
  .
  .
</wsdl:definitions>

```

2. Add the binding entry in the `composite.xml` file:

```

<reference ui:wSDLLocation="CombinedRequestReply.wsdl"
  name="CombineRequestReply">
  <interface.wSDL interface="http://xmlns.oracle.com/pcbpel
    /adapter/aq/EnqueueRequest/#wsdl.interface(Enqueue_ptt) "
    callbackInterface="http://xmlns.oracle.com/pcbpel
    /adapter/aq/DequeueReply/#wsdl.interface(Dequeue_ptt) "
    xmlns:ns="http://xmlns.oracle.com/sca/1.0"/>
  <binding.jca config="CombinedRequestReply_aq.jca"/>
</reference>

```

3. Remove any unused WSDL references from the `.bpel` file.

9.7.9 Upgrading Database Adapters Used for Stored Procedures

As described in [Section 9.1, "About Using the Oracle JDeveloper 11g Migration Wizard for Oracle SOA Suite Applications"](#), Oracle recommends that you apply the latest patch sets to your Oracle SOA Suite environment and use the latest 10g Release 3 (10.1.3) Oracle JDeveloper before upgrading to 11g.

However, if you are upgrading from 10g Release 3 (10.1.3.3) or earlier, note the following if you are using a database adapter for stored procedures.

- In Oracle SOA Suite 10g Release 3 (10.1.3.3) and earlier, the XSD generator of the adapter generated an XSD that contained unqualified elements. You can identify unqualified elements by viewing the `InputParameters` root element in the XSD file. The unqualified element appears as follows:

```

<db:InputParameters ... ></db:InputParameters>

```

The `db:` namespace is specified in the definition of the schema, which is the same as the target namespace.

- In 10g Release 3 (10.1.3.4) and later, the XSD generator generates qualified elements, which appear as follows in the `InputParameters` root element:

```
<InputParameters ..>/InputParameters>
```

Note that the `db:` namespace has been removed.

- If you were using a version of Oracle JDeveloper 10g Release 3 (10.1.3) that was incompatible with the Oracle SOA Suite version, you could also encounter a scenario where the elements are removed:

```
<null>(some value)</null>
```

The solution is as follows:

- First, make sure that you are using a version of Oracle JDeveloper that matches the version of your Oracle SOA Suite environment.
- Second, after you upgrade to Oracle JDeveloper and Oracle SOA Suite 10g Release 3 (10.1.3.4) or to Oracle SOA Suite and Oracle JDeveloper 11g, regenerate the XSD.

After you upgrade and regenerate the XSD, you must then modify any related assign activities and any XPath expressions accordingly.

9.8 Upgrade Tasks Associated with Human Workflow Tasks

The following upgrade tasks should be considered when you are upgrading Oracle SOA Suite applications that use human workflow tasks:

- [Upgrading Task Details for an Associated Human Workflow Task](#)
- [Upgrading Multiple Projects That Share a Common Human Workflow Task](#)

9.8.1 Upgrading Task Details for an Associated Human Workflow Task

When you are upgrading Oracle SOA Suite 10g applications that include task details for a human task, note the following:

- In Oracle SOA Suite 10g Release 3 (10.1.3), task details were generated inside same composite project and was deployed during composite deployment.
- In Oracle SOA Suite 11g, task details are generated as a separate project that must be deployed separately.

As a result, during the upgrade of Oracle Human Workflow, Oracle JDeveloper 11g automatically creates a new project for the task details UI and upgrades the task detail pages. You must then deploy this project separately, after deploying the associated composite application.

9.8.2 Upgrading Multiple Projects That Share a Common Human Workflow Task

If you are upgrading an application that contains multiple Oracle JDeveloper projects that share a common human workflow task, then be sure to review the upgraded application carefully in Oracle JDeveloper 11g.

In some cases, because the projects are moved into a common directory during the upgrade, data associated with the workflow task in one project may overwrite data from another project if the workflow task in the second project uses the same name.

9.9 Upgrade Tasks Associated with Oracle Business Rules

When you upgrade projects that use Oracle Business Rules, refer to the following documentation resources:

- [Section 9.3.1.1, "Upgrading to the Oracle Fusion Middleware Java API for Oracle Business Rules"](#)
- [Appendix C, "Upgrading Oracle Business Rules Dictionaries and Projects"](#)

9.10 Upgrading Oracle SOA Suite Client Applications

While you are in the process of upgrading your Oracle SOA Suite applications to 11g, and while your Oracle SOA Suite 10g environment is still up and running, you should begin to review, upgrade, and test the client applications that depend upon your Oracle SOA Suite environment.

Use the following list to analyze your client applications for required updates that will allow them to work with your newly upgraded 11g Oracle SOA Suite environment:

- Review the client applications to understand which remote Oracle SOA Suite APIs they are using.

For more information, refer to [Section 9.3.1, "Understanding Oracle SOA Suite API Changes for Oracle Fusion Middleware 11g"](#).

- Search your client applications for any references to Oracle SOA Suite HTTP URLs. The syntax of the Oracle SOA Suite 11g HTTP URLs has changed from 10g.

To obtain the new URL that a client can use to access a deployed Oracle SOA Suite 11g application, use the home page of the application deployment in Oracle Enterprise Manager Fusion Middleware Control. For more information, see "Monitoring SOA Composite Applications" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite*.

- If any of your client applications are using Web Services Addressing (WS-Addressing), then you must modify the client.

Oracle SOA Suite 11g supports WS-Addressing 1.0 for clients. As a result, you must either use a SOAP processor that understands WS-Addressing 1.0, or if you are using a SOAP processor that does not support WS-Addressing 1.0, then you must manually modify the 11g WSDL and change the client to get or set the WS-Addressing 1.0 headers.

Refer to [Example 9–5](#) and [Example 9–6](#) for a before and after example of how to modify a WSDL so it can be invoked by a SOAP processor that does not support WS-Addressing 1.0.

- If the clients using WS-Addressing are running on a Oracle SOA Suite 10g environment that you do not plan to upgrade at the same time, then you must apply the following patch to the Oracle SOA Suite environment before the client applications running on the 10g server can invoke 11g BPEL processes.

For more information, see "Applying Patch Sets to Address Specific Upgrade Interoperability Issues" in the *Oracle Fusion Middleware Upgrade Planning Guide*.

- Review your client applications for references to message targets, such as JMS queues and data source. Any references to these resources must be updated to point to the upgraded, Oracle SOA Suite 11g environment.
- If a client application is using Oracle Service Bus (OSB) to invoke an Oracle BPEL Process Manager process with the optimized transport, you must modify the client

application so it uses SOAP/HTML to invoke the Oracle BPEL Process Manager process.

For more information, see the Oracle Service Bus 10g Release 3 (10.3) documentation, which is available on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/documentation/bea.html>

Example 9-5 Example of WSDL Code That Can Be Used by a SOAP Processor that Supports WS-Addressing 1.0

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions
  name="UnitedLoan"
  targetNamespace="http://services.otn.com"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:s1="http://www.autoloan.com/ns/autoloan"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://services.otn.com">
  <plnk:partnerLinkType name="UnitedLoan">
    <plnk:role name="UnitedLoanProvider">
      <plnk:portType name="tns:UnitedLoan"/>
    </plnk:role>
    <plnk:role name="UnitedLoanRequester">
      <plnk:portType name="tns:UnitedLoanCallback"/>
    </plnk:role>
  </plnk:partnerLinkType>
  <wsp:Policy xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
    oasis-200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="wsaddr_policy"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <wsaw:UsingAddressing xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
      xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"/>
  </wsp:Policy>
  <documentation>
    <abstractWSDL>http://stadg55.us.oracle.com:7001/soa-infra/services/default/
      UnitedLoan!1.0/UnitedLoan.wsdl
    </abstractWSDL>
  </documentation>
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.autoloan.com/ns/autoloan"
        schemaLocation="http://stadg55.us.oracle.com:7001/soa-infra/services/
          default/UnitedLoan!1.0*05ed6481-93e6-4b46-b2d7-a21fa27a36a9/
          UnitedLoan_client?XSD=AutoLoanTypes.xsd"/>
    </schema>
  </types>
  <message name="UnitedLoanResultMessage">
    <part name="payload" element="s1:loanOffer"/>
  </message>
  <message name="UnitedLoanRequestMessage">
    <part name="payload" element="s1:loanApplication"/>
  </message>
  <portType name="UnitedLoanCallback">
    <operation name="onResult">
      <input message="tns:UnitedLoanResultMessage"/>
    </operation>
  </portType>
  <portType name="UnitedLoan">
```

```

        <operation name="initiate">
            <input message="tns:UnitedLoanRequestMessage" />
        </operation>
    </portType>
    <binding name="UnitedLoanBinding" type="tns:UnitedLoan">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <wsp:PolicyReference
            xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
            URI="#wsaddr_policy" wsdl:required="false" />
        <operation name="initiate">
            <soap:operation style="document" soapAction="initiate" />
            <input>
                <soap:body use="literal" namespace="http://services.otn.com" />
            </input>
        </operation>
    </binding>
    <binding name="UnitedLoanCallbackBinding"
        type="tns:UnitedLoanCallback">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
        <operation name="onResult">
            <soap:operation style="document" soapAction="onResult" />
            <input>
                <soap:body use="literal" namespace="http://services.otn.com" />
            </input>
        </operation>
    </binding>
    <service name="UnitedLoan">
        <port name="UnitedLoanPort" binding="tns:UnitedLoanBinding">
            <soap:address
                location="http://stadg55.us.oracle.com:7001/soa-infra/services/
                default/UnitedLoan!1.0*05ed6481-93e6-4b46-b2d7-a21fa27a36a9/
                UnitedLoan_client" />
        </port>
    </service>
</definitions>

```

Example 9–6 Example of WSDL Code Modified So It Can Be Used by a SOAP Processor That Does Not Support WS-Addressing 1.0

```

<?xml version="1.0" encoding="UTF-8" ?>
<definitions
    name="UnitedLoan"
    targetNamespace="http://services.otn.com"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:s1="http://www.autoloan.com/ns/autoloan"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:wsa="http://www.w3.org/2005/08/addressing"
    xmlns:tns="http://services.otn.com"
>
    <plnk:partnerLinkType name="UnitedLoan">
        <plnk:role name="UnitedLoanProvider">
            <plnk:portType name="tns:UnitedLoan" />
        </plnk:role>
        <plnk:role name="UnitedLoanRequester">
            <plnk:portType name="tns:UnitedLoanCallback" />
        </plnk:role>
    </plnk:partnerLinkType>
</types>

```

```

<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://www.autoloan.com/ns/autoloan"
    schemaLocation="http://stadg55.us.oracle.com:7001/
    soa-infra/services/default/
    UnitedLoan!1.0*05ed6481-93e6-4b46-b2d7-a21fa27a36a9/
    UnitedLoan_client?XSD=AutoLoanTypes.xsd" />
</schema>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://www.w3.org/2005/08/addressing"
    schemaLocation="http://www.w3.org/2006/03/
    addressing/ws-addr.xsd" />
</schema>
</types>
<message name="UnitedLoanResultMessage">
  <part name="payload" element="s1:loanOffer" />
</message>
<message name="UnitedLoanRequestMessage">
  <part name="payload" element="s1:loanApplication" />
</message>
<message name="WSAMessageIDHeader">
  <part name="MessageID" element="wsa:MessageID" />
</message>
<message name="WSAReplyToHeader">
  <part name="ReplyTo" element="wsa:ReplyTo" />
</message>
<message name="WSARelatesToHeader">
  <part name="RelatesTo" element="wsa:RelatesTo" />
</message>
<portType name="UnitedLoanCallback">
  <operation name="onResult">
    <input message="tns:UnitedLoanResultMessage" />
  </operation>
</portType>
<portType name="UnitedLoan">
  <operation name="initiate">
    <input message="tns:UnitedLoanRequestMessage" />
  </operation>
</portType>
<binding name="UnitedLoanBinding" type="tns:UnitedLoan">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="initiate">
    <soap:operation style="document" soapAction="initiate" />
    <input>
      <soap:header message="tns:WSAReplyToHeader"
        part="ReplyTo"
        use="literal" />
      <soap:header message="tns:WSAMessageIDHeader"
        part="MessageID"
        use="literal" />
      <soap:body use="literal" namespace="http://services.otn.com" />
    </input>
  </operation>
</binding>
<binding name="UnitedLoanCallbackBinding"
  type="tns:UnitedLoanCallback">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="onResult">
    <soap:operation style="document" soapAction="onResult" />
    <input>
      <soap:header message="tns:WSARelatesToHeader"

```

```

        part="RelatesTo"
        use="literal"/>
        <soap:body use="literal" namespace="http://services.otn.com"/>
    </input>
</operation>
</binding>
<service name="UnitedLoan">
    <port name="UnitedLoanPort" binding="tns:UnitedLoanBinding">
        <soap:address location="http://stadg55.us.oracle.com:7001/
            soa-infra/services/default/
            UnitedLoan!1.0*05ed6481-93e6-4b46-b2d7-a21fa27a36a9/
            UnitedLoan_client"/>
    </port>
</service>
</definitions>

```

9.11 Upgrading Oracle Web Services Manager (WSM) Policies

In Oracle WSM 10g, you specify *policy steps* at each policy enforcement point. Each policy step is a fine-grained operational task that addresses a specific security operation, such as authentication and authorization; encryption and decryption; security signature, token, or credential verification; and transformation. Each operational task is performed on either the Web service request or response.

For more details about the Oracle WSM 10g policy steps, see “Oracle Web Services Manager Policy Steps” in the *Oracle Web Services Manager Administrator’s Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

In Oracle WSM 11g, you attach *policies* to Web service endpoints. Each policy consists of one or more *assertions*, defined at the domain-level, that define the security requirements. A set of predefined policies and assertions are provided out-of-the-box. For more details about the predefined policies, see “Predefined Policy Reference” in *Security and Administrator’s Guide for Oracle Web Services*.

9.11.1 Before You Upgrade

Before you upgrade Oracle WSM policies, you must perform the following tasks:

- Install Oracle WSM 11g. For more information, see the *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
- Upgrade your Oracle Containers for J2EE (OC4J) 10g Web services to Oracle WebLogic Server 11g Web services.

For more information, see “Task 6: Upgrade the Application Web Services” in the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

9.11.1.1 A Note About Oracle WSM 10g Gateways

As described in “Examining the Rearchitecture of Oracle WSM in Oracle Fusion Middleware” in the *Oracle Fusion Middleware Security and Administrator’s Guide for Web Services*, Oracle Fusion Middleware 11g Release 1 (11.1.1) does not include a Gateway component.

You can continue to use the Oracle WSM 10g Gateway components with Oracle WSM 10g policies in your applications. For information about Oracle WSM 10g interoperability, see “Interoperability with Oracle WSM 10g Security Environments” in the *Oracle Fusion Middleware Security and Administrator’s Guide for Web Services*.

9.11.1.2 A Note About Third-party Software

As described in "Examining the Rearchitecture of Oracle WSM in Oracle Fusion Middleware" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*, Oracle WSM 10g supported policy enforcement for third-party application servers, such as IBM WebSphere and Red Hat JBoss. Oracle Fusion Middleware 11g Release 1 (11.1.1) only supports Oracle WebLogic Server.

You can continue to use the third-party application servers with Oracle WSM 10g policies. For information about Oracle WSM 10g interoperability, see "Interoperability with Oracle WSM 10g Security Environments" in the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

9.11.2 Upgrading Oracle WSM 10g Predefined Policies

Table 9–3 describes the most common Oracle WSM predefined policy upgrade scenarios based on the following security requirements: authentication and authorization, message protection, transport, and logging. A comparison of the steps required to implement each security requirement in both the Oracle WSM 10g and Oracle WSM 11g environments is provided.

For more information about:

- Attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.
- Oracle WSM 10g policy steps, see "Oracle Web Services Manager Policy Steps" in *Oracle Web Services Manager Administrator's Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

Table 9–3 Upgrading Oracle WSM 10g Predefined Policies

| Security Requirements | Oracle WSM 10g | Oracle WSM 11g |
|--|---|---|
| Anonymous authentication with message protection (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: Sign Message and Encrypt. ■ Web service: Decrypt and Verify Signature. | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_message_protection_client_policy. Web service: oracle/wss10_message_protection_service_policy. 2. Leave the default configuration set for message signing and encryption. 3. Disable the Include Timestamp configuration setting. |
| Anonymous authentication with message integrity (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: Sign Message. ■ Web service: Verify Signature. | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_message_protection_client_policy. Web service: oracle/wss10_message_protection_service_policy. 2. Configure the policy assertion for message signing only. 3. Disable the Include Timestamp configuration setting. |

Table 9–3 (Cont.) Upgrading Oracle WSM 10g Predefined Policies

| Security Requirements | Oracle WSM 10g | Oracle WSM 11g |
|---|---|--|
| Anonymous authentication with message confidentiality (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: XML Encrypt. ■ Web service: XML Decrypt. | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_message_protection_client_policy. Web service: oracle/wss10_message_protection_service_policy. 2. Configure the policy assertion for message encryption only. 3. Disable the Include Timestamp configuration setting. |
| Username token with message protection (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: Insert WSBasic Credentials and Sign Message and Encrypt. ■ Web service: Decrypt and Verify Signature, Extract Credentials (configured as WSBasic), and FileAuthenticate. <p>Note: You can substitute File Authenticate with LDAP Authenticate, Oracle Access Manager Authenticate, Active Directory Authenticate, or SetMinder Authenticate.</p> | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_username_token_with_message_protection_client_policy. Web service: oracle/wss10_username_token_with_message_protection_service_policy. 2. Leave the default configuration set for message signing and encryption. 3. Disable the Include Timestamp configuration setting. 4. Configure the Authentication and Identity Assertion provider. |
| Username token with message protection (WS-Security 1.0) and file authorization | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: Insert WSBasic Credentials and Sign Message and Encrypt. ■ Web service: Decrypt and Verify Signature, Extract Credentials (configured as WSBasic), FileAuthenticate and File Authorize. <p>Note: You can substitute File Authenticate with LDAP Authenticate, Active Directory Authenticate, or SetMinder Authenticate.</p> <p>Similarly, you can substitute File Authorize with LDAP Authorize, Active Directory Authorize, or SetMinder Authorize.</p> | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_username_token_with_message_protection_client_policy. Web service: oracle/wss10_username_token_with_message_protection_service_policy and oracle/binding_authorization. 2. Leave the default configuration set for message signing and encryption. 3. Disable the Include Timestamp configuration setting. 4. Configure the Authentication and Identity Assertion provider. |
| ID propagation with SAML token (sender vouches) with message protection (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ■ Client: SAML—Insert WSS 1.0 Sender-Vouches Token and Sign and Encrypt. ■ Web service: XML Decrypt and SAML—Verify WSS 1.0 Token. | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_saml_token_with_message_protection_client_policy. Web service: oracle/wss10_saml_token_with_message_protection_service_policy. 2. Disable the Include Timestamp configuration setting. 3. Leave the default configuration set for message signing and encryption. |

Table 9–3 (Cont.) Upgrading Oracle WSM 10g Predefined Policies

| Security Requirements | Oracle WSM 10g | Oracle WSM 11g |
|--|---|---|
| HTTP basic authentication | Attach policy steps as follows: <ul style="list-style-type: none"> ▪ Client: N/A. ▪ Web service: Extract Credentials (configure as HTTP). | Attach policies as follows: <ul style="list-style-type: none"> ▪ Client: oracle/wss_http_token_client_policy. ▪ Web service: oracle/wss_http_token_service_policy. |
| Oracle Access Manager security (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ▪ Client: Insert Oracle Access Manager Token. ▪ Web service: Extract Credentials and Oracle Access Manager Authenticate Authorize. | Attach policies as follows: <ul style="list-style-type: none"> ▪ Client: oracle/wss_oam_token_client_policy. ▪ Web service: oracle/wss_oam_token_service_policy. |
| Mutual authentication with message protection (WS-Security 1.0) | Attach policy steps as follows: <ul style="list-style-type: none"> ▪ Client: Insert WSBASIC Credentials and Sign Message and Encrypt. ▪ Web service: Decrypt and Verify Signature, Extract Credentials (configured as WSBASIC), and FileAuthenticate. <p>Note: You can substitute File Authenticate with LDAP Authenticate, Oracle Access Manager Authenticate, Active Directory Authenticate, or SetMinder Authenticate.</p> | <ol style="list-style-type: none"> 1. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss10_x509_token_with_message_protection_client_policy. Web service: oracle/wss10_x509_token_with_message_protection_service_policy. 2. Leave the default configuration set for message signing and encryption. 3. Disable the Include Timestamp configuration setting. 4. Configure the Authentication and Identity Assertion provider. |
| Username token over SSL | <ol style="list-style-type: none"> 1. Configure the application server for SSL. 2. Attach policy steps as follows: <ul style="list-style-type: none"> Client: Insert WSBASIC Credentials. Web service: Extract Credentials and File Authenticate. | <ol style="list-style-type: none"> 1. Configure the application server for SSL. 2. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss_username_token_over_ssl_client_policy. Web service: oracle/wss_username_token_over_ssl_client_service_policy. 3. Disable the Include Timestamp configuration setting. |
| ID propagation with SAML token (sender vouches) over SSL (WS-Security 1.0) | <ol style="list-style-type: none"> 1. Configure the application server for SSL. 2. Attach policy steps as follows: <ul style="list-style-type: none"> Client: SAML—Insert WSS 1.0 Sender-Vouches Token. Web service: SAML—Verify WSS 1.0 Token. | <ol style="list-style-type: none"> 1. Configure the application server for SSL. 2. Attach policies as follows: <ul style="list-style-type: none"> Client: oracle/wss_saml_token_over_ssl_client_policy. Web service: oracle/wss_saml_token_over_ssl_client_service_policy. 3. Disable the Include Timestamp configuration setting. |
| Log information | Attach the following policy step to the client or Web service: Log | Attach the following policy to the client or Web service: oracle/log_policy |

9.11.3 Upgrading Oracle WSM Custom Policies

In Oracle WSM 10g, you create, develop, and deploy custom *policy steps* using the procedures described in the *Oracle Web Services Manager Extensibility Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

In Oracle WSM 11g, you create, develop, and deploy custom *policy assertions*. You will need to redefine your custom policy steps as custom policy assertions using the procedures described in "Creating Custom Assertions" in *Security and Administrator's Guide for Oracle Web Services*.

9.12 Upgrading Oracle Containers for J2EE (OC4J) Security Environments

In OC4J 10g, you configure your security environment by modifying the contents of the XML-based deployment descriptor files. For complete details about securing OC4J environments, see *Oracle Application Server Web Services Security Guide* at:

<http://www.oracle.com/technology/documentation/>

In Oracle WSM 11g, you attach *policies* to Web service endpoints. Each policy consists of one or more *assertions*, defined at the domain-level, that define the security requirements. A set of predefined policies and assertions are provided out-of-the-box. For more details about the predefined policies, see "Predefined Policy Reference" in *Security and Administrator's Guide for Oracle Web Services*.

The following sections describe the most common OC4J upgrade scenarios based on the following security requirements: authentication, message protection, transport, and logging. A comparison of the steps required to implement each security requirement in both the OC4J 10g and Oracle WSM 11g environments is provided.

- [Section 9.12.1, "Before You Upgrade"](#)
- [Section 9.12.2, "Anonymous Authentication with Message Protection \(WS-Security 1.0\)"](#)
- [Section 9.12.3, "Anonymous Authentication with Message Integrity \(WS-Security 1.0\)"](#)
- [Section 9.12.4, "Anonymous Authentication with Message Confidentiality \(WS-Security 1.0\)"](#)
- [Section 9.12.5, "Username Token with Message Protection \(WS-Security 1.0\)"](#)
- [Section 9.12.6, "ID Propagation Using SAML Token \(Sender Vouches\) with Message Protection \(WS-Security 1.0\)"](#)
- [Section 9.12.7, "ID Propagation Using SAML Token \(Holder of Key\) with Message Protection \(WS-Security 1.0\)"](#)
- [Section 9.12.8, "Mutual Authentication with Message Protection \(WS-Security 1.0\)"](#)
- [Section 9.12.9, "Username token over SSL"](#)
- [Section 9.12.10, "ID Propagation with SAML Token \(Sender Vouches\) over SSL \(WS-Security 1.0\)"](#)
- [Section 9.12.11, "Log Information"](#)

Note: For information about configuring attaching policies in Oracle Fusion Middleware 11g, see *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

The next section describes the prerequisites required before you upgrade.

9.12.1 Before You Upgrade

Before you upgrade the OC4J security environment, you must perform the following tasks:

- Install Oracle WSM 11g. For more information, see the *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite*.
- Review "Task 6: Upgrade the Application Web Services" in the *Oracle Fusion Middleware Upgrade Guide for Java EE*.

This section provides general information about upgrading OC4J Web services to Oracle WebLogic Server.

9.12.2 Anonymous Authentication with Message Protection (WS-Security 1.0)

The following sections describe how to implement authentication with message protection that conforms to the WS-Security 1.0 standard, and compare the steps required for the OC4J 10g and Oracle WSM 11g environments.

9.12.2.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

Web Service Client (with sample data)

Define the <signature> and <encrypt> elements in the client deployment descriptor. For example:

```
<signature>
  <signature-method>RSA-SHA1</signature-method>
  <tbs-elements>
    <tbs-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <add-timestamp created="true" expiry="28800" />
</signature>
<encrypt>
  <recipient-key alias="orakey"/>
  <encryption-method>AES-128</encryption-method>
  <keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
  <tbe-elements>
    <tbe-element local-part="Body" name-space=
```

```
        "http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
    </tbe-elements>
</encrypt>
```

Web Service (with sample data)

Define the <verify-signature> and <decrypt> elements in the service deployment descriptor. For example:

```
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <verify-timestamp expiry="28800" created="true" />
</verify-signature>
<decrypt>
  <tbe-elements>
    <tbe-element
      name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" mode="CONTENT" />
  </tbe-elements>
</decrypt>
```

9.12.2.2 Oracle WSM 11g

Perform the following steps:

1. Attach policies as follows:

Client: oracle/wss10_message_protection_client_policy.

Web service: oracle/wss10_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Leave the configuration set for message body signing and encryption.

9.12.3 Anonymous Authentication with Message Integrity (WS-Security 1.0)

The following sections describe how to implement authentication with message integrity that conforms to the WS-Security 1.0 standard, and compare the steps required for the OC4J 10g and Oracle WSM 11g environments.

9.12.3.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the Oracle Application Server 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

Web Service Client (with sample data)

Define the <signature> element in the client deployment descriptor. For example:

```
<signature>
  <signature-method>RSA-SHA1</signature-method>
  <tbs-elements>
    <tbs-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <add-timestamp created="true" expiry="28800" />
</signature>
```

Web Service (with sample data)

Define the <verify-signature> element in the service deployment descriptor. For example:

```
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <verify-timestamp expiry="28800" created="true" />
</verify-signature>
```

9.12.3.2 Oracle WSM 11g

Perform the following steps:

1. Attach policies as follows:

Client: oracle/wss10_message_protection_client_policy.

Web service: oracle/wss10_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Configure the policy assertion for message body signing only.

9.12.4 Anonymous Authentication with Message Confidentiality (WS-Security 1.0)

The following sections describe how to implement authentication with message confidentiality that conforms to the WS-Security 1.0 standard, and compare the steps required for the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

9.12.4.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <encrypt> element in the client deployment descriptor. For example:

```
<encrypt>
  <recipient-key alias="orakey" />
  <encryption-method>AES-128</encryption-method>
  <keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
  <tbe-elements>
    <tbe-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
  </tbe-elements>
</encrypt>
```

Web Service (with sample data)

Define the <decrypt> element in the service deployment descriptor. For example:

```
<decrypt>
  <tbe-elements>
    <tbe-element
      name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" mode="CONTENT" />
  </tbe-elements>
</decrypt>
```

9.12.4.2 Oracle WSM 11g

Perform the following steps:

1. Attach policies as follows:

Client: oracle/wss10_message_protection_client_policy.

Web service: oracle/wss10_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Configure the policy assertion for message body encryption only.

9.12.5 Username Token with Message Protection (WS-Security 1.0)

The following sections describe how to implement username token with message protection that conforms to the WS-Security 1.0 standard, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

9.12.5.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

Web Service Client (with sample data)

Define the <username-token>, <signature>, and <encrypt> elements in the client deployment descriptor. For example:

```

<username-token password-type="PLAINTEXT" add-nonce="false"
  add-created="false" />
<signature>
  <signature-method>RSA-SHA1</signature-method>
  <tbs-elements>
    <tbs-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      local-part="UsernameToken" />
  </tbs-elements>
  <add-timestamp created="true" expiry="28800" />
</signature>
<encrypt>
  <recipient-key alias="orakey" />
  <encryption-method>AES-128</encryption-method>
  <keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
  <tbe-elements>
    <tbe-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
    <tbe-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      local-part="UsernameToken" />
  </tbe-elements>
</encrypt>

```

Web Service (with sample data)

Define the <verify-username-token>, <verify-signature>, and <decrypt> elements in the service deployment descriptor. For example:

```

<verify-username-token password-type="PLAINTEXT" require-nonce="false"
  require-created="false" />
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space=
      "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <verify-timestamp expiry="28800" created="true" />
</verify-signature>
<decrypt>
  <tbe-elements>
    <tbe-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" mode="CONTENT" />
  </tbe-elements>
</decrypt>

```

9.12.5.2 Oracle WSM 11g

Perform the following steps:

1. Attach policies as follows:

Client: oracle/wss10_username_token_with_message_protection_client_policy.

Web service: oracle/wss10_username_token_with_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Leave the configuration set for message body signing and encryption.
3. Configure the Authentication and Identity Assertion provider.

9.12.6 ID Propagation Using SAML Token (Sender Vouches) with Message Protection (WS-Security 1.0)

The following sections describe how to implement ID propagation using SAML token sender vouches with message protection that conforms to the WS-Security 1.0 standard, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide 10g (10.1.3.1.0)* at:

<http://www.oracle.com/technology/documentation/>

9.12.6.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <saml-token>, <signature>, and <encrypt> elements in the client deployment descriptor. For example:

```
<saml-token issuer-name="www.oracle.com" name="weblogic"
  name-format="UNSPECIFIED">
  <subject-confirmation-method>
    <confirmation-method>SENDER-VOUCHES</confirmation-method>
  </subject-confirmation-method>
</saml-token>
<signature>
  <signature-method>RSA-SHA1</signature-method>
  <tbs-elements>
    <tbs-element local-part="Body"
      name-space="http://schemas.xmlsoap.org/soap/envelope/" />
    <tbs-element name-space=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <add-timestamp created="true" expiry="28800" />
</signature>
```

```

<encrypt>
  <recipient-key alias="orakey" />
  <encryption-method>AES-128</encryption-method>
  <keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
  <tbe-elements>
    <tbe-element local-part="Body" name-space="
      http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
  </tbe-elements>
</encrypt>

```

Web Service (with sample data)

Define the <verify-saml-token>, <verify-signature>, and <decrypt> elements in the service deployment descriptor. For example:

```

<verify-saml-token>
  <subject-confirmation-methods>
    <confirmation-method>SENDER-VOUCHES</confirmation-method>
  </subject-confirmation-methods>
</verify-saml-token>
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space="
      http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <verify-timestamp expiry="28800" created="true" />
</verify-signature>
<decrypt>
  <tbe-elements>
    <tbe-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" mode="CONTENT" />
  </tbe-elements>
</decrypt>

```

9.12.6.2 Oracle WSM 11g

Attach policies as follows:

- **Client:** oracle/wss10_saml_token_with_message_protection_client_policy.
- **Web service:** oracle/wss10_saml_token_with_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

9.12.7 ID Propagation Using SAML Token (Holder of Key) with Message Protection (WS-Security 1.0)

The following sections describe how to implement ID propagation using SAML token holder of key with message protection that conforms to the WS-Security 1.0 standard, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

9.12.7.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <saml-token>, <signature>, and <encrypt> elements in the client deployment descriptor. For example:

```
<saml-token issuer-name="www.oracle.com" name="weblogic"
  name-format="UNSPECIFIED">
  <subject-confirmation-method>
    <confirmation-method>HOLDER-OF-KEY</confirmation-method>
  </subject-confirmation-method>
</saml-token>
<signature>
  <signature-method>RSA-SHA1</signature-method>
  <tbs-elements>
    <tbs-element local-part="Body"
      name-space="http://schemas.xmlsoap.org/soap/envelope/" />
    <tbs-element name-space=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <add-timestamp created="true" expiry="28800" />
</signature>
<encrypt>
  <recipient-key alias="orakey" />
  <encryption-method>AES-128</encryption-method>
  <keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
  <tbe-elements>
    <tbe-element local-part="Body" name-space=
      "http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
  </tbe-elements>
</encrypt>
```

Web Service (with sample data)

Define the <verify-saml-token>, <verify-signature>, and <decrypt> elements in the service deployment descriptor. For example:

```
<verify-saml-token>
  <subject-confirmation-methods>
    <confirmation-method>HOLDER-OF-KEY</confirmation-method>
  </subject-confirmation-methods>
</verify-saml-token>
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
</verify-signature>
```

```

        local-part="Timestamp" />
    </tbs-elements>
    <verify-timestamp expiry="28800" created="true" />
</verify-signature>
<decrypt>
    <tbe-elements>
        <tbe-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
            local-part="Body" mode="CONTENT" />
    </tbe-elements>
</decrypt>

```

9.12.7.2 Oracle WSM 11g

Attach policies as follows:

- **Client:** oracle/wss10_saml_hok_with_message_protection_client_policy.
- **Web service:** oracle/wss10_saml_hok_with_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

9.12.8 Mutual Authentication with Message Protection (WS-Security 1.0)

The following sections describe how to implement mutual authentication with message protection that conforms to the WS-Security 1.0 standard, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

9.12.8.1 OC4J 10g

Edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <x509-token>, <signature>, and <encrypt> elements in the client deployment descriptor. For example:

```

<x509-token />
<signature>
    <signature-method>RSA-SHA1</signature-method>
    <tbs-elements>
        <tbs-element local-part="Body"
            name-space="http://schemas.xmlsoap.org/soap/envelope/" />
        <tbs-element name-space="
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
            local-part="Timestamp" />
    </tbs-elements>
    <add-timestamp created="true" expiry="28800" />
</signature>
<encrypt>
    <recipient-key alias="orakey" />
    <encryption-method>AES-128</encryption-method>

```

```

<keytransport-method>RSA-OAEP-MGF1P</keytransport-method>
<tbe-elements>
  <tbe-element local-part="Body"
    name-space="http://schemas.xmlsoap.org/soap/envelope/" mode="CONTENT" />
</tbe-elements>
</encrypt>

```

Web Service (with sample data)

Define the <verify-x509-token>, <verify-signature>, and <decrypt> elements in the service deployment descriptor. For example:

```

<verify-x509-token />
<verify-signature>
  <tbs-elements>
    <tbs-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" />
    <tbs-element name-space="
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      local-part="Timestamp" />
  </tbs-elements>
  <verify-timestamp expiry="28800" created="true" />
</verify-signature>
<decrypt>
  <tbe-elements>
    <tbe-element name-space="http://schemas.xmlsoap.org/soap/envelope/"
      local-part="Body" mode="CONTENT" />
  </tbe-elements>
</decrypt>

```

9.12.8.2 Oracle WSM 11g

Perform the following steps:

1. Attach policies as follows:

Client: oracle/wss10_x509_token_with_message_protection_client_policy.

Web service: oracle/wss10_x509_token_with_message_protection_service_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

2. Leave the configuration set for message body signing and encryption.
3. Configure the Authentication and Identity Assertion provider.

9.12.9 Username token over SSL

The following sections describe how to implement username token over SSL, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

9.12.9.1 OC4J 10g

Configure the application server for SSL and edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <username-token> and <signature> elements in the client deployment descriptor. For example:

```
<username-token password-type="PLAINTEXT" add-nonce="true"
  add-created="true" />
<signature>
  <add-timestamp created="true" expiry="28800" />
</signature>
```

Web Service (with sample data)

Define the <verify-username> element in the service deployment descriptor. For example:

```
<verify-username-token password-type="PLAINTEXT" require-nonce="false"
  require-created="false" />
<signature>
  <verify-timestamp expiry="28800" created="true" />
</signature>
```

9.12.9.2 Oracle WSM 11g

Perform the following step:

1. Configure the application server for SSL.
2. Attach policies as follows:

Client: oracle/wss_username_token_over_ssl_client_policy.

Web service: oracle/wss_username_token_over_ssl_service_policy

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

9.12.10 ID Propagation with SAML Token (Sender Vouches) over SSL (WS-Security 1.0)

The following sections describe how to implement ID propagation with SAML token sender vouches over SSL that conforms to WS-Security 1.0, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

For more information about the deployment descriptor elements, see "OracleAS Web Services Security Schema" in *Oracle Application Server Web Services Security Guide* in the 10g Release 3 (10.1.3.1.0) documentation library at:

<http://www.oracle.com/technology/documentation/>

9.12.10.1 OC4J 10g

Configure the application server for SSL and edit the deployment descriptors for the Web service and client, as described in the following sections.

Web Service Client (with sample data)

Define the <saml-token> and <signature> elements in the client deployment descriptor. For example:

```
<saml-token name="weblogic" issuer-name="www.oracle.com"
  name-format="UNSPECIFIED">
  <subject-confirmation-method>
    <confirmation-method>SENDER-VOUCHES</confirmation-method>
  </subject-confirmation-method>
</saml-token>
<signature>
  <add-timestamp created="true" expiry="28800" />
</signature>
```

Web Service (with sample data)

Define the <verify-saml-token> element in the service deployment descriptor. For example:

```
<verify-saml-token>
  <subject-confirmation-methods>
    <confirmation-method>SENDER-VOUCHES-UNSIGNED</confirmation-method>
  </subject-confirmation-methods>
</verify-saml-token>
<signature>
  <verify-timestamp expiry="28800" created="true" />
</signature>
```

9.12.10.2 Oracle WSM 11g

Perform the following steps:

1. Configure the application server for SSL.
2. Attach policies as follows:
 - Client:** oracle/wss_saml_token_over_ssl_client_policy.
 - Web service:** oracle/wss_saml_token_over_ssl_service_policy.

9.12.11 Log Information

The following sections describe how to enable the collection of log information, and compare the steps required in the OC4J 10g and Oracle WSM 11g environments.

9.12.11.1 OC4J 10g

Configure the logging and auditing interceptor.

9.12.11.2 Oracle WSM 11g

Attach the following policy to the Web service or client: oracle/log_policy.

For more information about attaching policies in Oracle Fusion Middleware 11g, see "Attaching Policies to Web Services" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

Considerations When Upgrading Oracle WebCenter Applications

This chapter describes tasks specific to upgrading Oracle WebCenter 10.1.3.x applications to Oracle Fusion Middleware 11g.

For information about the high-level tasks required to upgrade any Oracle SOA Suite, Oracle WebCenter, and ADF application, see [Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications."](#)

Use the following sections to understand tasks specific to upgrading Oracle WebCenter applications:

- [About Migration of WebCenter Applications](#)
- [Prerequisite Tasks for Migrating a WebCenter Application](#)
- [How To Migrate WebCenter Applications](#)
- [How to Migrate Portlet Producers](#)
- [How to Migrate Wiki Data](#)

10.1 About Migration of WebCenter Applications

Oracle WebCenter 10.1.3.x applications and Oracle WebCenter 11g applications are based on different application templates. This section provides an overview of the different WebCenter application templates and lists the high-level migration tasks.

10.1.1 About WebCenter Application Templates

Application templates used for creating WebCenter applications differ in Oracle JDeveloper 10.1.3.x and Oracle JDeveloper 11g. In JDeveloper 10.1.3.x, WebCenter applications are created by using the **WebCenter Application [Portlet, Content Repository, JSF]** application template. WebCenter applications based on this template include both WebCenter consumer applications and portlet producer applications.

Note: In this document, a WebCenter consumer application refers to an application that provides the user interface and contains JSP pages and components required in a web application. A portlet producer application refers to an application that contains Oracle WebCenter server-side producer components and the portlets that can be consumed by a WebCenter consumer application.

By contrast, Oracle WebCenter 11g applications are based on one of the following application templates, depending on the purpose of the application: **WebCenter Application** or **Portlet Producer Application**. A WebCenter consumer application is created by using the **WebCenter Application** template. A portlet producer application is created by using the **Portlet Producer Application** template.

In JDeveloper 10.1.3.x, WebCenter applications consist of three projects: **Model**, **Portlets**, and **ViewController**. In JDeveloper 11g, applications created by using the **WebCenter Application** template contain two projects, **Model** and **ViewController**, while applications created by using the **Portlet Producer Application** template contain only one project, **Portlets**.

A WebCenter 10.1.3.x application, which is based on the **WebCenter Application [Portlet, Content Repository, JSF]** application template, may be created to serve any of the following purposes:

- To be used as a consumer application
- To be used as a portlet producer application
- To be used both as a consumer and a portlet producer application

10.1.2 About Migration Tasks

Based on the purpose of your WebCenter 10.1.3.x application and Oracle WebCenter Services it uses, you may require to perform the following tasks to migrate your application to Oracle WebCenter 11g:

- Migrate the WebCenter consumer application.
- Migrate the portlet producer application and customizations, if the WebCenter application contains portlets.
- Migrate the data related to Oracle WebCenter Services, if your WebCenter consumer application uses these services.

Migration of WebCenter consumer applications takes place within JDeveloper. Migration of some portlet producer applications and some Oracle WebCenter Services takes place both within and outside JDeveloper.

Note: In Oracle WebCenter 11g, Oracle WebCenter Services are referred as WebCenter Web 2.0 Services.

10.2 Prerequisite Tasks for Migrating a WebCenter Application

Before you attempt to migrate a WebCenter 10.1.3.x application:

1. Verify that the application is running successfully on Oracle Application Server 10g.
For information, see [Section 8.2, "Task 2: Verify that the Applications Are Up and Running Successfully on Oracle Application Server 10g."](#)
2. Verify that your Oracle WebCenter 10.1.3.x environment is upgraded to Oracle WebCenter 11g environment.
For information, see [Section 8.3, "Task 3: Verify That You Have Upgraded Your 10g Environment to 11g."](#)
3. Install Oracle JDeveloper 11g.
For information, see [Section 8.4, "Task 4: Install and Start Oracle JDeveloper 11g."](#)

4. Install the Oracle WebCenter extension (11.1.1).

To work with WebCenter applications, you must install the Oracle WebCenter extension. The Oracle WebCenter extension is a JDeveloper add-in that provides all WebCenter capabilities in JDeveloper.

To install Oracle WebCenter 11g extension:

- a. Start JDeveloper 11g.
- b. If the Select Default Roles dialog displays, select **Default Role** to enable all technologies, and click **OK**.
- c. If this is the first time you started JDeveloper 11g, then it prompts whether you want to migrate settings from a previous version of JDeveloper.

Note: If you intend to migrate application settings from JDeveloper 10.1.3.x, then close and remove your WebCenter applications from the IDE in JDeveloper 10.1.3.x before you start JDeveloper 11g for the first time.

You can remove applications from the IDE by right-clicking an application name in the Application Navigator and selecting **Delete** from the shortcut menu. Removing an application removes it only from the IDE and not from the disk.

Click **Yes** to automatically migrate portlet customization data, which includes portlet customizations from preconfigured portlet producers like OmniPortlet and WebClipping. (Figure 10–1)

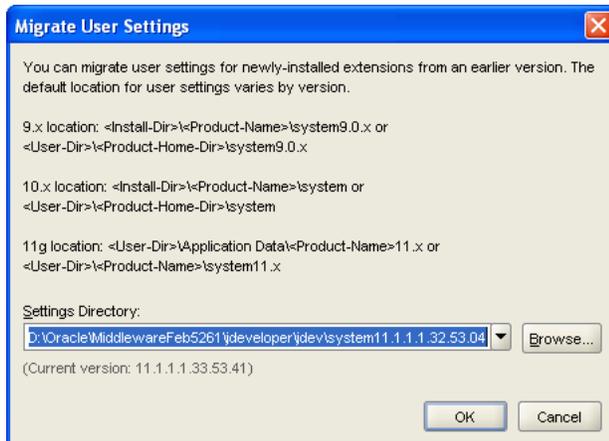
If you select **No** and later migrate your applications, then portlet customizations are not migrated from the previous version of JDeveloper. In such a case, you must manually migrate customizations later. For information about migration of customizations, see [Section 10.4.2.3, "Migrating Preconfigured Portlet Producers."](#)

Figure 10–1 *Migrate User Settings Dialog - Migrating Settings*



- d. The **Setting Directory** drop-down list shows the path to the previous version of JDeveloper. If you want to specify a different location, click the **Browse** button and specify the path. (Figure 10–2)
- e. Click **OK**.

Figure 10–2 Migrate User Settings Dialog - Settings Directory



- f. From the **Help** menu, select **Check for Updates**.
- g. On the Welcome page of the Check for Updates wizard, click **Next**.
- h. On the Source page, select **Search Update Centers** and then click **Next**.
- i. On the Updates page, search for the WebCenter extension, select it, and then click **Finish**.
- j. Click **Yes** when prompted to restart JDeveloper 11g.

For more information on obtaining and installing Oracle WebCenter extension, see the Oracle WebCenter page on Oracle Technology Network (OTN) at:

<http://webcenter.oracle.com>

10.3 How To Migrate WebCenter Applications

Table 10–1 lists the tasks involved in migrating a WebCenter 10.1.3.x consumer application to Oracle WebCenter 11g.

Table 10–1 Task Flow for Migrating a WebCenter 10.1.3.x Consumer Application

| Task | Sub Task | When to be Performed? |
|---|---|--|
| Preparing Your Application for Migration | Backing Up Your WebCenter Applications | Always |
| | Preparing Applications That Rely on Oracle Content DB | If your WebCenter 10.1.3.2 or 10.1.3.3 application contains Oracle Content Database (Oracle Content DB) connections. |
| | Preparing Applications That Contain Portlets | If your WebCenter application contains portlets. |
| Migrating Your WebCenter Consumer Application | Migrating Your WebCenter Application | Always |
| Performing Post Migration Tasks | Migrating Oracle Portal Connections | If your WebCenter application contains Oracle Portal connections. |
| | Configuring ADF Security | If your WebCenter application is secured |

Table 10–1 (Cont.) Task Flow for Migrating a WebCenter 10.1.3.x Consumer Application

| Task | Sub Task | When to be Performed? |
|------|---|--|
| | Downloading the Content Integration Suite API | If your WebCenter 10.1.3.4 application uses the <code>cis-client-8.0.0.jar</code> client API |
| | Migrating Producer Registrations of Preconfigured Portlet Producers | If preconfigured portlet producers are not deployed to the default port, 6688. |
| | Redeploying Your Applications | Always |

10.3.1 Preparing Your Applications for Migration

When you open a WebCenter 10.1.3.x application in JDeveloper 11g, the migration utility automatically upgrades the application to Oracle WebCenter 11g. However, you must back up your application before you migrate it. Further, you may need to perform certain tasks before migration depending on whether your application contains portlets or relies on Oracle Content DB.

This section describes the following tasks:

- [Backing Up Your WebCenter Applications](#)
- [Preparing Applications That Rely on Oracle Content DB](#)
- [Preparing Applications That Contain Portlets](#)

10.3.1.1 Backing Up Your WebCenter Applications

Before you migrate your WebCenter application, you must save a copy of the application to another location.

Note: It is important to back up your application because after you migrate a WebCenter 10.1.3.x application to Oracle WebCenter 11g, you cannot open it in an earlier release of JDeveloper 11g. Also, the changes made to the application during migration cannot be reverted.

If you use a source control system, then it may not be necessary to save a copy of your application.

Note: Make sure that you keep your JDeveloper 10.1.3.x installation when you migrate your applications. To prepare your applications for migration, you may require to perform certain tasks in JDeveloper 10.1.3.x.

10.3.1.2 Preparing Applications That Rely on Oracle Content DB

Oracle WebCenter 11g supports Oracle Content Server and Oracle Portal as external content repositories. If your WebCenter application relies on Oracle Content DB for content integration, then you must:

- Upgrade your application to Oracle WebCenter 10.1.3.4 to get Oracle Content Server support. This is required only if the application you intend to migrate is a WebCenter 10.1.3.2 or 10.1.3.3 application.

- Migrate all your Oracle Content DB data to Oracle Content Server 10.1.3.4.1.
For information about installing and configuring Oracle Content Server 10.1.3.4.1, see *Oracle Fusion Middleware Installation Guide for Oracle WebCenter*.

Note: If your application does not contain Oracle Content DB connections, then you can skip this section.

10.3.1.2.1 Upgrading Your Application in JDeveloper 10.1.3.4 To upgrade your WebCenter 10.1.3.2 or 10.1.3.3 application to Oracle WebCenter 10.1.3.4:

1. Open your application in JDeveloper 10.1.3.4 and follow the instructions in the migration wizard.
2. Edit the existing content repository data control in the migrated application to point to Oracle Content Server. For this, you must change the connection type to Oracle Content Server and redefine the custom attribute definitions. In the migrated WebCenter 10.1.3.4 application, you must use the same name and value for custom attributes that you used in your WebCenter 10.1.3.2 or 10.1.3.3 application.

For information about content repository connections, see the:

- "Managing Content Repositories" chapter in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter* - Describes how administrators can configure content repository connections for WebCenter application deployed to a managed server.
- "Integrating Content" chapter in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter* - Describes how developers can set up content repository connections within JDeveloper whilst building and testing new WebCenter applications.

After you have performed these tasks, your WebCenter 10.1.3.4 application is ready to be migrated to Oracle WebCenter 11g.

10.3.1.2.2 Migrating Oracle Content DB Data You can migrate Oracle Content DB data at any time during the application migration procedure.

To migrate Oracle Content DB data to Oracle Content Server by using a WebDav client:

1. Open the following WebDAV locations by using a user account having access to all the content:
 - Location on Oracle Content DB: `http://server:host/content/dav`
 - Location on Oracle Content Server:
`http://server:host/content-server-root/idcplg/webdav`
2. Copy the content from the Oracle Content DB location to the Oracle Content Server location.

When you migrate content by using WebDav, there are a few limitations. These are:

- The Trash folder gets copied.
- For the versioned documents, only the latest version gets copied.
- Metadata and access control setting are not preserved for the copied content.

10.3.1.3 Preparing Applications That Contain Portlets

In JDeveloper 11g, by default, WebCenter applications do not contain the Portlets project. If your WebCenter 10.1.3.x application contains portlets, then after you migrate the application, the application continues to include the Portlets project. Therefore, you must manually remove the Portlets project from your WebCenter application before you migrate it.

Note: If your application does not contain portlets, then you can skip this section.

To remove the Portlets project from your WebCenter 10.1.3.x application:

1. Start JDeveloper 10.1.3.x.

Note: To remove the Portlets project, you must use JDeveloper 10.1.3.x, and not JDeveloper 11g.

2. Open the WebCenter 10.1.3.x application that you want to migrate.
3. In the **Application Navigator**, select the **Portlets** project.
4. From the **File** menu, select **Erase from Disk**.
5. Click **Yes** to delete the project from the application.
6. Save your application.

10.3.2 Migrating Your WebCenter Application

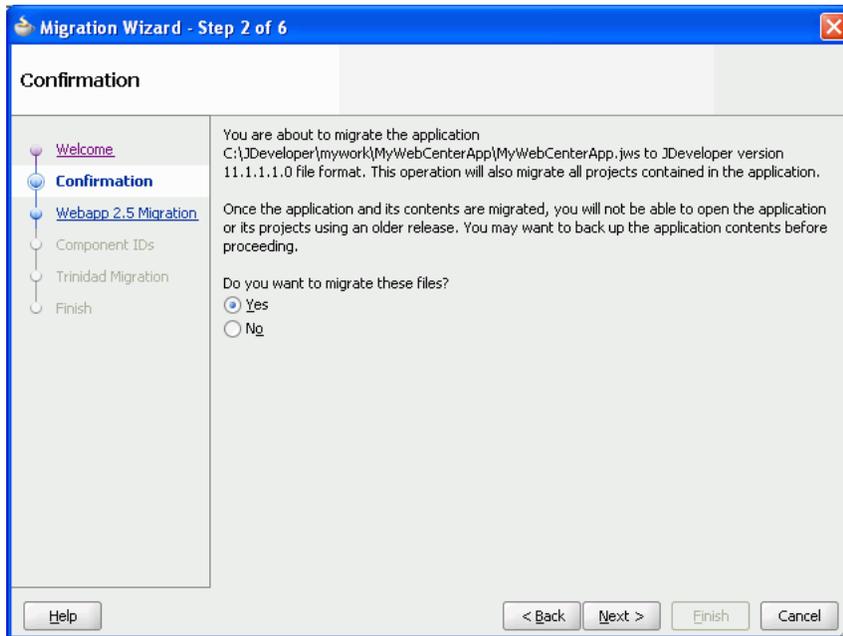
After you have prepared your WebCenter 10.1.3.x application, you can migrate it to Oracle WebCenter 11g.

To migrate your WebCenter 10.1.3.x application to Oracle WebCenter 11g:

1. Start JDeveloper 11g.
2. Open your WebCenter 10.1.3.x application.
This invokes the Migration Wizard because the application being opened is a WebCenter 10.1.3.x application and the JDeveloper version is 11g.
3. On the Welcome page, click **Next**.
4. On the Confirmation page, the **Yes** option is selected by default, as shown in [Figure 10–3](#). Click **Next** to confirm that you want to migrate the application and its projects.

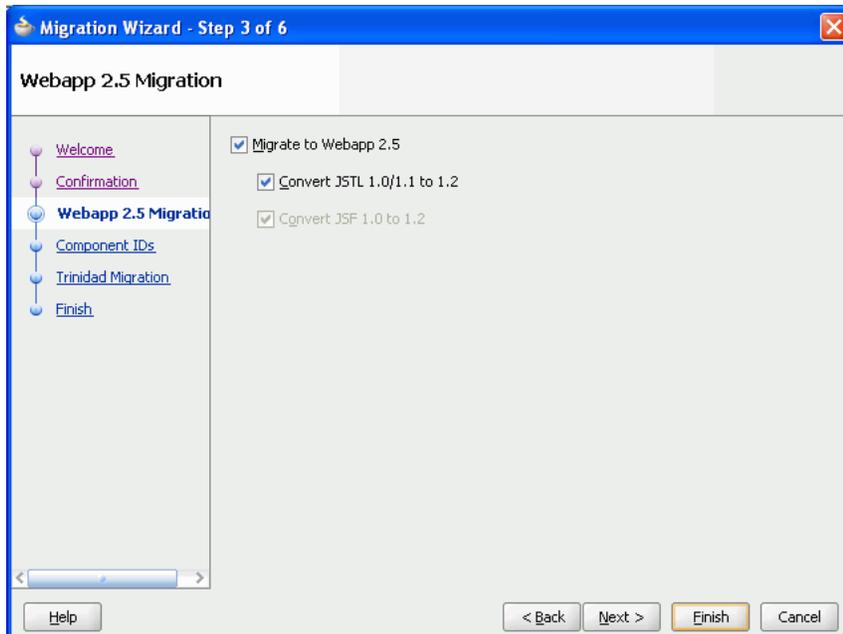
If you select **No**, the migration process is aborted and the application is not opened in JDeveloper.

Figure 10–3 Migration Wizard - Confirmation Page

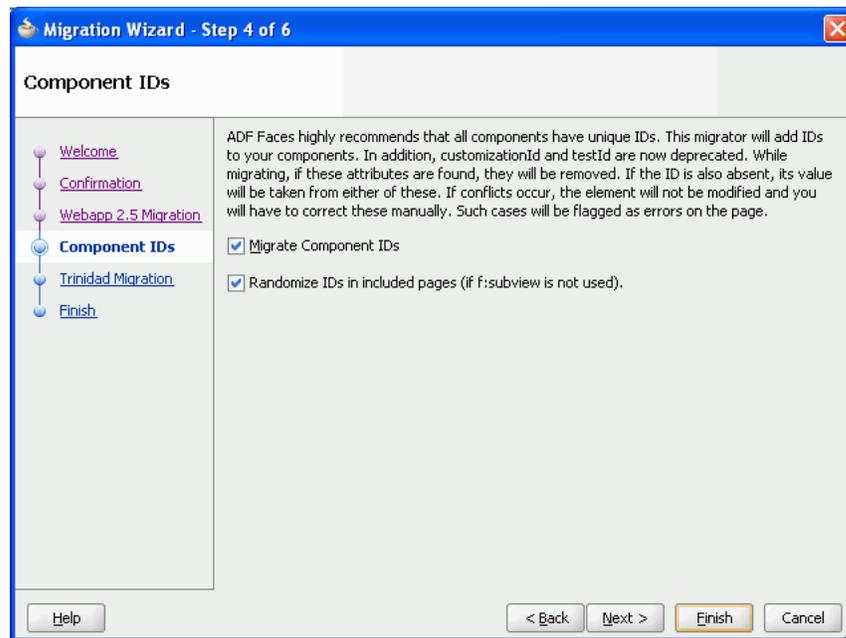


5. On the Webapp 2.5 Migration page, specify whether you want to migrate projects created using JavaServer Pages Standard Tag Library (JSTL) version 1.0 or 1.1, as shown in Figure 10–4. To accept the default setting, click **Next**.

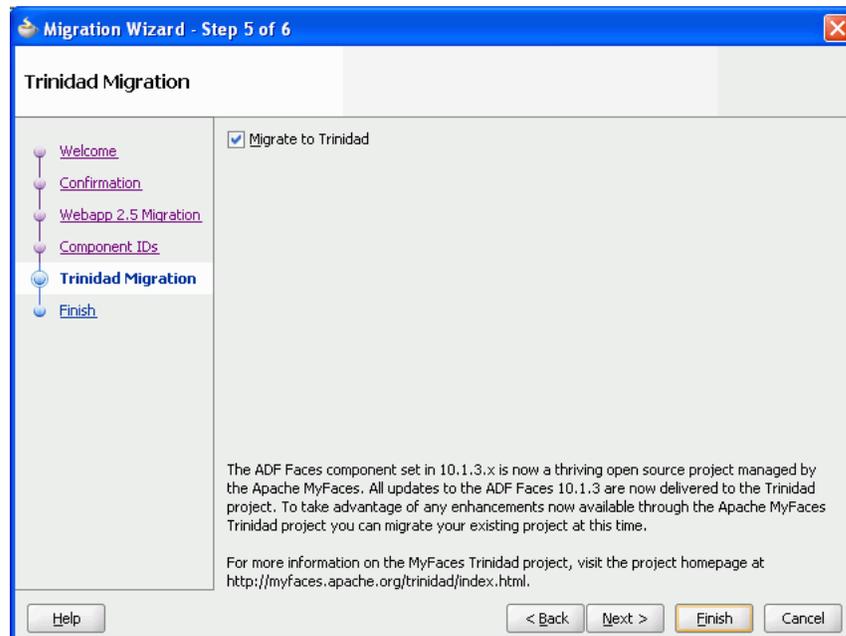
Figure 10–4 Migration Wizard - Webapp 2.5 Migration Page



6. On the Component IDs page, specify whether you want to migrate and randomize component IDs. Click **Next** to accept the default settings. (Figure 10–5)

Figure 10–5 Migration Wizard - Component IDs Page

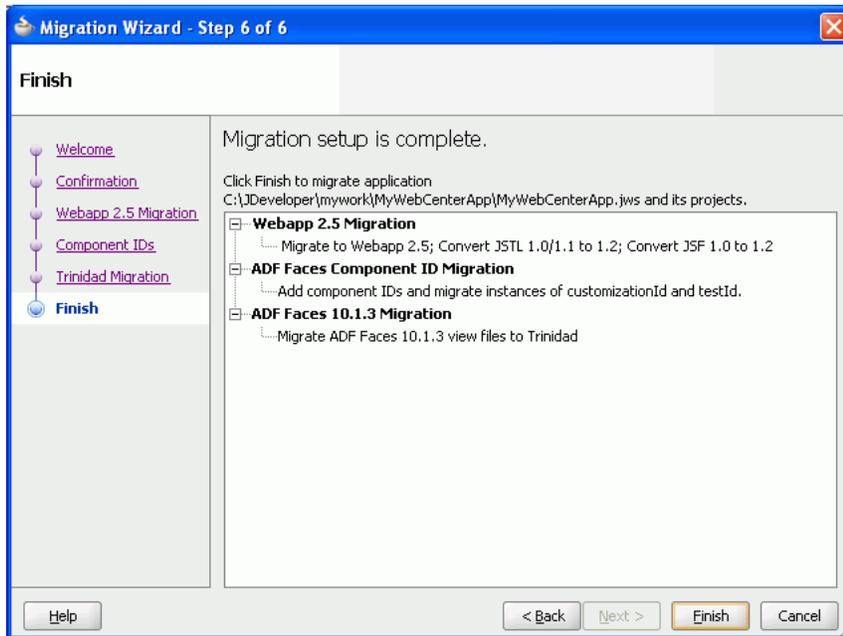
7. On the Trinidad Migration page, click **Next**. (Figure 10–6)

Figure 10–6 Migration Wizard - Trinidad Migration Page

8. Click **Finish** to begin migrating your WebCenter application. (Figure 10–7)

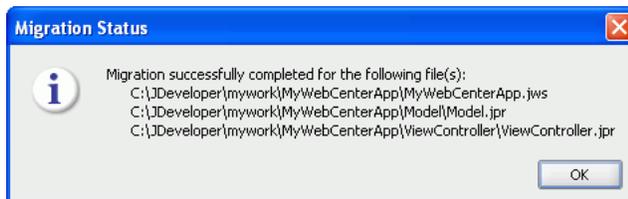
It might take a while for the migration process to complete, depending on the size of the application. A progress dialog displays while the migration process executes. When application migration is complete, the Migration Status dialog displays a list of projects that have been migrated, as shown in Figure 10–8.

Figure 10–7 Migration Wizard - Finish Page



9. Click OK.

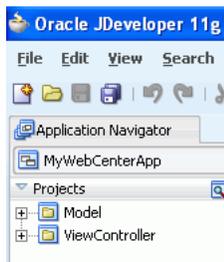
Figure 10–8 Migration Status



The migrated application is opened and its projects are listed in the Application Navigator. Notice that [Figure 10–8](#) and [Figure 10–9](#) do not list the Portlets project in the list of migrated projects.

If there are any errors during migration of an application, errors are listed in the Message - Log window.

Figure 10–9 Projects Migrated to JDeveloper 11g



10.3.3 Performing Post Migration Tasks

While migrating your application, the migration utility in JDeveloper 11g makes various changes to your application to configure it for Oracle WebCenter 11g. Some of these changes are related to the following:

- Customizable components
- External applications
- Portlet component changes

For information about these changes, see [Appendix B, "Additional Oracle WebCenter Migration Details."](#)

After migrating your WebCenter consumer application, you may require to perform various post migration tasks. These include:

- [Migrating Oracle Portal Connections](#)
- [Configuring ADF Security](#)
- [Downloading the Content Integration Suite API](#)
- [Migrating Producer Registrations of Preconfigured Portlet Producers](#)
- [Redeploying Your Applications](#)

10.3.3.1 Migrating Oracle Portal Connections

For content repository connections based on the Oracle Portal adapter, the data source needs to be migrated to a database connection, to be stored in `connections.xml`. When you start JDeveloper 11g for the first time, it prompts whether you want to migrate settings from a previous version of JDeveloper.

- If you choose **Yes**, then database connections are automatically migrated and are accessible from IDE Connections in JDeveloper 11g.
- If you choose **No**, then database connections are not migrated.

When you migrate your WebCenter application, the migration utility uses the migrated database connections if you chose **Yes** at the JDeveloper prompt for migrating settings. If you did not choose to migrate settings from a previous version of JDeveloper, then the migration utility attempts to create a database connection by using the information stored in the application's `data-sources.xml` file(s). In this case, the password for the database connection is not available. Therefore, after migrating your application, you must specify the password for the database connection by using the Edit Database Connection wizard.

In the unlikely event where the database connection used by the Oracle Portal adapter was not automatically migrated or created, then after migrating the application, you must create a new database connection by using the Create Database Connection wizard, which is accessible through the Edit Content Repository Connection wizard. For information about how to create a database connection, see the "How to Create a Content Repository Connection Based on the Oracle Portal Adapter" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

10.3.3.2 Configuring ADF Security

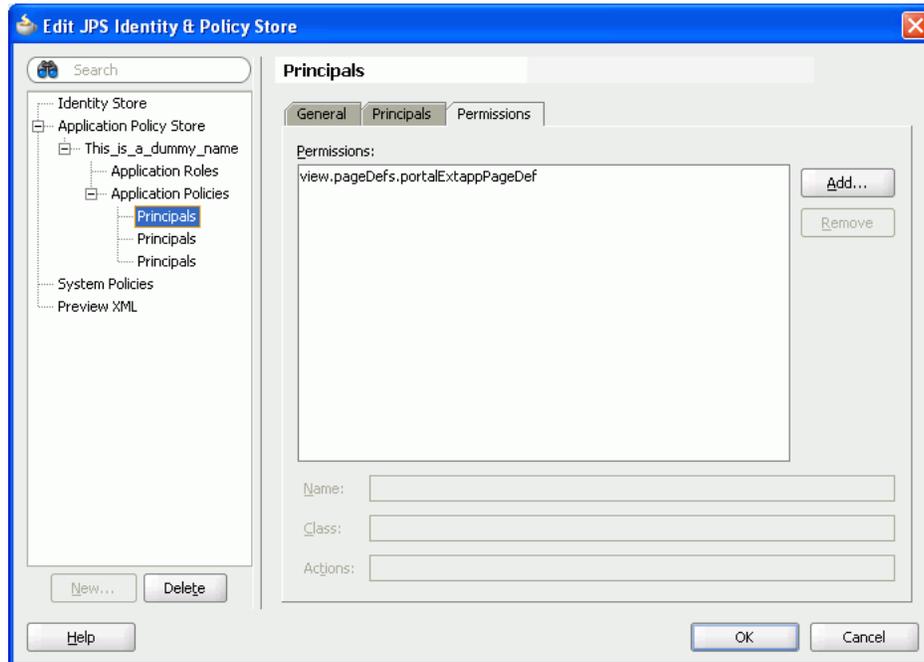
If your WebCenter 10.1.3.x application is ADF-secured, then you must reconfigure security after migrating the application.

When you migrate an ADF-secured WebCenter application, ADF security policies are migrated. The policies defined in the

`approval/.adf/META-INF/app-jazn-data.xml` are migrated to `approval/src/META-INF/jazn-data.xml`. However, the users and enterprise roles defined in the policies are not migrated.

Figure 10–10 shows the properties in `jazn-data.xml` of a migrated WebCenter application.

Figure 10–10 Security Settings of a Migrated WebCenter Application



To reconfigure ADF security in a migrated WebCenter application:

- You must create a new realm in the jazn-data editor and then re-create the required users and enterprise roles defined in the policies used in your WebCenter 10g application. It is recommended that you name the realm as `jazn.com`.
- You can optionally reconfigure application authorization data to use application roles. Oracle WebCenter 11g applications support application roles. You can reconfigure ADF security to use application roles instead of enterprise roles.

For information about how to configure ADF security, see *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

10.3.3.3 Downloading the Content Integration Suite API

Oracle WebCenter 10.1.3.4 includes `cis-client-8.0.0.jar`, a client API for Oracle Content Server. This API is not packaged with Oracle WebCenter 11g. If this API is not available to your migrated WebCenter 10.1.3.4 applications, then you may get compilation errors.

The `cis-client-8.0.0.jar` client API is available as part of Content Integration Suite 10g Release 3. If your WebCenter 10.1.3.4 applications use `cis-client-8.0.0.jar`, then after migrating your applications, you must download the API from the following location on OTN:

<http://www.oracle.com/technology/software/products/content-management/index.html>

On this OTN page, click the [Content Integration Suite 10gR3](#) link and download the ContentIntegrationSuite_10gR3_20081218.zip file, which contains the API.

10.3.3.4 Migrating Producer Registrations of Preconfigured Portlet Producers

If your WebCenter 10.1.3.x application uses portlets from preconfigured portlet producers like Web Clipping and OmniPortlet, then during migration port numbers of preconfigured portlet producers are updated. All producer registrations from preconfigured portlet producers that existed in Oracle Application Server 10.1.3.x environment with port 6688 are migrated to port 7101. If you did not use default port numbers in your Oracle Application Server 10.1.3.x environment, then you must manually change those port numbers to appropriate port numbers.

When you migrate a WebCenter application, port changes are not made to registrations of portlet producers other than preconfigured portlet producers.

10.3.3.5 Redeploying Your Applications

After migrating an application in JDeveloper 11g, you must recompile the application. Next, you must redeploy the application and verify the deployed application. For information, see [Chapter 8, "Overview of Upgrading Oracle SOA Suite, WebCenter, and ADF Applications."](#)

10.4 How to Migrate Portlet Producers

If your WebCenter 10.1.3.x application uses portlets, then in addition to migrating your WebCenter consumer application you may need to migrate portlet producers to Oracle WebCenter 11g.

[Table 10–2](#) lists the tasks involved in migrating portlet producers used by your WebCenter 10.1.3.x applications.

Table 10–2 Task Flow for Migrating Portlet Producers

| Task | SubTask | When to be Performed? |
|---|---|--|
| Preparing Your Application for Migration | Determining WebCenter Consumer Application and Portlet Producer Compatibility | Always |
| Migrating Your Portlet Producer Application | Migrating Portlet Producers Created in JDeveloper | If your WebCenter application contains portlets from portlet producers developed in JDeveloper 10.1.3.x. |
| | Migrating Portlet Producers Created Outside of JDeveloper | If your WebCenter application contains portlets from portlet producers developed outside of JDeveloper. |
| | Migrating Preconfigured Portlet Producers | If your WebCenter application contains portlets from preconfigured portlet producers. |

Table 10–2 (Cont.) Task Flow for Migrating Portlet Producers

| Task | SubTask | When to be Performed? |
|---------------------------------|--|---|
| Performing Post Migration Tasks | Migrating Customizations | If your WebCenter application contains portlets other than the portlets from preconfigured portlet producers. |
| | Redeploying a Portlet Producer Application | Always |

10.4.1 Determining WebCenter Consumer Application and Portlet Producer Compatibility

In most cases, WebCenter consumer applications or portlet producer applications deployed to Oracle Application Server 10.1.3.x are compatible with applications upgraded to Oracle Fusion Middleware 11g. For example, a WebCenter consumer application that has been migrated and deployed to Oracle WebLogic Server in Oracle Fusion Middleware 11g may still use portlets from PDK-Java producer applications running on Oracle Application Server 10.1.3.x.

Oracle Fusion Middleware 11g supports backward compatibility for WebCenter consumer applications and portlet producer applications. Based on the compatibility supported, you may decide to migrate only your WebCenter consumer applications or portlet producer applications, or both. [Table 10–3](#) lists the compatibility between different versions of WebCenter consumer applications and portlet producer applications. In the table, the term 11g refers to an application originally created in Oracle WebCenter 11g or to a WebCenter 10.1.3.x application that has been migrated to Oracle WebCenter 11g.

[Table 10–3](#) shows that a WebCenter 10.1.3.x consumer application and a PDK-Java or a WSRP 1.0 WebCenter 11g portlet producer application are compatible. In such a case where backward compatibility is supported, you may choose not to migrate your WebCenter 10.1.3.x consumer application to Oracle WebCenter 11g. The table further shows that a WebCenter 11g consumer application and a PDK-Java or a WSRP 1.0 WebCenter 10.1.3.x portlet producer application are compatible. In such a case, you may choose not to migrate your 10.1.3.x portlet producer application.

However, WebCenter 10.1.3.x consumer applications are not compatible with WSRP 2.0 Oracle WebCenter 11g portlet producer applications. Similarly, WebCenter 11g consumer applications are not compatible with WSRP 2.0 WebCenter 10.1.3.x portlet producer applications. In such cases, all 10.1.3.x WSRP 2.0 applications must be migrated to Oracle Fusion Middleware 11g for these applications to work together.

Table 10–3 Compatibility Between Different Versions of WebCenter Consumer and Portlet Producer Applications

| Version of WebCenter Consumer Application | Version of Portlet Producer Application | Is this Combination of Applications Supported in Oracle Fusion Middleware 11g? |
|---|---|--|
| 10.1.3.x | PDK-Java 11.1.1.x | Yes |
| 10.1.3.x | WSRP 1.0 deployed to Oracle Application Server 11.1.1.x | Yes |

Table 10–3 (Cont.) Compatibility Between Different Versions of WebCenter Consumer and Portlet Producer Applications

| Version of WebCenter Consumer Application | Version of Portlet Producer Application | Is this Combination of Applications Supported in Oracle Fusion Middleware 11g? |
|---|---|--|
| 10.1.3.x | WSRP 2.0 deployed to Oracle Application Server 11.1.1.x | No |
| 11g | PDK-Java 10.1.3.x | Yes |
| 11g | WSRP 1.0 deployed to Oracle Application Server 10.1.3.x | Yes |
| 11g | WSRP 2.0 deployed to Oracle Application Server 10.1.3.x | No |

10.4.2 Migrating Your Portlet Producer Applications

To migrate portlet producer applications, you may need to perform various tasks depending on the type of portlet producers used in your WebCenter 10.1.3.x application. These tasks include:

- [Migrating Portlet Producers Created in JDeveloper](#)
- [Migrating Portlet Producers Created Outside of JDeveloper](#)
- [Migrating Preconfigured Portlet Producers](#)

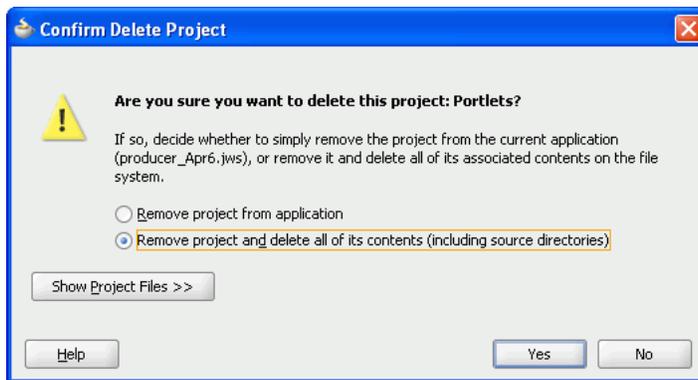
10.4.2.1 Migrating Portlet Producers Created in JDeveloper

If your WebCenter 10.1.3.x application contains portlets, and was created by using JDeveloper, then you must create a portlet producer application in Oracle WebCenter 11g to migrate portlet producers. You must also manually migrate the Portlets project of the WebCenter 10.1.3.x application to the newly created portlet producer application.

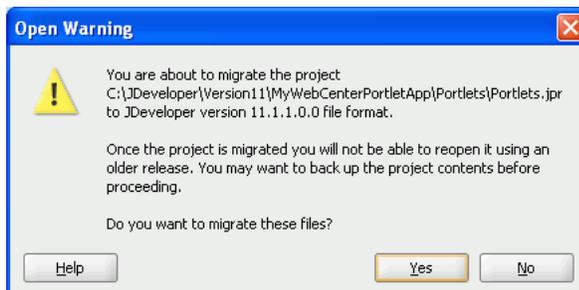
Note: If your WebCenter 10g application does not contain any portlet producers, its Portlets project will be empty.

To migrate a portlet producer application created in JDeveloper 10.1.3.x:

1. Create a portlet producer application in JDeveloper 11g by using the Portlet Producer Application template. For information, see the "Preparing Your Development Environment" chapter in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.
2. In the newly created portlet producer application, select the **Portlets** project.
3. From the **File** menu, select **Delete Project**.
4. Select the **Remove project and delete all of its contents (including source directories)** option and click **Yes**, as shown in [Figure 10–11](#).

Figure 10–11 Deleting the Portlets Project

5. Click **Yes**, if the message to confirm the delete operation displays.
6. Save your application.
7. On the file system, copy the `Portlets` directory from the backed up copy of your WebCenter 10.1.3.x application to the directory of the newly created portlet producer application.
8. In JDeveloper 11g, open the `portlets.jpr` of the portlet producer application. JDeveloper prompts whether you want to migrate the `Portlets` project.
9. Click **Yes** to migrate the 10.1.3.x **Portlets** project to the 11g portlet producer application, as shown in [Figure 10–12](#).

Figure 10–12 Migrating the Portlets Project

After the **Portlets** project is migrated, the Migration Status dialog prompts that the `portlets.jpr` has been migrated successfully.

10. Click **OK**.

10.4.2.2 Migrating Portlet Producers Created Outside of JDeveloper

If you have a JSR 168 Java portlet producer application or a PDK-Java portlet application packaged as a WAR file or as a WAR file within an EAR file, then you must update the application with Oracle WebCenter 11g specific elements, such as Oracle WebLogic Server application descriptors.

The following sections describe how to migrate a PDK-Java or JSR 168 application, which is in the form of an EAR or a WAR archive file, to Oracle WebCenter 11g.

10.4.2.2.1 Creating a JDeveloper Application from an EAR File If you have a JSR 168 portlet application or a PDK-Java portlet application packaged as a WAR file within an EAR

file, then to migrate that EAR to Oracle WebCenter 11g, you first need to create a JDeveloper 10.1.3.x application based on the archive file. You can then migrate the application by opening it in JDeveloper 11g.

To migrate an EAR file:

1. Open JDeveloper 10.1.3.x.

Note: You must use JDeveloper 10.1.3.x to prepare the EAR file for migration.

2. From the **File** menu, select **New**.
3. In the New Gallery dialog, expand **General**, select **Applications**, then **Application from EAR File**, and click **OK**.
4. Select **Application from EAR File**, and then click **OK**.
5. On the Welcome page in the Create Application from EAR File wizard, click **Next**.
6. On the Location page, in the **EAR File** field, enter the path to the EAR file.
7. Select the **Copy Files to Application** checkbox.
8. Select **Finish** to create the application in JDeveloper.
9. Save the application.
10. Save a backup copy of the application.
11. Open JDeveloper 11g.
12. Open the WebCenter 10.1.3.x application that you created from the EAR file.
13. Follow the instructions to migrate the application to JDeveloper 11g.
14. Save the application.

The application can now be redeployed.

10.4.2.2.2 Creating a JDeveloper Application from a WAR File To create a JDeveloper file from a JSR 168 or a PDK-Java application packaged as a WAR file:

1. Open JDeveloper 10.1.3.x.

Note: You must use JDeveloper 10.1.3.x to prepare the WAR file for migration.

2. From **File** menu, select **New**.
3. In the New Gallery dialog, expand **General**, select **Applications**, then **Application**, and click **OK**.
4. In the Create Application wizard, click **OK**.
5. In the Create Project wizard, click **Cancel** so that no new project is created in the application.
6. In the **Application Navigator**, select the newly created application.
7. From the **File** menu, select **New**.
8. In the New Gallery, expand **General**, select **Project**, then **Project from WAR File**, and click **OK**.

9. On the Welcome page in the Create Project from WAR File wizard, click **Next**.
10. On the WAR Location page, in the **WAR File** field, specify the path to the WAR file.
11. Click **Finish** to create the project.
12. Save the application.
13. Back up your application.
14. Open JDeveloper 11g.
15. Open the WebCenter 10.1.3.x application that you created from the WAR file.
16. Follow the instructions to migrate the application in JDeveloper 11g.
17. Save the application.

The application can now be redeployed.

10.4.2.3 Migrating Preconfigured Portlet Producers

Oracle WebCenter 11g provides various preconfigured portlet producers. These include OmniPortlet, Web Clipping, Rich Text Portlet, WSRP Parameter Form Portlet, sample WSRP portlet producers, and sample PDK-Java portlet producers.

Oracle WebCenter allows you to customize portlets. Customizations can include preferences such as user data and portlet and producer settings. Customizations can be stored in a database or a file system. Customizations related to preconfigured portlet producers are saved in the customization store of these producers, and not within your application projects. By default, customizations for preconfigured portlet producers are stored at the following location:

ORACLE_HOME/portal/portletdata

So, when you migrate a WebCenter application containing portlets from preconfigured portlet producers, you must ensure that portlet customizations are also migrated.

10.4.2.3.1 Migrating Customizations from the Default Location After installing JDeveloper 11g, when you start it the first time, it prompts whether you want to migrate settings from a previous release. If you select **Yes**, then portlet customizations of preconfigured portlet producers, along with other JDeveloper system properties, are automatically migrated from a previous JDeveloper installation to JDeveloper 11g. However, this migrates portlet customizations only from the default location, *ORACLE_HOME/portal/portletdata*.

If you choose not to migrate portlet customizations and select **No**, then you must migrate customizations later manually. The manual procedure involves copying the *JDEV10.1.3.x_ORACLE_HOME/portal/portletdata* directory to the *JDEV11_ORACLE_HOME/jdeveloper/portal/portletdata* directory.

10.4.2.3.2 Migrating Customizations from a Nondefault Location You may choose to store portlet customizations at a different location instead of the default location, *ORACLE_HOME/portal/portletdata*.

To migrate customizations from a nondefault location, you must perform either of the following tasks:

- Copy the customization store directory to the required path and configure your migrated portlet producer to point to the new location of the customization store. If customizations are stored in a database, then configure your migrated portlet producer to access that database.

- Configure your migrated portlet producer to access the customization store from the old location.

For information about storing customizations at a nondefault location, see [Appendix B.3, "Preconfigured Portlet Producer: Customization Store Location."](#)

10.4.3 Performing Post Migration Tasks for Portlet Producers

After migrating portlet producers, you migrate customizations and redeploy the portlet producer applications.

10.4.3.1 Migrating Customizations

After you have migrated Oracle PDK-Java and WSRP portlet producers, you must migrate their customizations if the customizations are not shared or accessible to the migrated portlet producers. These customizations are of portlet producers other than the preconfigured portlet producers.

To migrate such customizations, you can use the preference store migration utility. For information, see the "Portlet Preference Store Migration Utilities" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

10.4.3.2 Redeploying a Portlet Producer Application

You can deploy your migrated portlet producer application to any Oracle WebLogic Server managed server configured to support Oracle WebCenter portlet producers. For deployment, you can use Oracle Enterprise Manager Fusion Middleware Control, Oracle WebLogic Server Administration Console, or Oracle WebLogic Scripting Tool (WLST). For information, see the "Deploying Portlet Producer Applications" section in the *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter*.

You can also deploy portlet producer applications to an Oracle WebLogic Server instance directly from a development environment by using JDeveloper, provided that you have the required credentials to access the WebLogic server. For information, see the "Deploying a Portlet Application to an Oracle WebLogic Managed Server Instance" section in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

10.5 How to Migrate Wiki Data

Oracle WebCenter 11g includes Oracle WebCenter Wiki and Blog Server 11g that enables you to integrate wikis and blogs into WebCenter applications. When you migrate a WebCenter 10.1.3.x application containing Wiki and Blog services, you must also migrate the wiki data from Oracle Wiki Server 10.1.3.2 or Oracle WebCenter Wiki and Blog Server 10.1.3.4 to Oracle WebCenter Wiki and Blog Server 11g.

Note: You can migrate wiki data any time during the migration procedure of a WebCenter application.

If your application is not configured for the Wiki or the Blog service, then you can skip this section.

You can configure Oracle WebCenter Wiki and Blog Server 11g to use either a file system-based repository or a database repository. Oracle WebCenter Wiki and Blog Server 10.1.3.4 also supports both these repositories, but Oracle WebCenter Wiki 10.1.3.2 supports only a file system-based repository. In a file system-based repository, a domain is represented by a directory and the wiki pages within the domain are represented by files within that directory.

Note: The following features previously available in Oracle Wiki Server 10.1.3.x are no longer available on Oracle Wiki and Blog Server 11g:

- Tags on wiki pages
- Discussions for wiki pages

To integrate the tags and discussions features into your wiki pages, you can use the Tags and Discussions services supported by Oracle WebCenter 11g.

10.5.1 Migrating Content to a File System-Based Repository

To migrate content from Oracle Wiki Server 10.1.3.2 or Oracle WebCenter Wiki and Blog Server 10.1.3.4 file system-based repository to Oracle WebCenter Wiki and Blog Server 11g file system-based repository:

1. On the wiki and blog server 10.1.3.2 or 10.1.3.4, navigate to the `$SOURCE_WIKI_HOME/pages` directory, where `$SOURCE_WIKI_HOME` refers to the deployment directory of the wiki and blog server.
2. Copy the files in the `pages` directory to the `$WIKI_HOME/pages` directory on Oracle WebCenter Wiki and Blog Server 11g, where `$WIKI_HOME` refers to the deployment directory of the wiki and blog server 11g.

If Oracle WebCenter Wiki and Blog Server 11g is configured to use the file system-based repository, then it recognizes the newly copied directories as new domains, and the files within those directories as new wiki pages.

Note: Attachments associated with wiki pages are stored in a separate directory named `attachments`. In addition to the `pages` directory, you must also copy the `attachments` directory to your Oracle WebCenter Wiki and Blog Server 11g.

By default, the attachment feature is disabled on Oracle WebCenter Wiki and Blog Server 11g. You can enable this feature by using the Administration mode.

3. Restart the server to which Oracle WebCenter Wiki and Blog Server 11g is deployed.

10.5.2 Migrating Content to a Database Repository

To migrate wiki content from Oracle Wiki Server 10.1.3.2 or Oracle WebCenter Wiki and Blog Server 10.1.3.4 file system-based repository to Oracle WebCenter Wiki and Blog Server 11g database repository:

1. Migrate the wiki content to the file-based repository on Oracle WebCenter Wiki and Blog Server 11g. For information, see [Section 10.5.1, "Migrating Content to a File System-Based Repository."](#)

This gets the wiki 10.1.3.x files on Oracle WebCenter Wiki and Blog Server 11g.

2. Log on to Oracle WebCenter Wiki and Blog Server 11g as an administrator.
3. Click the **Administration** link to access the Administration mode. ([Figure 10-13](#))

Figure 10–13 Administration Link on Oracle WebCenter Wiki and Blog Server 11g

4. Under **Domain Related**, click **Import Domain** to import the domain directory into the database.

Note: You can import only one domain at a time.

5. On the Import Domain page, specify the following details: domain name, description, start page name, and the location of the Oracle Wiki Server 10.1.3.2 domain that you intend to import. (Figure 10–14)

Figure 10–14 Importing a Domain

6. Click **Import Domain**.
7. Repeat step 3 and 4 to import other domains.
8. Copy the `$SOURCE_WIKI_HOME/attachments` directory from Oracle Wiki Server 10.1.3.2 to the `$WIKI_HOME/attachments` directory on Oracle WebCenter Wiki and Blog Server 11g.
9. Restart the server to which Oracle WebCenter Wiki and Blog Server 11g is deployed.

To migrate content from Oracle WebCenter Wiki and Server 10.1.3.4 configured to use a database-based repository, you can either point to the same Oracle database (the schemas are the same), or you can export the Oracle WebCenter Wiki and Server 10.1.3.4 database by using Oracle tools and then import it into a new database.

Example: Upgrading and Redeploying SOA Order Booking

This appendix provides the step-by-step procedure for upgrading a complex Oracle SOA Suite (SOA) application from Oracle Application Server 10g Release 3 (10.1.3.3) to Oracle Fusion Middleware Release 11 (11.1.1.0.0).

This procedure is meant to serve as an overall example of the Oracle SOA Suite environment and application upgrade process.

Specifically, this appendix contains the following topics:

- [About the SOA Order Booking Application](#)
- [Task 1: Verify That the SOA Order Booking and Its External Partners are Up and Running](#)
- [Task 2: Install and Upgrade the Oracle Fusion Middleware SOA Suite](#)
- [Task 3: Use the Oracle WebLogic Suite Administration Console to Configure Required Resources](#)
- [Task 4: Install Oracle JDeveloper 11g](#)
- [Task 5: Set Up Connections in Oracle JDeveloper Release 11](#)
- [Task 6: Open the SOA Order Booking Application in Oracle JDeveloper 11g and Migrate the Application Projects](#)
- [Task 7: Compile and Deploy Each of the Application Projects on Oracle Fusion Middleware 11g](#)
- [Task 8: Test the SOA Order Booking Application in Fusion Middleware Control](#)

A.1 About the SOA Order Booking Application

Oracle provided the SOA Order Booking application with Oracle Application Server 10g Release 3 (10.1.3.1.0) as a sample application that demonstrates many of the key features and components of the Oracle Application Server 10g Release 3 (10.1.3.1.0) release.

The SOA Order Booking application is described in detail in the *Oracle SOA Tutorial* and in the Oracle SOA Suite Quick Start Guide, which are both available in the Oracle Application Server 10g Release 3 (10.1.3.1.0) Documentation Library on the Oracle Technology Network:

<http://www.oracle.com/technology/documentation/index.html>

The Oracle Application Server 10g Release 3 (10.1.3.4) patch set introduced a set of improvements and fixes to common problems. In addition, Oracle released a new version of the SOA Order Booking that was tested and improved for 10g Release 3 (10.1.3.4).

This appendix assumes you are familiar with the SOA Order Booking application and the technologies it demonstrates.

Figure A-1 shows the login page of the SOA Order Booking Demo, deployed on Oracle Application Server 10g Release 3 (10.1.3.4).

Figure A-1 SOA Order Booking Application Deployed on 10g Release 3 (10.1.3.4)



A.2 Task 1: Verify That the SOA Order Booking and Its External Partners are Up and Running

Before you can migrate the SOA Order Booking application to 11g, you must review the following prerequisites:

- [Task 1a: Verify That the SOA Order Booking Is Successfully Deployed on Oracle Application Server 10g Release 3 \(10.1.3.4\)](#)
- [Task 1b: Verify That SOA Order Booking Is Currently Running on Oracle Application Server 10g Release 3 \(10.1.3.4\)](#)
- [Task 1c: Verify That All External Partners Are Available](#)

A.2.1 Task 1a: Verify That the SOA Order Booking Is Successfully Deployed on Oracle Application Server 10g Release 3 (10.1.3.4)

Before you can migrate the SOA Order Booking application to 11g, it is assumed that:

- You have already installed and configured Oracle Application Server 10g Release 3 (10.1.3.1.0), and you have applied the 10g Release 3 (10.1.3.4) patch set.
- You have already installed and configured Oracle JDeveloper 10g Release 3 (10.1.3.4).
- You have downloaded the updated 10g Release 3 (10.1.3.4) version of the SOA Order Booking demo application.

You can download the 10g Release 3 (10.1.3.4) SOA Order Booking demo application from the following location on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/products/ias/bpel/>

- You have used the instructions in the 10g Release 3 (10.1.3.1.0) *Oracle SOA Suite Tutorial* or *Oracle SOA Suite Quick Start Guide* to deploy the SOA Order Booking application on Oracle Application Server 10g Release 3 (10.1.3.4).

This includes installing the database schemas for the SOA Order Booking 10g Release 3 (10.1.3.4) application in an available database.

A.2.2 Task 1b: Verify That SOA Order Booking Is Currently Running on Oracle Application Server 10g Release 3 (10.1.3.4)

Before you migrate the SOA Order Booking application to 11g, test the application on 10g Release 3 (10.1.3.4) and make sure that it is functioning properly. This ensures that the database where the SOA Order Booking schemas are installed is up and running and the external services and external partners are up and running.

For example, you should be able to log in to the application, browse the list of products (Figure A-2), and place an order. You should also be able to monitor the BPEL processes using the BPEL Worklist Manager.

Figure A-2 Browsing Items in the SOA Order Booking Application

The screenshot shows the 'Global Company' online shopping application. The header includes the company logo, 'Your Online Shopping MegaStore', and navigation links for 'Home', 'Browse Items', 'Orders', and 'Cart'. The user is logged in as 'sking@soademo.org'. The main content area is titled 'Browse and Select Items' and displays a table of products. The table has columns for 'Select', 'Name', 'Product Id', 'Description', 'Category', 'Price', and 'Producer'. There are 10 products listed, including HD Television, PlayStation 2 Video Game System, Treo 650 Phone/PDA, Treo 700w Phone/PDA, Tungsten E PDA, Xbox Video Game System, Xbox 360 Video Game System, Playstation Portable, Nintendo DS, and Razer Cellular Phone.

| Select | Name | Product Id | Description | Category | Price | Producer |
|--------------------------|---------------------------------|------------|--|-----------|-----------|-------------|
| <input type="checkbox"/> | HD Television | 128723 | A High-Definition Flat Screen Television that uses DLP Technology. | Video | \$1999.00 | Azimuth |
| <input type="checkbox"/> | PlayStation 2 Video Game System | 223453 | Playstation II Interactive Video Gaming System. | Games | \$199.00 | Playstation |
| <input type="checkbox"/> | Treo 650 Phone/PDA | 429832 | A fully featured Treo Phone and PDA combination. | Handhelds | \$299.00 | Treo |
| <input type="checkbox"/> | Treo 700w Phone/PDA | 423322 | A fully featured Treo Phone and Windows CE Handheld. | Handhelds | \$399.00 | Treo |
| <input type="checkbox"/> | Tungsten E PDA | 592039 | Tungsten handheld Personal Digital Assistant. | Handhelds | \$195.00 | Tungsten |
| <input type="checkbox"/> | XBox Video Game System | 621135 | XBox Interactive Video Gaming System | Games | \$159.00 | XBox |
| <input type="checkbox"/> | XBox 360 Video Game System | 218407 | Next generation Xbox 360 Interactive Video Gaming System. | Games | \$299.00 | XBox |
| <input type="checkbox"/> | Playstation Portable | 124039 | Playstation Portable HandHeld Video Gaming System. | Handhelds | \$199.00 | MyTag |
| <input type="checkbox"/> | Nintendo DS | 606945 | Nintendo DS Handheld Gaming System | Handhelds | \$129.00 | DS |
| <input type="checkbox"/> | Razer Cellular Phone | 927384 | Ultra-sleek Razer Cellular Phone | Handhelds | \$229.00 | Razer |

© Oracle Corp., 2006 [Contact Us](#) [About this sample](#)

For more information, see "Running and Monitoring the SOA Order Booking Application" in the Oracle Application Server 10g Release 3 (10.1.3) *Oracle SOA Suite Quick Start Guide*.

The *Oracle SOA Suite Quick Start Guide* is included in the Oracle Application Server 10g Release 3 (10.1.3) documentation library, which is available on the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/documentation/>

A.2.3 Task 1c: Verify That All External Partners Are Available

Before you can upgrade the SOA Order Booking application to 11g, you must ensure that the external partners (in the form remote Web services) are up and running and available.

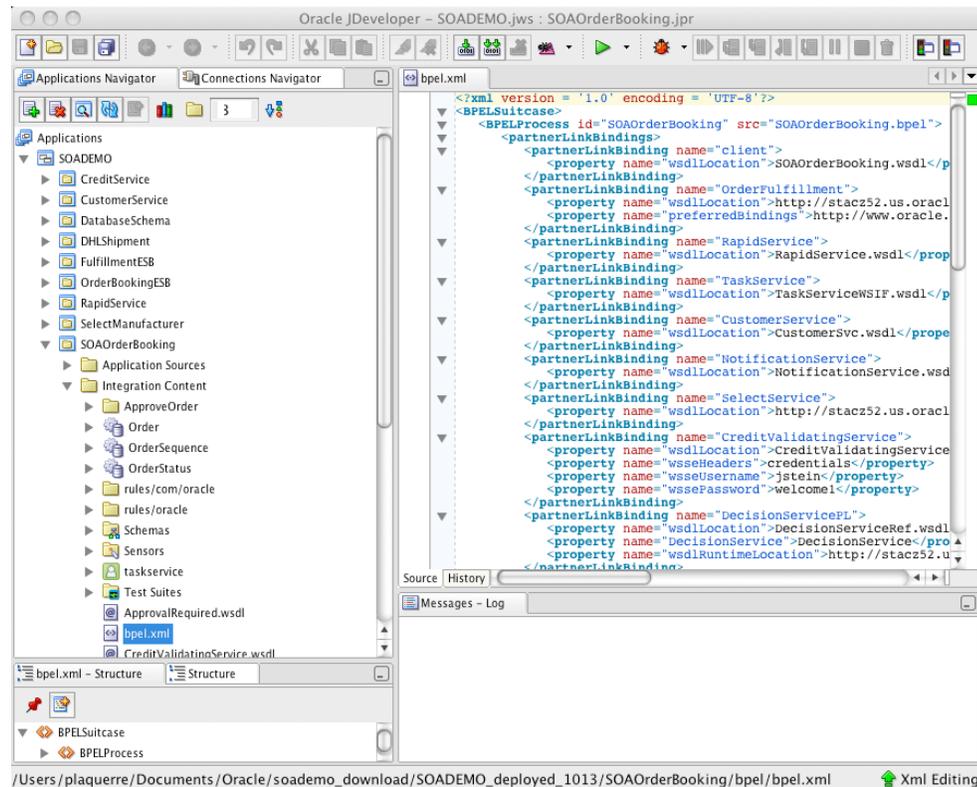
Oracle JDeveloper 11g must be able to access these external partners, as well as the Oracle BPEL Process Manager and Oracle Enterprise Service Bus services deployed on 10g; otherwise, it can not properly migrate the projects in your application to 11g.

To verify that the required Web services are available, you should review each of the projects in your Oracle SOA Suite applications before you migrate them to 11g. For example, you can review the Oracle BPEL Process Manager configuration files for each project and verify that the Web services URIs referenced in those configuration files are valid and working.

For example, in the SOA Order Booking application, you can review the contents of the SOAOrderBooking project, as follows:

1. Open the SOA Order Booking 10g Release 3 (10.1.3.4) application in Oracle JDeveloper 10g Release 3 (10.1.3).
2. Expand the **SOAOrderBooking** project, and then expand the **Integration Content** folder.
3. Expand the **bpel** folder and double-click the **bpel.xml** file (Figure A–3).

Figure A–3 Selecting a *bpel.xml* File to Test WSDL URLs



4. Inside the *bpel.xml* file, search for the following entry:

```
<partnerLinkBinding name="OrderFulfillment">
  <property name="wsdlLocation">
```

```

http://host.domain:port/esb/wsdl/Fulfillment/OrderFulfillment?wsdl
    </property>
    <property name="preferredBindings">
        http://www.oracle.com/esb/
    </property>
</partnerLinkBinding>

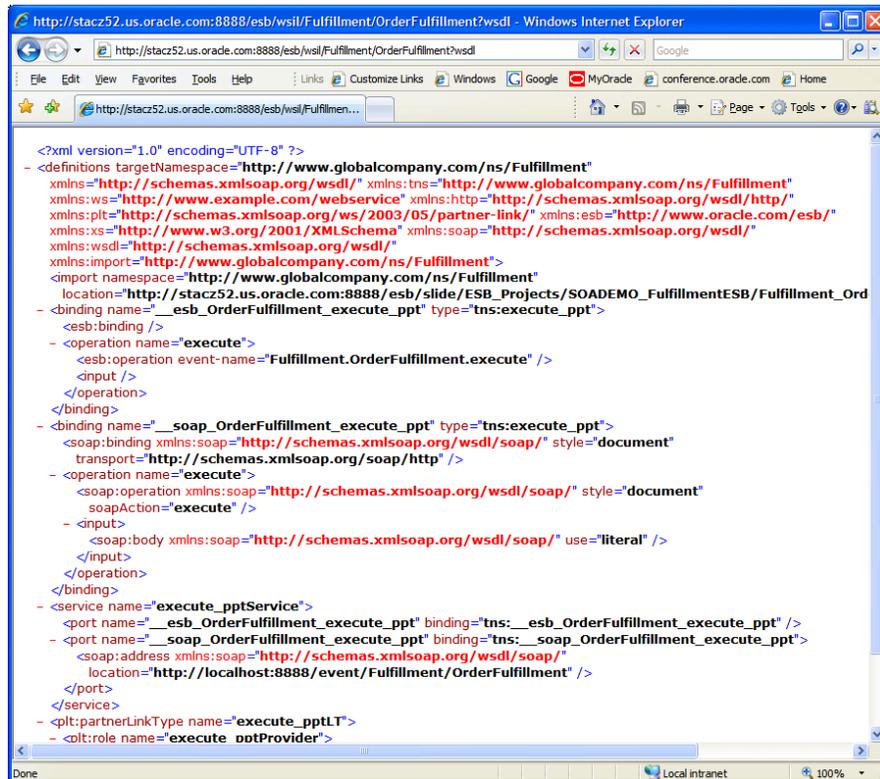
```

5. Copy the URL in the `wsdlLocation` element to your browser and verify that the URL is valid and displays the output shown in Figure A-4.
6. If the Web Services URL do not display correctly, troubleshoot and solve the problem before attempting to migrate the application to 11g.

For example, make sure the database where the SOA Order Booking schemas are installed is up and running and that all components of the application have been deployed successfully.

7. Repeat this procedure for the other URLs in the `bpel.xml` file.
8. Perform a similar test for the WSDL URLs in the following projects:
 - SOAOrderBooking:
 - CreditValidatingService.wsdl
 - CustomerSvc.wsdl
 - RapidService.wsdl
 - OrderBookingESB:
 - OrderBooking_OrderBookingProcess.esbsvc
 - OrderBooking_OrderBookingService.esbsvc

Figure A-4 OrderFulfillment WSDL URL in Internet Explorer



A.3 Task 2: Install and Upgrade the Oracle Fusion Middleware SOA Suite

After you verify that the SOA Order Booking application is running successfully on your Oracle Application Server 10g Release 3 (10.1.3) environment, you can then install a new Oracle Fusion Middleware 11g environment and upgrade the environment.

Follow the instructions in [Chapter 6, "Upgrading an Oracle SOA Suite Environment"](#) to prepare the Oracle Fusion Middleware environment for the SOA Order Booking migration and redeployment.

This task includes:

- Installing and configuring the Oracle SOA Suite 11g middle tiers
- Upgrading the Oracle SOA Suite schemas to 11g

A.4 Task 3: Use the Oracle WebLogic Suite Administration Console to Configure Required Resources

The SOA Order Booking demo application requires that a data source and a set of JMS resources be available on the Oracle WebLogic Server managed server where SOA Order Booking is deployed.

Refer to the following sections for information on how to use the Oracle WebLogic Server Administration Console to create these resources:

- [Task 3a: Create the SOADEMO Data Source](#)
- [Task 3b: Configure the DbAdapter Application](#)
- [Task 3c: Create the SOA Order Booking JMS Resources](#)

A.4.1 Task 3a: Create the SOADEMO Data Source

Before you can deploy the SOA Order Booking sample application on Oracle WebLogic Server, you must first define an Oracle WebLogic Server data source that references the database where the SOA Order Booking schemas are installed.

These are the same schemas you installed previously, when you deployed the SOA Order Booking demo application on Oracle Application Server 10g Release 3 (10.1.3.4), as described in [Section A.2, "Task 1: Verify That the SOA Order Booking and Its External Partners are Up and Running"](#).

To create the data source:

1. Log in the Administration Console.
2. In the Domain Structure tree, expand **Services > JDBC**, then select **Data Sources**.
3. On the Summary of Data Sources page, click **New**.
4. Oracle WebLogic Server displays the JDBC Data Source Properties page.

Enter the following properties on this page:

- Name: `soademoDS`
 - JNDI Name: `jdbc/soademoDS`
 - Database Type: **Oracle**
 - Database Driver: **Oracle's Driver (Thin XA)**
5. Click **Next** to display the Transaction Options page:

- Select **Supports Global Transactions**.
 - Select **Logging Last Resource**.
6. Click **Next** to display the Connection Details page ().
 - **Database Name:** *Oracle_database_system_idenfifer_(SID)*
 - **Host Name:** *Host_where_the_database_is_installed*
 - **Port:** *Database_listening_port_for_example_1521*
 - **Database User Name:** *soademo*
 - **Password and Confirm Password:** *soademo*
 7. Click **Next** to test the database connection.
 8. Click **Next** to display the Select Targets page, and select the **soa_server1**.
 9. Click **Finish** to deploy the data source on **soa_server1**.

A.4.2 Task 3b: Configure the DbAdapter Application

To connect to the database, the SOA Order Booking application requires some changes to the default DbAdapter application that is installed and configured as part of a Oracle SOA Suite 11g domain.

Refer to the following sections for instructions on how to configure the DbAdapter for the SOA Order Booking application:

- [Creating a New Outbound Connection Pool for the DbAdapter Application](#)
- [Configuring the New Connection Pool for the DbAdapter Application](#)

A.4.2.1 Creating a New Outbound Connection Pool for the DbAdapter Application

Use the following procedure to create a new outbound connection pool in the DbAdapter application:

1. Create a directory where the deployment plan for the DbAdapter application can be saved.

For example:

```
MW_HOME/Oracle_SOA1/soa/DBPlan
```

2. In the Domain Structure tree, click **Deployments**.
3. Click the DbAdapter application (click the name, not the check box).
4. Click the **Configuration** tab, and then click the **Outbound Connection Pools** tab.
5. Click **New**.
6. Select the radio button for `javax.resource.cci.ConnectionFactory` and click **Next**.
7. Enter the JNDI Name as follows: `eis/DB/soademoDS`

Note that this JNDI name must match the value you enter in your database connection, which you create when upgrading your application with Oracle JDeveloper later in this procedure.

8. Click **Finish**.

At this point, you should be asked for the deployment plan location.

9. Click on the path to the directory you created earlier and enter the deployment plan name `Plan.xml`.
10. Verify that the correct deployment plan is selected.
11. Click **Next** and then click **Finish**.

A.4.2.2 Configuring the New Connection Pool for the DbAdapter Application

Use the following procedure to configure the new connection pool so it references the data source:

1. Click the **Configuration** tab, expand the connection factory, and click your new connection pool (click on the name, not the check box).
2. In the Properties table, select the box to the right of **xADDataSource**.
The edit box appears.
3. Enter the name of the data source name that you created previously:
`jdbc/soademoDS`
4. Press ENTER to apply the value and click **Save**.
You must use the ENTER key for the value to be entered in the field, and you must click **Save** to save your changes.
5. Return to the Deployments page and click **Deployments** in the left navigation bar.
6. Select the check box next to **DbAdapter**.
7. Click **Update**.
8. Select **Redeploy this application** and confirm the deployment plan location, and click **Finish**.
9. Verify your changes to ensure that the configuration is correct; otherwise, deployment errors will occur later:
 - Confirm that the connection pool is added, by returning to the **DbAdapter > Configuration > Outbound Connection Pools** page and expanding the connection factory.
 - Confirm the value of the **xADDataSource** property that you entered previously.
 - Return to the DbAdapter Outbound Connection Pools tab, and review the connection factory property value for the `eis/DB/soademoDS` connection pool.

A.4.3 Task 3c: Create the SOA Order Booking JMS Resources

The SOA Order Booking sample application uses message queues to transmit order and fulfillment details. As a result, you must configure a JMS queue before you can deploy the application successfully.

You can use WLST commands and scripts to configure JMS, but these instructions show how to use the Oracle WebLogic Administration Console to configure JMS.

The following sections describe how to configure JMS for the SOA Order Booking Demo sample application:

- [Creating the JMS Queue for the SOA Order Booking Demo](#)
- [Creating a JMS Connection Factory for the SOA Order Booking Demo](#)

- [Creating an Outbound Connection Pool for the JmsAdapter Application](#)
- [Configuring the JmsAdapter Connection Pool](#)
- [Redeploying the Modified JmsAdapter Application](#)

A.4.3.1 Creating the JMS Queue for the SOA Order Booking Demo

To create the JMS destinations required for the SOA Order Booking Demo:

1. In the Oracle WebLogic Administration Console, navigate to the Summary of JMS Modules page and click the **SOAJMSModule** JMS module.

This module is created and configured during the Oracle SOA Suite 11g installation and configuration.

2. On the JMS Module Configuration page, click **New**.
3. From the list of resource types, select **Queue**.
4. On the JMS Destination Properties page, enter a name and JNDI location for the the destination queue.

For the purposes of this procedure, enter the following values:

Name: demoFulfillmentQueue

JNDI Name: jms/demoFulfillmentQueue

5. Click **Next**.
6. Select **SOASubDeployment** from the **SubDeployments** drop-down menu.
7. Select the **SOAJMSServer** JMS Server and click **Finish**.

A.4.3.2 Creating a JMS Connection Factory for the SOA Order Booking Demo

To create the connection factories required by the SOA Order Booking Demo sample application:

1. In the Oracle WebLogic Administration Console, navigate to the Summary of JMS Modules page and click the **SOAJMSModule** JMS module.
2. On the JMS Module Configuration page, click **New**.
3. From the list of resource types, select **Connection Factory** and click **Next**.
4. On the Connection Factory Properties page, enter a name and a JNDI location for the JMS connection factory.

Use the following values:

Name: demoCF

JNDI Name: jms/demoCF

5. Click **Next**.
6. Click **Finish**.

You should see both the queue and the connection factory listed in the Summary of Resources.

A.4.3.3 Creating an Outbound Connection Pool for the JmsAdapter Application

Use the following procedure to create a new connection pool. The connection pool is configured in the JmsAdapter application and uses a Deployment Plan:

1. Create a directory for the JmsAdapter application deployment plan.

For example:

`MW_HOME/Oracle_SOA1/soa/JMSPlan`

2. In the Oracle WebLogic Server Administration Console, click **Deployments**.
3. Click **JmsAdapter** (click the name, not the check box).
4. Click the **Configuration** tab and then the **Outbound Connection Pools** tab
5. Click **New**.
6. Select the factory `oracle.tip.adapter.jms.IJmsConnectionFactory`.
7. Click **Next**.
8. Enter `eis/demo/Queue`.
9. Click **Finish**.

At this point, you should be asked for the deployment plan location.

10. Click the path to the directory you created earlier and enter `Plan.xml` for the plan name.

If there is a plan file selected with the radio button at the bottom of the screen, the one you enter will take precedence.
11. Click **Next** and then click **Finish**.

A.4.3.4 Configuring the JmsAdapter Connection Pool

Use the following procedure to configure the connection pool you created in the previous section:

1. In the Oracle WebLogic Server Administration Console, click **Deployments**.
2. Click **JmsAdapter** (click the name, not the check box).
3. Click the **Configuration** tab and then the **Outbound Connection Pools** tab
4. Expand the `oracle.tip.adapter.jms.IJmsConnectionFactory` connection factory.
5. Click `eis/demo/Queue`.
6. Change the value of the **ConnectionFactoryLocation** property to `jms/demoCF` and press ENTER.

You must use the ENTER key for the value to be entered in the field.

7. Click **Save**.

You must use the Save button to save the value.

A.4.3.5 Redeploying the Modified JmsAdapter Application

Use the following procedure to redeploy the JmsAdapter application:

1. Click **Deployments** in the Oracle WebLogic Server Administration Console navigation tree.
2. Select the check box next to **JmsAdapter**.
3. Click **Update** at the top of the Deployment table.
4. Verify that the correct deployment plan is selected.

5. Select **Redeploy this application** and confirm the deployment plan location, and click **Finish**.

A.5 Task 4: Install Oracle JDeveloper 11g

You can install Oracle JDeveloper 11g from the ZIP file on the Oracle Fusion Middleware Companion CD or by downloading the Oracle JDeveloper Studio Edition Version 11.1.1.1.0.0 ZIP file from the Oracle Technology Network (OTN):

<http://www.oracle.com/technology/software/products/jdev/index.html>

You can install JDeveloper on the same machine that is running Oracle Fusion Middleware 11g, or you can install it on a different machine from the one where Oracle Fusion Middleware is running.

For complete information, see *Oracle Fusion Middleware Installation Guide for Oracle JDeveloper*.

A.6 Task 5: Set Up Connections in Oracle JDeveloper Release 11

In Oracle JDeveloper Release 11 (11.1.1.0.0), you set up connections to the database, to Oracle Application Server. These connections enable you to view the data in the SOADEMO schema, and to deploy the applications to Oracle Fusion Middleware from JDeveloper.

This task includes the following subtasks:

- [Task 5a: Start Oracle JDeveloper 11g](#)
- [Task 5b: Verify That You Have the Required Oracle JDeveloper Extensions](#)
- [Task 5c: Create a Connection to the SOADEMO Schema](#)
- [Task 5d: Create a Connection to Oracle WebLogic Server](#)

A.6.1 Task 5a: Start Oracle JDeveloper 11g

To start Oracle JDeveloper Studio on Windows, use any of the following methods:

- From your **Start Menu**, select **All Programs**, then select **Oracle Fusion Middleware 11.1.1.x.x**, and then select **Oracle JDeveloper Studio 11.1.1.x.x**.
- You can also start Oracle JDeveloper from the command line by running one of the following commands:

```
install_dir\jdeveloper\jdeveloper.exe
install_dir\jdeveloper\jdev\bin\jdevw.exe
install_dir\jdeveloper\jdev\bin\jdev.exe (to display a console window
                                         for internal diagnostic information)
```

To start Oracle JDeveloper on UNIX systems, run the file:

```
install_dir/jdeveloper/jdev/bin/jdev
```

A.6.2 Task 5b: Verify That You Have the Required Oracle JDeveloper Extensions

To upgrade your Oracle SOA Suite applications to 11g, you must have the Oracle SOA Composite Editor extension for Oracle JDeveloper 11g.

To verify that the extension is installed, select **About** from the Oracle JDeveloper **Help** menu, and click the **Version** tab. You should see an entry in the list of components called SOA Composite Editor.

If this component does not appear on the Version tab of the About dialog box, close the About dialog and select **Check for Updates** from the **Help** menu. Use the Check for Updates wizard to locate and install the latest version of the SOA Composite Editor extension.

A.6.3 Task 5c: Create a Connection to the SOADEMO Schema

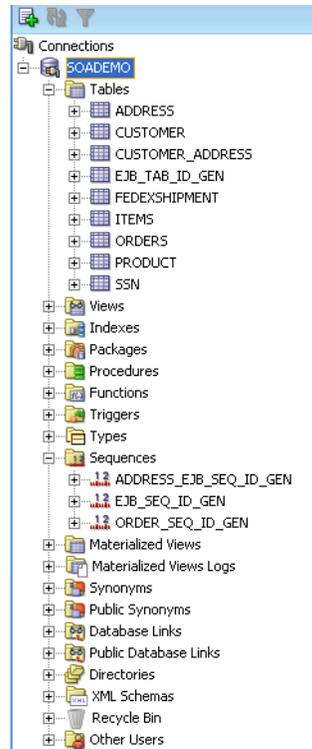
In JDeveloper, create a database connection to the SOADEMO schema:

Note: In this tutorial, the database connection is named **SOADEMO**. You can use a different name if you want, but using the same naming conventions will make it easier to follow the instructions.

1. In Oracle JDeveloper, select **View > Database Navigator**.
2. Right-click the **IDE Connections** node and select **New Connection**.
3. In the resulting dialog box, enter the following values:
 - **Connection Name:** enter **SOADEMO**.
 - **Connection Type:** select **Oracle (JDBC)**.
 - **Username:** enter **SOADEMO**.
 - **Password:** enter **soademo**.
 - **Role:** leave blank.
 - **Save Password:** select the check box.
 - **Enter Custom JDBC URL:** clear the check box.
 - **Driver:** select **thin**.
 - **Host Name:** enter the name of the machine running the database where you installed the SOADEMO schema.
 - **JDBC Port:** enter the port number for the database. The default value is 1521.
 - **SID:** enter the system identifier for the database. The default value is ORCL.

If you are unsure about the database connection values, check with your database administrator.
4. Click **Test Connection**.

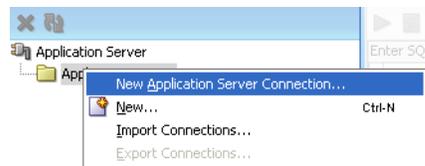
If the test is not successful, check that the database is available and that the connection values are correct.
5. Click **Finish**. The connection appears below the **Database** node in the Connection Navigator.
6. You can now examine the schema from Oracle JDeveloper. In the Connection Navigator, expand **Database > soademo > SOADEMO**. Expand the **Sequences** and **Tables** nodes and verify that the tables match those listed in Section 1.4, "Contents of the SOADEMO Schema" of the 10g Release 3 (10.1.3.1.0) *Oracle SOA Tutorial*.



A.6.4 Task 5d: Create a Connection to Oracle WebLogic Server

In JDeveloper, create a connection to the Oracle WebLogic Server domain where you installed and configured Oracle SOA Suite 11g:

1. In Oracle JDeveloper, select **View > Application Server Navigator**.
2. Right-click the **Application Servers** folder under the Application Server node and select **New Application Server**.



3. Click **Next** on the Welcome page of the Application Server Connection Wizard.
4. Enter the following values:

- **Connection Name:** enter `SoademoApplicationServer`.
- **Connection Type:** select **WebLogic 10.3**.

Click **Next**.

5. In Step 2, Authentication, enter the following values:
 - **Username:** Enter **weblogic**. This is the name of the default Oracle WebLogic Server administration user.
 - **Password:** Enter the password for the `weblogic` user. This is the password that you set when you configured the Oracle SOA Suite 11g domain.

6. Click **Next** and enter the host, port, and name of the Oracle WebLogic Server domain you installed and configured in [Section A.3, "Task 2: Install and Upgrade the Oracle Fusion Middleware SOA Suite"](#).
7. Click **Next**. The Test Connection page appears.
8. Click **Test Connection**.
9. Click **Finish**.

The connection appears below the **Application Servers** node in the Connection Navigator.

A.7 Task 6: Open the SOA Order Booking Application in Oracle JDeveloper 11g and Migrate the Application Projects

The following sections describe how you can open an application created in a previous version of Oracle JDeveloper and use the Migration Wizard to upgrade the application to Oracle Fusion Middleware 11g:

- [Opening and Migrating the Application](#)
- [Viewing the Migration Log Files](#)

Important: The SOA Order Booking application and its projects rely upon external partners. These remote Web services must be up and running and available to the SOA Order Booking application before you migrate the application to Oracle Fusion Middleware 11g.

You cannot successfully open the application in Oracle JDeveloper 11g if the external partner URI's are not available. For more information, see [Task 1: Verify That the SOA Order Booking and Its External Partners are Up and Running](#).

A.7.1 Opening and Migrating the Application

To open and migrate the application to Oracle JDeveloper 11g:

1. Make a backup copy of the soademo folder that contains the 10g Release 3 (10.1.3.4) SOA Order Booking Demo application source files.

The contents of the soademo folder will be modified during the upgrade. The backup copy allows you to restore the original 10g Release 3 (10.1.3) version of the demo if necessary.

Note that for this procedure, the working folder (the folder containing the files you will upgrade) is referred to as the SOADEMO_HOME directory.

2. Review the contents of the SOADEMO_HOME directory and remove the following file if it exists:

`SOADEMO_HOME/SOAOrderBooking/bpel/rules/com`

This includes the subdirectories under the `bpel/rules/com/` directory.

3. Select **Open** from the File menu and select the Oracle JDeveloper 10g Release 3 (10.1.3.1.0) `soademo.jws` file.

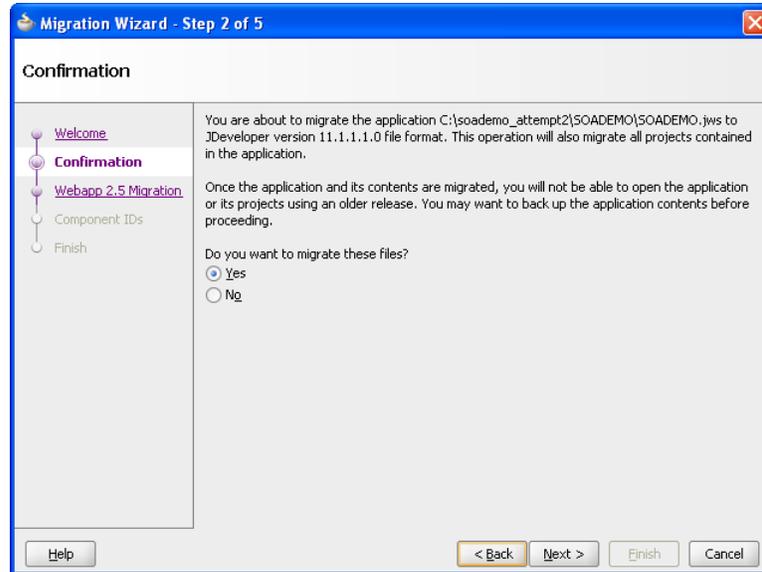
Oracle JDeveloper displays Step 1, the Welcome page, of the Migration Wizard.

Click **Next**.

4. In Step 2 (Figure A-5), select **Yes** to migrate the project files to Oracle JDeveloper Release 11 (11.1.1.0.0).

Click **Next**.

Figure A-5 Step 2 of the Oracle JDeveloper 11g Migration Wizard

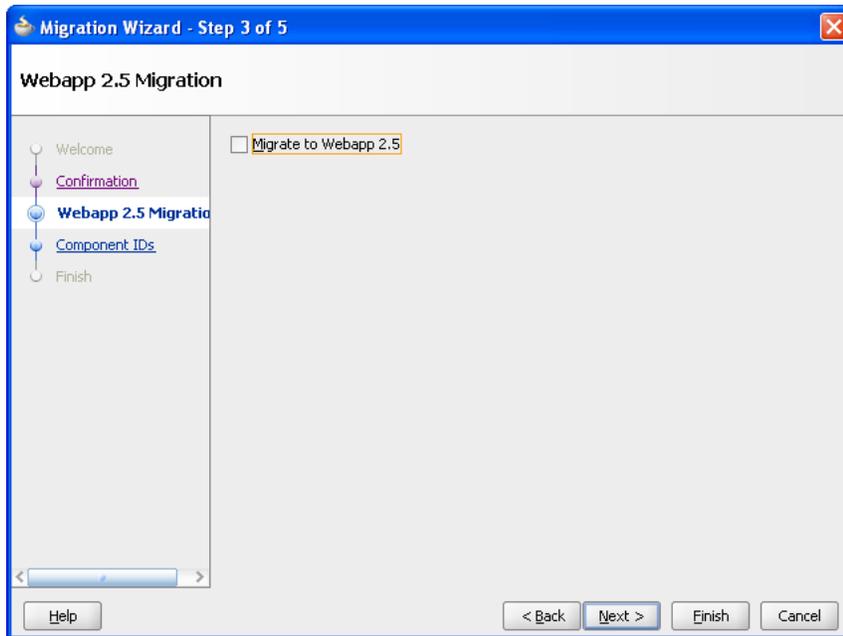


5. In Step 3, clear the **Migrate to Webapp 2.5** and **Convert JSTL 1.0/1.1 to 1.2**.

Note: Clearing the **Migrate to WebApp 2.5** check box is important because the SOA Order Booking demo application uses Web services that you will upgrade manually later in this procedure. To be sure the Migration wizard does not attempt to upgrade these automatically, be sure to clear the **Migrate to WebApp 2.5** check box.

Click **Next**.

Figure A-6 Clearing the WebApp 2.5 Option in the Oracle JDeveloper Migration Wizard



6. In Step 4, review the items on the page, accept the default selections, and click **Next**.
7. In Step 5, review the setup options you selected, and click **Finish** to upgrade the application to Oracle JDeveloper 11g.

JDeveloper displays a Migration Status dialog box that indicates the progress of the migration. In addition, information about the migration is displayed in the logging pane in the main Oracle JDeveloper window.

8. When the migration is complete, click **Finish**.

A.7.2 Viewing the Migration Log Files

After the migration, you can review the log files generated during the application migration. The log files describe the various changes that Oracle JDeveloper 11g makes to the projects as they are opened and migrated to Oracle JDeveloper 11g.

The log files are located in the following directory:

```
JDEV_HOME\upgrade\logs\
```

You can identify the different log files for each application by viewing the date and time stamp in each log file name. For example:

```
SOAOrderBooking2009-04-29-12-46-35PM.log
```

In addition, the Messages tab in the Oracle JDeveloper main window displays links to the individual log files for each project that was migrated. Click any of the links to display a migration log file.

A.8 Task 7: Compile and Deploy Each of the Application Projects on Oracle Fusion Middleware 11g

After you have opened the SOADEMO application and migrated the projects to Oracle JDeveloper 11g, you can now recompile each of the projects and redeploy them one by one.

Review, compile, and redeploy each project in the following order:

- [About the CreditService, RapidService, and CustomerService Projects](#)
- [SelectManufacturer](#)
- [DHLShipment](#)
- [FulfillmentESB](#)
- [CreditService](#)
- [RapidService](#)
- [CustomerService](#)
- [SOAOrderBooking](#)
- [OrderBookingESB](#)
- [ApproveOrderTaskForm](#)

A.8.1 About the CreditService, RapidService, and CustomerService Projects

The CreditService, RapidService, and CustomerService projects in the SOA Order Booking application are Java EE Web services that are referenced by the other projects in the application.

As a result, for the purposes of this example, it is not necessary to upgrade these three projects to 11g. Instead, you can continue to use the Oracle Application Server 10g versions of these services, which are currently running in your 10g environment.

If you choose to upgrade these three Web services projects, note the following:

- The instructions for modifying the Web services projects in this Appendix should be adequate if you are familiar and experienced with developing Java EE applications and Web services; however, they are not designed to serve as a tutorial for upgrading Java EE projects.
- If you choose to upgrade the CustomerService Web services project, some additional changes will be required to the SOAOrderBooking project, which are not documented in detail in this Appendix.

A.8.2 SelectManufacturer

The SelectManufacturer project is an asynchronous BPEL process. This means that it contains a receive activity to initiate the BPEL process flow and an invoke activity to call back the client asynchronously with the results (that is, the quote) at the end of the flow.

To compile and redeploy the SelectManufacturer project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make SelectManufacturer.jpr**.

The project should compile successfully.

3. Right-click the project again and select **Deploy > SelectManufacturer > to > SoademoApplicationServer**.

The project should deploy successfully.

A.8.3 DHLShipment

The DHLShipment project was created as an Oracle Enterprise Service Bus (ESB) service in Oracle Application Server 10g Release 3 (10.1.3). It should require no changes after opening the project in Oracle JDeveloper 11g.

To compile and redeploy the DHLShipment project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make DHLShipment.jpr**.

The project should compile successfully.

3. Right-click the project again and select **Deploy > SelectManufacturer > to > SoademoApplicationServer**.

The project should deploy successfully.

A.8.4 FulfillmentESB

The FulfillmentESB project was created in 10g Release 3 (10.1.3) as an Oracle Enterprise Service Bus (ESB) project.

After an order has been approved, the SOAOrderBooking project invokes the FulfillmentESB project to determine the shipping method for the order.

Orders are sent to the USPS through a file adapter. The order information is written to a file. For Fedex, orders are sent through a database adapter. The order information is written to a database. The FulfillmentESB project also sends orders to a JMS queue.

Before you can compile and deploy this project, you need to make some minor changes in Oracle JDeveloper 11g:

- [Modifying the FulfillmentESB Project in Oracle JDeveloper 11g](#)
- [Compile and Deploy the FulfillmentESB Project](#)

A.8.4.1 Modifying the FulfillmentESB Project in Oracle JDeveloper 11g

Make the following changes to the FulfillmentESB project:

1. Expand the **FulfillmentESB** project and double-click **composite.xml**.
2. Oracle JDeveloper displays the FulfillmentESB composite in the SOA Composite Editor ([Figure A-4](#)).

For more information, see "Introduction to the SOA Composite Editor" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

3. Double-click the icon that represents the **Fulfillment.FulfillmentBatch** reference in the right swimlane.

This external reference represents the Fulfillment.FulfillmentBatch JMS adapter.

When you double-click this reference, Oracle JDeveloper displays the Adapter Configuration Wizard.

4. Follow the instructions in the Adapter Configuration Wizard to reconfigure the JMS Adapter for your Oracle Fusion Middleware 11g environment.

You can accept all the default settings in the wizard, except for the following:

- Select **Oracle Weblogic JMS** as the Oracle Enterprise Messaging service.
- Select the Application Server connection that you created in "[Task 5d: Create a Connection to Oracle WebLogic Server](#)".
- Click the **Browse** button next to the queue name and select **demoFulfillmentQueue**.

This should cause `jms/demoFulfillmentQueue` to be entered into the Queue Name field.

- Set the JNDI name to `eis/demo/Queue`.

5. When you return to the Composite Editor, double-click the icon that represents the **Fulfillment.FedexShipment** reference in the right swimlane.

This external reference represents a database adapter.

When you double-click this reference, Oracle JDeveloper displays the Adapter Configuration Wizard.

6. Follow the instructions in the Adapter Configuration Wizard to reconfigure the database adapter for your Oracle Fusion Middleware 11g environment.

You can accept all the defaults in the wizard, except for the following changes:

- Select `soademo` as your database connection.
- Set the JNDI name to `eis/DB/soademoDS`.

7. Update the **Fulfillment.USPSShipment** external reference (which represents a file adapter), as follows:

- a. Create a directory on disk for the file adapter; for example:

```
MW_HOME/Oracle_SOA1/soa/USPSShipment
```

- b. Return to the Composite Editor for the **FulfillmentESB** project, and double-click the **FulfillmentUSPSShipment** external reference.

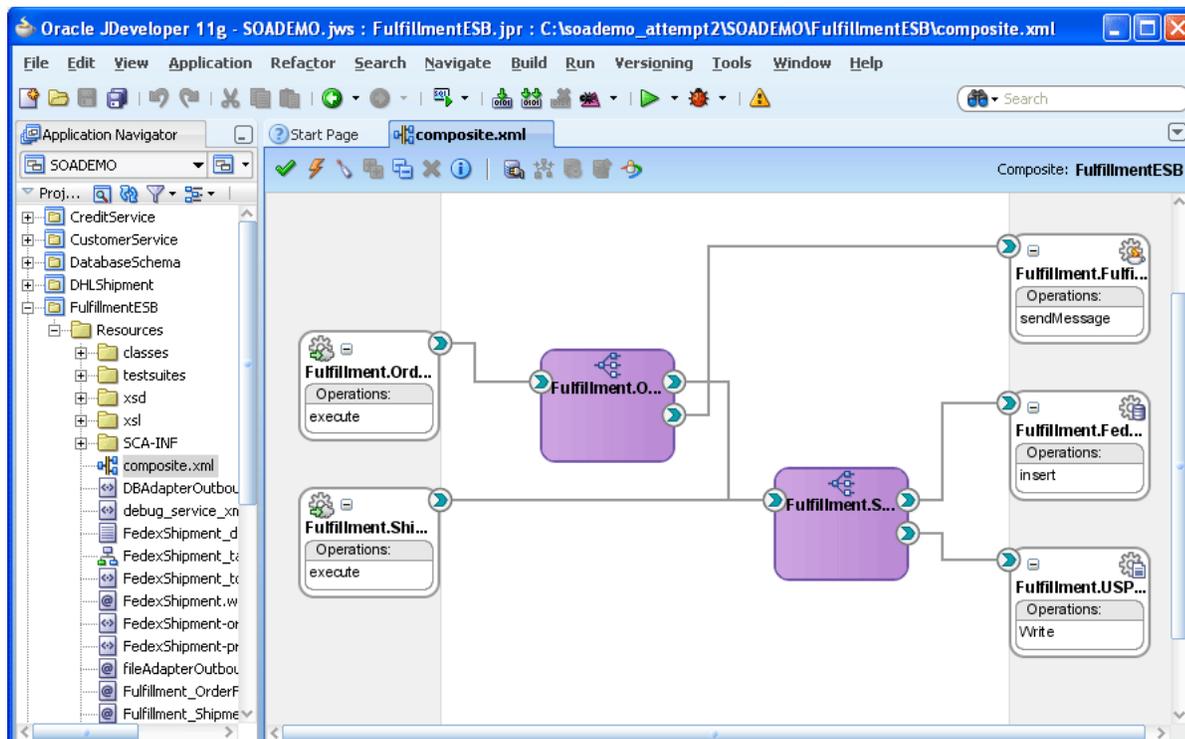
Oracle JDeveloper displays the Adapter Configuration Wizard.

- c. Follow the instructions on the screen until the File Configuration screen appears.

- d. In the **Directory for Outgoing Files (physical path)** field, enter the path to the directory you created in Step 7a.

- e. Click **Next** and follow the rest of the adapter configuration instructions.

Figure A-7 Composite Editor for the FulfillmentESB Project of the SOA Order Booking Application



A.8.4.2 Compile and Deploy the FulfillmentESB Project

To compile and redeploy the Fulfillment project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make FulfillmentESB.jpr**.
3. Right-click the project again and select **Deploy > FulFillmentESB > to > SoademoApplicationServer**.

The project should compile successfully.

The project should deploy successfully.

A.8.5 CreditService

The CreditService project checks whether a customer's credit card is valid or not. In this SOA Order Booking application, the code simply checks the value of the credit card number. If the credit card number is less than 12345678, then the card is invalid. Otherwise, it is valid.

This project is a Java EE Web services project. Before you proceed, refer to [Section A.8.1, "About the CreditService, RapidService, and CustomerService Projects"](#).

Before you can compile and deploy this project, you must upgrade the Web service from OC4J JAX-RPC to Oracle WebLogic Server JAX-RPC:

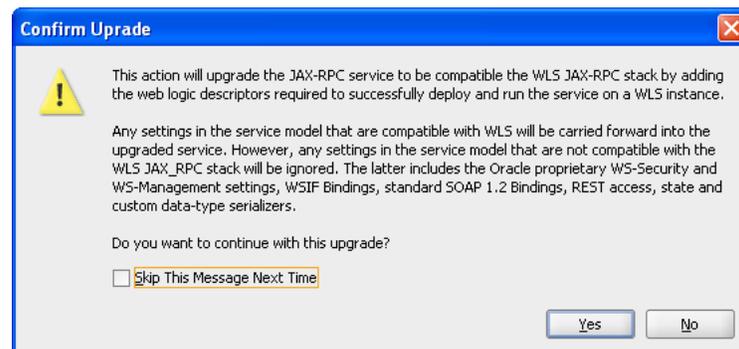
- [Upgrading the CreditService Web Services](#)
- [Compiling and Deploying the CreditService Project](#)

A.8.5.1 Upgrading the CreditService Web Services

To upgrade the OC4J JAX-RPC Web service to Oracle WebLogic Server JAX-RPC:

1. In the Application Navigator, expand the CreditService project, and select the following:
Application Sources > org.soademo.creditservice > CreditService
2. Right-click the Credit Service Web service and select **Upgrade Web Service to WLS JAX-RPC Configuration** from the context menu.
Oracle JDeveloper displays the Confirm Upgrade dialog box.
3. Click Yes to upgrade the CreditService Web service to Oracle WebLogic Server JAX-RPC.
4. When the upgrade is complete, save the project.

Figure A-8 Confirmation Dialog Box When Upgrading the CreditService Web Service



A.8.5.2 Compiling and Deploying the CreditService Project

To compile and redeploy the CreditService project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make CreditService.jpr**.
The project should compile successfully.
3. Right-click the project again and select **Deploy > CreditService > to > SoademoApplicationServer**.
The project should deploy successfully.

A.8.6 RapidService

The RapidService project represents a supplier that provides price quotes for customer orders. The other supplier which it competes with is SelectManufacturer. The supplier that provides the lower quote for a customer's order gets to fulfill the order.

This project is a Java EE Web services project. Before you proceed, refer to [Section A.8.1, "About the CreditService, RapidService, and CustomerService Projects"](#).

Before you can compile and deploy this project, you must modify it using Oracle JDeveloper 11g:

- [Upgrade the RapidService Web Service from OC4J JAX-RPC to Oracle WebLogic Server JAX-WS](#)
- [Compiling and Deploying the RapidService Project](#)

A.8.6.1 Upgrade the RapidService Web Service from OC4J JAX-RPC to Oracle WebLogic Server JAX-WS

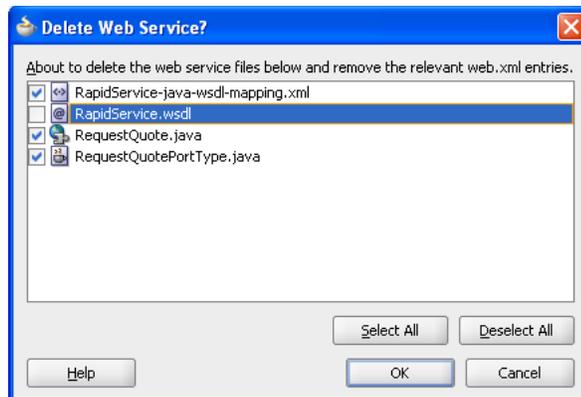
Before you can compile and deploy this project, you must upgrade the OC4J JAX-RPC Web services to Oracle WebLogic Server JAX-WS.

The following procedure uses the "top-down" approach to regenerate the Web service using the existing WSDL:

1. Remove the existing JAX-RPC Web service files from the RapidService project:
 - a. In Oracle JDeveloper 11g application navigator pane, open the RapidService project and expand following node:

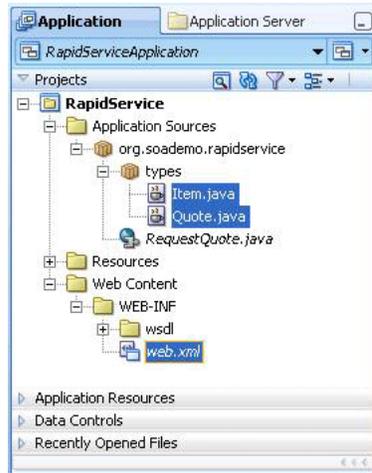
Application Sources > org.soademo.rapidservice
 - b. Right-click the **RapidService** node and select **Delete** from the context menu.
 - c. In the Delete Web Service? dialog box, click **Select All** to select all the files shown in the dialog box.
 - d. Clear the check box for the `RapidService.wsdl` file, as shown in [Figure A-9](#).
 - e. Click **OK**.

Figure A-9 Deleting the RapidService Web Service



- f. Expand the contents of the **Types** node and delete `Item.java`, `Quote.java`.
- g. Expand the contents of the **Web Content > WEB-INF** node, and then delete `web.xml`.

[Figure A-10](#) shows the files under RapidService that you must select and then delete.

Figure A-10 Removing the RapidService Web Services Files and Related Files

2. Remove JAX-RPC Web Services from the classpath:
 - a. Right-click the RapidService project and select **Project Properties** from the context menu.
Oracle JDeveloper displays the Project Properties window.
 - b. Select **Libraries and Classpath**, select **JAX-RPC 10.1.3 Web Services**, click **Remove**, and the click **OK**.
3. Generate the new JAX-WS Web service:
 - a. Right click the RapidService.wsdl file under the following location in the RapidService project:
Application Sources > org.soademo.rapidservice > RapidService
 - b. Select **Create Web Service** from the context menu.
Oracle JDeveloper displays the Create Java Web Service From WSDL wizard.
 - c. Follow the instructions on the screen to create the new JAX-WS Web service:
Be sure to select **Java EE 1.5, with support for JAX-WS Annotations** on the Select Deployment Platform screen.
4. Modify the poItemsQuote method in the RequestQuoteImpl.java file as follows:

```
public Quote poItemsQuote(@WebParam(name = "param0", targetNamespace =
"http://rapidservice.soademo.org/types/")
    List<Item> param0) {
    Long totalPrice = new Long(0);
    for (int i = 0; i < param0.size(); i++) {
        Item localItem = param0.get(i);
        totalPrice += localItem.getQuantity() * 110;
    }
    Quote priceQuote = new Quote();
    priceQuote.setSupplierName("RapidDistributors");
    priceQuote.setSupplierPrice(totalPrice.toString());
    return priceQuote;
}
```

5. Modify the orderQuote method in the RequestQuoteImpl.java file, as follows:


```
public Quote orderQuote(@WebParam(name = "param0", targetNamespace =
```

```
        "http://rapidservice.soademo.org/types/")
String param0, @WebParam(name = "param1", targetNamespace =
    "http://rapidservice.soademo.org/types/")
String param1, @WebParam(name = "param2", targetNamespace =
    "http://rapidservice.soademo.org/types/")
String param2, @WebParam(name = "param3", targetNamespace =
    "http://rapidservice.soademo.org/types/")
String param3, @WebParam(name = "param4", targetNamespace =
    "http://rapidservice.soademo.org/types/")
String param4) {
    Quote priceQuote = new Quote();
    priceQuote.setSupplierName("RapidDistributors");
    priceQuote.setSupplierPrice("5000");
    return priceQuote;
}
```

6. Save your changes to the RapidService project.

A.8.6.2 Compiling and Deploying the RapidService Project

To compile and redeploy the Fulfillment project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make RapidService.jpr**.
The project should compile successfully.
3. Right-click the project again and select **Deploy > RapidService > to > SoademoApplicationServer**.

The project should deploy successfully.

A.8.7 CustomerService

The CustomerService project is an EJB implementation that provides methods that enable client applications, such as the SOADemo-Client application, to retrieve customer information from the database and add customers to the database.

This project is a Java EE Web services project. Before you proceed, refer to [Section A.8.1, "About the CreditService, RapidService, and CustomerService Projects"](#).

Before you can compile and deploy the project, there some changes you have to make in Oracle JDeveloper 11g.

- [Modifying Specific CustomerService Project Class Files](#)
- [Modifying the persistence.xml File in the CustomerService Project](#)
- [Compiling and Deploying the CustomerService Project](#)

A.8.7.1 Modifying Specific CustomerService Project Class Files

To modify the CustomerService project in Oracle JDeveloper 11g, you must modify the following files in the `/src/org/soademo/customerservice/` directory of the project:

- **business/Customerservice.java:**
Remove "throws RemoteException" from the `findCustomerById()` method.
- **client/CustomerServiceClient.java** and **business/CustomerServiceClient.java**

- a. Locate the section of the class file shown in [Example A-1](#) and modify it so it reflects the changes shown in [Example A-2](#).
- b. Change the lookup to the following:

```
CustomerService customerService =
    (CustomerService)context.lookup(
    "CustomerService#org.soademo.customerservice.business.CustomerService");
```

- **business/CustomerserviceBean.java:**

Modify this file so its contents reflects the changes shown in [Example A-3](#).

Example A-1 Original CustomerService.java File

```
private static Context getInitialContext() throws NamingException {
    Hashtable en = new Hashtable();
    // Standalone OC4J connection details
    env.put( Context.INITIAL_CONTEXT_FACTORY,
            "oracle.j2ee.rmi.RMIInitialContextFactory" );
    env.put( Context.SECURITY_PRINCIPAL, "oc4jadmin" );
    env.put( Context.SECURITY_CREDENTIALS, "welcome1" );
    env.put(Context.PROVIDER_URL, "ormi://localhost:23791/CustomerService");
    return new InitialContext( env );
}
```

Example A-2 Modified CustomerService.java File

```
private static Context getInitialContext() throws NamingException {
    Hashtable env = new Hashtable();
    env.put( Context.INITIAL_CONTEXT_FACTORY,
"weblogic.jndi.WLInitialContextFactory" );
    env.put( Context.SECURITY_PRINCIPAL, "weblogic" );
    env.put( Context.SECURITY_CREDENTIALS, "weblogic" );
    env.put(Context.PROVIDER_URL, "t3://edith:7101");
    return new InitialContext( env );
}
```

Example A-3 Updated CustomerServiceBean.java Class File

```
package org.soademo.customerservice.business;
import javax.ejb.Remote;
import javax.ejb.RemoteHome;
import javax.ejb.Stateless;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import org.soademo.customerservice.persistence.Customer;
@Remote
@WebService(serviceName = "CustomerSvc",
            targetNamespace = "http://www.globalcompany.com/ns/customer")
@Stateless(name="CustomerService", mappedName = "CustomerService")
public class CustomerServiceBean implements CustomerServiceLocal,
    CustomerService {
    @PersistenceContext(unitName = "customerServiceUnit" )
    private EntityManager em;
    public CustomerServiceBean() {
    }
    /**
     * @param entity
```

```

    * @return
    */
    public Object mergeEntity(Object entity) {
        return em.merge(entity);
    }
    public Object persistEntity(Object entity) {
        em.persist(entity);
        return entity;
    }
    public Object refreshEntity(Object entity) {
        em.refresh(entity);
        return entity;
    }
    public void removeEntity(Object entity) {
        em.remove(em.merge(entity));
    }
    /** <code>select object(cust) from Customer cust
        where cust.custid = :custid</code> */
    @WebMethod
    public Customer findCustomerById(@WebParam(name = "custid",
        targetNamespace = "http://www.globalcompany.com/ns/customer")
        String custid){
    //    public Customer findCustomerById(String custid) {
        return
    (Customer)em.createNamedQuery("Customer.findCustomerById").setParameter("custid",
        custid).getSingleResult();
    }
    //    public String getCustomerStatus(String CustomerID) {
        @WebMethod
        public String getCustomerStatus(@WebParam(name = "CustomerID",
            targetNamespace = "http://www.globalcompany.com/ns/customer")
            String CustomerID){return
            findCustomerById(CustomerID).getStatus();
        }
    //    public String addNewCustomer(Customer customer) {
        @WebMethod
        public String addNewCustomer(@WebParam(name = "customer",
            targetNamespace = "http://www.globalcompany.com/ns/customer")
            Customer customer){
            em.persist(customer);
            //making the change to return the customer id
            // that client application can use to query orders
            return customer.getCustid();
            //return "New customer added sucessfully to customer database";
        }
    //    public Customer findCustomerByEmail(String email, String password) {
        @WebMethod
        public Customer findCustomerByEmail(@WebParam(name = "email",
            targetNamespace = "http://www.globalcompany.com/ns/customer")
            String email, @WebParam(name = "password",
            targetNamespace = "http://www.globalcompany.com/ns/customer")
            String password){
            return(Customer)em.createNamedQuery(
                "Customer.findCustomerByEmail").setParameter("email",
                email).setParameter("password",
                password).getSingleResult();
        }
    }
}

```

A.8.7.2 Modifying the persistence.xml File in the CustomerService Project

Modify the `src/META-INF/persistence.xml` so it uses Kodo explicitly in Oracle WebLogic Server.

Kodo is an implementation of Sun Microsystems's Java Persistence API (JPA) specification and Java Data Objects (JDO) specification for transparent data objects. Kodo is available as a stand-alone product and is integrated within WebLogic Server.

Add the following entry inside the `<persistence-unit>` element in the `persistence.xml` file:

```
<provider>kodo.persistence.PersistenceProviderImpl</provider>
```

A.8.7.3 Compiling and Deploying the CustomerService Project

To compile and redeploy the Fulfillment project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make CustomerService.jpr**.
The project should compile successfully.
3. Right-click the project again and select **Deploy > CustomerService > to > SoademoApplicationServer**.

The project should deploy successfully.

A.8.8 SOAOrderBooking

The SOAOrderBooking project, which is a BPEL project, represents the main flow in the SOA Order Booking application. It sends the order information to the appropriate services at the appropriate times. For example, it contacts CreditService to check the customer's credit card, and if the credit card is acceptable, it contacts the suppliers (Select Manufacturer and Rapid Distributors) to get price quotes for the order.

Before you can compile and deploy this project from Oracle JDeveloper 11g, you must make some changes:

- [Modifying the SOAOrderBooking Project](#)
- [Optionally, Changing SOAOrderBooking to Use Upgraded Web Services](#)
- [Compiling and Deploying the SOAOrderBooking Project](#)

A.8.8.1 Modifying the SOAOrderBooking Project

Make the following changes to the SOAOrderBooking project in Oracle JDeveloper 11g:

1. Expand the SOAOrderBooking project, and open the **SOA Content > ApproveOrder** folder.
2. Double-click the **ApproveOrder.task** file, which defines the human approval task.
3. Double-click the Manager icon under the Assignment and Routing section, select **User** from the **Identification Type** drop-down menu, and enter `weblogic` in the associated field.

4. Open the **SOA Content** folder, and double-click **composite.xml**.

Oracle JDeveloper displays the SOAOrderBooking composite in the SOA Composite Editor.

5. Double-click the icon that represents the **Order** reference.

This external reference represents the Order database adapter.

When you double-click this reference, Oracle JDeveloper displays the Adapter Configuration Wizard.

6. Follow the instructions in the wizard, and accept the default settings, except for the following changes:

- Select `soademo` as your database connection.
- Set the JNDI name to `eis/DB/soademoDS`.

7. Perform Steps 5 and 6 for the `OrderSequence` reference and the `OrderStatus` external references, which both also represents database adapters that must connect to the `soademo` database and schema.

8. Double click the **SelectService** reference.

Oracle JDeveloper displays the Update Reference dialog box.

Click on the WSDL icon next to the WSDL URL field to display the SOA Resource Browser.

Select **Resource Palette** from the drop-down menu, and then select **SelectManufacturer** in the Resource Palette.

9. Repeat Step 8 for the **OrderFulfillment** reference, but this time select the **Fulfillment.OrderFulfillment_ep** entry point in `FulfillmentESB` from the resource palette.

10. Save the change you made to the project.

A.8.8.2 Optionally, Changing SOAOrderBooking to Use Upgraded Web Services

If you upgraded and redeployed the `RapidService`, `CreditService`, and `CustomerService` projects earlier in this procedure, you can now modify the `SOAOrderBooking` project to use the new 11g Web services, rather than the ones that are deployed and still running in your Oracle Application Server 10g environment.

Alternatively, you can continue to reference the existing 10g Web services. For more information, see [Section A.8.1, "About the CreditService, RapidService, and CustomerService Projects"](#).

To modify the `SOAOrderBooking` project so it uses the newly upgraded `RapidService`, `CreditService`, and `CustomerService` Web services:

1. Locate the new WSDL URL for the upgraded Web services projects:

- a. Open your Web browser and log in to the Fusion Middleware Control.

For example: `host24.mycompany.com:7001/em`

For more information, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

- b. Open the **Application Deployments** folder in the Target Navigation Pane.
- c. Click the **SOADEMO-RAPIDSERVICE-RapidService-WS** application in the list of deployed applications.

Fusion Middleware Control displays the Application Deployment page for the application you selected.

- d. Click the Test icon for the **SOADEMO-RAPIDSERVICE-RapidService-WS** in the Web Services table in the Entry Points section of the page.

- e. On the Test Web Services page, note the WSDL URL for the Web service.
 - f. Repeat steps c through e for the CreditService and CustomerService applications.
2. Modify the SOAOrderBooking composite with the new WSDL URLs:
 - a. Expand the **SOAOrderBooking** project and double-click **composite.xml**.
 - b. Oracle JDeveloper displays the SOAOrderBooking composite in the SOA Composite Editor.
 - c. Double-click the icon that represents the **RapidService** external reference in the right swimlane.
 This external reference represents the RapidService Web service.
 When you double-click this reference, Oracle JDeveloper displays the Update References dialog box.
 - d. Enter the new 11g WSDL URL in the **WSDL URL** field and click **OK**.
 - e. Repeat Steps c and d for the **CreditValidationService** and **CustomerService** external references.
 - f. Save your changes to the project.
 3. If you upgraded the CustomerService project to 11g, then you must also modify the SOAOrderBooking.bpel file to reflect the changes made to the CustomerService while upgrading it to 11g.
 This involves reconnecting the Web services binding, and updating the customerServiceRequest, customerServiceResponse, and subsequent variable queries.

A.8.8.3 Compiling and Deploying the SOAOrderBooking Project:

To compile and redeploy the SOAOrderBooking project:

1. Right-click the project in the Application Navigator.
2. From the context-menu, select **Make**.
 The project should compile successfully.
3. Right-click the project again and select **Deploy > SOAOrderBooking > to > SoademoApplicationServer**.
 The project should deploy successfully.

A.8.9 OrderBookingESB

The OrderBookingESB project is the entry point to the SOA Order Booking application.

The OrderBookingESB project invokes the SOAOrderBooking project, which is a BPEL project that defines the main flow of the SOA Order Booking application.

It was created in 10g Release 3 (10.1.3) as an Oracle Enterprise Service Bus (ESB) project.

Before you can compile and deploy this project from Oracle JDeveloper 11g, you must make some changes:

- [Modifying the OrderBookingESB Project](#)
- [Compiling and Deploying the OrderBookingESB Project:](#)

A.8.9.1 Modifying the OrderBookingESB Project

To modify the OrderBookingESB project in Oracle JDeveloper 11g:

1. Expand the **OrderBookingESB** project and double-click **composite.xml**.
2. Oracle JDeveloper displays the OrderBookingESB composite in the SOA Composite Editor (Figure A-4).
3. Double-click the icon that represents the **OrderBooking** external reference in the right swimlane.

This external reference represents the OrderBooking Web service.

When you double-click this reference, Oracle JDeveloper displays the Update References dialog box.

4. Click the WSDL icon next to the WSDL URL field to display the SOA Resource Browser.

Select **Resource Palette** from the drop-down menu, expand the application server connection, and select the **SOAOrderBooking > Client** resource from the Resource Palette.

5. Save your changes to the project.

A.8.9.2 Compiling and Deploying the OrderBookingESB Project:

To compile and redeploy the OrderBookingESB project:

1. Right-click the project in the Application Navigator.
 2. From the context-menu, select **Make**.
- The project should compile successfully.
3. Right-click the project again and select **Deploy > SOAOrderBookingESB > to > SoademoApplicationServer**.

The project should deploy successfully.

A.8.10 ApproveOrderTaskForm

The ApproveOrderTaskForm project is created automatically when you open and upgrade the SOA Order Booking application in Oracle JDeveloper 11g.

No changes are required to this project. However, to deploy this project, you must use the deploy it as an application and not as a project:

1. Select the ApproveOrderTaskForm project.
2. From the **Application** menu, select **Deploy > ApproveOrderTaskFlow > to > SoademoApplicationServer**.

A.9 Task 8: Test the SOA Order Booking Application in Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control provides a Web-based management console that you can use to manage your Oracle SOA Suite composite applications.

To test the SOA Order Booking application in Fusion Middleware Control:

1. Log in to the Fusion Middleware Control for the Oracle WebLogic Server domain where you deployed your SOA Order Booking application projects:

The URL for Fusion Middleware Control is usually:

```
http://hostname:7001/em
```

For more information, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

2. In the Target Navigation Pane, expand the **SOA** folder, and then expand the **soa-infra (soa_server1)** node to display the SOA composite applications deployed to the domain.

3. Click the **OrderBookingESB** composite application.

Fusion Middleware Control displays the Dashboard tab of the OrderBookingESB composite application home page, which you can use to manage and monitor a deployed SOA composite application.

4. Click **Test** in the button bar across the top of the page.

Fusion Middleware Control displays the Test Web Services page, which displays the WSDL location, Operations list, and two tabs, Request and Response tabs.

5. Scroll down to the Input Arguments section of the page.

From this section of the page, you can enter input data to test the composite application. The SOA Order Booking application ZIP file contains two text files that contain sample data you use to test the SOA Order Booking application.

6. Locate the following input sample test file in the root directory of the SOADEMO_HOME:

```
sampleInput2.txt
```

7. Open `sampleInput2.txt` and select and copy the entire contents of the file.

8. Select **XML View** from the drop-down menu in the Input Arguments section of the Test Web Service page.

9. Select and delete the XML content displayed on the Request tab in the Input Arguments section of the page.

[Figure A-11](#) shows the XML content selected in the Input Arguments section of the page.

10. Paste the contents of the `sampleinput2.txt` file into the text field, and click **Test Web Service** at the bottom of the page.

After a few seconds, a message appears indicating that the Web service invocation was successful. It also displays the return value obtained by the service.

11. Click the **Launch Message Flow Trace** link to see the flow of your composite and the status of each service, component, and reference.

Fusion Middleware Control displays the Flow Trace page ([Figure A-12](#)), which shows each step in the process flow. Using this view, you can monitor the application as it progresses through each component and task in the application.

You can also click a component to drill down into the Instance Details for the component.

12. For further testing, you can perform this same procedure using `sampleinput1.txt`, which is in the same directory as `sampleinput2.txt` and submits an order of more than \$1000.

For `sampleinput1.txt`, you'll notice that the work flow stops and does not continue, until you log into the following URL and approve the order:

http://domain:port/integration/worklistapp

Figure A-11 Selecting XML Content in the Input Argument Section of the SOA Composite - Test Web Service Page

The screenshot displays the Oracle Enterprise Manager 11g Fusion Middleware Control interface. The left-hand navigation pane shows a tree structure under 'Farm_soa_domain' with 'SOA' expanded to 'soa-infra (soa_server1)', where 'OrderBookingESB [1.0]' is selected. The main content area shows the configuration for the 'OrderBookingESB [1.0]' SOA Composite. The 'Quality of Service' section includes radio buttons for 'WSDL Default', 'None', and 'Custom' for both 'WS-RM' and 'WS-Addressing'. The 'HTTP Transport Options' section has a checked 'Enable SOAP Action' with a value of 'initiate'. The 'Additional Test Options' section includes checkboxes for 'Enable Stress Test' and input fields for 'Concurrent Threads' (5), 'Loops per Thread' (10), and 'Delay in Milliseconds' (1000). The 'Input Arguments' section is expanded to 'XML View', showing a SOAP message payload with various XML elements like 'PurchaseOrder', 'ShipTo', 'BillTo', and 'UserContact'.

Web service invocation payload in XML format

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://www.globalcompany.com/ns/OrderBooking">
    <ns1:SOAOrderBookingProcessRequest xmlns:ns2="http://www.globalcompany.com/ns/order">
      <ns2:PurchaseOrder>
        <ns2:CustID></ns2:CustID>
        <ns2:ID></ns2:ID>
        <ns2:ShipTo>
          <ns2:Name></ns2:Name>
          <ns2:Address></ns2:Address>
        </ns2:ShipTo>
        <ns2:BillTo>
          <ns2:Name></ns2:Name>
          <ns2:Address></ns2:Address>
        </ns2:BillTo>
        <ns2:UserContact>
          <ns2:PhoneNumber></ns2:PhoneNumber>
          <ns2:EmailAddress></ns2:EmailAddress>
        </ns2:UserContact>
        <ns2:OrderItems></ns2:OrderItems>
      </ns2:PurchaseOrder>
    </ns1:SOAOrderBookingProcessRequest>
  </soap:Body>
</soap:Envelope>
```

Figure A-12 Flow Trace of the SOA Order Booking Application in Fusion Middleware Control

Trace

Click a component instance to see its detailed audit trail.

Show Instance IDs:

| Instance | Type | State | Time | Composite Instance |
|---|----------------------------|-----------|--------------------------|-----------------------------|
| [-] OrderBooking.OrderBookingService_ep | Service | Completed | May 14, 2009 12:30:47 PM | OrderBookingESB of 20001 |
| [-] OrderBooking.OrderBookingService | Mediator Component | Completed | May 14, 2009 12:30:54 PM | OrderBookingESB of 20001 |
| [-] OrderBooking.OrderBookingProcess | Reference | Completed | May 14, 2009 12:30:52 PM | OrderBookingESB of 20001 |
| client | Service | Completed | May 14, 2009 12:30:53 PM | SOAOrderBooking of 20002 |
| [-] SOAOrderBooking | BPEL Component | Completed | May 14, 2009 12:31:58 PM | SOAOrderBooking of 20002 |
| [-] OrderSequence | Reference | Completed | May 14, 2009 12:31:10 PM | SOAOrderBooking of 20002 |
| [-] Order | Reference | Completed | May 14, 2009 12:31:13 PM | SOAOrderBooking of 20002 |
| [-] CustomerSvc | Reference | Completed | May 14, 2009 12:31:14 PM | SOAOrderBooking of 20002 |
| [-] CreditValidatingService | Reference | Completed | May 14, 2009 12:31:20 PM | SOAOrderBooking of 20002 |
| [-] OrderBookingRules | Decision Service Component | Completed | May 14, 2009 12:31:25 PM | SOAOrderBooking of 20002 |
| [-] RapidService | Reference | Completed | May 14, 2009 12:31:27 PM | SOAOrderBooking of 20002 |
| [-] SelectService | Reference | Completed | May 14, 2009 12:31:30 PM | SOAOrderBooking of 20002 |
| [-] SelectService | Service | Completed | May 14, 2009 12:31:31 PM | SelectManufacturer of 20003 |
| [-] SelectManufacturer | BPEL Component | Completed | May 14, 2009 12:31:36 PM | SelectManufacturer of 20003 |
| [-] OrderFulfillment | Reference | Completed | May 14, 2009 12:31:39 PM | SOAOrderBooking of 20002 |
| [-] Fulfillment.OrderFulfillment_ep | Service | Completed | May 14, 2009 12:31:39 PM | FulfillmentESB of 20004 |
| [-] Fulfillment.OrderFulfillment | Mediator Component | Completed | May 14, 2009 12:32:04 PM | FulfillmentESB of 20004 |
| [-] Fulfillment.Shipment | Mediator Component | Completed | May 14, 2009 12:31:47 PM | FulfillmentESB of 20004 |
| [-] Fulfillment.USPSShipment | Reference | Completed | May 14, 2009 12:31:46 PM | FulfillmentESB of 20004 |
| [-] Fulfillment.FulfillmentBatch | Reference | Completed | May 14, 2009 12:32:02 PM | FulfillmentESB of 20004 |

Additional Oracle WebCenter Migration Details

This appendix describes the changes that the migration utility in Oracle JDeveloper 11g automatically makes to a WebCenter 10.1.3.x application while migrating the application. It also describes how you can change the location of customization store for preconfigured portlet producers.

The appendix contains the following sections:

- [Automated Changes in Migrated WebCenter Applications](#)
- [Content Integration Connections: Migration of Credentials to External Applications](#)
- [Preconfigured Portlet Producer: Customization Store Location](#)

B.1 Automated Changes in Migrated WebCenter Applications

To migrate a WebCenter 10.1.3.x application to Oracle WebCenter 11g, you use JDeveloper 11g. While migrating an application, the migration utility in JDeveloper 11g automatically makes certain changes, such as those related to customizable components, external applications, and portlet components.

This section describes the following changes:

- [Customizable Component Changes](#)
- [External Application Changes](#)
- [Portlet Component Changes](#)

B.1.1 Customizable Component Changes

In WebCenter applications, you can use Oracle Composer to add components to pages that users can customize at runtime. Oracle Composer provides components like Panel Customizable and Show Detail Frame that you can add to your applications to support page personalization and customization.

When you migrate a WebCenter 10.1.3.x application that uses customizable components, JDeveloper 11g automatically makes various changes related to customizable components to prepare the application to run on Oracle WebLogic Server. These changes include:

- General Changes
- Panel Customizable Changes
- Show Detail Frame Changes

B.1.1.1 General Changes

When you migrate your WebCenter application, JDeveloper 11g makes the following general changes to your application:

- The namespace of customizable components in JSPX files is renamed from `http://xmlns.oracle.com/adf/faces/customizable` to `http://xmlns.oracle.com/adf/faces/customizable/html`.
- From `web.xml`, entries for the following are removed:
 - Context parameter `oracle.mds.web-app-root`
 - Filter `mdsFilter`
 - Filter mapping for `mdsFilter`
 - Servlet `jsp2`, servlet mapping for `jsp2`, and resource proxy
- The Java Customizable Components library is removed and the WebCenter Customizable Components library is added.
- The old Customizable Components tag libraries are removed and the new version Customizable Components (HTML) is added.
- In import statements and declarations, if any, in Java files, the class name of Panel Customizable is changed to `oracle.adf.view.html.component.customizable.PanelCustomizable`.
- In import statements and declarations, if any, in Java files, the class name of Show Detail Frame is changed to `oracle.adf.view.html.component.customizable.ShowDetailFrame`.

B.1.1.2 Panel Customizable Changes

A Panel Customizable component is a container for a group of Oracle ADF components, task flows, and portlets that you want to make customizable at runtime. When you migrate a WebCenter application, JDeveloper 11g removes various attributes from Panel Customizable because they are no longer supported in Oracle WebCenter 11g. [Table B-1](#) lists Panel Customizable attribute changes in a migrated WebCenter application.

Table B–1 Panel Customizable Attribute Changes in a Migrated WebCenter Application

| Attribute | Attribute Availability in a Migrated WebCenter Application |
|--|---|
| DisplayHeader ExpansionMode Text Icon IsSeededInteractionAvailable IsMinimizable IsMaximizable IsMovable IsShowContentEnabled ContentInlineStyle DisclosureListener Background isEditable isHelpAvailable | Removed. All these attributes are removed from a migrated WebCenter application. A header is no longer supported for Panel Customizable. Therefore, <code>DisplayHeader</code> and certain other attributes, such as <code>ExpansionMode</code> , <code>Text</code> , and so on, that are dependent on the header being available are also removed. |
| DisplayScrollBar | Removed. However, to render scrollbars for the Panel Customizable content area, you can set the <code>Layout</code> attribute to <code>Scroll</code> . |
| Width Height | Deprecated. The usage of the <code>Width</code> and <code>Height</code> attributes is discouraged. It is recommended that you use the <code>inlineStyle</code> attribute to specify the width or height of Panel Customizable. For example, you can set <code>inlineStyle="width:300px;height:500px;"</code> . |

[Example B–1](#) shows the sample Panel Customizable code in the source view of a WebCenter 10.1.3.x application.

Example B–1 Panel Customizable Code in a WebCenter 10.1.3.x Application

```
<cust:panelCustomizable id="panelCustomizable1"
  text="panelCustomizable 1"
  displayHeader="true"
  isMaximizable="true"
  isMinimizable="true"
  width="100%"
  height="50px">
  displayScrollBar="auto"
  isShowContentEnabled="true"/>
```

[Example B–2](#) shows the sample Panel Customizable code after a WebCenter 10.1.3.x application has been migrated to Oracle WebCenter 11g. Notice that attributes such as `isMinimizable` and `displayHeader` are removed from the Panel Customizable code.

Example B–2 Panel Customizable Code of a WebCenter 11g Application

```
<cust:panelCustomizable id="panelCustomizable1"
xmlns="http://xmlns.oracle.com/adf/faces/customizable/html"/>
```

B.1.1.3 Show Detail Frame

A Show Detail Frame component renders a border or chrome around a child component in your WebCenter application pages. It provides a header with an Actions menu that contains User Interface (UI) controls to customize the display of the child component. [Table B-2](#) lists the changes that JDeveloper makes to the attributes of the Show Detail Frame component when you migrate your WebCenter application.

Table B-2 Changes Made to Show Detail Frame Attributes in Migrated WebCenter Applications

| Attribute | Description |
|------------------------------|--|
| isMaximizable | This attribute is removed from the migrated application. |
| isSeededInteractionAvailable | <p>IsSeededInteractionAvailable=true is migrated to the following:</p> <ul style="list-style-type: none"> ■ showMoveAction=menu ■ showMinimizeAction=chrome ■ showRemoveAction=chrome <p>IsSeededInteractionAvailable=false is migrated to the following:</p> <ul style="list-style-type: none"> ■ showMoveAction=none ■ showMinimizeAction=none ■ showRemoveAction=none |
| isMovable | This attribute is renamed to showMoveAction. The migration utility migrates isMovable=None Menu to showMoveAction=none menu, respectively. |
| isMinimizable | This attribute is renamed to showMinimizeAction. The migration utility migrates the isMinimizable=false true to showMinimizeAction=none chrome, respectively. |
| expansionMode | This attribute is removed from the migrated application if expansionMode=maximized. If the value was set to normal or minimized, then the attribute is retained. |
| ContentInlineStyle | This attribute is renamed to ContentStyle. |

[Example B-3](#) shows the sample code of Show Detail Frame in the source view of a WebCenter 10.1.3.x application page. [Example B-4](#) shows the sample Show Detail Frame code after this application is migrated to Oracle WebCenter 11g. Notice that isMaximizable is removed and the isSeededInteractionAvailable, isMinimizable, isMovable, and ContentInlineStyle attributes are renamed.

Example B-3 Show Detail Frame Code of a WebCenter 10.1.3.x Application

```
<cust:showDetailFrame id="showDetailFrame1"
  text="showDetailFrame 1"
  isSeededInteractionAvailable="false"
  isMaximizable="false"
  isMinimizable="true"
  isMovable="true">
  contentInlineStyle="background-color:rgb(0,255,255)"/>
```

Example B-4 Show Detail Frame Code of a WebCenter 11g Application

```
<cust:showDetailFrame id="showDetailFrame1"
  text="showDetailFrame 1"
```

```

displayActions="never"
showMoveAction="none"
showMinimizeAction="none"
contentStyle="background-color:rgb(0,255,255)"
xmlns="http://xmlns.oracle.com/adf/faces/customizable/html"/>

```

B.1.2 External Application Changes

You can associate external applications with portlet producers. Oracle WebCenter Framework defines an external application as any application that implements its own authentication process. That is, an application that does not take part in the single sign-on process of Oracle WebCenter Framework. For more information, see the "Working with External Applications" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

B.1.2.1 External Applications Changes in Oracle Application Server 10g and Oracle Fusion Middleware 11g

Compared to Oracle Application Server 10g, in Oracle Fusion Middleware 11g, external applications have undergone some changes. In Oracle Fusion Middleware 11g, external applications let you specify shared and public credentials. External applications also let you define an internal name (that serves the purpose of external application identifier and cannot be changed after creation) and a display name (that can be changed). Another distinguishing feature is that in Oracle Application Server 10g, external applications use Metadata Services (MDS) to store metadata; in Oracle Fusion Middleware 11g, external applications are stored as connections similar to other services like content repository connections.

[Table B-3](#) compares how the distinguishing attributes of external applications map between Oracle Application Server 10.1.3.x and Oracle Fusion Middleware 11g. The table shows sample values for the name, ExtAppId, and GUID attributes.

Table B-3 Distinguishing External Applications Attributes in Oracle Application Server 10.1.3.x and Oracle Fusion Middleware 11g

| External Application Attribute | Sample Attribute Value in Oracle Application Server 10.1.3.x | Sample Attribute Value in Oracle Fusion Middleware 11g |
|--------------------------------|--|--|
| Name | app1 | app1 (Also used as the display name of the external application) |
| extAppId | /oracle/adf/externalApplications/extApp9213895574200898560.xml | app1 |
| SharedCredEnabled | Attribute not supported | False |
| PublicCredEnabled | Attribute not supported | False |
| loginEnabled | Attribute not supported | True |
| GUID | Attribute not supported | ExtAppbc638395-2860-41ae-a9ff-edfc6b79110f |

B.1.2.2 External Application Migration Changes

If your WebCenter 10.1.3.x application relies on external applications, then when you migrate that application, the associated external applications are also migrated automatically. For example, when migrating a WebCenter 10.1.3.x application that

uses PDK-Java producers dependent on external applications, the external applications are automatically migrated. During WebCenter application migration, the migration utility performs the tasks for a migrated external application:

- Uses the name of the 10.1.3.x external application as the internal name (`ExtAppId`) of the migrated application.
- Updates the producer metadata with the new external application ID. An external application is referenced in the portlet producer metadata through its ID.
- Migrates the external application metadata from MDS store configured in the WebCenter application's `adf-config.xml` to `connections.xml`.
- Adds the automated login configuration to the project that contains external application portlets.
- Adds the new attributes, listed in [Table B-3](#), to the migrated external application.
- Removes the credential provisioning page configured for the project containing external application portlets and the managed bean and navigation rule defined in each project. In Oracle Fusion Middleware 11g, the credential provisioning page is packaged in the external application library.
- Configures external application libraries, WebCenter External Application Service and WebCenter External Application Service View.

B.1.3 Portlet Component Changes

When you migrate a WebCenter application that uses portlets, JDeveloper automatically makes various changes related to portlet tags in the migrated application.

B.1.3.1 Portlet Tag Changes

In WebCenter 10.1.3.x applications, the `adfp:portlet` tag represents a portlet in the namespace `http://xmlns.oracle.com/adf/faces/portlet`.

In WebCenter 11g applications, portlet tags are converted to Trinidad HTML Portlet tags. The `adfp:portlet` portlet tag is migrated to `adfph:portlet` in the namespace `http://xmlns.oracle.com/adf/faces/portlet/html`.

B.1.3.2 Attribute Changes

[Table B-4](#) lists the attributes that are renamed or removed from a migrated WebCenter application containing portlets.

Table B-4 Attribute Changes in a Migrated WebCenter Application Containing Portlets

| Attribute | Change in the Migrated Application |
|---|---|
| <code>isSeededInteractionAvailable</code> | This attribute is renamed to <code>displayActions</code> . The migration utility migrates <code>isSeededInteractionAvailable=false true</code> to <code>displayActions=onHover always</code> , respectively. |
| <code>isMinimizable</code> | This attribute is renamed to <code>showMinimizeAction</code> . The migration utility migrates <code>isMinimizable=false true</code> to <code>showMinimizeAction=none chrome</code> , respectively. |
| <code>isMaximizable</code> | This attribute is renamed to <code>showMaximizeAction</code> . This change is applicable only for HTML portlets. The attribute is not available for rich-text portlets. The migration utility migrates <code>isMaximizable=false true</code> to <code>showMaximizeAction=none chrome</code> , respectively. |

Table B–4 (Cont.) Attribute Changes in a Migrated WebCenter Application Containing

| Attribute | Change in the Migrated Application |
|----------------------------------|---|
| <code>isMovable</code> | This attribute is renamed to <code>showMoveAction</code> . The migration utility migrates <code>isMovable=False True</code> to <code>showMoveAction=none menu</code> , respectively. |
| <code>contentInlineStyle</code> | This attribute is renamed to <code>contentStyle</code> . |
| <code>text</code> | This attribute is renamed to <code>title</code> . |
| <code>portletType</code> | This attribute is removed. |
| <code>isLinkModeAvailable</code> | This attribute is removed. |
| <code>showRemoveAction</code> | This attribute is removed from an HTML portlet. In Oracle WebCenter 11g, this attribute is available only for rich-text portlets. |
| <code>expansionMode</code> | In JDeveloper 11g, the values allowed for this attribute are <code>normal</code> and <code>minimized</code> . In a WebCenter 10.1.3.x application, if the value is set to <code>maximized</code> , then the attribute is removed from the migrated application. |

B.2 Content Integration Connections: Migration of Credentials to External Applications

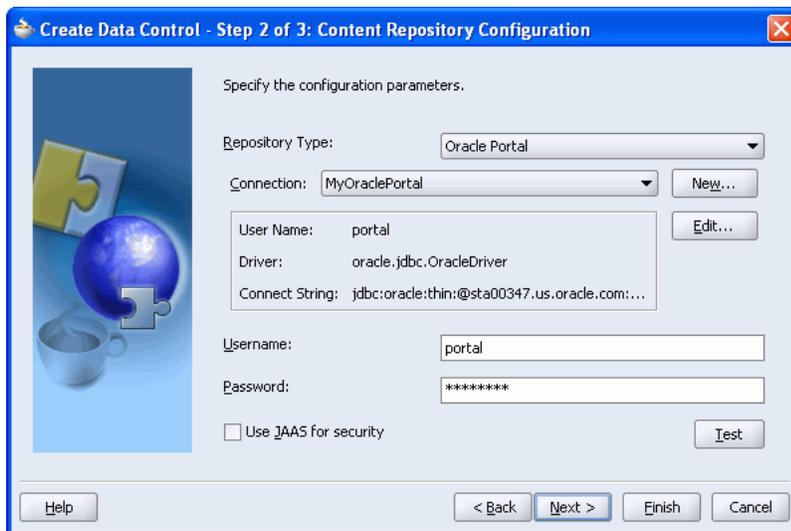
Content integration capabilities of Oracle WebCenter enable you to integrate decentralized content located across various content repositories, such as Oracle Content Server, Oracle Portal, or your file system, into your WebCenter applications. This section describes the authentication methods supported in Oracle WebCenter 10.1.3.x and Oracle WebCenter 11g applications. It also describes how credentials are migrated to external applications for content repository connections when you migrate a WebCenter 10.1.3.x application.

To integrate content from a content repository into your WebCenter application, you need to create a connection to the required repository. In Oracle WebCenter 11g, while creating a content repository connection, you can set the authentication method to any of the following:

- **Identity Propagation:** In this method, no credentials are passed to the content repository. Instead, the repository connector uses the current user's identity as determined from the Java security context. This method is generally used when the application and the content repository use the same identity store to authenticate users.
- **External Application:** This method can be used in all other cases where the current user's identity should not be propagated directly to the content repository.

In Oracle WebCenter 10.1.3.x, the authentication method for connecting to a content repository is determined through the **Use JAAS for security** check box. You select the **Use JAAS for security** check box if you want to transfer the user's identity to the repository without passing the credentials and to use the trust relationship between the WebCenter application and the content repository. If you do not use the JAAS security, then you need to specify credentials and leave the check box unselected.

Figure B–1 shows the content repository connection details of a WebCenter 10.1.3.x application that does not use JAAS security. In Figure B–1, the **Use JAAS for security** check box is not selected and credentials are specified in the **Username** and **Password** fields.

Figure B-1 Authentication Method of a WebCenter 10.1.3.x Application

If you migrate a WebCenter 10.1.3.x application in which JAAS security is used (that is the **Use JAAS for security** checkbox is selected), then **Identity Propagation** is automatically set as the authentication method for the connection in the migrated application. (The **Use JAAS for security** feature is same as **Identity Propagation**.)

If the **Use JAAS for security** check box was not selected in your WebCenter 10.1.3.x application, then when the application is migrated, the authentication method is set to **External Application**. Further, a new external application with public credentials is created automatically by using credentials from the 10.1.3.x content repository connection. [Figure B-2](#) shows the content repository connection details of a migrated application. It shows the **External Application** authentication method selected and a new external application created for the migrated application. [Figure B-3](#) shows the public credentials automatically specified for the newly created external application.

Figure B–2 Content Repository Connection Configuration of a Migrated Application

The screenshot shows the 'Edit Content Repository Connection' dialog box. The title bar reads 'Edit Content Repository Connection'. The main text says 'Configure the content repository connection.' Below this, there are two radio buttons: 'Application Resources' (selected) and 'IDE Connections'. The 'Connection Name' field contains 'PortalContent'. The 'Repository Type' dropdown menu is set to 'Oracle Portal'. There is a checkbox for 'Set as primary connection for Documents service' which is unchecked. The 'Database Connection' dropdown is empty, with plus, edit, and search icons to its right. Below are fields for 'User Name:', 'Driver:', and 'Connect String:'. The 'Login Timeout (ms):' field is empty. The 'Authentication' section has two radio buttons: 'Identity Propagation' (unchecked) and 'External Application' (selected). The 'External Application' dropdown is set to 'PortalContent_extapp', with plus and edit icons to its right. Below this is a checkbox for 'Specify login credentials for the current JDeveloper session:' which is unchecked. There are 'User Name:' and 'Password:' fields. At the bottom, there is a 'Test Connection' button, a text area, and 'Help', 'OK', and 'Cancel' buttons.

Figure B–3 Public Credentials Specified for an External Application

The screenshot shows the 'Edit External Application - Step 5 of 5' dialog box, specifically the 'Public Credentials' tab. The title bar reads 'Edit External Application - Step 5 of 5'. The main text says 'Specify the public credentials, if required. The specified public credentials will be used by unauthenticated users for accessing this external application.' There is a checkbox for 'Specify Public Credentials' which is checked. Below this, there is a section titled 'Public Credentials' with two input fields: 'Username:' containing 'portal' and 'Password:' containing a masked password (represented by dots). On the left side, there is a navigation pane with links for 'Name', 'General', 'Additional Fields', 'Shared Credentials', and 'Public Credentials' (which is selected). At the bottom, there are 'Help', '< Back', 'Next >', 'Finish', and 'Cancel' buttons.

Note: If your application contains a content repository connection that uses credentials, and during migration if there is a problem in retrieving credentials from the credential store or in creating the external application, then you must edit the migrated content repository connection. In the Edit Content Repository Connection wizard, you must select External Application as the authentication method, and then create an external application with the required credentials.

B.3 Preconfigured Portlet Producer: Customization Store Location

By default, customizations related to preconfigured portlet producers are stored at the following location:

```
ORACLE_HOME/portal/portletdata
```

You may choose to store portlet customizations at a location different. This section describes how to change the location of a customizations store for various preconfigured portlet producers.

B.3.1 WSRP Portlet Producers

To specify a file preference store location for portlet customizations of your WSRP portlet producers, update your producer's `WEB-INF/web.xml` with the following entry:

```
<env-entry>
  <env-entry-name>oracle/portal/wsrp/server/fileStoreRoot</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>{YOUR_PORTLET_DATA_PATH}</env-entry-value>
</env-entry>
```

Where `YOUR_PORTLET_DATA_PATH` refers to the path where you want to store portlet customizations.

If you want to use a database preference store for your WSRP portlet producer, then you must set up the database preference store. For information, see the "Setting Up a Preference Store" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

B.3.2 Web Clipping Portlet Producer

To specify the Web Clipping customization store, update the following entry in `mds-config.xml`:

```
<property name="metadata-path" value="{YOUR_PORTLET_DATA_PATH}"/>
```

For information about Web Clipping portlet configuration, see the "Web Clipping Portlet Configuration Tips" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

B.3.3 OmniPortlet and other PDK-Java Portlet Producers

For OmniPortlet, you specify the customization store location in `provider.xml`. The file is available at the following location on Oracle WebLogic Server:

```
FMW_HOME/user_projects/domains/wc_domain/servers/WLS_Portlet/tmp/_WL_
user/portalTools_11.1.1.1.0/RANDOMLY_GENERATED_
```

DIRECTORY/war/WEB-INF/providers/omniPortlet/provider.xml

To specify a file preference store location for portlet customizations of your OmniPortlet portlet producers, update the following entry in `provider.xml`:

```
<preferenceStore class="oracle.webdb.reformlet.ReformletFilePreferenceStore">
  <name>omniPortletprefStore</name>
  <useHashing>true</useHashing>
  <rootDirectory>{YOUR_PORTLET_DATA_PATH}</env-entry-value</rootDirectory>
</preferenceStore>
```

You can change the file preference store location for other PDK-Java portlet producers in the same way.

If you use a database preference store for OmniPortlet and other PDK-Java portlet producer, then you must set up the database preference store. For more information, see the "Setting Up a Preference Store" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter*.

Upgrading Oracle Business Rules Dictionaries and Projects

You can migrate Oracle Business Rules dictionaries and projects from previous releases. Before you can use a dictionary created with Oracle Business Rules Release 10.1.3.x with Oracle Business Rules for Oracle Fusion Middleware 11g Release 1 (11.1.1), you must migrate the dictionary to the new dictionary format. Depending on how the Release 10.1.3.x dictionary was created, there are several possible migration paths to migrate a dictionary from Oracle Business Rules Release 10.1.3.x format to Oracle Fusion Middleware 11g Release 1 (11.1.1).

This appendix includes the following sections:

- [Section C.1, "Using Oracle JDeveloper to Migrate an Oracle Business Rules Dictionary"](#).
- [Section C.2, "Using Rule Migrator Tool to Migrate an Oracle Business Rules Dictionary"](#)
- [Section C.3, "Using MigrateRuleRepository with Oracle Business Rules SDK to Migrate a Dictionary"](#)
- [Section C.4, "Oracle Business Rules Manual Migration Tasks"](#)

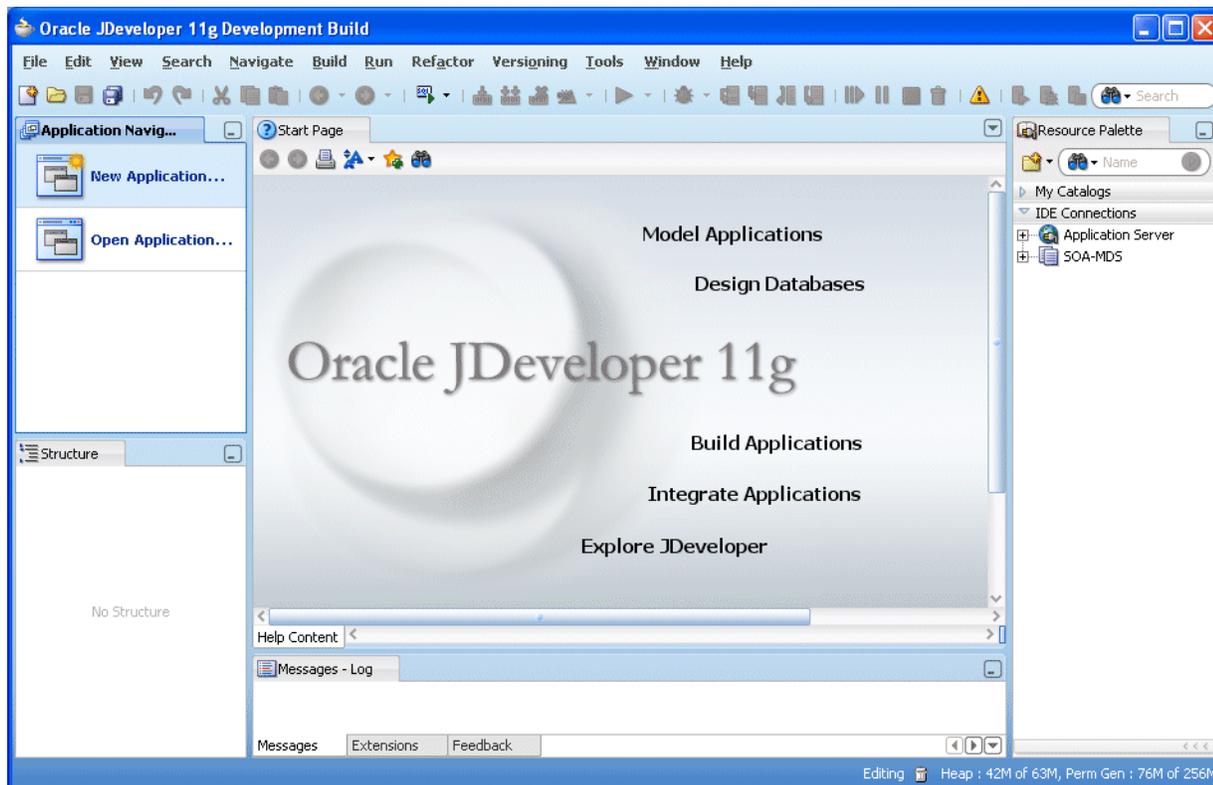
C.1 Using Oracle JDeveloper to Migrate an Oracle Business Rules Dictionary

When a dictionary is part of an application created in a 10.1.3.x BPEL Decision Service, you can migrate to the Oracle Fusion Middleware 11g Release 1 (11.1.1) dictionary format when you open the BPEL process in Oracle JDeveloper.

To use Oracle JDeveloper to migrate a dictionary:

1. Start Oracle JDeveloper, as shown in [Figure C-1](#), this displays the Oracle JDeveloper start page.

Figure C-1 Creating an Application in Oracle JDeveloper



2. From the **File** menu select **Open**.
3. Navigate to your project and select **Open**.

JDeveloper invokes the Oracle BPEL Process Manager migration utility to convert the project, including its rules dictionary.

4. Examine the **Messages - Log** window for any errors.

In some cases, you may need to manually edit the migrated rules dictionary using the Oracle Business Rules Designer to complete the migration. For more information, see [Section C.4, "Oracle Business Rules Manual Migration Tasks"](#).

5. From the **File** menu select **Save All**.

C.2 Using Rule Migrator Tool to Migrate an Oracle Business Rules Dictionary

You can use the `MigrateRuleRepository` command-line tool to migrate a dictionary from 10.1.3.x to Oracle Fusion Middleware 11g Release 1 (11.1.1) format. Note that most migration scenarios require that you perform manual migration tasks because 10.1.3.x has concepts that have been superseded in Oracle Fusion Middleware 11g Release 1 (11.1.1) by updated and more powerful constructs. For more information on the manual migration tasks, see [Section C.4, "Oracle Business Rules Manual Migration Tasks"](#).

The command-line tool is a Java class called the following:

```
oracle.rules.tools.migrator.MigrateRuleRepository
```

It is located in the following directory inside the SOA Oracle home:

```
SOA_HOME/soa/modules/oracle.rules.migration_11.1.1/migrator.jar
```

To run the command-line tool, you must have a specific set of JAR files available in the classpath. The list of required JAR files and libraries is included in [Section C.3.1, "How to Migrate a Dictionary with Oracle Business Rules SDK"](#).

For example, to set the classpath and run the migrator on Linux:

```
> export CLASSPATH=required_jar_files
> java -cp $CLASSPATH oracle.rules.tools.migrator.MigrateRuleRepository OPTIONS
```

The following shows a sample MigrateRuleRepository command:

```
MigrateRuleRepository -all
    -origType FILE_REPO
    -origLocation /scratch/MyRepository
    -destLocation /scratch/MyMigratedDictionaries
    -importedClasspath /scratch/foo.jar:/scratch/myclasses
    -xsdDirectory /scratch/myschemas
    -xsdGeneration /scratch/jaxb_classes
```

[Table C-1](#) lists the required MigrateRuleRepository command-line arguments.
[Table C-2](#) lists the optional MigrateRuleRepository command-line arguments.
[Table C-3](#) lists the WebDAV MigrateRuleRepository command-line arguments.

Table C-1 Required Migrate Rule Repository Options

| Option | Description |
|--|--|
| -origType {FILE_REPO WEBDAV_REPO} | Specify one of these strings to indicate the type of the origin repository. |
| -origLocation <i>origin-repo</i> | Argument specifies the Release 10.1.3.x repository containing a dictionary to be migrated. This is specified either with string that is either a URL to a WebDAV repository or a file path. |
| -destLocation <i>dir-name</i> | Specifies the destination directory in which the migrated dictionaries are placed. The migrated dictionaries are placed in a directory structure constructed with the -destLocation, -destPackageName and -destDictionaryName arguments. For example, <i>dir-name/dest-pkg/dest-dict-name.xml</i> The <i>pkg-name</i> is migrated into directories created in <i>dir-name</i> . |
| -importedClasspath <i>classpath</i> | Specifies the classpath to use for importing Java fact types during migration. |
| -xsdDirectory <i>schema-dir</i> | Specifies the schema directory which contains all schemas used in XML fact types. |
| -xsdGeneration <i>jaxb-class-destination</i> | Specifies the directory to generate JAXB classes into when importing schemas. |
| -all | Migrate all dictionaries in the origin repository. Either use -all, or specify the -origDictionaryName and the -origVersionName |
| -origDictionaryName <i>origin-dict</i> | Do not use with the -all option Specifies the name of the dictionary to be migrated. |
| -origVersionName <i>origin-version</i> | Do not use with -all option. Specifies the version of dictionary to be migrated. |

Table C–2 Optional Migrate Rule Repository Options

| Option | Description |
|---|---|
| <code>-destPackageName</code> <i>dest-pkg</i> | Specifies the package name to use when saving the migrated result dictionary in Oracle Business Rules Oracle Fusion Middleware 11g Release 1 (11.1.1) format. The migrated dictionaries are placed in a directory structure constructed with the <code>-destlocation</code> , <code>-destPackageName</code> and <code>-destDictionaryName</code> arguments. |
| <code>-destDictionaryName</code> <i>dest-dict</i> | Specifies the dictionary name to use when saving the Oracle Business Rules Oracle Fusion Middleware 11g Release 1 (11.1.1) format result dictionary. The migrated dictionaries are placed in a directory structure constructed with the <code>-destlocation</code> , <code>-destPackageName</code> and <code>-destDictionaryName</code> arguments. |

Table C–3 WebDAV Only Migrate Rule Repository Options

| Option | Arguments and Description |
|--|---|
| <code>-originPasswordSource</code> <i>webdav-password-src</i> | Argument specifies the Oracle Wallet file containing the WebDAV authentication information. |
| <code>-originProxyHost</code> <i>webdav-proxy-host</i> | When a proxy is involved, this option is required. Argument specifies the host to use as proxy for WebDAV access, if necessary. |
| <code>-originProxyPort</code> <i>webdav-proxy-port</i> | When a proxy is involved the argument specifies the port to use on proxy host for WebDAV access, if necessary. |

C.3 Using MigrateRuleRepository with Oracle Business Rules SDK to Migrate a Dictionary

Using the `MigrateRuleRepository` programmatic interface with Oracle Business Rules SDK. Note that most migration scenarios require that you perform manual migration tasks.

C.3.1 How to Migrate a Dictionary with Oracle Business Rules SDK

To migrate using Java with Oracle Business Rules SDK, you use the `oracle.rules.tools.migrator.MigrateRuleRepository` class.

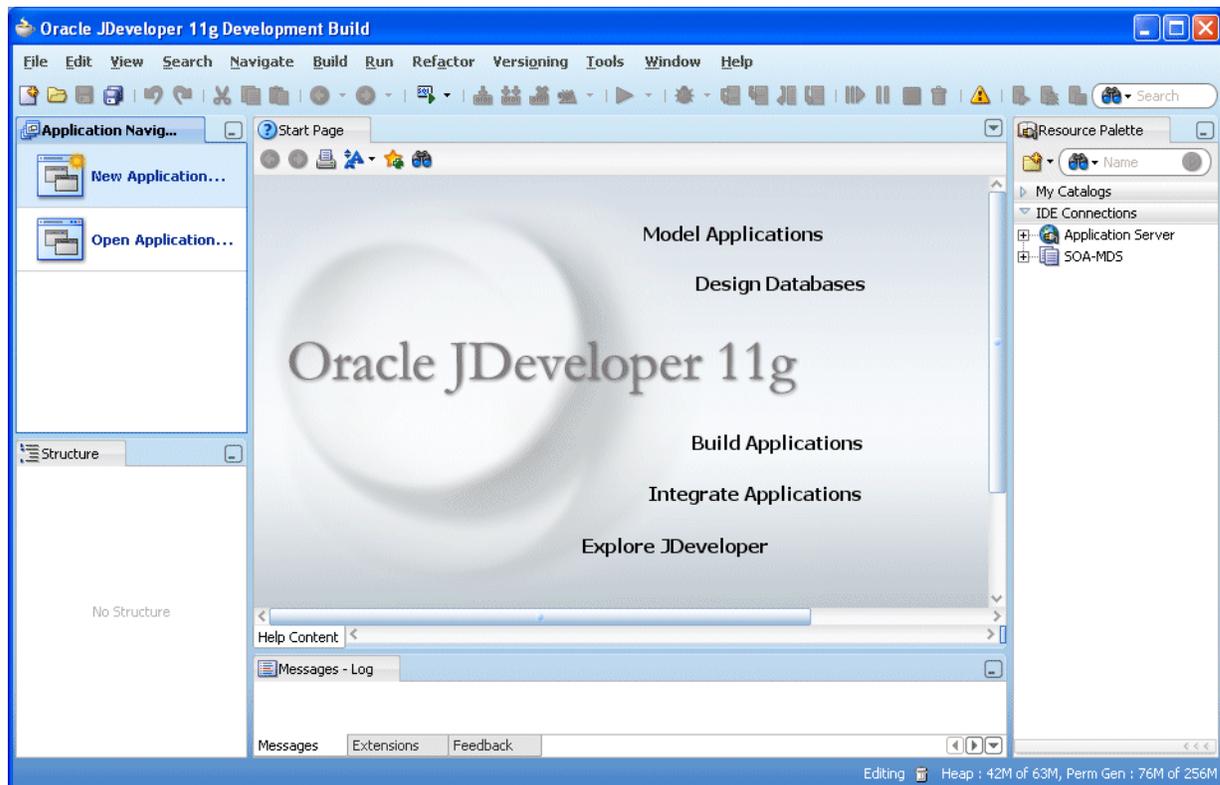
In the following example, you migrate the dictionary in the `how-to-rules-java` sample located here:

http://www.oracle.com/technology/products/ias/business_rules/index.html.

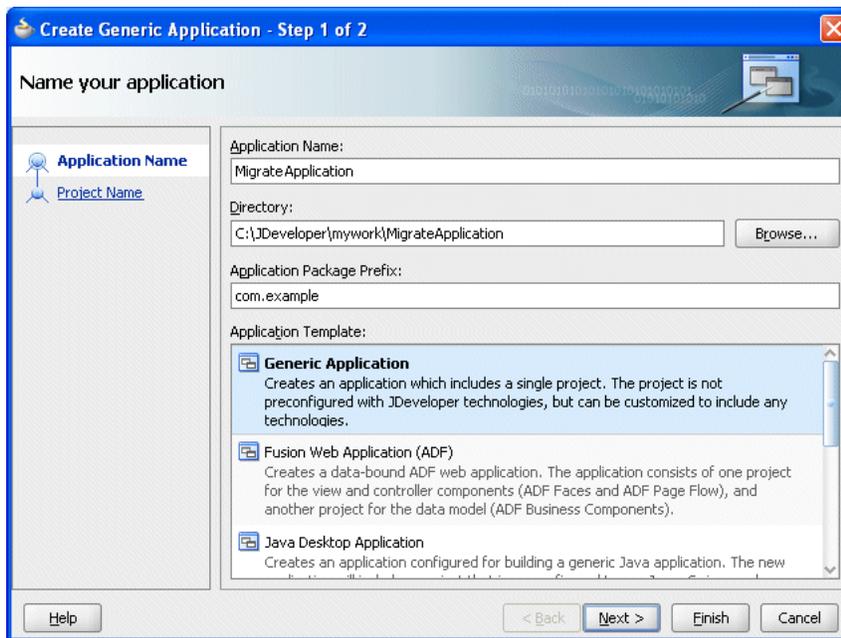
To migrate a dictionary using Oracle Business Rules SDK:

1. Start Oracle JDeveloper, as shown in [Figure C–2](#), this displays the Oracle JDeveloper start page.

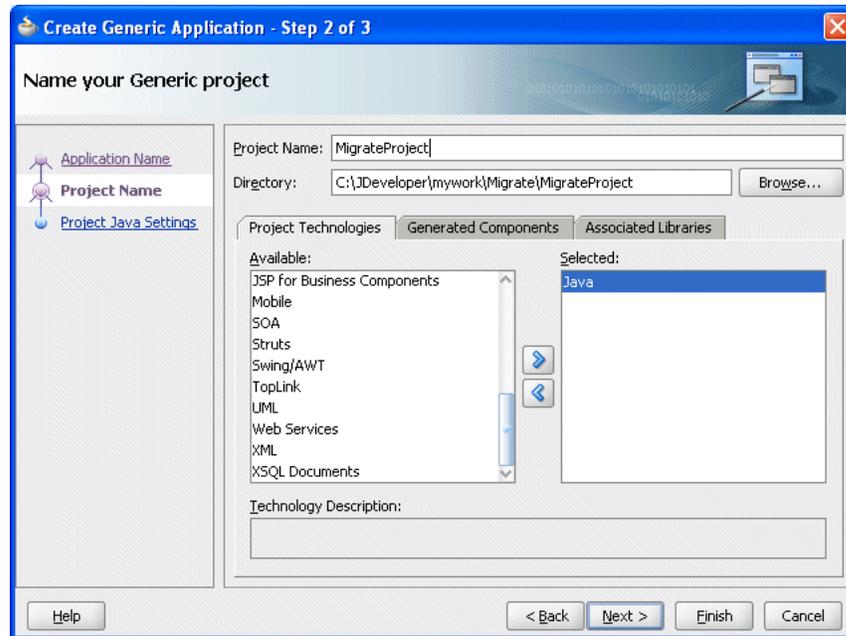
Figure C-2 Creating an Application in Oracle JDeveloper



2. In the Application Navigator, click **New Application** if no applications have been created, or if applications have already been created click **Applications** and from the drop-down list choose **New Application**.
3. In the Create Application wizard, enter the name and location for the new application:
 - a. In the **Application Name** field, enter an application name. For example, enter `MigrateApplication`.
 - b. Enter or browse for a directory name, or accept the default.
 - c. Enter an application package prefix or accept the default, no prefix.
This should be a globally unique prefix and is commonly a domain name owned by your company. The prefix, followed by a period, applies to objects created in the initial project of an application.
In this sample, you can use the prefix `com.example`.
 - d. For an Oracle Business Rules project, select **Generic Application** for the application template, as shown in [Figure C-3](#).

Figure C-3 Adding the Migrate Application

4. Click **Next**.
5. In the Create Generic Application wizard - Name your Generic project page, enter the name and location for the new project as shown in [Figure C-4](#):
 - In the **Project Name** field, enter an application name. For example, enter `MigrateProject`.
 - Enter or browse for a directory name, or accept the default.
 - On the Project Technologies tab, in the Available list, select **Java** and click **Move** to add it to the Selected area.

Figure C-4 Specifying Technologies in a Project

6. Click **Finish**.
7. In Oracle JDeveloper, select the project named **MigrateProject**.
8. Right-click and from the drop-down list select **Project Properties**.
9. Select the **Libraries and Classpath** item.
10. Click **Add JAR/Directory**.

You must update your project classpath with the following JAR files.

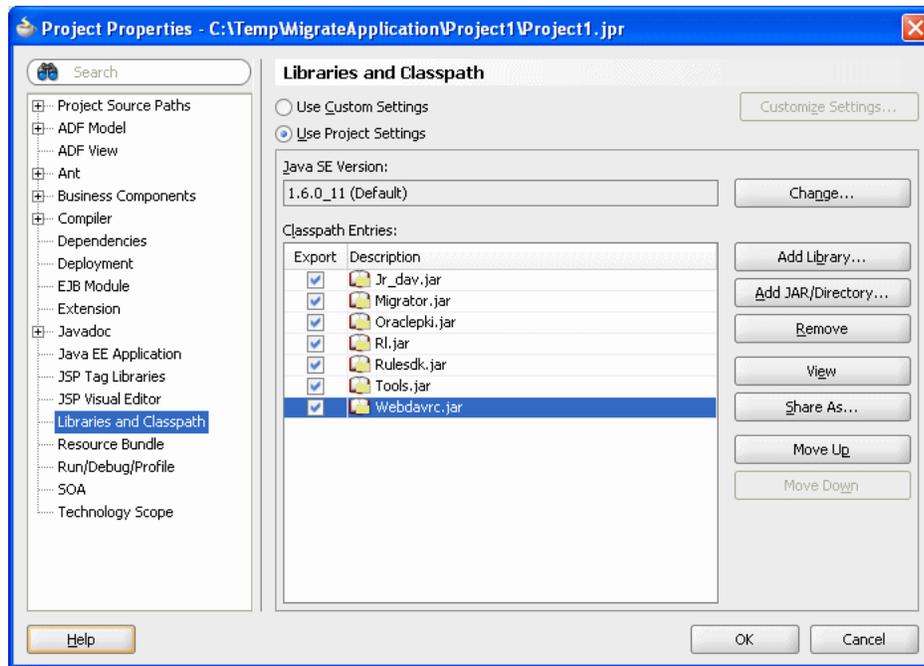
Note that there may be a JAVA_HOME installed next to JDEV_HOME, in your ORACLE_HOME directory.

```

JAVA_HOME/lib/tools.jar
JDEV_HOME/soa/modules/oracle.rules.migration_11.1.1/migrator.jar
JDEV_HOME/soa/modules/oracle.rules.migration_11.1.1/rulesdk.jar
JDEV_HOME/modules/oracle.adf.model_11.1.1/jr_dav.jar
JDEV_HOME/modules/oracle.pki_11.1.1/oraclepki.jar
JDEV_HOME/soa/modules/oracle.rules_11.1.1/r1.jar
JDEV_HOME/soa/modules/oracle.rules.migration_11.1.1/webdavrc.jar

```

11. In the Add Archive or Directory dialog, select the required JAR file and click **Select** as [Figure C-5](#) shows.

Figure C–5 Project Properties Dialog: migrator.jar

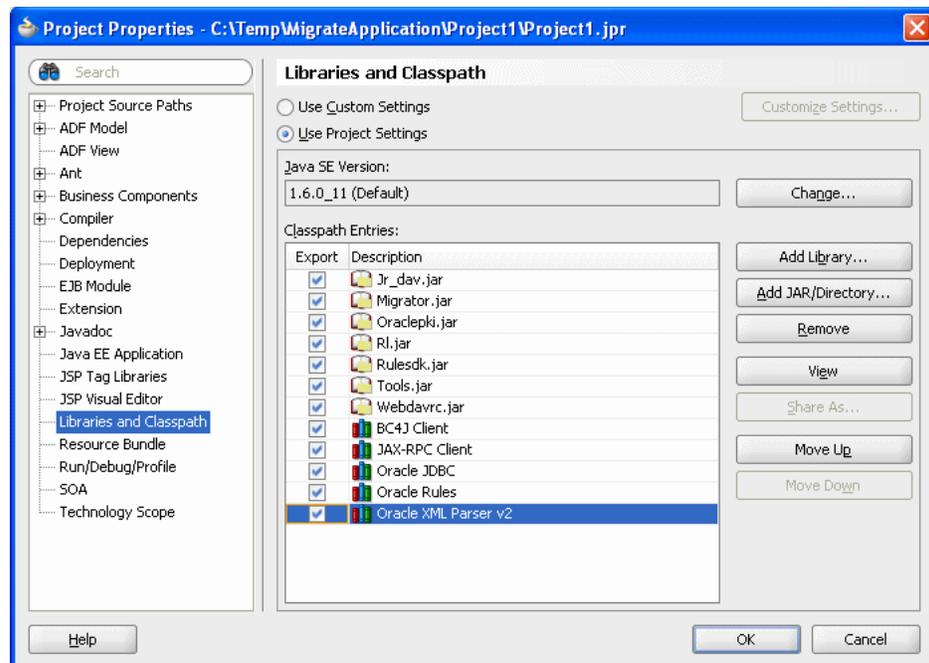
12. Repeat from step 10 for the remaining JAR files.

13. Click **Add Library**.

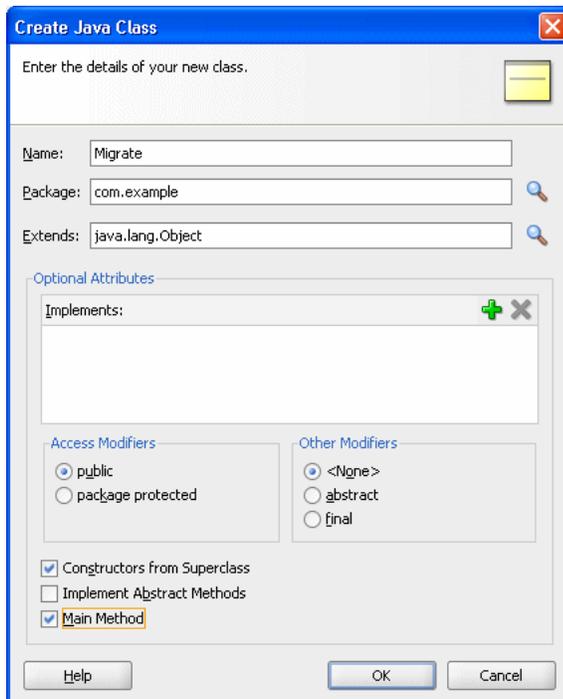
You must add the following libraries to your project classpath:

- **BC4J Client**
- **Oracle JDBC**
- **Oracle Rules**
- **Oracle XML Parser v2**
- **JAX-RPC Client**

14. In the Add Library dialog, select the required library and click **OK** as [Figure C–6](#) shows.

Figure C–6 Project Properties Dialog: Oracle Rules Library

15. Repeat from step 13 for the remaining libraries.
16. Click **OK**.
17. In Oracle JDeveloper, select the project named **MigrateProject**.
18. Right-click and from the drop-down list select **New**.
19. In the New Gallery, in the **Categories** area, select **General**.
20. In the New Gallery, in the **Items** area, select **Java Class**.
21. Click **OK**.
22. In the Create Java Class window, configure the following properties as shown in [Figure C–7](#):
 - Enter the **Name** value `Migrate`.
 - Enter the **Package** value `com.example`.
 - Check the following check boxes:
 - **Public**
 - **<None>**
 - **Constructors from Superclass**
 - **Main Method**

Figure C-7 Creating the Migrate.java Class

23. Click **OK**.

Oracle JDeveloper displays the Java Class, as shown in [Example C-1](#).

Example C-1 Code Created for New Migrate.java Class

```
package com.example;

public class Migrate {
    public Migrate() {
    }

    public static void main(String[] args) {
        Migrate migrate = new Migrate();
    }
}
```

24. Use the `MigrateRuleRepository` API to add code like that shown in [Example C-2](#).

For more information about the `MigrateRuleRepository` API, see [Section C.4.3, "What You May Need to Know About Manual Migration"](#).

Example C-2 Migrate.java Completed

```
package com.example;

import java.io.PrintWriter;

import oracle.rules.tools.migrator.MigrateRuleRepository;

public class Migrate {
    public Migrate() {
        try {

            // Create a migrator instance
```

```

MigrateRuleRepository conv = new MigrateRuleRepository();

// Set the output buffer
conv.setOutLog(new PrintWriter(System.out));

// Set input properties (SDK format)
conv.setOriginLocation("C:\\Temp\\how-to-rules-java\\dict\\CarRepository");
conv.setOriginType(MigrateRuleRepository.FILE_REPO);
conv.setOriginDictionaryName(MigrateRuleRepository.MIGRATE_ALL);
conv.setOriginVersionName(MigrateRuleRepository.MIGRATE_ALL);
conv.setImportedClassPath("C:\\Temp\\how-to-rules-java\\lib\\car.jar");

// Set output properties
conv.setDestinationLocation("C:\\Temp\\how-to-rules-java\\dict\\CarRepositorySDK2");

conv.migrate();
    String results = conv.getTotalStats();
    System.out.print(results);
}
catch (Exception e)
{
    System.out.print(e);
}
}

public static void main(String[] args) {
    Migrate migrate = new Migrate();
}
}

```

25. If the repository you are migrating contains XML Fact Types, you need to specify the location of the schema file that defines these XML Fact Types, and also the output directory that JAXB should use when converting schema files into Java classes.

Since there are no XML Fact Types in this demo, these lines of code are not needed:

```

conv.setXSDLocation("C:\\Temp\\how-to-rules-java\\schemas"); // or other
    directory, as appropriate
conv.setXSDOutputDirectory("C:\\Temp\\how-to-rules-java\\jaxbOutput"); // or
    other directory, as appropriate

```

26. In the Application Navigator, right-click `Migrate.java` and select **Make**.
27. In the Application Navigator, right-click `Migrate.java` and select **Run**.
28. Examine the **Messages - Log** window for any errors.

In some cases, you may need to manually edit rules to complete the migration.

29. Retrieve the migrated dictionary from the destination location.

In this example, from

```
C:\Temp\how-to-rules-java\dict\CarRepositorySDK2.
```

C.3.2 What You May Need to Know About the MigrateRuleRepository API

For complete details on the `MigrateRuleRepository` API, see the *Oracle Fusion Middleware Java API Reference for Oracle Business Rules*.

As shown in [Example C-3](#), a typical migration program uses the `MigrateRuleRepository` API to set:

- [Input Properties](#)
- [Output Properties](#)

C.3.2.1 Input Properties

Input properties define characteristics of the dictionary that you want to migrate.

[Example C-3](#) shows a migration program setting typical input properties.

Example C-3 Input Properties

```
// Set input properties (SDK format)
conv.setOriginLocation( "C:\\scratch\\SDK1_Repository" );
conv.setOriginType(MigrateRuleRepository.FILE_REPO);
conv.setOriginDictionaryName(MigrateRuleRepository.MIGRATE_ALL);
conv.setOriginVersionName(MigrateRuleRepository.MIGRATE_ALL);
```

You must set the following mandatory input properties using `MigrateRuleRepository` method:

- `setOriginLocation` to specify the fully qualified path to the rules dictionary to migrate.
- `setOriginType` to specify the type of repository that contains the rules dictionary. You can choose any of the following constants:
 - `MigrateRuleRepository.FILE_REPO`: the rules dictionary is stored in the host file system as a simple file.
 - `MigrateRuleRepository.WEBDAV_REPO`: the rules dictionary is stored in a WebDAV-managed repository.
- `setOriginDictionaryName` to specify the name of the dictionary contained by the specified rules dictionary to migrate. Specify either the dictionary name as a `String` or `MigrateRuleRepository.MIGRATE_ALL` to migrate all dictionaries contained by the specified dictionary.
- `setOriginVersionName` to specify the version of the specified rules dictionary to migrate. Specify either the dictionary version as a `String` or `MigrateRuleRepository.MIGRATE_ALL` to migrate all versions of the specified dictionary.

You may use other `MigrateRuleRepository` methods to define optional input properties as required.

C.3.2.2 Output Properties

Output properties define characteristics of the Oracle Fusion Middleware 11g Release 1 (11.1.1) Oracle Business Rules dictionary that the `MigrateRuleRepository` class migrates to, given the input properties you specify (see [Section C.3.2.1, "Input Properties"](#)). [Example C-4](#) shows a migration program setting typical input properties.

Example C-4 Output Properties

```
// Set output properties (SDK2 format)
conv.setDestinationLocation( "C:\\scratch\\SDK2_Migrated_Dictionaries" );
```

You must set the following mandatory output properties using `MigrateRuleRepository` method:

- `setDestinationLocation` to specify the fully qualified file name of the new Oracle Fusion Middleware 11g Release 1 (11.1.1) Oracle Business Rules based rules dictionary.

You may use other `MigrateRuleRepository` methods to define optional output properties as required.

For example, by default, the new Oracle Fusion Middleware 11g Release 1 (11.1.1) Oracle Business Rules dictionary uses the same package name as that used in the original dictionary. To override this behavior, you can specify a new package name using `MigrateRuleRepository` method `setDestinationPackageName`.

C.4 Oracle Business Rules Manual Migration Tasks

Before you can use an a dictionary created with Oracle Business Rules Release 10.1.3.x, you must migrate it to the format required for Oracle Fusion Middleware 11g Release 1 (11.1.1). The easiest way to migrate a dictionary that is part of a Decision Service in a BPEL process is to use `JDeveloper`. Alternatively, you can migrate using the `MigrateRuleRepository` command-line tool. Certain migration scenarios require you to perform manual migration tasks.

C.4.1 How to Migrate JAXB 1.0 to JAXB 2.0

To take advantage of the significant usability improvements present in JAXB 2.0, you must perform the following migration tasks manually.

The most significant difference between JAXB 1.0 and JAXB 2.0 is the classes which are generated. JAXB 1.0 was created before Java 1.5 introduced annotations, so it relies on a series of interfaces and abstract classes. The default in Oracle Business Rules Release 10.1.3.x was to use JAXB 1.0. In Oracle Fusion Middleware 11g Release 1 (11.1.1), the default is to use JAXB 2.0. Rules Designer only allows XML schemas to be imported using the JAXB 2.0. Migrated dictionaries continue to use JAXB 1.0, but the schema cannot be modified or reimported as JAXB 1.0.

A migrated dictionary which uses JAXB 1.0 objects still works with Oracle Fusion Middleware 11g Release 1 (11.1.1). However, JAXB 2.0 provides significant new features and many improvements for working with Oracle Business Rules.

Consider the XSD fragment that [Example C-5](#) shows.

Example C-5 XSD Fragment

```
<element name="foo">
  <complexType>
    <sequence>
      <element name="bar" type="string"/>
      <element name="baz" type="string"/>
    </sequence>
  </complexType>
</element>
```

This element uses an anonymous complex type to define a data structure. JAXB 1.0 generates the following classes:

- `FooType`
- `Foo` extends `FooType`, `javax.xml.bind.Element`
- `FooImpl` extends `FooTypeImpl` implements `Foo`
- `FooTypeImpl` extends `oracle.xml.jaxb.JaxbNode` implements `FooType`

The only way to create instances of these classes is the static factory methods in `ObjectFactory`.

By contrast, JAXB 2.0 generates a single annotated class, `Foo`, which can be instantiated using either `new` or an `ObjectFactory` method.

In the previous release, rules were written using the `FooType` fact type, whereas in 11g Release 1 (11.1.1), they are written using the `Foo` fact type.

To migrate from `FooType` to `Foo` fact types, perform the following manual migration task.

To migrate JAXB 1.0 to JAXB 2.0:

1. Use Rule Migrator to migrate.

For more information, see [Section C.2, "Using Rule Migrator Tool to Migrate an Oracle Business Rules Dictionary"](#).

2. Delete all classes in `SCA-INF/classes`.
3. Record the aliases assigned to all of the fact types used in your rules.
For example, `FooType` fact alias "My Foo Type".
4. Delete from the data model the XML Fact Types you want to upgrade to JAXB 2.0.

This operation can result in a large number of validation warnings, one for every reference to the now-deleted fact types.

However, the SDK stores references to both the ID and `String` value of all referenced elements.

If the reference ID is set and it exists, then the `String` value is updated to reflect its current alias. So, if the alias of a fact type is changed, all of the places where that fact type is referenced get the new value because of the reference id.

If the element with the ID is removed from the dictionary, the `String` value retains the last known name of the referenced element. This allows you to, for example:

1. Delete a fact type with alias "FactType1"
2. Import a new fact type.
3. Set the alias of the new fact type to "FactType1" and all of the references to the deleted fact type automatically change to the new fact type.
5. Import the schema.
In 11g Release 1 (11.1.1), JAXB 2.0 is the default.
6. Set the alias of the fact type to the previous alias.
For example, change the alias of `Foo` to `FooType`.
7. Run validation on the dictionary.
8. If desired, set the alias of the fact type back to the original alias.
For example, change the alias `FooType` back to `Foo`.
9. Confirm that all of the references are now be updated.

C.4.2 How to Migrate RL Functions

In the previous release, the only option for creating a Function was to write free form RL. These functions relied on generated RL names, which have changed in 11g Release 1 (11.1.1): the RL names are now automatically computed from the alias so they are RL-compliant, instead of relying on the user to provide a valid name.

You must re-write any function which accesses globals to comply with this change by using Actions. It is easier to write functions using Actions because they are validated at design-time.

In Oracle Business Rules Release 10.1.3.x dictionaries that contain free-form RL functions, it was a common practice to refer to Java classes which were not in the data model. For example, many dictionaries include functions which use ObjectFactory to create instances of JAXB classes. In Oracle Fusion Middleware 11g Release 1 (11.1.1), to include this class in the data model you cannot simply reimport the schema and import ObjectFactory because it is imported as a JAXB 2.0 class. The solution in this case is to use the ObjectFactory class which is in the "classes" directory in your project from when the dictionary was migrated. This ObjectFactory class should be imported as a Java class and can then be referenced in Oracle Fusion Middleware 11g Release 1 (11.1.1) functions.

For more information, see [Section C.4.3, "What You May Need to Know About Manual Migration"](#).

To migrate RL functions:

1. For each free form RL function, re-write the RL to use Actions.

In 11g Release 1 (11.1.1) you may use Actions both in Function bodies and in rules. There are several new Action forms which in most cases eliminate the need for free form function bodies. Action forms available include:

- Call
- Assert and Assert new
- Retract
- Assign and Assign New
- Modify
- Synchronized
- Expression
- Assert
- If, Else If, and Else
- For and While
- Return, Throw, Try, Catch, and Finally
- RL

2. Validate your dictionary.

For more information, see "Understanding Decision Table Validation" in the *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.

C.4.3 What You May Need to Know About Manual Migration

In the previous release, the only option for XML schema import was JAXB 1.0. In 11g Release 1 (11.1.1), Oracle Business Rules provides support for JAXB 1.0, but JAXB 2.0 is the default. If you choose to continue to use JAXB 1.0, generated RL may throw `MultipleInheritanceException`. For more information, see "JAXB 1.0 Dictionaries and RL `MultipleInheritanceException`" in the *Oracle Fusion Middleware User's Guide for Oracle Business Rules*.

The SDK does not do any validity checking on free form RL Function bodies. As a result, any errors in the functions are not revealed until the RL is executed.

[Example C-6](#) shows what such an error looks like in a BPEL process:

Example C-6 Invalid RL Function Runtime Error

```
[2008-10-01T04:52:15.043-07:00] [server_soa] [ERROR] [] [oracle.soa.services.rules] [tid: 35]
[ecid:0000HmsG4oAEshQ6ubV4EH18sYBt0000JA,0:2:100000051] [composite_name:ThreePattern]
[component_name: CreditRatingRules] [component_instance_
id:90558a4f-8781-4ac7-821c-49f46dd2f8fc] [composite_instance_id: 90033] <.>
Error caching the Decision Services metadata.[]
Error caching the decision services metadata for path
default/ThreePattern!1.0*c04f9ced-55ff-42fc-a6ee-f64a2055baf2/CreditRatingRules.
Check the underlying exception and correct the error. This is most likely due
to a rule modeling issue. Validate the rule dictionary in rule designer and
fix any errors and warnings. Contact oracle support if error is not fixable.
ORABPEL-36109 Error caching the Decision Services metadata.
Error caching the decision services metadata for path
default/ThreePattern!1.0*c04f9ced-55ff-42fc-a6ee-f64a2055baf2/CreditRatingRules.
Check the underlying exception and correct the error. This is most likely due to a rule
modeling issue. Validate the rule dictionary in rule designer and fix any errors and
warnings. Contact oracle support if error is not fixable.
    at
oracle.bpel.services.rules.impl.DecisionServiceCache.cacheDecisionServiceMetadata(DecisionSer
viceCache.java:1039)
    at
oracle.bpel.services.rules.impl.DecisionServiceCache.prepare(DecisionServiceCache.java:343)
    at
oracle.bpel.services.rules.impl.DecisionServiceImpl.preProcess(DecisionServiceImpl.java:1164)
...

Caused by: oracle.rules.rl.exceptions.UndefinedException: symbol 'CreditRatingRules.rating_
param' is undefined at line 25 column 8 in main
    at
oracle.rules.rl.exceptions.ExceptionFactory.createUndefinedException(ExceptionFactory.java:38
6)
    at oracle.rules.rl.analyze.Expr.qname(Expr.java:1742)
    at oracle.rules.rl.analyze.Expr.primary(Expr.java:814)
    at oracle.rules.rl.analyze.Expr.expr(Expr.java:195)
...
```

Numerics

- 10.1.3.x dictionary format
 - migrating from, C-1
- 10g Release 2 (10.1.2)
 - information for users of, 2-1
 - information on upgrading from, 2-1
- 10g Release 3 (10.1.3), 2-1, 3-1, 3-3
- 10g Release 3 (10.1.3.1.0), 2-1
- 10g Release 3 (10.1.3.2.0), 2-1
- 10g Release 3 (10.1.3.4), 2-1
 - verifying SOA Order Booking on, A-2
- 10g Users
 - information for, 3-1
- 11.1.1 dictionary format
 - migrating to, C-1

A

- adapters
 - See* technology adapters
- ADF
 - See* Oracle ADF
- ADF runtime libraries, 3-1
- ADF security, 10-11
- ant-sca-compile.xml, 9-10
- ant-sca-deploy.xml, 9-10
- ant-sca-upgrade.xml, 9-10, 9-11, 9-12
- APIs
 - available in the JRF template, 5-2
- Application Developer
 - Oracle Fusion Middleware software suite, 5-2
- Application Development Framework Runtime, 5-3
- Application Programming Interfaces (APIs)
 - changes for 11g, 9-3
 - Oracle Business Rules, 9-3
- application upgrade
 - log files generated during, 8-4
 - opening and upgrading the application in
 - JDeveloper, 8-3
 - overview, 8-1
 - redeploying on 11g, 8-5
 - reviewing and recompiling, 8-4
 - upgrading SOA applications, 9-1
 - verifying on 10g, 8-2
- ApplyJRF

- WLST command, 5-3
- AS Common Schemas, 6-3
 - schema category, 7-2
- auditDetailThreshold, 9-17
- auditLevel, 9-17
- authentication methods
 - External Application, B-7
 - Identity Propagation, B-7

B

- B2B
 - See* Oracle B2B
- B2B Schema
 - option on the Upgrade Assistant Specify Operation screen, 6-9
- backing up WebCenter application, 10-5
- BAM Schema
 - option on the Specify Operation screen, 6-11
- Basic Installation option
 - for Oracle Application Server 10.1.3, 6-5
- bpelcClasspath, 9-17
- bpel-config.xml, 9-17
- bpel.xml
 - locating external references in 10g, A-4
 - verifying properties in, 9-15
- build.properties
 - recreating after upgrade, 9-7
- build.xml
 - recreating after upgrade, 9-7

C

- Callout library jar files
 - upgrading to 11g, 6-13
- caseSensitive
 - property in the workflow-identity-config.xml file, 6-12
- channel passwords
 - upgrading to 11g, 6-14
- Check for Updates
 - JDeveloper command, 9-2, A-12
- cis-client-8.0.0.jar, downloading, 10-12
- client applications
 - upgrading, 9-32
- clustering, 6-1

- compatibility between WebCenter consumer and portlet producer applications, 10-14
- Composite Editor
 - in Oracle JDeveloper, A-20
- composite services, 9-11
- composite.xml, 9-8, 9-15
- Content Integration Suite API, 10-12
- content repository connections, credential migration, B-7
- correlation sets
 - in an asynchronous service, 9-16
- cross references
 - upgrading with command-line upgrade tool, 9-11
- customizable component migration changes
 - about, B-1
 - general, B-2
 - Panel Customizable, B-2
 - Show Detail Frame, B-4
- customization store
 - changing location
 - OmniPortlet and other PDK-Java producers, B-10
 - Web Clipping, B-10
 - WSRP portlet producers, B-10
 - default location, B-10
- customizations
 - non-preconfigured portlet producers, 10-19
 - preconfigured portlet producers, 10-18
 - migrating from default location, 10-18
 - migrating from non-default location, 10-18

D

- data source
 - creating in WebLogic Server, A-6
- database
 - upgrading in preparation for WebCenter upgrade, 7-1
 - verifying supported version for 11g, 6-2
 - verifying version requirements, 7-1
- DataObjectOperations, 9-25
- DataObjectOperations10131, 9-25
- datasourceJndi, 9-17
- DbAdapter application
 - configuring for SOA Order Booking, A-7
- deliveryPersistPolicy, 9-18
- dependency services, 9-13
- dependent JAR files
 - reviewing projects for, 9-6
- deployment descriptor
 - bpel.xml, 9-15
- dictionary
 - migrating, C-1
- directory structure
 - of Oracle WebCenter in 10g and 11g, 3-5
- document protocol revision and document type
 - upgrading to 11g, 6-13
- domain templates, 5-2
- domain value maps
 - deploying metadata in a MAR file, 9-12

- upgrading with command-line upgrade tool, 9-11
- domain.xml, 9-17
- dspEngineThreads, 9-18
- dspInvokeAllocFactor, 9-18
- dspInvokeThreads, 9-18
- dspMaxThreads, 9-18
- dspMinThreads, 9-19
- dspSystemThreads, 9-19
- DVMs
 - See also* domain value maps, 9-11

E

- EDI EDIFACT
 - Oracle B2B document type, 6-14
- EDI X12
 - Oracle B2B document type, 6-14
- enterprise deployment
 - upgrading, 3-3, 6-1
- Enterprise Link for Oracle BAM, 9-26
- ESB
 - See* Oracle Enterprise Service Bus
- esb_config.ini, 9-15
- exchange protocol parameter identifiers
 - upgrading to 11g, 6-13
- exchange protocol parameters for the MLLP channel
 - upgrading to 11g, 6-14
- expirationMaxRetry, 9-19
- External Application authentication method, B-7
- external application migration changes, B-5
- external partners
 - verifying availability during upgrade, 6-2
- external references
 - verifying before upgrade, A-4

F

- fault policies
 - upgrading in a BPEL project, 9-14
- fault-policy.xml, 9-15
- fileAdapterOutboundHeader.wsdl, 9-28
- filterExpression, 9-22
- flow chart
 - of upgrade process, 1-1
- Fusion Middleware Control
 - See* Oracle Enterprise Manager Fusion Middleware Control

H

- high availability, 6-1
- host and partner security certificates
 - upgrading to 11g, 6-13
- Human Workflow Applications
 - limitations when upgrading with the command-line upgrade tool, 9-9

I

- IBM WebSphere, 9-37
- Identity Propagation authentication method, B-7

- inbound and outbound partner agreements
 - in Oracle B2B, 6-14
- installing Oracle WebCenter extension, 10-3
- instanceKeyBlockSize, 9-19

J

- Java EE
 - features
 - configuring for SOA applications, 6-11
 - configuring for WebCenter applications, 7-4
 - required to support your SOA applications, 6-11
 - required to support your WebCenter applications, 7-4
- Java EE applications
 - upgrade tasks for all, 9-2
- Java EE components
 - documentation resources when upgrading, 8-4
- Java Required Files, 5-2
 - configuring or extending a domain with, 3-1
 - creating a domain with, 5-2
 - domain template, 3-1
 - extending a domain with, 5-2
- JDeveloper
 - See* Oracle JDeveloper
- JMS resources
 - creating for SOA Order Booking, A-8
- JRF
 - See also* Java Required Files, 3-1
- JSR 168 portlet application, 10-16

L

- largeDocumentThreshold, 9-19
- LDAP directory
 - associating with upgraded SOA components, 6-12
- log files
 - when upgrading applications in JDeveloper, 8-4
- long-running processes
 - approaches to upgrading, 4-1
 - definition of, 4-1
 - effect on upgrade process, 4-1
 - side-by-side upgrade approach, 4-2
 - upgrading to BPEL Process Manager 11g, 4-1
 - using a grace period and a hard stop of, 4-3

M

- maximum message size
 - WebLogic Server attribute
 - setting for B2B schema upgrade, 6-7
- metadata archive (MAR) file, 9-12
- Metadata Services
 - schema
 - installing for SOA Suite, 6-3
 - schema category, 7-2
- Middleware home, 3-3
- migrating dictionaries, C-1
- migrating WebCenter consumer application

- backing up an application, 10-5
- configuring ADF security, 10-11
- Migrating in JDeveloper, 10-7
- Oracle Portal connections, 10-11
- removing Portlets project, 10-7
- upgrading to Oracle WebCenter 10.1.3.4, 10-6
- Migration Wizard, 10-7
 - in JDeveloper, 8-3
 - limitations, 9-1
 - log files, A-16
 - using to upgrade SOA applications, 9-1
- minBPELWait, 9-20
- MLLP channels configured for inbound and outbound messages
 - in Oracle B2B, 6-15
- My Oracle Support, 2-1

N

- non-managed-connection, 9-28

O

- OC4J
 - applying configuration settings to 11g, 8-2
- OC4J security environments
 - Anonymous Authentication with Message Confidentiality (WS-Security 1.0), 9-43
 - Anonymous Authentication with Message Integrity (WS-Security 1.0), 9-42
 - Anonymous Authentication with Message Protection (WS-Security 1.0), 9-41
 - before you begin, 9-41
 - ID Propagation Using SAML Token (Holder of Key) with Message Protection (WS-Security 1.0), 9-47
 - ID Propagation Using SAML Token (Sender Vouches) with Message Protection (WS-Security 1.0), 9-46
 - ID Propagation with SAML Token (Sender Vouches) over SSL (WS-Security 1.0), 9-51
 - Mutual Authentication with Message Protection (WS-Security 1.0), 9-49
 - upgrading, 9-40
 - Username token over SSL, 9-50
 - Username Token with Message Protection (WS-Security 1.0), 9-44
- ODI Knowledge Modules, 9-27
- Oracle ADF
 - configuring a WebLogic Server domain for, 5-1
 - deciding upon a topology, 5-1
 - information for ADF 10g users, 3-1
 - runtime libraries for, 3-1
 - sample application upgrade procedure, 8-1
 - support in 10g compared to 11g, 3-2
 - upgrading an existing domain, 5-3
- Oracle Application Development Framework
 - See also* Oracle ADF, 3-1
- Oracle Application Server
 - summary of upgrade starting points, 2-1

- Oracle Application Server Integration B2B Callouts
 - Java API, 9-3
- Oracle Application Server Integration B2B Instance
 - Message Java API, 9-3
- Oracle B2B
 - exporting trading partner agreements
 - before upgrade, 6-8
 - inbound and outbound partner agreements, 6-14
 - MLLP channels configured for inbound and
 - outbound messages, 6-15
 - packaging of in 10g and 11g, 3-4
 - post-upgrade tasks
 - callout library jar files, 6-13
 - channel passwords, 6-14
 - document protocol revision and document
 - type, 6-13
 - exchange protocol parameter identifiers, 6-13
 - exchange protocol parameters for the MLLP
 - channel, 6-14
 - Host and Partner Security Certificates, 6-13
 - receipt recipient property of the AS2-1.1
 - channel, 6-14
 - RosettaNet outbound messages, 6-15
 - schema
 - upgrading, 6-7
 - schema upgrade
 - general information about, 6-7
 - patch set required for, 6-7
 - performing, 6-8
 - prerequisites, 6-7
 - upgrading RosettaNet documents, 6-14
 - upgrading the schema, 6-5
- Oracle BAM, 3-2
 - about upgrading from previous versions, 6-10
 - invoking Web services from alerts, 9-26
 - migrating users and LDAP information to
 - 11g, 9-23
 - missing images in reports after upgrade, 9-25
 - new support for Web services, 9-26
 - packaging of in 10g and 11g, 3-4
 - schema
 - upgrading, 6-5
 - schema upgrade, 6-10
 - patch set required for, 6-10
 - performing, 6-10
 - understanding, 6-10
 - upgrade tasks associated with, 9-23
 - upgrading Enterprise Link plans, 9-26
 - upgrading external data sources, 9-24
 - upgrading sensor actions, 9-26
 - using the ManualRuleFire Web service, 9-26
 - using Web services with, 9-25
 - verifying action form templates after
 - upgrade, 9-24
- Oracle BPEL Process Manager
 - additional upgrade considerations, 9-15
 - combining multiple projects into a single
 - composite, 9-10
 - correcting binding errors after upgrade, 9-13
 - correlation sets in an asynchronous service, 9-16
 - domain descriptor properties, 9-17
 - long running processes
 - See also long-running processes, 4-1
 - manual upgrade steps, 9-13
 - MBean properties in 11g, 9-17
 - providing missing binding information for
 - external references, 9-13
 - specifying domain descriptor properties in, 9-17
 - upgrade tasks associated with, 9-13
 - upgrading fault policies in, 9-14
 - upgrading test suites, 9-14
 - upgrading with no BPEL folder, 9-15
 - user-defined XPath functions, 9-16
 - using deployment plans after upgrade, 9-14
- Oracle BPEL Process Manager Client Java API, 9-3
- Oracle BPEL Process Manager Sensor API, 9-3
- Oracle BPEL Process Manager Workflow Services
 - API, 9-3
- Oracle Business Rules
 - RuleRepository, 9-4
- Oracle Business Activity Monitoring
 - See Oracle BAM
- Oracle Business Rules
 - Decision Point API, 9-4
 - development environment
 - accessing a dictionary in 10g, 9-4
 - accessing a dictionary in 11g, 9-4
 - generating RL code, 9-6
 - production environment
 - accessing a dictionary in 11g, 9-5
 - accessing a repository, 9-5
 - accessing a repository in 10g, 9-5
 - SDK
 - changes for 11g, 9-3
 - upgrade tasks associated with, 9-32
 - upgrading dictionaries and projects, C-1
 - upgrading the Java API for, 9-3
- Oracle Business Rules Java API, 9-3
- Oracle Content DB
 - migrating data to Oracle Content Server, 10-6
 - upgrading applications in JDeveloper
 - 10.1.3.4, 10-6
- Oracle Content Server, client API, 10-12
- Oracle database
 - verifying supported version for 11g, 6-2
 - verifying version requirements, 7-1
- Oracle Enterprise Manager Fusion Middleware
 - Control, 5-2
 - applying the JRF template with, 5-3
 - configuring Oracle Identity Management for use
 - with Oracle SOA Suite, 6-12
 - using to test a SOA application, A-30
- Oracle Enterprise Service Bus
 - domain value maps, 9-21
 - projects with multiple routing services with same
 - namespace, 9-22
 - upgrade tasks associated with, 9-21
 - upgrading filtering or setting message
 - headers, 9-22
 - upgrading routing rules not exposed as

- services, 9-21
- upgrading routing services to asynchronous routing, 9-22
- upgrading SOAP headers, 9-21
- upgrading with command-line tool, 9-11
- Oracle Fusion Middleware
 - installing in preparation for an ADF upgrade, 5-1
- Oracle Fusion Middleware Callout Java API Reference, 9-3
- Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite, 9-14
- Oracle Fusion Middleware Infrastructure Management Java API for Oracle SOA Suite upgrading to, 9-6
- Oracle Fusion Middleware Infrastructure Management Java API Reference for Oracle SOA Suite, 9-3
- Oracle Fusion Middleware Installation Guide for Application Developer, 5-2
- Oracle Fusion Middleware Installation Guide for Oracle JDeveloper, 5-1
- Oracle Fusion Middleware Instance Data Access Java API Reference, 9-3
- Oracle Fusion Middleware Java API Reference for Oracle Business Rules, 9-3
- Oracle Fusion Middleware Upgrade Assistant
 - Specify Operation screen, 6-6, 6-11
 - Specify Schema Name screen, 6-9, 6-11
 - Specify Source Database screen, 6-11
 - Specify Target Database screen, 6-9, 6-11
 - starting, 6-5
 - using to upgrade the Oracle B2B schema, 6-5
 - using to upgrade the Oracle BAM schema, 6-5
 - Welcome screen, 6-5
- Oracle Fusion Middleware Upgrade Guide for Java EE, 3-1, 5-2, 6-12, 7-4, 9-2
- Oracle Fusion Middleware Upgrade Planning Guide, 3-4, 7-1
- Oracle Fusion Middleware Workflow Services Java API Reference for Oracle BPEL Process Manager, 9-3
- Oracle home, 3-5
 - comparison between 10g and 11g, 3-3
 - for Oracle SOA Suite, 3-3
 - for Oracle WebCenter, 3-5
- Oracle HTTP Server, 3-2
- Oracle Human Workflow
 - case sensitivity after upgrade, 6-12
 - upgrade tasks associated with, 9-31
 - upgrading task details, 9-31
- Oracle Identity Management
 - configuring for use with Oracle SOA Suite 11g, 6-12
- Oracle JDeveloper, 3-2, 3-4, 5-2
 - Check for Updates command, 9-2, A-12
 - composite editor, A-20
 - installing, A-11
 - installing an ADF environment with, 5-1
 - installing and starting, 8-3
 - log files during application upgrade, 8-4
 - migrating settings, 10-3
- Migration Wizard, 8-3, 9-1
 - limitations, 9-1
- opening and upgrading applications in, 8-3
- required JDeveloper extensions, 9-2, A-11
- setting up connections in, A-11
- starting, A-11
- using with Oracle SOA Suite, 9-2
- using with the Oracle SOA Suite Command-Line Upgrade Tool, 9-8
- Oracle JDeveloper Studio, 5-1
- Oracle Portal connections, 10-11
- Oracle Service Bus (OSB), 9-32
- Oracle SOA Suite
 - APIs
 - changes for 11g, 9-3
 - clients using remote APIs, 9-32
 - Command-Line Upgrade Tool, 9-8
 - benefits of using, 9-8
 - upgrading BPEL projects, 9-9
 - upgrading ESB projects, 9-11
 - using with JDeveloper, 9-8
 - comparing installation on OC4J and WebLogic Server, 3-3
 - configuring to use Identity Management components, 6-12
 - deciding upon a topology, 6-1
 - directory structure in OC4J versus WebLogic Server, 3-3
 - information for 10g users, 3-2
 - installation on OC4J versus WebLogic Server, 3-2
 - installing middle tiers, 6-4
 - installing required schemas, 6-3
 - main steps required to install and configure, 6-4, 7-3
 - references to HTTP URLs, 9-32
 - sample application upgrade, 8-2
 - schemas required, 3-2
 - selecting schemas for, 6-3
 - upgrade tasks associated with all, 9-2
 - upgrading a SOA Suite enterprise deployment, 3-3
 - upgrading applications with external references, 6-2
 - upgrading client applications, 9-32
 - upgrading SOA applications, 9-1
 - using with JDeveloper, 9-2
- Oracle SOA Suite components
 - documentation resources when upgrading, 8-4
- Oracle SOA Suite, WebCenter, and ADF
 - overview of upgrading, 8-1
 - starting points for upgrade, 2-1
 - upgrade process summary, 1-1
- Oracle Technology Network (OTN), 8-1, 8-4, 9-25, 9-33, A-1
- Oracle Web Services Manager
 - before you upgrade, 9-36
 - gateways, 9-36
 - upgrading custom policies, 9-40
 - upgrading policies, 9-36

- upgrading predefined policies, 9-37
- Oracle WebCenter
 - comparison on OC4J versus WebLogic Server, 3-4
 - deciding upon a topology, 7-1
 - directory structure, 3-5
 - information for 10g users, 3-4
 - installing middle tiers, 7-3
 - schema category, 7-2
 - schema upgrade, 7-2
 - selecting schemas to install, 7-2
 - upgrading the environment, 7-1
 - using RCU to install required schemas, 7-2
 - verifying database requirements, 7-1
- Oracle WebCenter components
 - documentation resources when upgrading, 8-4
- Oracle WebCenter Services, 10-2
- Oracle WebCenter Wiki and Blog Server
 - about migrating wiki data, 10-19
 - administration mode, 10-20
 - importing domain, 10-21
 - migrating content to a database repository, 10-20
 - migrating content to a file system-based repository, 10-20
 - pages directory, 10-20
- Oracle WebLogic Server
 - Administration Console, 6-7, A-6
 - configuration wizard, 5-3
 - configuring a domain for Oracle ADF, 3-1, 5-1
 - configuring a domain for Oracle SOA Suite, 3-2
 - configuring a domain without JDeveloper, 5-2
 - upgrading existing instances to use the ADF Runtime, 5-3
- Oracle *MetaLink*, 2-1
- oracle.rules.sdk2 package, 9-4
- orauddi protocol
 - changing to after upgrade, 9-7

P

- Panel Customizable changes, B-2
- patch sets
 - applying latest before upgrade, 9-2
 - applying to 10g releases before upgrade, 2-1
 - Oracle JDeveloper 10.1.3.5, 9-25
 - required for clients using WS-addressing, 9-32
 - required for Oracle B2B schema upgrade, 6-7
 - required for Oracle BAM schema upgrade, 6-10
- PDK-Java portlet application, 10-16
- persistence.xml, A-27
- portlet producers
 - creating portlet producer application, 10-15
 - determining application compatibility, 10-14
 - migrating a WAR file, 10-17
 - migrating an EAR file, 10-16
 - migrating customizations
 - non-preconfigured portlet producers, 10-19
 - preconfigured portlet producers, 10-18
 - migrating producers created in JDeveloper, 10-15
 - migrating producers created outside of

- JDeveloper, 10-16
 - redeploying, 10-19
- portlet tag changes, B-6
- Portlets project
 - migrating, 10-15
 - removing, 10-7
- portlets.jpr, 10-16
- post-upgrade tasks
 - for Oracle B2B, 6-13
 - for SOA schemas, 6-12
- preconfigured portlet producers
 - about, 10-18
 - customization store
 - OmniPortlet and other PDK-Java producers, B-10
 - Web Clipping, B-10
 - WSRP portlet producers, B-10
 - customizations
 - migrating from default location, 10-18
 - migrating from non-default location, 10-18
 - default customization store, B-10
 - port registrations, 10-13
- preparing for migration
 - portlet producer applications, 10-14
- processCheckSecs, 9-20
- proxy settings for Web services
 - upgrading applications that require, 9-6

R

- RCU
 - See also* Repository Creation Utility, 3-2
- receipt recipient property of the AS2-1.1 channel
 - upgrading to 11g, 6-14
- Red Hat JBoss, 9-37
- Repository Creation Utility, 3-2, 3-4, 6-2, 7-2
 - starting, 6-3
 - using to install WebCenter schemas, 7-2
- RosettaNet documents
 - in Oracle B2B, 6-14
- RosettaNet outbound messages
 - in Oracle B2B, 6-15
- rules
 - migrating, C-1
- rulesdk2.jar, 9-4

S

- sample applications
 - demonstrating end-to-end upgrade, 8-1
- samples_102_responseForward sample application, 9-22
- sca_XrefDvmFiles10g_rev1.0.jar, 9-12
- sca-upgrade.xml, 9-6
- schema
 - Oracle B2B
 - upgrading, 6-7
 - upgrading the Oracle B2B, 6-5
- schemas
 - installing for Oracle SOA Suite, 6-2

- required by Oracle SOA Suite, 3-2
- required by Oracle WebCenter, 3-4
- selecting for SOA Suite, 6-3
- serviceKey Endpoints
 - verifying before upgrade, 9-7
- Show Detail Frame changes, B-4
- SOA applications
 - considerations when upgrading, 9-1
- SOA Composite Editor
 - JDeveloper extension, 9-2, A-12
- SOA Infrastructure
 - category of schemas in RCU, 6-3
- SOA Order Booking
 - about the sample application, A-1
 - clearing the WebApp 2.5 option in Migration Wizard, A-16
 - compiling and deploying projects in, A-17
 - configuring JMS resources for, A-8
 - configuring required WebLogic resources, A-6
 - creating the soademo data source in WebLogic Server, A-6
 - end-to-end application upgrade example, A-1
 - opening and upgrading in JDeveloper, A-14
 - sample application, A-1
 - sample application upgrade, 8-2
 - setting up connections for, A-11
 - testing in Fusion Middleware Control, A-30
 - verifying external references before upgrade, A-4
 - verifying on 10g Release 3 (10.1.3.4), A-2
- soademo.jws, A-14
- soa-infra-config.xml, 9-17
- soaversion.bat, 9-10
- soaversion.sh, 9-10
- Specify Operation screen
 - B2B Schema option, 6-9
 - in the Upgrade Assistant, 6-6, 6-11
- Specify Schema Name screen
 - in the Upgrade Assistant, 6-9, 6-11
- Specify Source Database screen
 - in the Upgrade Assistant, 6-11
- Specify Target Database screen
 - in the Upgrade Assistant, 6-9, 6-11
- SRDemo
 - sample upgrade procedure, 8-1
- starting points
 - for upgrade, 2-1
- statsLastN, 9-20

T

- targetNamespace, 9-22
- technology adapters
 - AQ outbound request-reply scenario, 9-30
 - missing descriptions after upgrade, 9-28
 - upgrade tasks associated with, 9-27
 - upgrading an inbound synchronous request-reply, 9-28
 - upgrading AQ adapter headers, 9-29
 - upgrading database adapters used for stored procedures, 9-30

- upgrading header variables, 9-27
- upgrading non-managed connection information, 9-28
- upgrading projects that use outbound dequeue, 9-29
- verifying with adapter configuration wizard, 9-27
- topology
 - deciding upon an Oracle ADF, 5-1
 - deciding upon an Oracle SOA Suite, 6-1
 - deciding upon an Oracle WebCenter, 7-1
- trading partner agreements
 - exporting before B2B upgrade, 6-8
- transaction=participate property, 9-16
- Troubleshooting, 6-9, 6-11

U

- ua
 - command to start Upgrade Assistant, 6-5
- UDDI registry, 9-8
- UDDI-registered resources
 - upgrading applications that use, 9-7
- upgrade
 - starting points for, 2-1
- Upgrade Assistant
 - See* Oracle Fusion Middleware Upgrade Assistant
- upgrade process
 - recommended tasks, 8-1
 - summary of, 1-1
- Upgrade Web Service to WLS JAX-RPC Configuration
 - JDeveloper command, A-21
- Use JAAS for security checkbox, B-7

W

- Web services
 - upgrading applications that require proxy settings for, 9-6
 - using the Upgrade Web Service to WLS JAX-RPC Configuration command, A-21
- Web Services Addressing, 9-32
- Web services URIs
 - verifying before upgrade, 6-2
- WebApp 2.5, A-16
- WebCenter
 - See* Oracle WebCenter
- WebCenter applications
 - ADF security, 10-11
 - application roles in Oracle WebCenter 10.1.3.x, 10-2
 - automated migration changes
 - customizable components, B-1
 - external applications, B-5
 - portlet component changes, B-6
 - migrating consumer applications, 10-4
 - migrating portlet producers, 10-13
 - migration prerequisite tasks, 10-2
 - migration tasks, 10-2
 - Oracle Portal connections, 10-11

- preparing for migration
 - consumer applications, 10-5
 - redeploying, 10-13
 - templates, 10-1
- WebCenter consumer application, 10-1
- WebCenter Web 2.0 Services, 10-2
- WebDav
 - limitations, 10-6
 - migrating Content DB data, 10-6
- WebTier and Utilities CD-ROM, 3-2
- Welcome screen
 - of the Upgrade Assistant, 6-5
- wiki data
 - about, 10-19
 - migrating to a database repository, 10-20
 - migrating to a file system-based repository, 10-20
- workflow-identity-config.xml
 - Oracle Human Workflow configuration file, 6-12
- workspace (.jws) file, 8-3
 - opening in JDeveloper, 8-3
- WS-Addressing, 9-32

X

- XPath functions
 - upgrading user-defined, 9-16
- XrefDvmFiles10g, 9-12