

Oracle® Fusion Middleware
WebCenter Sites Administrator's Guide
11g Release 1 (11.1.1.8.0)
E29636-02

December 2016

Oracle Fusion Middleware WebCenter Sites Administrator's Guide, 11g Release 1 (11.1.1.8.0)

E29636-02

Copyright © 2013, 2016 Oracle and/or its affiliates. All rights reserved.

Primary Author: Debosri Brahma

Contributor: Subha Gottumukkala

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

Contents

Preface	xxv
Audience	xxv
Documentation Accessibility	xxv
Related Documents	xxv
Conventions	xxv
What's New in This Guide	xxvii
New and Changed Features in 11g Release 1 (11.1.1.8.0)	xxvii
Deprecation Notice	xxviii

Part I Administering WebCenter Sites

1 Overview

1.1	Introduction	1-1
1.2	Online Sites	1-2
1.3	CM Sites	1-5
1.4	Configuration Components	1-6
1.4.1	CM Site Definition	1-7
1.4.2	Data Model	1-7
1.4.3	Users and Their Table-Level Permissions	1-8
1.4.4	Roles	1-9
1.4.5	Start Menus	1-11
1.4.6	Workflow Processes	1-11
1.4.7	Publishing System	1-12
1.4.8	User Interface Options	1-12
1.5	Summary	1-12
1.6	Content Management Models	1-13
1.6.1	1:1 Model	1-13
1.6.2	1:n Model	1-13
1.6.3	x:n Model	1-14
1.7	Oracle WebCenter Sites Environment	1-15
1.8	Administrator's Job	1-16
1.8.1	Collaborating	1-17
1.8.2	Planning	1-17
1.8.3	Development	1-19

1.8.3.1	At the Development System	1-19
1.8.3.2	At the Management System	1-20
1.8.3.3	At the Delivery System	1-21
1.8.3.4	At the Testing System	1-21
1.8.4	Implementation	1-21
1.9	The Focus of This Guide	1-21
1.10	Administrator's Prerequisites	1-21

2 Administrator's Interface

2.1	Types of WebCenter Sites Administrators	2-1
2.2	Logging in to the Administrator's Interface	2-2
2.3	Administrator Tabs	2-6
2.3.1	Admin Tab	2-7
2.3.2	Site Admin Tab	2-9
2.3.3	Workflow Tab	2-10
2.4	Non-Administrative Tabs	2-11
2.5	System Defaults	2-11
2.6	Sample Sites	2-11

3 Site Configuration Guidelines

3.1	Overview	3-1
3.2	Site Replication Option	3-3
3.3	Site Configuration Steps	3-4
3.3.1	Assumptions	3-4
3.3.2	Pre-Configuration Decisions	3-4
3.3.3	Configuration Steps at the Development System	3-5
3.3.4	Configuration Steps at the Management System	3-6

4 Working with ACLs and Roles

4.1	Overview	4-1
4.2	ACLs	4-2
4.2.1	System ACLs	4-3
4.2.2	Sample ACLs	4-4
4.2.3	Custom ACLs	4-4
4.3	Working with ACLs	4-4
4.3.1	Creating a New ACL	4-5
4.3.2	Editing a Custom ACL	4-6
4.3.3	Deleting a Custom ACL	4-7
4.3.4	Assigning ACLs to Custom Tables	4-8
4.3.5	Assigning ACLs to WebCenter Sites Pages (SiteCatalog Page Entries)	4-9
4.3.6	Setting the ACL Restriction Error Message	4-11
4.4	Roles	4-12
4.4.1	System Roles	4-12
4.4.2	Sample Roles	4-12
4.4.3	Custom Roles	4-12
4.5	Working with Roles	4-13

4.5.1	Creating a Role	4-13
4.5.2	Editing a Role	4-13
4.5.3	Deleting a Role	4-14

5 Configuring Users, Profiles, and Attributes

5.1	Overview	5-1
5.2	User Management Options	5-2
5.2.1	Native WebCenter Sites User Manager	5-2
5.2.2	LDAP Plug-In	5-2
5.3	Configuring Users in WebCenter Sites	5-2
5.4	Working with User Accounts	5-3
5.4.1	Creating a New User	5-3
5.4.2	Creating Common User Types	5-6
5.4.2.1	Making the User a General Administrator	5-6
5.4.2.2	Making the User a Site Administrator	5-7
5.4.2.3	Making the User a Workflow Administrator	5-8
5.4.2.4	Making the User a Content Contributor	5-8
5.4.3	Editing a User	5-9
5.4.4	Deleting a User from the System	5-9
5.5	Working with User Profiles and User Attributes	5-10
5.5.1	Creating and Editing a User Profile	5-11
5.5.2	Deleting a User Profile	5-12
5.5.3	Modifying, Adding, and Deleting User Attributes	5-12

6 Setting Up External Security

6.1	Overview	6-1
6.1.1	ACLs and Security	6-2
6.1.2	DefaultReader, secure.CatalogManager, and secure.TreeManager	6-3
6.1.3	BlobServer and Security	6-3
6.1.4	Security Goals	6-3
6.1.4.1	Security Goals for the Development System	6-4
6.1.4.2	Security Goals for the Management System	6-4
6.1.4.3	Security Goals for the Delivery System	6-4
6.2	Implementing Security	6-4
6.2.1	Properties That Configure Security Settings	6-5
6.2.2	Users and Passwords	6-5
6.2.3	SSL and Digital Certification	6-6
6.2.4	URLs and the Web Server (Delivery System Only)	6-6
6.2.5	WebCenter Sites Forms and Pages (Delivery System Only)	6-7
6.3	Testing Security	6-7

7 Implementing Security Using Cryptography and Encryption Key Storage

7.1	WebCenter Sites Security Overview	7-1
7.2	Considerations for External Key Stores	7-2
7.3	The SitesSecurityContext.xml Configuration File	7-3
7.4	Mixed Credential Stores	7-5

7.5	Sites Security Utilities	7-5
7.6	Using Java JCEKS Key Stores	7-7
7.7	ESAPI Security Validation	7-8

8 Assembling Content Management Sites

8.1	Working with Sites	8-1
8.1.1	Creating a Site	8-1
8.1.2	Obtaining Site Configuration Information	8-4
8.1.3	Editing a Site	8-4
8.1.4	Deleting a Site	8-4
8.2	Assigning and Managing Site Users	8-5
8.2.1	Granting Users Access to a Site (Assigning Roles to Users)	8-6
8.2.2	Viewing Site Users and Reassigning Roles	8-8
8.2.3	Deleting Users from a Site	8-9
8.3	Enabling and Managing Asset Types	8-9
8.3.1	Enabling Asset Types for a Site	8-9
8.3.2	Removing Asset Types from a Site	8-11
8.4	Enabling and Managing User Interfaces	8-12

9 Managing Access to Content Management Site Components

9.1	Overview	9-1
9.2	Managing Access to Asset Types via Start Menus	9-1
9.2.1	Start Menu Items	9-2
9.2.2	Creating Start Menu Items	9-4
9.2.2.1	Creating New Start Menu Items	9-5
9.2.2.2	Creating Search Start Menu Items	9-10
9.2.2.3	Editing Start Menu Items	9-13
9.2.2.4	Creating Start Menu Items for Sites Desktop	9-14
9.2.2.5	Creating Start Menu Items for Sites DocLink	9-15
9.2.2.6	Displaying a List of All Start Menu Items	9-15
9.3	Setting Access Permissions to Assets	9-15
9.3.1	Permission Structure	9-16
9.3.2	Types of Authorization	9-17
9.3.3	Summary	9-17
9.3.4	Setting Access Permissions	9-18
9.3.4.1	Asset Toolbar	9-18
9.3.4.2	Setting Access Permissions from the Administrator's Interface	9-19
9.3.4.3	Setting Access Permissions from Property Files	9-21
9.4	Configuring Other Options for Asset Types	9-22
9.5	Managing Access to the Tree (Administrator's interface only)	9-22
9.5.1	Displaying and Hiding the Tree	9-23
9.5.2	Creating Tree Tabs	9-24
9.5.2.1	Creating a New Tree Tab	9-24
9.5.2.2	Creating the 'Workflow Groups' Tab	9-27
9.5.3	Editing Tree Tabs	9-27
9.5.4	Deleting Tree Tabs	9-28
9.5.5	Changing Tree Tab Order	9-28

10 Creating and Managing Workflow Processes

10.1	Overview	10-1
10.1.1	Workflow Participants	10-2
10.1.2	Workflow States	10-2
10.1.3	Workflow Steps: Moving Assets from State to State	10-2
10.1.3.1	How Steps Work	10-3
10.1.3.2	Step Actions and Conditions	10-3
10.1.4	Multiple Paths for the Asset	10-4
10.1.5	Managing Deadlocks	10-4
10.1.5.1	Resolving Deadlocks	10-5
10.1.5.2	Preventing Deadlocks	10-5
10.1.5.3	Notifying Participants When Deadlocks Occur	10-5
10.1.6	Workflow Groups	10-6
10.1.6.1	Using Workflow Groups	10-6
10.1.6.2	Adding a Synchronize Step	10-6
10.1.6.3	Managing Group Deadlocks	10-6
10.1.7	Delegating and Clearing Assignments	10-6
10.1.8	Placing an Asset in Workflow	10-7
10.1.9	Restricting Access to Assets While They Are in Workflow	10-7
10.1.9.1	Function Privileges and Step Actions	10-8
10.1.9.2	Function Privileges: All or None	10-8
10.1.10	Deadlines	10-8
10.1.10.1	Setting and Overriding State Deadlines	10-8
10.1.10.2	Setting Process Deadlines	10-9
10.1.11	Scheduling a Deadline Calculation	10-9
10.1.12	How Does a Workflow Process End?	10-9
10.1.13	Roles Required to Configure Workflow Processes	10-10
10.2	Planning Your Workflow Processes	10-10
10.2.1	Start with a Sketch	10-10
10.2.2	Determine Roles and Participants	10-11
10.2.2.1	Planning Roles	10-11
10.2.2.2	Selecting Individual Participants	10-11
10.2.2.3	Cross-Site Assignments and Participants	10-12
10.2.3	Determine the Email Objects, Actions, and Conditions	10-12
10.2.3.1	About Email Objects	10-13
10.2.3.2	About Timed Actions	10-15
10.2.3.3	About Delegate Actions	10-16
10.2.3.4	About Step Actions	10-16
10.2.3.5	About Step Conditions	10-17
10.2.3.6	About Deadlock Actions	10-17
10.2.3.7	About Group Deadlock Actions	10-18
10.2.4	Determine the States	10-18
10.2.5	Determine the Steps	10-19
10.2.6	Determine the Function Privileges	10-21
10.2.6.1	Planning Function Privileges	10-24

10.2.6.2	Implementing Simplified Access Control	10-24
10.2.7	Determine Additional Workflow Process Details	10-25
10.3	Configuring Your Workflow Processes	10-26
10.3.1	Overview	10-26
10.3.2	Setting Up Email Objects	10-27
10.3.2.1	Enabling the Email Feature	10-27
10.3.2.2	Creating Email Objects	10-27
10.3.2.3	Editing Email Objects	10-28
10.3.2.4	Deleting Email Objects	10-29
10.3.3	Setting Up the Actions and Conditions	10-29
10.3.3.1	Creating Actions and Conditions	10-29
10.3.3.2	Configuring Approve for Publish Step Actions	10-30
10.3.3.3	Editing Actions and Conditions	10-31
10.3.3.4	Deleting Actions and Conditions	10-31
10.3.4	Setting Up the Timed Action Event	10-32
10.3.5	Setting Up the States	10-33
10.3.5.1	Creating Workflow States	10-33
10.3.5.2	Editing Workflow States	10-35
10.3.5.3	Deleting Workflow States	10-36
10.3.6	Setting Up the Workflow Processes	10-36
10.3.6.1	Creating Workflow Processes	10-36
10.3.6.2	Editing Workflow Processes	10-45
10.3.6.3	Editing Steps	10-45
10.3.6.4	Editing Function Privileges	10-45
10.3.6.5	Copying Workflow Processes	10-46
10.3.6.6	Deleting Workflow Processes	10-46
10.3.6.7	Deleting Steps	10-47
10.3.6.8	Deleting Function Privileges	10-47
10.3.7	Testing Your Workflow Process	10-47
10.4	Moving Your Work	10-48
10.5	Clearing Workflow Assignments	10-48

11 Replicating Content Management Sites

11.1	Site Launcher Overview	11-1
11.2	Preparing for Replication	11-2
11.2.1	Ensuring the Source Site Meets Replication Requirements	11-2
11.2.2	Planning the New Site	11-2
11.2.2.1	Copying vs. Sharing	11-2
11.2.2.2	Naming Assets	11-4
11.2.2.3	Planning Users	11-4
11.2.2.4	Choosing the Replication Time	11-4
11.3	Site Replication Steps	11-5
11.3.1	Ensure that Replication Requirements Are Satisfied	11-5
11.3.2	Enable the Source Site for Site Launcher	11-5
11.3.3	Replicate the Source Site	11-6
11.4	Post-Replication Tasks and Guidelines	11-9

12 Publishing with Oracle WebCenter Sites

12.1	Overview	12-1
12.2	Publishing Methods	12-2
12.3	Publishing Destinations	12-4
12.4	The Approval System	12-5
12.5	The Publishing Schedule	12-6
12.6	What Happens During a Publishing Session?	12-6
12.7	Obtaining Information About a Publishing Session	12-7

13 The Approval System

13.1	Overview	13-2
13.2	A Brief Look at Dependency Analysis	13-3
13.3	Terms and Definitions	13-6
13.3.1	Approval: Intent to Publish vs. Permission to Publish	13-6
13.3.2	Approval Dependencies	13-7
13.3.2.1	Examples of Approval Dependencies	13-10
13.3.2.2	Summary	13-12
13.3.3	Approval States	13-12
13.4	Rules for Dependency Analysis	13-17
13.5	Approval Dependencies and Parent-Child Relationships	13-18
13.6	Export to Disk Publishing	13-19
13.6.1	Evaluating Approval Dependencies	13-19
13.6.2	Test Rendering	13-20
13.7	Mirror to Server Publishing and Export to XML	13-21
13.8	Ensuring Version-Matched Content	13-21
13.9	Exists-Exact Dependencies in Mirror to Server and Export to XML Publishing	13-22
13.10	Exists-Exact Dependencies in Export to Disk Publishing	13-23
13.11	Exists/Exact Dependency Rules	13-23
13.11.1	Basic Asset Types	13-23
13.11.2	CSElement and SiteEntry Assets	13-24
13.11.3	Flex Families	13-24
13.11.4	Oracle WebCenter Sites: Engage Visitor Data Assets	13-25
13.12	Evaluating Published Content	13-26
13.13	Putting It All Together	13-26
13.14	Reference: Approval States	13-30

14 Various Topics in Export to Disk Publishing and the Approval Process

14.1	Export to Disk Publishing Terminology	14-1
14.1.1	Page	14-2
14.1.2	Publish Key	14-2
14.1.3	Publish Queue	14-2
14.1.4	Primary Asset	14-2
14.2	Approvals and Export to Disk Publishing	14-3
14.2.1	Approval Template	14-3
14.2.2	Approval Queue	14-4
14.2.3	Approval Dependencies	14-4

14.3	Export to Disk Publishing	14-5
14.3.1	Publish-Time Template	14-5
14.3.2	Starting Points	14-5
14.3.3	Compositional Dependencies	14-5
14.4	Frequently Asked Questions	14-6
14.5	Behavior and Functionality	14-6
14.5.1	Example 1: Template Dependencies	14-7
14.5.2	Example 2: Reference Dependencies	14-7
14.6	Approval and Publish-Time Templates: What Happens When They Differ?	14-8
14.6.1	Example 1	14-8
14.6.2	Example 2	14-9
14.6.3	Example 3	14-9
14.7	Sample Templates	14-10

15 The Export to Disk Publishing Process

15.1	Rendering Export to Disk Pages	15-1
15.2	How Export to Disk Publishing Works	15-2
15.3	Path Naming Convention	15-3
15.3.1	Export Path Variables and SIMPLEDIR	15-4
15.3.2	Export Path Construction	15-5
15.3.2.1	Forced Export Path	15-6
15.3.2.2	Export Path for Page Assets	15-6
15.3.2.3	Export Path for Assets Other than Page	15-6
15.3.2.4	Sample Export Paths	15-7
15.3.3	Paths for Links Within Exported Files	15-7
15.3.4	File Naming Conventions	15-8
15.3.5	Export Starting Points	15-10
15.4	Configuring Your System for Export to Disk Publishing	15-10
15.4.1	Create the Batch User Account (If One Does Not Exist)	15-11
15.4.2	Specify the Root Export Directory	15-12
15.4.3	Configure an Export Destination	15-13
15.4.4	Map a URL Prefix for Your Web Server	15-18
15.4.5	Create the Export Starting Points	15-18
15.4.6	Approve Your Assets	15-21
15.4.7	Publish and Test the Results	15-21
15.4.8	Set Up the Schedule	15-22

16 The Mirror to Server Publishing Process

16.1	How Mirror to Server Publishing Works	16-1
16.2	Before Configuring Mirror to Server Publishing	16-4
16.2.1	Users and Mirror to Server Publishing	16-4
16.2.2	CacheManager	16-4
16.2.3	Configuring a Mirror Destination	16-4
16.2.4	When to Use Mirror Destination Configuration	16-5
16.3	Configuring Your System for Mirror to Server Publishing	16-5
16.3.1	Create the Batch User Account (If One Does Not Exist)	16-6
16.3.2	Set Up the Destination System	16-6

16.3.3	Identify the Mirror User to the Source System	16-7
16.3.4	Identify the Local Proxy Server to the Source System (If One Exists)	16-8
16.3.5	Create a Mirror Destination	16-8
16.3.6	Initialize the Destination	16-12
16.3.7	Configure the Mirror Destination	16-13
16.3.8	Approve Your Assets	16-15
16.3.9	Publish the Assets	16-16
16.3.10	Test the Results	16-16
16.3.11	Set Up the Schedule	16-17
16.3.12	Turn Off Asset Invalidation on the Delivery System	16-17
16.4	Approval Query for Performance Enhancement	16-18
16.5	Retrieving Logs From Delivery WebCenter Sites Systems	16-19

17 The Export Assets to XML Publishing Process

17.1	The Export Assets to XML Publishing Method	17-1
17.1.1	The XML Output	17-2
17.1.2	File Naming Conventions for Export Assets to XML	17-3
17.2	Configuring Your System for Export Assets to XML	17-3
17.2.1	Create the Batch User Account (If One Does Not Exist)	17-3
17.2.2	Specify the Root Export Directory	17-4
17.2.3	Configure an Export Assets to XML Destination	17-4
17.2.4	Approve Your Assets	17-6
17.2.5	Publish and Test the Results	17-6
17.2.6	Set Up the Schedule	17-6

18 Additional Publishing Procedures

18.1	Migrating a Site from One System to Another	18-1
18.1.1	Moving a Site from a Development System to a Management System	18-2
18.1.2	Moving a Site to a Delivery System	18-2
18.2	Approving Multiple Assets	18-3
18.3	Creating Destinations	18-5
18.4	Editing Destinations	18-5
18.5	Deleting Destinations	18-7
18.6	Creating Export Starting Points	18-7
18.7	Scheduling Publish Events	18-7
18.8	Reading the Schedule Abbreviations	18-10
18.9	Editing Publish Events	18-11
18.10	Overriding the Schedule	18-11
18.11	Assigning Approval or Preview Templates	18-12
18.12	Monitoring Sessions in the Publishing Console	18-13
18.13	Verifying Publishing Readiness	18-14
18.14	Managing Publishing History Information	18-16
18.15	Publishing All Approved Assets	18-18
18.16	Troubleshooting	18-19
18.16.1	About Publishing System Error Messages	18-19
18.16.2	Numeric Messages	18-19

18.16.2.1	-2	18-20
18.16.2.2	-3	18-20
18.16.2.3	-103	18-21
18.16.2.4	-611	18-21
18.16.2.5	-612	18-21
18.16.2.6	-12011	18-21
18.16.2.7	-12044	18-21
18.16.2.8	-12045	18-22
18.16.2.9	-12046	18-22
18.16.2.10	-12047	18-22
18.16.2.11	-13054	18-22
18.16.2.12	-13055	18-22
18.16.3	Other Indicators of System or Configuration Issues	18-23
18.16.3.1	Publishing Session Does Not End and Displays an Odd Status	18-23
18.16.3.2	The System Stops Altogether	18-24
18.16.3.3	Pages Are Not Refreshed After the Publishing Session Ends	18-24
18.16.3.4	DB2 Systems, Troubles When Publishing Assets with Associations	18-24

19 RealTime Publishing Overview

19.1	Overview	19-1
19.2	How RealTime Publishing Works	19-6

20 Configuring the RealTime Publishing Process

20.1	Overview	20-1
20.2	Configuring Your System for RealTime Publishing	20-2
20.2.1	Create the Batch User Account (If One Does Not Exist)	20-3
20.2.2	Set Up the Destination System	20-3
20.2.3	If a Proxy Server Exists, Identify it to the Source System	20-4
20.2.4	Create a RealTime Destination Definition on the Source System	20-5
20.2.5	Initialize the Destination Database	20-9
20.2.6	Mirror the Site Configuration Data to the Destination Database	20-11
20.2.7	Test the Publishing Process	20-13
20.2.7.1	Approve Assets	20-14
20.2.7.2	Publish the Assets	20-14
20.2.8	Test the Published Site	20-14
20.2.9	Set Up the Schedule	20-15
20.2.10	Turn Off Asset Invalidation on the Delivery System	20-16
20.3	Adding a New RealTime Destination Definition	20-16
20.4	Synchronizing Site Configuration Data	20-16
20.5	Performance Enhancement	20-17

21 Working with RealTime Publishing Modes

21.1	Complete and Delayed Publishing	21-1
21.2	Publish Console	21-2
21.3	Working with Active Publishing Sessions	21-2
21.3.1	Monitoring a Publishing Session	21-3

21.3.2	Resuming a Delayed Publishing Session	21-4
21.3.3	Canceling a Publishing Session	21-8
21.4	Working with Scheduled Publishing Sessions	21-10
21.5	Working with Prior Publishing Sessions	21-11
21.5.1	Viewing Past Publishing Sessions	21-12
21.5.2	Redoing a Publishing Session	21-13
21.6	Working with Publishing Logs	21-15
21.6.1	Viewing and Searching Logs	21-15
21.6.2	Downloading Logs	21-17
21.6.3	Deleting Publish History	21-19

22 On Demand Publishing and Asset Unapproval

22.1	Using On-Demand Publishing	22-1
22.2	Unapproving Selected Assets from a Publishing Session	22-4

23 Configuring the User Interfaces

23.1	Setting the Locale for the WebCenter Sites Interfaces	23-1
23.1.1	System Default Locale for the WebCenter Sites Users	23-2
23.1.2	System Locale Properties	23-3
23.1.3	Single-Language Restrictions	23-4
23.1.4	Locale and Asset Types	23-5
23.1.5	Locale and the Article Asset Type	23-5
23.2	Sites Desktop	23-6
23.2.1	Overview	23-6
23.2.1.1	Sources of Data for a Word Asset	23-7
23.2.1.2	Preserving the Formatting of the Word Data	23-7
23.2.2	Configuring the Word Asset Types for Sites Desktop	23-8
23.2.2.1	Enabling Asset Types for Sites Desktop	23-9
23.2.2.2	Specifying Which Fields Are on the WebCenter Sites Toolbar	23-10
23.2.2.3	Creating Sites Desktop Start Menu Items	23-12
23.2.3	User Accounts and Sites Desktop	23-12
23.2.4	Installing the Sites Desktop Client Application	23-13
23.2.5	Specifying Locale for the Client Application	23-14
23.2.6	Testing the Sites Desktop Configuration	23-14
23.2.7	Configuring Word Templates for the Word Assets	23-15
23.3	Sites DocLink	23-15
23.3.1	Overview	23-15
23.3.2	Configuring the Sites DocLink Asset Types	23-16
23.3.2.1	Enabling Asset Types for Sites DocLink	23-16
23.3.2.2	Specifying Which Fields Are Available in Sites DocLink	23-17
23.3.2.3	Creating Sites DocLink Start Menu Items	23-20
23.3.3	Users and Sites DocLink	23-21
23.3.4	Installing the Sites DocLink Client Application	23-21
23.3.5	Testing the Sites DocLink Configuration	23-22
23.4	Web Mode	23-22
23.5	Maintaining Separate Browser Sessions for Preview	23-23

24 Configuring Vanity URLs

24.1	WebRoots	24-2
24.2	Generating Vanity URLs	24-7
24.2.1	Configuring Auto-Generated URLs	24-7
24.2.2	Working with URL Patterns	24-11
24.2.2.1	Functions	24-12
24.2.2.2	Examples of Using Patterns	24-13
24.2.2.3	Deleting Patterns	24-14
24.2.3	Publishing URL Patterns	24-14
24.3	Using Vanity URL with System Tools	24-15
24.3.1	Viewing All Vanity URLs	24-15
24.3.2	Resolving Vanity URL Conflicts Using System Tools	24-15
24.4	Resolving Vanity URLs Using a Rewriter Filter	24-16
24.5	Vanity URL for Mobile Sites	24-17
24.6	Using the Web Server with Vanity URLs	24-17
24.7	Using mod_rewrite with Vanity URLs	24-18
24.7.1	Setting Up mod_rewrite	24-18
24.7.2	Converting with mod_rewrite	24-21
24.7.3	Using the Sites Filter with mod_rewrite	24-22
24.7.4	Using the Rewriter Filter with mod_rewrite	24-23
24.7.5	Using Static Content with mod_rewrite	24-23

25 Configuring the Lucene Search Engine

25.1	Overview	25-1
25.1.1	Indexing for Search Functions	25-1
25.1.2	WebCenter Sites Search Functions	25-2
25.2	Setting Up Search Indices	25-4
25.2.1	Enabling the Lucene Search Engine	25-5
25.2.2	Adding Asset Types to the Search Index	25-5
25.2.3	Configuring Attributes for Asset Type Index	25-7
25.2.4	Binary File Indexing	25-9
25.2.4.1	Enabling Indexing of Binary Files	25-9
25.2.4.2	Disabling Indexing of Binary Files	25-9
25.3	Disabling the Lucene Search Engine	25-10
25.4	Maintaining Search Indexes	25-10
25.4.1	Pausing and Resuming Indexing	25-11
25.4.1.1	Pausing Global and Asset Type Indexing	25-11
25.4.1.2	Resuming Global and Asset Type Indexing	25-13
25.4.2	Re-indexing	25-13
25.4.3	Deleting Index Data	25-14
25.5	Writing Code that Queries the Search Index	25-16

26 Revision Tracking

26.1	Overview	26-1
26.1.1	Tracker Tables and Storage Directories	26-2
26.1.2	The RTInfo Table	26-2

26.1.3	Revision Tracking and the Two Asset Models	26-3
26.1.4	Implicit vs. Explicit Checkin and Explicit Checkout	26-3
26.1.5	Revision Tracking and Non-Asset Tables	26-4
26.1.6	How Many Versions?	26-4
26.2	Enabling Revision Tracking	26-4
26.2.1	Enabling Revision Tracking for Assets	26-5
26.2.2	Enabling Revision Tracking for Non-Asset Tables	26-5
26.2.3	Editing Revision Tracking Settings	26-6
26.2.3.1	Changing the Root Storage Directory	26-6
26.2.3.2	Changing the Number of Revisions	26-6
26.3	Disabling Revision Tracking	26-7
26.4	Unlocking Revisions	26-8

27 inCache Framework

27.1	What is inCache?	27-1
27.2	How inCache Works	27-2
27.3	Restarting a Node	27-3
27.4	Page Regeneration During RealTime Publishing	27-4
27.5	Double-Buffered Caching	27-4
27.6	inCache Features for Remote Satellite Server	27-4
27.6.1	Page Propagation	27-5
27.6.2	Page Regeneration in Background	27-5
27.7	Summary	27-5
27.8	Next Steps	27-5

28 inCache for Page Caching

28.1	Overview	28-1
28.2	Configuring Your System for inCache Page Caching	28-1
28.3	Tuning Options	28-5
28.3.1	Striping the Disk Cache	28-5
28.3.2	Configuring for Page Regeneration During RealTime Publishing	28-7
28.3.3	Setting Up Page Propagation	28-8
28.3.3.1	Enabling Page Propagation	28-9
28.3.3.2	Setting Up Page Propagation on Restart	28-10
28.3.4	Configuring for Pagelet Regeneration in Background	28-10

29 inCache for Asset Caching

29.1	Introduction to Asset Caching	29-1
29.2	AssetCache Container	29-1
29.3	Asset Caching Operations	29-2
29.3.1	Asset Loading	29-3
29.3.2	Comparing Asset Caching with inCache Page Caching	29-4
29.3.3	Flushing AssetCache	29-5
29.4	Types of Asset Caching	29-6
29.5	Customizing Asset Caching	29-7
29.6	Disabling Asset Caching	29-8

29.7	Cache Management Tool	29-8
------	-----------------------------	------

30 System Tools

30.1	Configure log4j	30-1
30.1.1	Setting Up Apache log4j	30-2
30.1.2	Viewing Loggers	30-2
30.1.3	Changing Logger Levels	30-4
30.1.4	Adding New Loggers	30-6
30.2	System Information	30-7
30.2.1	WebCenter Sites Information	30-8
30.2.2	Database Information	30-11
30.2.3	Thread Information	30-13
30.3	Cache Management	30-15
30.4	Cache Management - CAS Cache	30-15
30.4.1	Summary	30-15
30.4.2	Cluster Info	30-17
30.5	Cache Management - Resultset Cache	30-18
30.5.1	Enabling and Configuring Resultset Caching over inCache	30-18
30.5.2	Summary	30-19
30.5.3	Cluster Info	30-22
30.6	Cache Management - Sites Cache	30-22
30.6.1	Summary	30-22
30.6.2	Page Cache	30-24
30.6.3	Dependency	30-26
30.6.4	Asset Cache	30-28
30.6.5	URL Cache	30-29
30.6.6	Cluster Info	30-30
30.7	Cache Management - Co-Res Cache	30-30
30.7.1	Summary	30-30
30.7.2	Page Cache	30-32
30.7.3	Dependency	30-33
30.7.4	URL Cache	30-35
30.7.5	Cluster Info	30-36
30.8	Log Viewer	30-36
30.8.1	Viewing the Log	30-37
30.8.2	Tailing the Log	30-37
30.8.3	Downloading the Log	30-38
30.8.4	Searching the Log	30-38
30.9	File System Test	30-40
30.10	Slots	30-44
30.11	URL	30-46
30.12	Purge Voided Assets	30-46

31 System Defaults

31.1	ACLs	31-1
31.1.1	Permissions	31-1
31.1.2	Accessing ACLs	31-2

31.2	System ACLs	31-2
31.3	ACLs of Default Users	31-5
31.4	Required ACLs for Custom Users	31-6
31.5	System Roles	31-6
31.6	System Asset Types	31-7
31.7	Default Tree Tabs	31-9

32 System Data: WebCenter Sites Database

32.1	Cache Management Tables	32-1
32.2	Approval System Tables	32-2
32.3	Publishing System Tables	32-2
32.4	Workflow Tables	32-2
32.5	Basic Asset Tables	32-3
32.6	Flex Asset Tables	32-3
32.7	Visitor Tables (Oracle WebCenter Sites: Engage)	32-4
32.7.1	Managing the Attribute Tables	32-5
32.7.2	Managing the Session Objects Table (scratch)	32-5
32.7.3	Deleting Unnecessary .class Files	32-6

33 Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems

33.1	Overview	33-1
33.1.1	LDAP Schema	33-2
33.1.2	LDAP Connectivity for Site and Role Management	33-3
33.1.3	LDAP Users and Their Permissions to LDAP Servers	33-4
33.1.4	LDAP-Integrated Operations	33-4
33.2	User Management Operations	33-5
33.2.1	Creating an ACL	33-7
33.2.2	Editing an ACL	33-7
33.2.3	Deleting an ACL	33-7
33.2.4	Assigning ACLs to Tables	33-8
33.2.5	Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries)	33-8
33.2.6	Creating a User (Granting ACLs for Access Privileges)	33-8
33.2.7	Editing a User	33-9
33.2.8	Deleting a User	33-9
33.2.9	Creating/Editing a User Profile	33-10
33.2.10	Deleting a User Profile	33-10
33.2.11	Creating User Attributes	33-10
33.2.12	Editing/Deleting User Attributes	33-11
33.3	Site and Role Management Operations	33-11
33.3.1	Operations with Flat-Schema LDAP Servers	33-11
33.3.2	Operations with Hierarchical-Schema LDAP Servers	33-12
33.3.3	Creating a Site	33-13
33.3.4	Editing a Site	33-14
33.3.5	Deleting a Site	33-14
33.3.6	Granting Users Access to Sites	33-15
33.3.7	Removing Users' Access to Sites	33-15

33.3.8	Creating a Role	33-16
33.3.9	Editing a Role	33-16
33.3.10	Deleting a Role	33-17

Part II Administering the Web Experience Management Framework

34 Web Experience Management Framework

34.1	Overview	34-1
34.2	Administrative Roles and Permissions	34-4
34.2.1	General Administrator	34-4
34.2.2	Site Administrator	34-4
34.2.3	All Administrators	34-5
34.3	WebCenter Sites Applications	34-5
34.4	Sample Sites	34-5

35 Getting Started

35.1	Logging In and 'Quick Tour'	35-1
35.2	Next Steps	35-4

36 Creating and Authorizing Users

36.1	Creating Users	36-1
36.2	Authorizing Users to Work with Applications	36-7
36.3	Authorizing a Predefined User	36-14
36.4	Authorizing Developers to Register Applications	36-14
36.5	Ask Your Developers	36-15
36.5.1	Resources and Applications	36-15
36.5.2	Roles and Applications	36-16
36.5.3	Predefined Users	36-16

37 Configuring REST Security

37.1	REST Authorization	37-1
37.1.1	Security Model	37-1
37.1.2	Configuring REST Security	37-2
37.1.3	Privilege Resolution Algorithm	37-3
37.2	Authorizing Users to Access Application Resources	37-3
37.2.1	Viewing REST Security Configurations	37-3
37.2.2	Creating a Group	37-5
37.2.3	Adding Users to a Group	37-5
37.2.4	Configuring Security for REST Resources	37-7
37.3	REST Security Configuration Reference	37-9
37.3.1	Configuring REST Security for ACL Resources	37-10
37.3.2	Configuring REST Security for Application Resources	37-11
37.3.3	Configuring REST Security for Asset Resources	37-12
37.3.4	Configuring REST Security for Asset Type Resources	37-14
37.3.5	Configuring REST Security for Engage Resources	37-15
37.3.6	Configuring REST Security for Group Resources	37-17

37.3.7	Configuring REST Security for Indexed Asset Type Resources	37-18
37.3.8	Configuring REST Security for Role Resources	37-20
37.3.9	Configuring REST Security for Site Resources	37-21
37.3.10	Configuring REST Security for User Resources	37-22
37.3.11	Configuring REST Security for UserDef Resources	37-23
37.3.12	Configuring REST Security for UserLocale Resources	37-24
37.3.13	Configuring REST Security for Visitor Resources	37-26

38 Working with Sites

38.1	Managing WebCenter Sites CM Sites in the WEM Framework	38-1
38.2	Enabling Tree Tabs	38-4

39 WEM Admin Interface Quick Reference

39.1	Quick Tips for Managing the WEM Framework	39-1
39.2	Managing Sites	39-2
39.3	Managing Applications	39-4
39.4	Managing Users	39-4
39.5	Managing Roles	39-6
39.6	Managing Profiles	39-6

Part III Administering Oracle WebCenter Sites: Community-Gadgets

40 Enabling the Community and Gadgets Interfaces

40.1	Enabling the Community Interface	40-1
40.1.1	Enabling the Community Interface for Existing Site Users	40-2
40.1.2	Enabling the Community Interface for New Users	40-3
40.2	Enabling the Gadgets Interfaces	40-4
40.2.1	Enabling the Global Gadget Catalog Interface	40-4
40.2.2	Enabling the Gadgets User Interface	40-5
40.2.2.1	Enabling the Gadgets User Interface for Existing Site Users	40-5
40.2.2.2	Enabling the Gadgets User Interface for New Users	40-6

41 Working with the Cache Tool

41.1	Overview of the Cache Tool	41-1
41.2	Viewing Cache Information	41-1
41.2.1	Viewing Information in the Common Cache	41-3
41.2.2	Viewing Information in the Dependencies Cache	41-5
41.2.3	Viewing Information in the Shared Cache	41-6

Part IV Administering the Oracle WebCenter Sites: Gadgets Application

42 Overview of the WebCenter Sites: Gadgets Application

42.1	Overview of the Gadgets Application Interfaces	42-1
42.2	Gadgets Application Roles and Permissions	42-2
42.3	Making Gadgets Available for Deployment	42-4

43 Getting Started with the Gadgets Application

43.1	Logging In	43-1
43.2	Sample Gadgets	43-4
43.3	Next Steps	43-5

44 Registering Gadgets

44.1	Overview of Gadget Registration and Deployment	44-1
44.2	Gadget Sources	44-5
44.3	Registering Gadgets to the Global Catalog	44-5
44.4	Registering Gadgets to a Site Gadget Catalog	44-10
44.5	Deleting Gadgets from the Global Catalog or Site Gadget Catalog	44-11

45 Managing Gadgets

45.1	Sharing Gadgets from the Global Catalog	45-1
45.2	Modifying a Gadget's Metadata	45-3
45.2.1	Modifying a Gadget's Metadata in the Global Catalog	45-3
45.2.2	Modifying a Gadget's Metadata in a Site Gadget Catalog	45-4

Part V Administering the Community Blog Module

46 Enabling Blog Components

46.1	Enabling Blog Asset Types	46-1
46.2	Creating a Blog Tree Tab	46-2
46.3	Sharing Blog Assets	46-4
46.4	Creating Workflows for Blog Assets	46-5
46.5	Working with Blog Authors	46-5
46.5.1	Creating Author Assets	46-5
46.5.2	Setting Access Permissions to a Blog Author Asset	46-7
46.6	Publishing Blog Assets	46-9
46.7	Next Steps	46-9

Part VI Implementing Oracle WebCenter Sites: Site Capture

47 Overview of the Site Capture Application

47.1	Site Capture Model	47-1
47.1.1	Capture Modes	47-1
47.1.2	Crawlers	47-2
47.2	Logging in to the Site Capture Application	47-2
47.3	Using the Default Crawlers	47-4
47.3.1	Sample Crawler	47-5
47.3.2	FirstSiteII Crawler	47-5
47.3.3	Running a Default Crawler	47-5
47.4	Setting Up a Site Capture Operation	47-5
47.4.1	Creating a Starter Crawler Configuration File	47-6
47.4.2	Defining a Crawler	47-7

47.4.3	Editing the Crawler Configuration File	47-9
47.4.4	Starting a Crawl	47-10
47.4.4.1	Run the Crawler Manually in Static Mode	47-10
47.4.4.2	Run the Crawler Manually in Archive Mode	47-12
47.4.4.3	Schedule the Crawler for Archive Capture	47-15
47.4.4.4	Publish a Site in RealTime Mode	47-16
47.4.5	Managing Captured Data	47-16
47.5	Enabling Publishing-Triggered Site Capture	47-17
47.5.1	Integrating the Site Capture Application with Oracle WebCenter Sites	47-17
47.5.2	Configuring a RealTime Publishing Destination Definition for Site Capture	47-17
47.5.3	Matching Crawlers	47-18
47.5.4	Next Steps	47-18

48 Managing Downloaded Sites

48.1	Managing Statically Captured Sites	48-1
48.2	Managing Archived Sites	48-3
48.3	Summary	48-4
48.3.1	Creating and Editing Crawlers	48-5
48.3.2	Deleting a Crawler	48-5
48.3.3	Scheduling a Crawler	48-5
48.3.4	Monitoring a Static Crawl	48-6
48.3.5	Stopping a Crawl	48-6
48.3.6	Downloading Archives	48-6
48.3.7	Previewing Sites	48-6
48.3.8	Configuring Publishing Destination Definitions	48-6
48.3.9	Accessing Log Files	48-6

49 Site Capture File System

49.1	General Directory Structure	49-1
49.2	Custom Folders	49-3

Part VII Oracle WebCenter Content Integration

50 About WebCenter Content Integration

50.1	Administrator's Prerequisites	50-1
50.2	Basics of WebCenter Content Integration	50-1
50.3	WebCenter Content Data Model	50-2
50.4	Asset Type Mapping and Connector Rules	50-3
50.4.1	Asset Type Mapping	50-3
50.4.1.1	Content Identifier Attribute	50-4
50.4.1.2	Content Storage Attributes	50-4
50.4.1.3	Template Attributes	50-4
50.4.1.4	Administrator-Defined Specifications	50-5
50.4.2	Connector Rules	50-5
50.5	WebCenter Content Integration and Its Parts	50-7
50.6	First Steps	50-8

51 Enabling the WebCenter Content Integration Feature

51.1	Enabling the SitesIntegration Component	51-1
51.2	Enabling Access to the Connector Admin Tab	51-2
51.3	Next Steps	51-4

52 Setting Up Content Synchronization

52.1	Step 1: Enabling Content Items for Synchronization	52-2
52.2	Step 2: Determining Which Renditions to Synchronize	52-2
52.3	Step 3: Selecting or Creating Target Asset Types	52-3
52.4	Step 4: Connecting to WebCenter Content	52-3
52.5	Step 5: Creating an Active Rule	52-4
52.6	Step 6: Prioritizing Active Rules and Managing Rules	52-11
52.7	Step 7: Manually Running the Connector	52-12
52.8	Step 8: Monitoring the Connector	52-13
52.9	Step 9: Viewing Connector History	52-15
52.9.1	Viewing the Entire Connector History	52-15
52.9.2	Viewing Connector History for a Single Batch	52-17
52.10	Step 10: Verifying Imported Content	52-19
52.11	Step 11: If You Need to Modify Your Setup	52-19
52.12	Step 12: Scheduling the Connector	52-20
52.13	Step 13: Managing Synchronized Assets	52-22
52.13.1	Mapped Attributes	52-22
52.13.2	Orphaned Content	52-22

53 Resynchronization Options

53.1	Rolling Back the Synchronization Point	53-1
53.1.1	Rolling Back to the Original Starting Point	53-2
53.1.2	Rolling Back to a Particular Update	53-3
53.2	About the Generated Token Value	53-4

54 WCC Connector Configuration Files

54.1	About WCC Connector Configuration Files	54-1
54.2	User Configurable Parameters	54-1

Part VIII Implementing Oracle WebCenter Sites Content Integration Platform for File Systems and Microsoft SharePoint

55 Integrating with Oracle WebCenter Sites

55.1	Overview	55-1
55.2	System Defaults	55-3
55.2.1	Default Mapping Framework	55-3
55.2.2	Implications for Publishing and Synchronization	55-4
55.2.2.1	Default mappings.xml and Flex Families	55-5
55.2.2.2	Custom mappings.xml and Flex Families	55-5
55.3	Event Notification Option	55-5

56	Publishing	
56.1	Overview	56-1
56.1.1	Synchronization	56-1
56.1.2	Unpublishing	56-2
56.2	Publishing to WebCenter Sites	56-2
56.3	Examples	56-3
56.3.1	Publishing from a File System	56-3
56.3.2	Publishing from Microsoft SharePoint	56-4
56.4	When an Object is Published	56-4
56.5	Synchronization	56-4
56.6	Unpublish Command	56-5
57	Configuring Event Notification	
57.1	Overview	57-1
57.2	Sample Workflows	57-2
57.2.1	Installing Sample Workflows	57-2
57.2.2	Verifying Sample Workflows	57-3
57.2.3	Enabling Sample Workflows	57-3
58	Remapping	
58.1	When to Remap	58-1
58.2	Remapping Procedures	58-1
58.2.1	File Systems	58-1
58.2.2	Microsoft SharePoint Systems	58-3
58.2.2.1	Tags and Parameters for Mapping Microsoft SharePoint Data to Oracle WebCenter Sites	58-5
59	Adding Assets to a Flex Family	
59.1	Adding a New Attribute	59-1
59.2	Adding a New Child Definition Asset	59-3
60	File Systems: Default Mapping Specifications	
60.1	Overview	60-1
60.2	Default mappings.xml	60-3
60.3	FileSystem Flex Family Specifications	60-4
61	Microsoft SharePoint Systems: Default Mapping Specifications	
61.1	Overview	61-1
61.2	Default mappings.xml	61-3
61.3	SharePoint Flex Family Specifications	61-4

Part IX Implementing Oracle WebCenter Sites Content Integration Platform for EMC Documentum

62 Publishing to Oracle WebCenter Sites

62.1	Overview of the Publishing Process	62-1
62.1.1	System Architecture and Process Flow	62-2
62.1.2	Mapping Framework	62-3
62.2	Publishing Procedures	62-4
62.2.1	Publishing via the Default Mapping	62-5
62.2.2	Publishing via Customized Mappings	62-6
62.3	Maintaining the Integrated Systems	62-7
62.3.1	Tuning the Synchronization Process	62-7
62.3.2	Unpublishing	62-8

63 Archiving to EMC Documentum

63.1	Overview of the Archival Process	63-1
63.1.1	System Architecture and Process Flow	63-2
63.1.2	CS DataStore	63-3
63.1.3	Mapping Framework	63-5
63.1.3.1	Object Types in the Documentum Workspace	63-5
63.1.3.2	mappings.xml	63-5
63.1.3.3	Mappings, Publishing, and Synchronization	63-6
63.2	Steps for Archiving WebCenter Sites Assets to Documentum	63-6
63.2.1	Prepare the Documentum System to Store WebCenter Sites Assets	63-7
63.2.2	Configure the Path to the CS DataStore	63-7
63.2.3	Add Metadata to mappings.xml	63-7
63.2.4	RealTime Publish the Site	63-8
63.2.5	Archive the CS DataStore on Documentum	63-8
63.2.6	Archive Visitor-Generated Content	63-9
63.3	Testing Synchronization	63-10
63.4	Tuning the Synchronization Process	63-10

64 Configuring Event Notification

64.1	Overview of Event Notification	64-1
64.2	Configuring Event Notification Workflows for WebCenter Sites	64-1
64.2.1	Installing Default Workflows on WebCenter Sites	64-3
64.2.2	Verifying the Installation of Default Workflows	64-3
64.2.3	Enabling Default Workflows	64-4
64.3	Configuring Event Notification Workflows for EMC Documentum	64-4

65 Default Mapping Specifications for Publishing

65.1	Mapping Framework	65-1
65.2	Documentum Flex Family Specifications	65-2

Preface

This guide provides procedures for administering Oracle WebCenter Sites.

Audience

This document is intended for general administrators of WebCenter Sites, who also have experience with installing and configuring enterprise-level software. Also required is a strong understanding of the WebCenter Sites flex asset model and RealTime publishing, in addition to a clear understanding of EMC Documentum and its data models.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle WebCenter Sites Release 11gR1 documentation set:

- *Oracle WebCenter Sites Release Notes*
- *Oracle Fusion Middleware WebCenter Sites User's Guide*
- *Oracle Fusion Middleware WebCenter Sites Developer's Guide*
- Oracle Fusion Middleware WebCenter Sites Property Files Reference
- Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software

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.

What's New in This Guide

This preface introduces the new and changed features of WebCenter Sites 11g Release 1 (11.1.1.8.0) and other significant changes that are described in this guide, and provides pointers to additional information. This preface also lists deprecated features.

New and Changed Features in 11g Release 1 (11.1.1.8.0)

WebCenter Sites 11g Release 1 (11.1.1.8.0) includes the following new and changed features, described in this guide:

- **New and Changed Features**

- Updated Admin Interface. WebCenter Sites forms now offer an improved user experience for administrators to create and manage data. This includes a rearrangement of nodes on the administrative tree tabs for more intuitive navigation.
- WebCenter Sites Security Cryptography. This new feature provides for secure server access to protect your system from internal or external security attacks. All access is protected through cryptographic encryption of sensitive information whether located in files or within the server itself. See [Chapter 7, "Implementing Security Using Cryptography and Encryption Key Storage."](#)
- Vanity URLs. This new feature supports short, readable, easy-to-control URLs that typically either contain the name of, or describe the web page to which they point. A vanity URL can be either auto-generated or created by a content contributor in the WebCenter Sites Contributor interface. See [Chapter 24, "Configuring Vanity URLs."](#)
- WebCenter Content Integration. This new feature is a GUI-controlled connector. Its function is to synchronize documents and digital assets (such as images and videos) from WebCenter Content to WebCenter Sites, where they are managed as native assets. See [Part VII, "Oracle WebCenter Content Integration"](#) for more information.

- **Document Consolidation**

This guide consolidates administrator's documentation that was once available in separate books. Topics that have been added to this guide are the following:

- [Part II, "Administering the Web Experience Management Framework"](#) describes WEM Framework administration through the WEM Admin interface, which supports centralized application management and user authorization. The WEM Framework is used to develop and integrate applications with Oracle WebCenter Sites.

- [Part III, "Administering Oracle WebCenter Sites: Community-Gadgets"](#) provides instructions on enabling the Community and Gadgets interfaces and contains information about viewing the Community-Gadgets data that is stored in the production and management caches.
- [Part IV, "Administering the Oracle WebCenter Sites: Gadgets Application"](#) contains information about registering gadgets to the global catalog and site gadget catalogs. It also contains instructions for managing registered gadgets.
- [Part V, "Administering the Community Blog Module"](#) provides administrators with instructions for enabling content providers to use the Community Blogs module to create blogs and modify their author profiles.
- [Part VI, "Implementing Oracle WebCenter Sites: Site Capture"](#) describes the web application for downloading dynamically published websites. It shows you how to navigate the Site Capture application and its file system to manage crawl sessions and downloaded websites.
- [Part VII, "Oracle WebCenter Content Integration"](#)

Note: This part is new. It describes a new feature, WebCenter Content Integration, and provides instructions for using this feature to synchronize documents and digital assets (such as images and videos) from WebCenter Content to WebCenter Sites, where they are managed as native assets.

- [Part VIII, "Implementing Oracle WebCenter Sites Content Integration Platform for File Systems and Microsoft SharePoint"](#) contains procedures for publishing content to the Oracle WebCenter Sites web application from file systems and Microsoft SharePoint.
- [Part IX, "Implementing Oracle WebCenter Sites Content Integration Platform for EMC Documentum"](#) contains procedures for configuring and using the platform to transfer content between Oracle WebCenter Sites and EMC Documentum.

Deprecation Notice

The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0):

- SOAP-based web services. This feature is replaced by REST services.
- Mirror publishing. This feature is replaced by RealTime publishing.
- Static publishing
- WebCenter Sites Desktop
- WebCenter Sites DocLink

Part I

Administering WebCenter Sites

The Oracle Fusion Middleware WebCenter Sites Administrator's Guide describes all facets of the administration of websites, from creation to delivery. The maintenance of these websites, including creating and maintaining user roles, assets, and publishing is detailed here.

Part I contains the following chapters:

- [Chapter 1, "Overview"](#)
- [Chapter 2, "Administrator's Interface"](#)
- [Chapter 3, "Site Configuration Guidelines"](#)
- [Chapter 4, "Working with ACLs and Roles"](#)
- [Chapter 5, "Configuring Users, Profiles, and Attributes"](#)
- [Chapter 6, "Setting Up External Security"](#)
- [Chapter 7, "Implementing Security Using Cryptography and Encryption Key Storage"](#)
- [Chapter 8, "Assembling Content Management Sites"](#)
- [Chapter 9, "Managing Access to Content Management Site Components"](#)
- [Chapter 10, "Creating and Managing Workflow Processes"](#)
- [Chapter 11, "Replicating Content Management Sites"](#)
- [Chapter 12, "Publishing with Oracle WebCenter Sites"](#)
- [Chapter 13, "The Approval System"](#)
- [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process"](#)
- [Chapter 15, "The Export to Disk Publishing Process"](#)
- [Chapter 16, "The Mirror to Server Publishing Process"](#)
- [Chapter 17, "The Export Assets to XML Publishing Process"](#)
- [Chapter 18, "Additional Publishing Procedures"](#)
- [Chapter 19, "RealTime Publishing Overview"](#)
- [Chapter 20, "Configuring the RealTime Publishing Process"](#)
- [Chapter 21, "Working with RealTime Publishing Modes"](#)
- [Chapter 22, "On Demand Publishing and Asset Unapproval"](#)
- [Chapter 23, "Configuring the User Interfaces"](#)

- Chapter 24, "Configuring Vanity URLs"
- Chapter 25, "Configuring the Lucene Search Engine"
- Chapter 26, "Revision Tracking"
- Chapter 27, "inCache Framework"
- Chapter 28, "inCache for Page Caching"
- Chapter 29, "inCache for Asset Caching"
- Chapter 30, "System Tools"
- Chapter 31, "System Defaults"
- Chapter 32, "System Data: WebCenter Sites Database"
- Chapter 33, "Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems"

As the administrator, you will be working with Content Management (CM) sites throughout your job, configuring the sites, replicating them, and managing their day-to-day use. Because they are a critical part of your work, you need to understand what CM sites are, and how they relate to the online site. This chapter provides an overview.

This chapter also outlines the administrator's job in the Oracle WebCenter Sites environment and offers guidelines for planning configuration tasks.

This chapter contains the following sections:

- [Section 1.1, "Introduction"](#)
- [Section 1.2, "Online Sites"](#)
- [Section 1.3, "CM Sites"](#)
- [Section 1.6, "Content Management Models"](#)
- [Section 1.7, "Oracle WebCenter Sites Environment"](#)
- [Section 1.8, "Administrator's Job"](#)
- [Section 1.9, "The Focus of This Guide"](#)
- [Section 1.10, "Administrator's Prerequisites"](#)

Note: The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Sites Desktop, Sites DocLink.

1.1 Introduction

As the Oracle WebCenter Sites administrator, your job is to build the foundation of your online site, specifically, its back end. The back end itself is a site (or set of sites) in the Oracle WebCenter Sites installation—a site where business users work to create their electronic assets, manage them, and deploy them to their audiences. Building the back end involves configuring the business users, linking them to content-entry forms as well as other authoring tools, and providing the users with publishing and delivery systems for serving the online site to browsers.

When the back end is configured, its users log on and fill in content-entry forms. The content they enter is then saved to tables in the Oracle WebCenter Sites database. When ready for delivery, the content is drawn programmatically from the database tables, formatted, and served to browsers as the online site. [Figure 1–2, "Component Website and Relations"](#) and [Figure 1–2, "Component Website and Relations"](#) illustrate the process flow, using a Mirror to Server publishing scenario.

In our scenario, [Figure 1-1, "CM Site"](#) depicts the smallest unit that can function as a back end: a user who has permissions to write to the Oracle WebCenter Sites database and to invoke the publishing-delivery systems. This unit, Oracle WebCenter Sites defines as a CM site. Oracle WebCenter Sites imposes no limit on the number of CM sites that you can configure for the back end of the online site, or the number of ways in which the sites can be configured.

Throughout your job as the Oracle WebCenter Sites administrator, you will be working with CM sites—configuring them, replicating them, and managing their day-to-day use. Because they are a critical part of your work, you need to understand how CM sites are defined, how they must be configured, and how they relate to the online site. This chapter provides an overview.

1.2 Online Sites

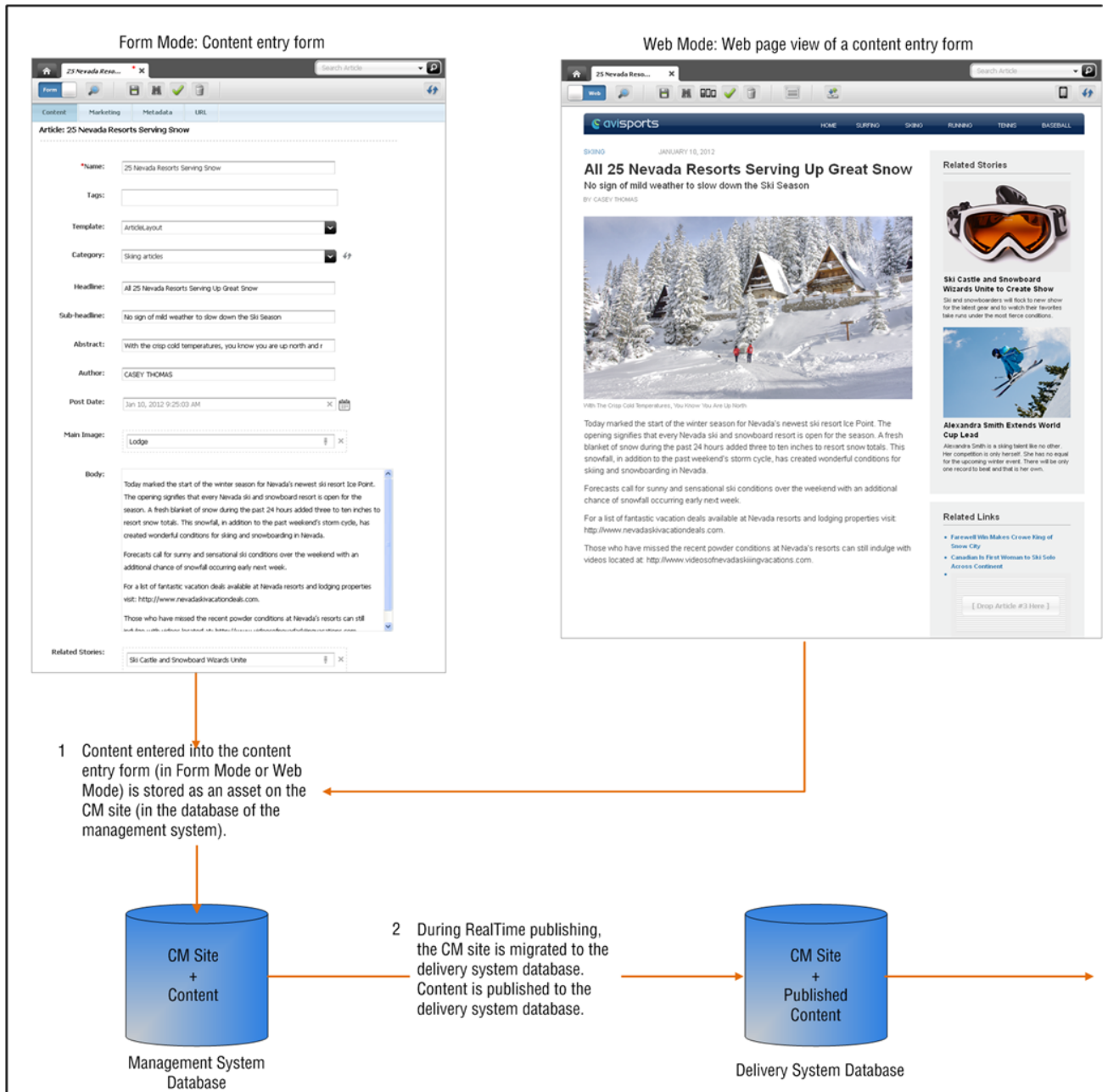
Although most Internet users don't need a definition of "online site," it is important to describe what an online site powered by Oracle WebCenter Sites is. Later in this chapter, we explain how online sites can be modeled by CM sites.

A WebCenter Sites-powered online site is the set of pages that an organization displays to its target audience of customers, clients, and casual visitors via Oracle WebCenter Sites. The online site can be accessible to the general public or it can be a password-protected site. It can also be a completely exclusive site, such as a corporate network or departmental network, operating strictly within the private domain.

Regardless of its nature, an online site originates from either a single CM site, or many CM sites, depending on which model you choose. Throughout our product guides, we use the term "online site" generically to refer to websites.

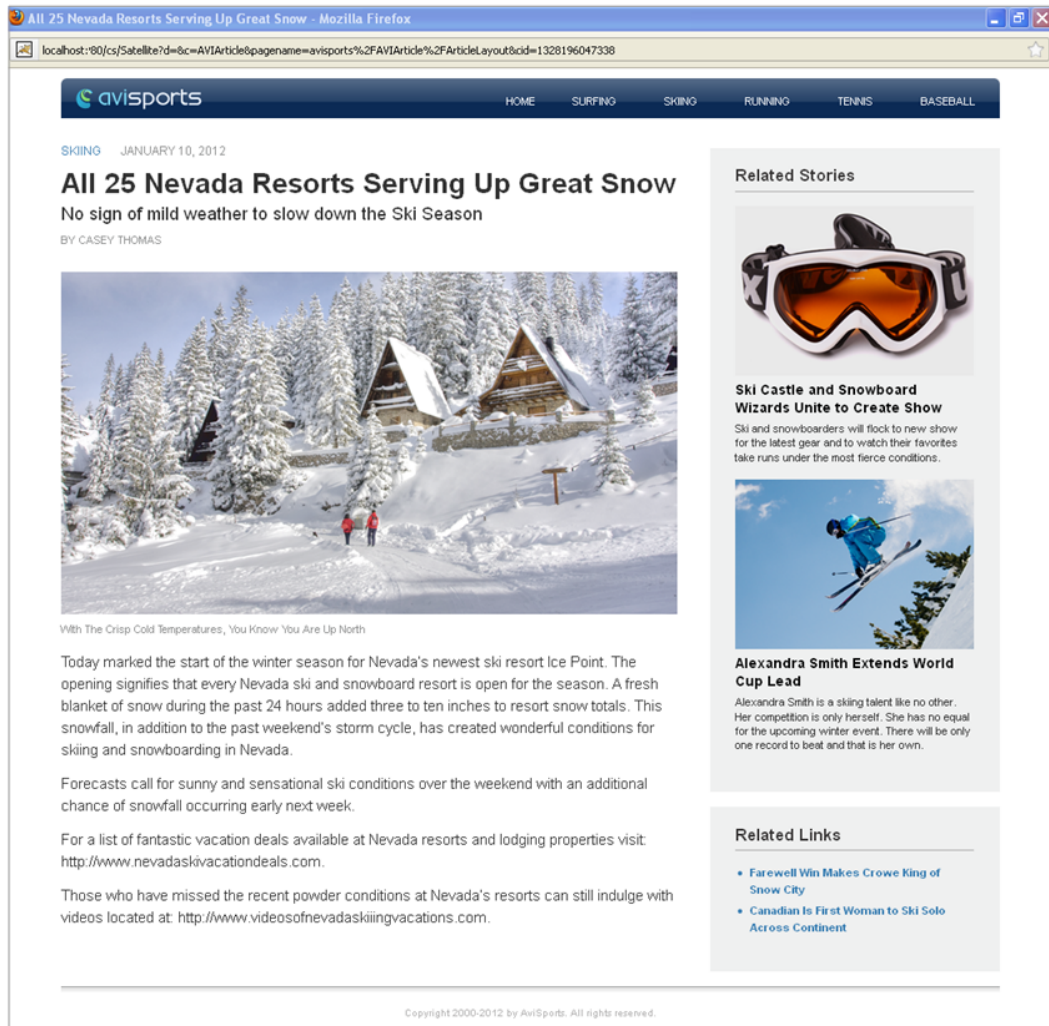
This content was created by a user populating one of the content-entry forms below.

Figure 1-1 CM Site



This online site, a single page, was created from the asset shown in Figure 1-1.

Figure 1–2 Component Website and Relations



4 Published content is formatted by templates, laid out by templates, and delivered to the online site

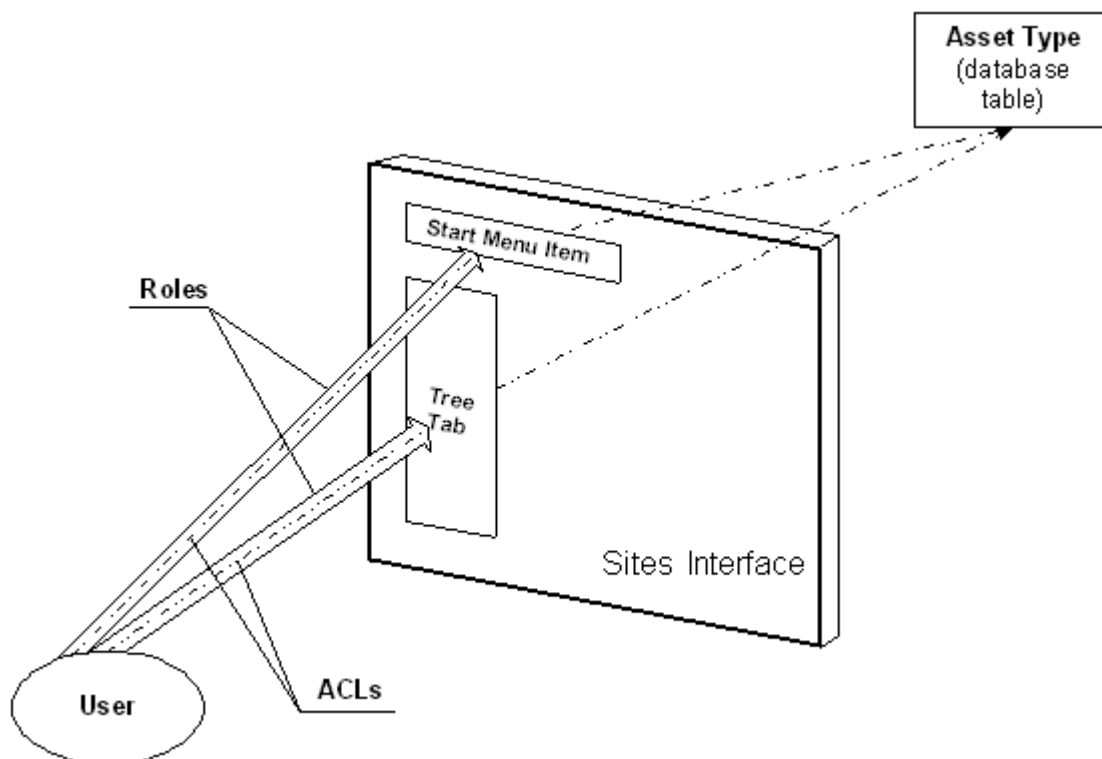
1.3 CM Sites

A CM site is a content management unit within Oracle WebCenter Sites. It is the source of content for the online site and can represent either an entire online site or one of its sections.

Note: From this point forward, when the term "site" is used without a qualifier, it means "CM site."

A site is an object that you must configure in order to first define the authors and managers of the online site, and then provide them with the permissions and content management tools they will need: content-entry forms, content-rendering templates, workflow processes, start menu items, publishing methods, and a delivery system. The process of configuring a site involves creating not only the site components, but also associating the components with each other. Making the associations defines a site, similar to the one shown in [Figure 1-3](#).

Figure 1-3 User Roles and ACLs



The site is identified by a name (also configured by the administrator), stored in Oracle WebCenter Sites' database, and listed in authorized users' interfaces. Users with the correct roles have access to the site. Within the site, roles manage the users' access to specific functions (such as a Start Menu Item) in the Oracle WebCenter Sites interface. ACLs manage the user's permissions to content (database tables). Through the interface functions (made accessible by roles), the user is able to actuate his permissions (as defined in the ACLs), and therefore operate on the database tables in order to author and manage specific types of content.

A Oracle WebCenter Sites system typically has many such sites, each one unique in its function and composition. When users log in to Oracle WebCenter Sites, they must also select a site where they will work. The site can be independent of other sites, or it can share assets with other sites, providing that the sites have a common set of users.

Note: In this release, Oracle WebCenter Sites offers a site-replication utility called Site Launcher that speeds up the site creation process. Instead of creating sites from scratch, you can replicate established sites as necessary, modify the replicates, and spin them off as new sites in the Oracle WebCenter Sites environment.

1.4 Configuration Components

Configuring a site involves using Oracle WebCenter Sites' administrator interface to access a system-wide configuration pool, select (or create) components that will make up the site, and associate the components with each other so they can function together to produce either the online site or one of its sections.

[Table 1-1, "Site Components"](#) lists site components, most of which are required. The optional components vary according to business needs and user preferences. Whereas developers are responsible for the code-based components, the administrator is responsible for all other components.

Site components are generally complex constructs, especially the data model. They are described in the sections that follow. Data modeling is described in detail in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Table 1-1 Site Components

Component	Required	Created by	See page ...
CM site definition	Yes	WebCenter Sites administrator	6
Data model	Yes	Developers	7
Users	Yes	WebCenter Sites administrator	7
Roles	Yes	WebCenter Sites administrator	9
Start menu items	Yes	WebCenter Sites or its administrator	11
Workflow processes	No	WebCenter Sites administrator	12
Publishing system	Yes	WebCenter Sites administrator	12
User interface options	No	WebCenter Sites administrator	12

This section contains the following topics:

- [Section 1.4.1, "CM Site Definition"](#)
- [Section 1.4.2, "Data Model"](#)
- [Section 1.4.3, "Users and Their Table-Level Permissions"](#)
- [Section 1.4.4, "Roles"](#)
- [Section 1.4.5, "Start Menus"](#)
- [Section 1.4.6, "Workflow Processes"](#)
- [Section 1.4.7, "Publishing System"](#)

- [Section 1.4.8, "User Interface Options"](#)

1.4.1 CM Site Definition

A CM site definition is one of the components of a CM site. It consists of a site name and optionally a description, both specified by the administrator. The site name can represent an online site, a business topic, the work of a department, or yet another type of content. In any case, the site name establishes a business theme that users in the site are responsible for developing and maintaining.

When you specify a site name, Oracle WebCenter Sites creates a node to represent the site in its interface. It also appends several default sub-nodes for linking components to the site: an Asset Types sub-node for linking the data model to the site, a Users sub-node for linking users to the site, and two other nodes for enabling the optional user interfaces Desktop and DocLink. Descriptions of these components are given in the next sections.

For content providers, the site is a visibility control mechanism. The site provides authorized users access to certain content in the Oracle WebCenter Sites installation. When a site is properly configured, its name is displayed in the content providers' interfaces, allowing the content providers to select the site and navigate within it according to their roles and permissions.

1.4.2 Data Model

The data model is a component of a site. It comprises a set of asset types (database tables) and asset type definitions, coded by developers for content providers' use. To help developers equip content providers with the broadest possible set of content management options, Oracle WebCenter Sites supports three kinds of asset types.

- Content asset types, which are structured repositories for content and by design reflect the business theme established by the site definition. For example, if you defined a site named "Social Events," suitable content asset types could be "Selected Moments in History," "The 20th Century's Greatest Events," and so on, since they pertain to the theme suggested by the site name.

From asset types, developers code asset type definitions. These are expressed in the WebCenter Sites interface as content-entry forms (such as the one in [Figure 1-1, "CM Site"](#)), whose fields prompt users for information that will be delivered to the online site (or reserved for internal use, if necessary). The set of fields defines the asset type; users' field entries define the asset (an instance of the asset type).

- Design asset types, which are used by developers to code template assets, which render the content assets. (We distinguish template assets from content assets in order to distinguish presentation code from content.)
- Management asset types, which are also used by developers to create tools such as simple searches and database queries that help content providers manage their content.

Ordinarily, developers test the asset types they create and pass them on to you so that you can link them to the site definition (through the Asset Types node) and complete another step in the site configuration process

Note: Oracle WebCenter Sites provides a number of default asset types and allows developers to create their own. The asset types, their definitions, and the assets themselves are stored in the Oracle WebCenter Sites database as tables or table entries, and loosely referred to as content in this guide.

For information about data modeling, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. We recommend that administrators read the guide to gain a basic understanding of asset types.

1.4.3 Users and Their Table-Level Permissions

Users are site components. In this guide, they are often referred to as content providers—people who use the developers' data model to author and manage content.

Content providers are the subject matter experts. They can be:

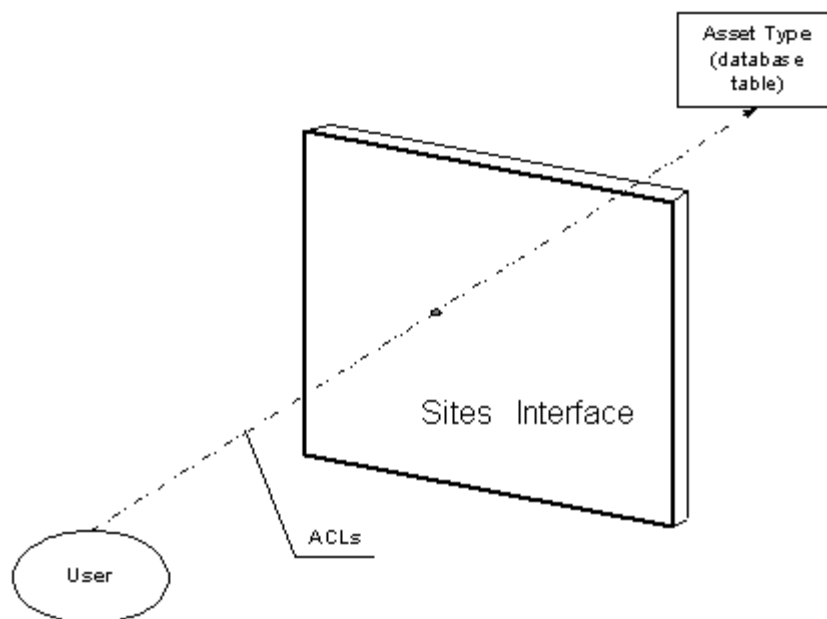
- Authors of online content. Copywriters and designers would fall into this group.
- Reviewers, who examine and edit the content that other users submit to them in order to ensure its quality. Examples of content reviewers include editors and art directors who review and modify the copy and designs that are submitted to them.
- Content publishers, who ensure that content is ready to be delivered to the online site, and approve the content for delivery. An editor-in-chief could be a content publisher.
- Content managers, who oversee the authoring, review, and publishing processes.

Each content provider must be identified to Oracle WebCenter Sites through a user account, which consists of a user name, a password, and Access Control Lists (ACLs), the foundation of Oracle WebCenter Sites' security system.

An ACL is a set of permissions to database tables. The permissions (such as read and write) are granted when the same ACL is assigned to both the table and the user. If no ACLs are common to a table and a user, the user has no permissions to the table.

For example, the system table named SystemUsers contains user account information. The table is assigned three ACLs: SiteGod, UserReader, and UserEditor. If a user is assigned one of the ACLs—UserReader, in our example—he can read the table. If the same user is assigned a second ACL—UserEditor—he can also edit the table. If the user is not assigned any of the ACLs, he has no permissions to the table.

Be aware, that while ACLs give the user permissions to operate on tables, they do not give the user the means to operate on the tables. For example, in our preceding scenario, the user's permissions to read and edit the table translate into permissions to use the View and Edit functions in the Oracle WebCenter Sites interface. However, the functions are hidden from the user, unless the same roles are assigned to both the user and the functions. For information about roles, see [Section 1.4.4, "Roles."](#)

Figure 1–4 User ACLs

In general, Oracle WebCenter Sites uses ACLs at two levels:

- At the system security level, to provide authentication functionality and therefore prevent hackers from entering the Oracle WebCenter Sites environment.
- At the interface level, to control users' permissions to database tables and, therefore, control the ability to use (but not view) interface functions through which the permissions are actuated.

Oracle WebCenter Sites provides a number of default system ACLs and pre-assigns them to system tables. You can re-use the ACLs, or configure your own and assign them to custom tables, as necessary. For more information about ACLs, see [Chapter 4, "Working with ACLs and Roles"](#) and [Chapter 31, "System Defaults."](#)

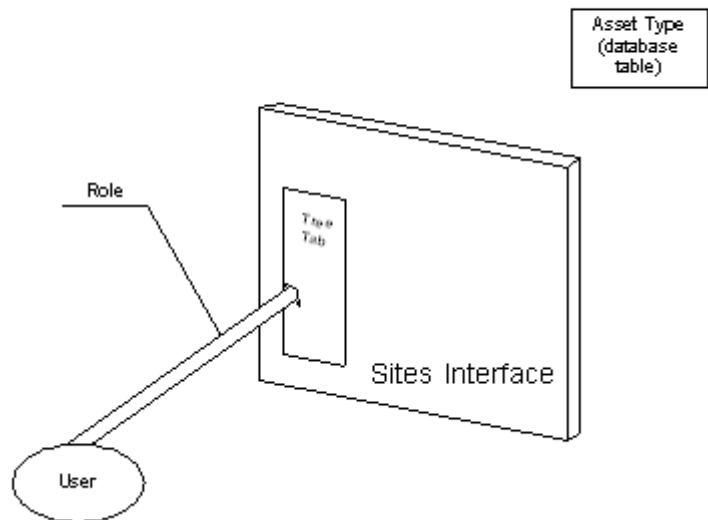
1.4.4 Roles

Whereas ACLs give the user permission to operate on database tables, roles give the user the means to operate on database tables. Roles determine whether the user has access to a site, and whether interface functions, such as **Edit**, **Delete**, and **Start Workflow** are exposed in the user's interface. If the functions are hidden, the user's permissions to database tables (as specified in the ACLs) cannot be actuated, leaving the user unable to operate on the tables. Roles also define groups of users, such as authors and editors. They are used to describe the groups' permissions (and therefore the users' permissions) to sites, the sites' content, to collateral (such as start menus for creating and locating content), and to workflow processes.

Roles are implemented in the same way as ACLs; that is, for a function to be displayed in the user's interface, the function and the user must be assigned the same role. To illustrate, we continue our previous scenario, where a user is given editorial permissions to the SystemUser table through the UserReader and UserEditor ACLs. To exercise those permissions, the user needs access to the Edit function in the Oracle WebCenter Sites interface. Edit is located in the Admin tab. To view the Admin tab, however, the user must be assigned the same role as the tab. By default, the tab is assigned the GeneralAdmin role; the same role must also be assigned to the user.

Figure 1-5 User Roles

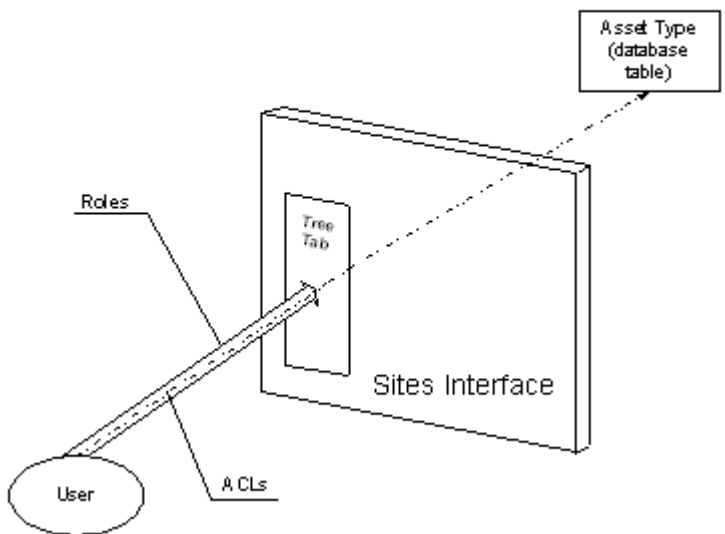
Roles, without ACLs, give the user the means (but not the permission) to operate on database tables



To summarize, the user gains full access to the database table only when he is assigned the ACLs of the table, and the role of the Admin tab. In practice, as in our scenario, roles and ACLs must be compatibly assigned—role assignments must support the permissions that are granted by ACL assignments.

Figure 1-6 User Roles and ACLs

ACLs + roles give the user the permission and the means to operate on database tables



In addition to displaying interface functions to site users, roles provide a way of grouping users according to their responsibilities on the site. Users with similar responsibilities can be assigned the same role(s). For example, administrative users

need access to the Admin tab. All the administrators can be assigned the GeneralAdmin role and thereby be given access to the Admin tab.

Oracle WebCenter Sites defines several default system roles, all of which are pre-assigned to various functions in the Sites interface. Oracle WebCenter Sites also allows you to configure and assign your own roles. When choosing role names, consider the responsibilities of the users on the site and select the role names accordingly. Note that unlike ACLs (which are mapped to database tables), roles are mapped to sites and functions on the sites (which means that roles must be assigned on a per-user, per-site basis)

Note: Unlike ACLs, roles are exposed to content providers for enlisting other content providers into workflow processes.

1.4.5 Start Menus

A start menu is a site component. Items on the menu provide a way of coupling a user with the asset types he is to work with on the site. The coupling is accomplished by means of roles.

Oracle WebCenter Sites defines several start menus: New, which allows the user to create assets on the site; Search, which allows the user to look for and edit assets on the site; start menus for Sites Desktop, which enables Microsoft Word as an alternative interface to WebCenter Sites content-entry forms; and start menus for Sites DocLink which enables the document management interface.

A start menu item specifies:

- The site(s) to which the start menu item applies
- The asset type that users on the sites can work with
- The roles that are allowed to create or search for assets of those types
- The workflow processes (if any) in which the roles can participate

In other words, a start menu item determines which roles can create and search for assets of a specific type on a site, and which workflow processes the roles can participate in.

Oracle WebCenter Sites creates start menu items automatically and gives you the option to configure your own.

1.4.6 Workflow Processes

Workflow processes are optional constructs, mandated only by business needs, and used to regulate collaborations among content providers.

A workflow process enlists qualified users with complementary expertise to perform a sequence of operations that starts with the creation of a content asset, continues with review of the asset, and culminates in approval of the asset. Once approved, the asset is published to the delivery system by one of the users and finally, delivered as content to the online site.

Once workflows are configured, they can be specified in start menu items as processes that automatically begin the moment newly created assets are saved. Workflow processes can also be omitted from start menu items, to be invoked by users as necessary. For more information about workflow processes, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

1.4.7 Publishing System

The publishing system is a site component. The publishing system gives users the means to migrate sites and their content from one system to another. The publishing methods that can be configured are: RealTime, Mirror to Server, Export to Disk, and export to XML. The structure of the site (that is, the database schema) is migrated by means of either the Mirror to Server or the RealTime publishing method. The content itself can be published at the administrator's discretion.

1.4.8 User Interface Options

Oracle WebCenter Sites supports a number of optional interfaces through which users can interact with content:

- **Oracle WebCenter Sites Desktop (Sites Desktop)**, which provides users with the familiar Microsoft Word interface as an alternative to the content-entry forms that are native to Oracle WebCenter Sites. Users can create their content in MS-Word documents, import the Word documents as assets into the Sites database, and from there, recall and edit the documents.

Note that Sites Desktop requires the content in Word documents to be structured. For example, when using Oracle WebCenter Sites Desktop to author content, the user opens a Word document, enters content, and structures the content by tagging it with the same field names as defined in the equivalent content-entry form. The tagging utility is embedded in the Word interface, and the selection of fields is determined by the administrator. When the Word document is saved, the content in its fields is parsed to the appropriate database table(s).

- **Oracle WebCenter Sites DocLink (Sites DocLink)**, which supports unstructured content in the flex asset family.

Sites DocLink provides a drag-and-drop interface for uploading and downloading unstructured content—documents, graphics, or other single binary files that are managed as flex assets. Sites DocLink also presents the hierarchical structure of any flex asset family in the Oracle WebCenter Sites database as folders and files in the Windows Explorer application.

- **Web Mode of the Contributor interface**, which supports the editing of content directly on the rendered page. Regular Oracle WebCenter Sites users can make quick edits in context, while infrequent users can accomplish their work without having to learn the Oracle WebCenter Sites interface.

1.5 Summary

It is important to understand that the components of a site are not site-specific; the associations among them are.

Site components are constructs that either exist or must be created in a system-wide pool. From this pool, you select the components that must interact with each other in order to create a site; you then associate the components with each other, and assign them a site name. Because the components are system-wide, they are re-usable—components that are used to create one site can be used to build other sites.

Note that to content providers, site components appear to be site-specific. However, any assets that content providers create on the site are indeed specific to that site. The assets can be copied to other sites, or shared among them.

The options of sharing and copying site components play an important role in the modeling and replication of online sites, as explained in [Section 1.6, "Content](#)

[Management Models.](#)"

1.6 Content Management Models

As the administrator, one of your biggest decisions, whether you make it alone or with collaborators, is how to model the online site in the Oracle WebCenter Sites interface: as a single site or a set of sites. Which content management model you choose depends largely on the size of your online site and the nature of its content.

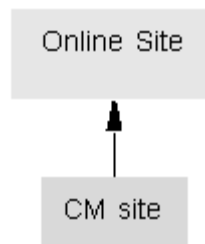
This section contains the following topics:

- [Section 1.6.1, "1:1 Model"](#)
- [Section 1.6.2, "1:n Model"](#)
- [Section 1.6.3, "x:n Model"](#)

1.6.1 1:1 Model

The 1:1 model maps the online site directly to a single CM site. The CM site is the sole source of the online site.

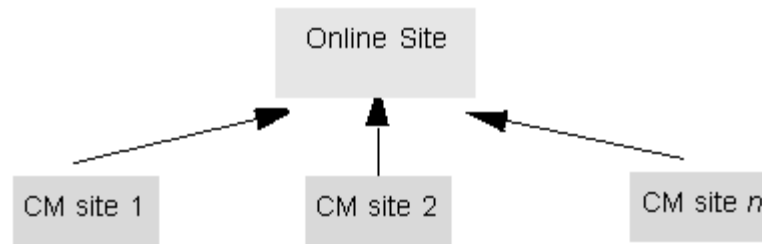
Figure 1–7 1:1 Model



The 1:1 model works efficiently for small online sites, where content tends to be limited, uniform, and managed by few users. For enterprise-level sites, however, pages often number in the millions and differ significantly in subject matter, content, presentation, and scope. Organizing large sites requires a model that can handle the complexities of the task.

1.6.2 1:n Model

The 1:n model maps the online site to multiple CM sites. Each section of the online site, whether logical or physical, maps to a certain CM site as its source. The CM sites can function independently of each other or overlap each other by sharing components and content.

Figure 1–8 1:n Model

For example, in a certain catalog site, content contributors who enter data about household goods never enter data about yard goods, so two separate sites are used to represent household goods and yard goods. Similarly, in a publication site where writers work independently of each other, the sports writers have a site that represents the sports news section, while the financial writers have a separate site that represents the financial news section. In each design, two sites contribute to one online site.

In a 1:n model, each site is unique; it has its own content types, templates, content providers, roles, workflow processes, and publishing mechanisms. Each site is the source of content for its corresponding online section.

When using a 1: n model, administrators have the option to configure sites to be independent of each other or to overlap each other by sharing components and content. An independent model works well when certain content needs to be segregated from other content. For example, the content is sensitive and must be handled only by certain content providers; or the content is so dissimilar that it needs to be handled by different specialists.

Overlapping sites, on the other hand, are used to support collaborations among remotely related content providers, such as those residing in different departments, or business units.

For example, consider the following scenario. A growing e-business specializing in mountain climbing disseminates mountain-climbing news and publishes a photograph album. It also sells gear and answers FAQs. The online site is clearly segmented into four sections: news, photos, gear, and FAQs. On many occasions, however, the news and photo sections share photographic content. In this scenario, one of your options is to create four sites: news, photos, gear, and FAQs such that news and photos share content. (Another one of your options is to create three independent sites: one for news and photos, one for gear, and one for FAQs.)

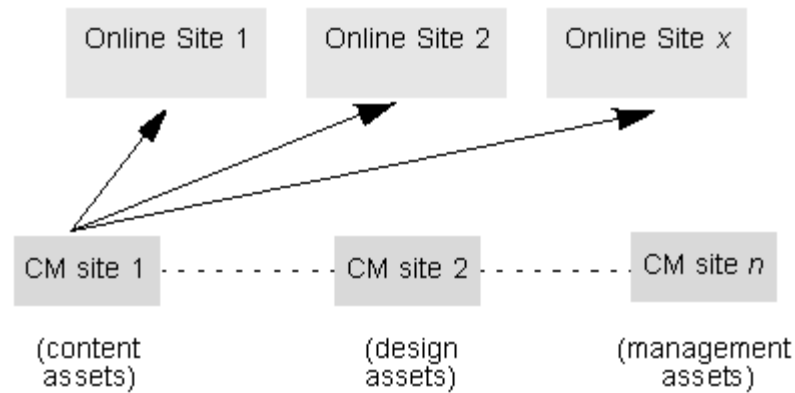
1.6.3 x:n Model

The x:n model maps multiple online sites to multiple CM sites. Of the three possibilities, this model is the most complex, but offers the greatest advantage to enterprise-level e-businesses.

In this model, content assets are localized on a given site, while the design and management assets are configured on separate sites from which they are shared to all other sites. This type of design allows the administrator to separate content from presentation and business logic.

Note: To help administrators and other users understand sites and their relation to the online site, Oracle WebCenter Sites provides sample sites named avisports and FirstSiteII. Each sample site maps 1:1 to its online site. If the sample sites are installed, you can log in to the sites by following the login procedure in [Chapter 2, "Administrator's Interface"](#).

Figure 1–9 *x:n Model*

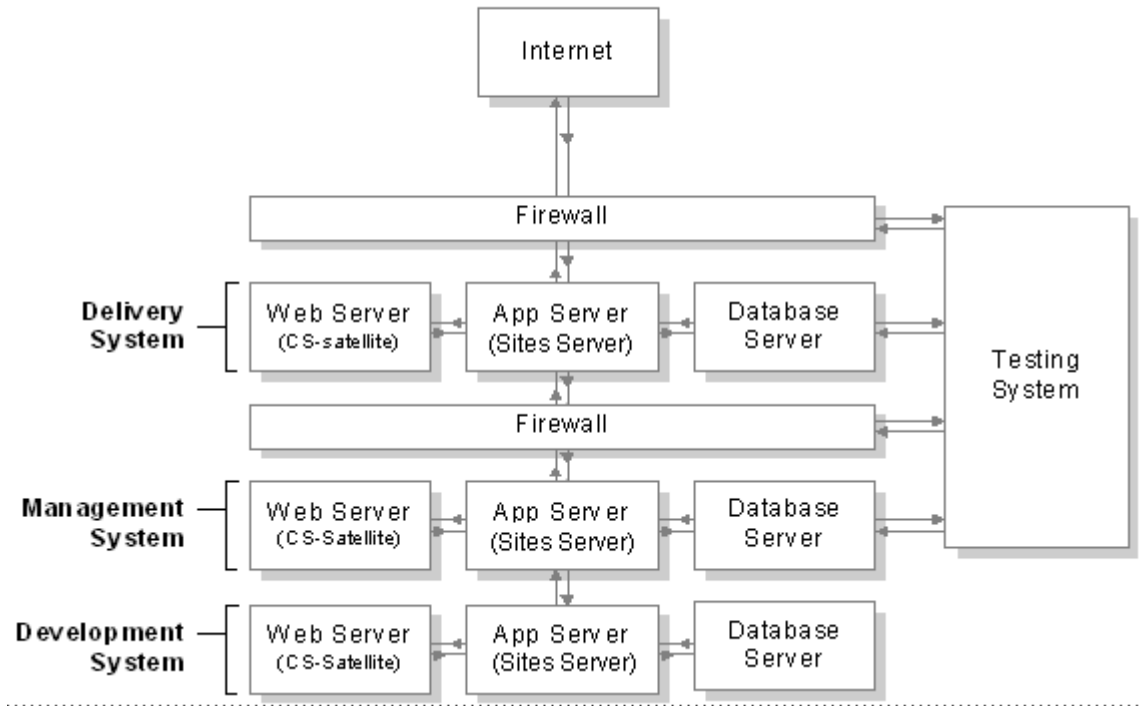


1.7 Oracle WebCenter Sites Environment

Site configuration is a collaborative process that takes place in a Oracle WebCenter Sites environment rather than an isolated Oracle WebCenter Sites system. An enterprise-level environment typically consists of four different WebCenter Sites-powered systems: development, management, delivery, and testing, as [Figure 1–10, "Oracle WebCenter Sites Environment"](#) shows. Each system runs on its own database and at some point interacts with the other systems. One of the systems, testing, is an optional system that large organizations typically install for increased quality assurance.

As the administrator, you will work with all of the systems at one time or another. Your collaborators are developers, who work exclusively at the development system with you and with users whom you appoint as collaborators. This section summarizes the possible systems and your function in each of them.

Figure 1–10 Oracle WebCenter Sites Environment



Note: The names of the systems in your WebCenter Sites environment might vary from the names used in this guide. Generally, the management system is also called staging; the delivery system is also called production; and the testing system is also called QA or QA testing.

- The development system is responsible for planning and creating the framework of the online site: the data model that will be used by the CM sites and the data presentation templates.
- The management system is the staging area, responsible for configuring site components (other than the data model), assembling the components (including the data model) into sites, making the sites available to users, and managing users' day-to-day activities on the sites. The users' responsibility is to develop and manage content for delivery to the online site.
- The delivery system is the production area, which receives content for the online site from the management system and serves it to the target audience.
- The testing system is where QA tests both the management and delivery systems, and the online site itself before its launch. When the testing system is absent, the development system doubles as the testing system.

The rest of this chapter describes your collaborators, your planning strategies, and your functions at each of the systems.

1.8 Administrator's Job

This section contains the following topics:

- [Section 1.8.1, "Collaborating"](#)
- [Section 1.8.2, "Planning"](#)
- [Section 1.8.3, "Development"](#)
- [Section 1.8.4, "Implementation"](#)

1.8.1 Collaborating

Before development begins you typically collaborate with a team comprising many different specialists:

- Site designers
- XML and JSP developers
- Java application developers
- Database administrators
- System network administrators
- Marketers and advertising staff
- Business managers
- Product managers, if you are developing a commerce site
- Content providers

The job of the team is to establish functional requirements and design specifications for the management and delivery systems:

- Page design
- Caching strategy
- Security strategy
- Format vs. content
- Data model
- Content management model

Much of the preceding information is outlined in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. The information presented in this guide is in the context of the administrator's job.

1.8.2 Planning

Planning is critical, as the decisions that you and your collaborators make will determine how developers will code system security, the data model, and the delivery system. Your job, in particular, concerns the management system, as described below.

During the planning stage, your job is to establish with collaborators (mostly business managers) the requirements and design of the management system, and relay the information to developers. Items to consider are listed below. The list is not meant to be exhaustive, but to help you start gathering and organizing information that is most critical to a successful start of the development cycle.

- Content management model
 - Determine which of the models (1:1, $n:1$, and $x:n$) is most appropriate for your operation

- Determine which content assets must be shared and which must copied. Do the same for template assets.
- Site replication

Determine whether you will be replicating sites. If so, you must configure the source sites to conform to the requirements that are set by the replication process.

Developers must know your decision in order to code correctly. For example, if sites are to be replicated, the names of templates that will be copied to the new sites must not be hard-coded.
- Data modeling
 - What types of assets should be created
 - Which asset types should use revision tracking
 - Which users must have access to which asset types? on which sites?
 - What types of templates need to be created by developers
- User management methods

In addition to its native user manager, Oracle WebCenter Sites supports LDAP plug-ins, both hierarchical and flat-schema. Determine which method is appropriate for your environment. (Procedures in this guide use Oracle WebCenter Sites' native user manager system throughout.)
- User management models

Determine the following:

 - How many users are needed
 - Who the users are
 - What permissions the users must be given to the database tables:
 - Which system ACLs the users must be assigned (without ACLs, no user accounts can be created).
 - Whether custom ACLs need to be created.

(Note that you may also need to create ACLs to assign to visitors of the online site in order to restrict them from accessing certain database tables.)
 - How many roles are needed
 - What types of roles are needed (writers, editors, illustrators). Make sure that the roles are compatible with the users' ACLs.
 - To which users the roles should be assigned.
- Workflow processes

Workflow processes can be optionally attached to asset types to ensure that newly created assets of those types are automatically engaged in workflow.

 - Should workflow be implemented
 - If so, which asset types need workflow processes
 - How must the workflow processes be designed
 - Which users and roles need to participate in the workflow processes
- Publishing system options
 - Which type of publishing needs to be configured

- Whether publishing schedules need to be approved
- User interface options
 - Determine which interfaces the users will need:
 - Contributor (Web Mode and Form Mode)
 - Admin
 - WEM Admin
 - Sites Desktop
 - DocLink

Use both this book and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* to help you make these decisions.

1.8.3 Development

When the specifications of the online and sites are established, you are ready to begin your work. As the Oracle WebCenter Sites administrator, you will be working on all systems in the WebCenter Sites environment during the development stage. (Once the online site is running, however, you will spend most of your time at the management system, a smaller fraction at the delivery system, and perhaps none at all at the development system.)

At each system, you have a specific set of jobs. While developers are responsible for the data model and other code-based components, you are responsible for all other aspects.

This section contains the following topics:

- [Section 1.8.3.1, "At the Development System"](#)
- [Section 1.8.3.2, "At the Management System"](#)
- [Section 1.8.3.3, "At the Delivery System"](#)
- [Section 1.8.3.4, "At the Testing System"](#)

1.8.3.1 At the Development System

The development system is where coding takes place.

- Your job is to provide developers with the specifications of the management system and to assist with content management operations, such as creating sites.
- The developers' job is to:
 - Create sites with the same names as those that will be used on the management system
 - Code the data model (bearing in mind site-replication requirements:
 - Content asset types
 - Design asset types
 - Management asset types
 - Create sample assets of each type.
 - Test the data model.
 - Code templates and, in general, the framework of the online site.

Once the data model is complete and tested to the satisfaction of all collaborators, developers migrate the code to the management system

Note: In your WebCenter Sites environment, developers might choose to not migrate the data model, but to re-create the data model on the management system.

Note that once development is complete, the development system continues to operate. One of its ongoing functions is to revise the data model in response to the evolving needs of online site visitors, content providers, and administrators.

1.8.3.2 At the Management System

Configuration tasks that require no coding are typically completed at the management system. Here, your administrative work is not collaborative unless you choose to make it so. For example, because no data modeling or coding takes place at the management system, you have no need for assistance from developers. However, you can appoint developers (or other users) to be the site or workflow administrators, to manage specific sites or workflow processes for the system

Note: On occasion, you will need help from developers; for example, you might want to add custom functionality to the tree tabs in Oracle WebCenter Sites' interface.

At the management system, your job is the following:

- Create the sites

Note: In this release of Oracle WebCenter Sites, you can quickly spin off new sites by using the Site Launcher feature to replicate source sites. You can then modify the replicates as necessary. For information about site replication and selecting or creating the appropriate source sites, see [Chapter 11, "Replicating Content Management Sites."](#)

- Configure the users
 - Create their ACLs in Oracle WebCenter Sites, as necessary
 - Create user accounts, either through Oracle WebCenter Sites, or external user managers, such as LDAP plug-ins
 - Create roles in Oracle WebCenter Sites, as necessary
 - Assign users their roles for each site
- Assemble the sites by
 - Associating the users with the sites
 - Associating the correct asset types and publishing system with the sites
- Enable the users by
 - Assigning users their roles for each site
 - Using roles to associate users and asset types with start menu items
 - Using roles to associate users with interface functions such as tree tabs

- Creating tree tabs, as necessary
- Create and manage workflows
- Enable revision tracking
- Configure site and workflow administrators, as necessary
- Configure the publishing process
- Configure the users' interfaces, as necessary

1.8.3.3 At the Delivery System

The delivery system is configured by developers and typically does not involve an administrator until the system is ready for use. For Mirror to Server and RealTime delivery, the delivery system is the entire Oracle WebCenter Sites installation. For Export to Disk and Export to XML delivery, the delivery system is simply the web server component of the Oracle WebCenter Sites installation.

1.8.3.4 At the Testing System

The testing system is where QA analysts test the performance of both the management system and the delivery system, as well as the online site itself before its launch. If the testing system is absent, the development system doubles as the testing system. Ordinarily, you are not involved in testing.

1.8.4 Implementation

When the management and delivery systems are in use and the online site is running, you work almost exclusively at the management system, where you:

- Manage the users
 - Adjust their permissions to database tables
 - Add, delete, and re-organize users
 - Create workflows
 - Configure the users' interfaces
- Manage the publishing system
- And otherwise respond to the needs of online site visitors, content providers, and the e-business

1.9 The Focus of This Guide

This guide focuses on the configuration tasks that you complete at the management system during the development stage, as described in [Section 1.8.3, "Development."](#) It is assumed that the data model resides on the management system by the time you begin your configuration tasks.

1.10 Administrator's Prerequisites

To function in the jobs that were described in the previous section, you must be equipped with a certain amount of information. This section summarizes the information you need to know or be familiar with.

- Technical aspects of content management

Although you are not required to have a specific technical knowledge of programming and system architecture, a basic knowledge of the following is desirable:

- Java
- JSPs
- Publishing
- Databases and database management
- Browsers
- Your collaborators
For detailed information, see [Section 1.8.1, "Collaborating"](#) and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
- Online site specifications
Online site specifications are typically prepared by site designers or the corporate communications department to define the site's audience, the site's size, its message and goals, the nature of the content, and the presentation style. Having an understanding of the online site will help you model the online site, plan sites, and collaborate efficiently with developers.
- The Oracle WebCenter Sites environment
As the Oracle WebCenter Sites administrator, you can work with up to four different systems: development, management, delivery, and testing. Make sure you have a good understanding of the systems and your function at each of them. For detailed information, refer to the sections [Section 1.7, "Oracle WebCenter Sites Environment,"](#) [Section 1.8, "Administrator's Job,"](#) and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
- Oracle WebCenter Sites' administrative interface and system defaults
The administrative interface provides you with site configuration tools, and access to system defaults such as ACLs, roles, and asset types. For information about the administrator interface, see [Chapter 2, "Administrator's Interface."](#) For information about system defaults, see [Chapter 31, "System Defaults"](#) and [Chapter 32, "System Data: WebCenter Sites Database."](#)

Administrator's Interface

Oracle WebCenter Sites defines three kinds of administrators, each with a different range of administrative rights. This chapter summarizes the administrators and shows you how to log in to Oracle WebCenter Sites as the general administrator. It describes the administrator's interface, its tabs, the nodes within the tabs that allow you to manage sites, and important system defaults that are accessible either from the interface or the back end.

This chapter contains the following sections:

- [Section 2.1, "Types of WebCenter Sites Administrators"](#)
- [Section 2.2, "Logging in to the Administrator's Interface"](#)
- [Section 2.3, "Administrator Tabs"](#)
- [Section 2.4, "Non-Administrative Tabs"](#)
- [Section 2.5, "System Defaults"](#)
- [Section 2.6, "Sample Sites"](#)

2.1 Types of WebCenter Sites Administrators

Oracle WebCenter Sites defines three types of administrators:

- The general administrator, for whom this guide is written.
The general administrator typically manages all systems in the WebCenter Sites environment and therefore has unrestricted access to each system's interfaces. The jobs of the general administrator at each system are outlined in [Chapter 1, "Overview."](#)
- Site administrators, who administrate only certain sites on the management system.
- Workflow administrators, who create workflow processes on the management system

Each administrator is given permissions to relevant parts of the WebCenter Sites interface.

Procedures in this guide are specific to the general administrator. Whenever steps can be executed by the site or workflow administrators, this guide indicates which steps.

Finally, this guide is not restricted to any particular system in the WebCenter Sites environment. The concepts and procedures it contains are valid at all systems in the WebCenter Sites environment.

Note: As this guide is written for the general administrator, the word "you," whenever it is used, refers strictly to the general administrator.

2.2 Logging in to the Administrator's Interface

This section shows you how to log in to Oracle WebCenter Sites as the general administrator.

Note: If you are not the general administrator and need to log in as a site administrator or workflow administrator, obtain your user name and password from the general administrator before following the steps below. Also bear in mind that because your permissions to the Oracle WebCenter Sites interfaces are limited by your role, only certain parts of this guide are relevant to you.

To log in to Oracle WebCenter Sites as the general administrator

1. Open your browser.
2. Set your browser so that it checks for newer versions of stored pages with every visit to the page.
3. Navigate to the following URL:

`http://<server>:<port>/<context>/login`

where <server> is the host name or IP address of the machine running Oracle WebCenter Sites, <port> is the port number for the server, and <context> is the context path of the Oracle WebCenter Sites web application. Depending on how the system was set up, you might also need to include the port number. The system can be development, management, delivery, testing, or another system in your WebCenter Sites environment.

Oracle WebCenter Sites displays the following login dialog:

Figure 2–1 Login Dialog

The screenshot shows the Oracle WebCenter Sites login interface. At the top, there is a dark header bar with "ORACLE WebCenter Sites" on the left and "Version: 11gR1" on the right. Below the header, the main content area has a light blue background. On the left side, the "ORACLE" logo is displayed. To the right of the logo, the text "Secure User Login" is centered. Below this text, there are two input fields: "Username" and "Password". To the right of the "Password" field, there is a link that says "Forgot password?". Below the input fields, there is a prominent "Login" button. At the bottom right, there is a checkbox labeled "Remember me".

4. Enter your user name and password

Note: If you are the general administrator of a new installation that had no previous administrator, log in as the default system user, with user name fwadmin and password xceladmin and then immediately change the password. For instructions on changing your password, follow the procedure in [Section 5.4.1, "Creating a New User."](#)

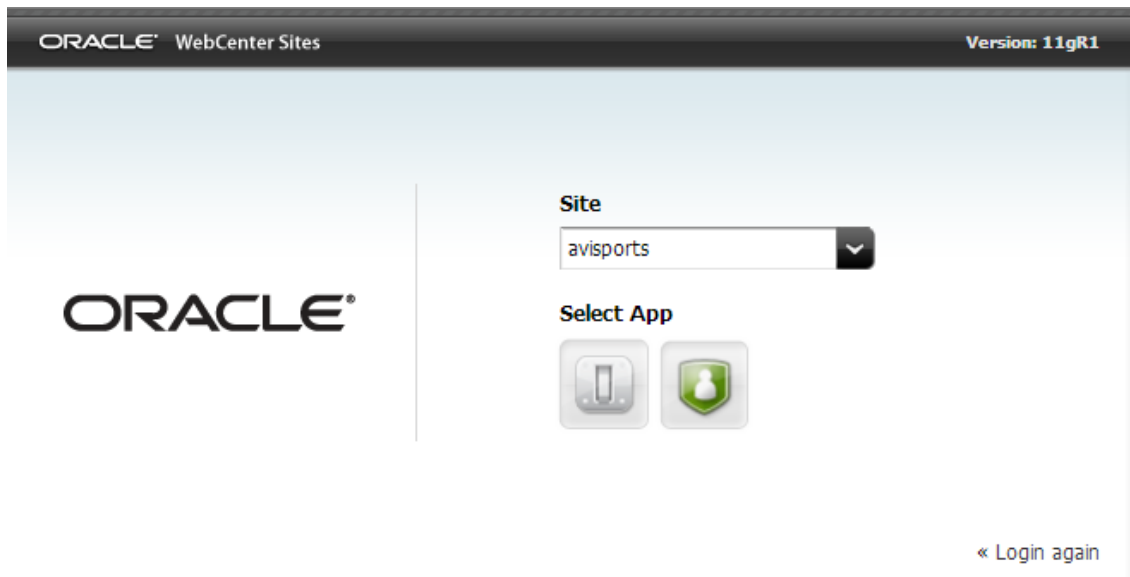
Because you have logged on as the default administrator, your user name has the xceladmin ACL (Access Control List) assigned to it, in addition to being a part of the REST_Admin security group. See [Chapter 5, "Configuring Users, Profiles, and Attributes"](#) for information about user accounts and their ACLs.

If you create any other general administrators, they must have the xceladmin ACL assigned to them, and they must be part of the REST_Admin security group.

5. Click **Login**.

6. WebCenter Sites responds according to the number of sites that have been created in its interface. To continue, complete one of the following steps:
- If no sites have been created, you must log in to the AdminSite, which will be listed in the drop-down list. (Note that the situation of no sites is rare. Most new installations run the sample sites that are provided with Oracle WebCenter Sites.) Continue with step 7.
 - If one site has been created, you still need to select the site from the list to log in to that site. Continue with step 7.
 - If two or more sites have been created, they are listed in your WebCenter Sites interface, along with the sample sites that your installation engineer has elected to install. Select the site that you want to work on. Continue with step 7.
7. Once a site is selected, you will see two icons appear. The left icon is to log in to the Admin interface, and the right icon is to log in to the Contributor interface.

Figure 2–2 Login Dialog

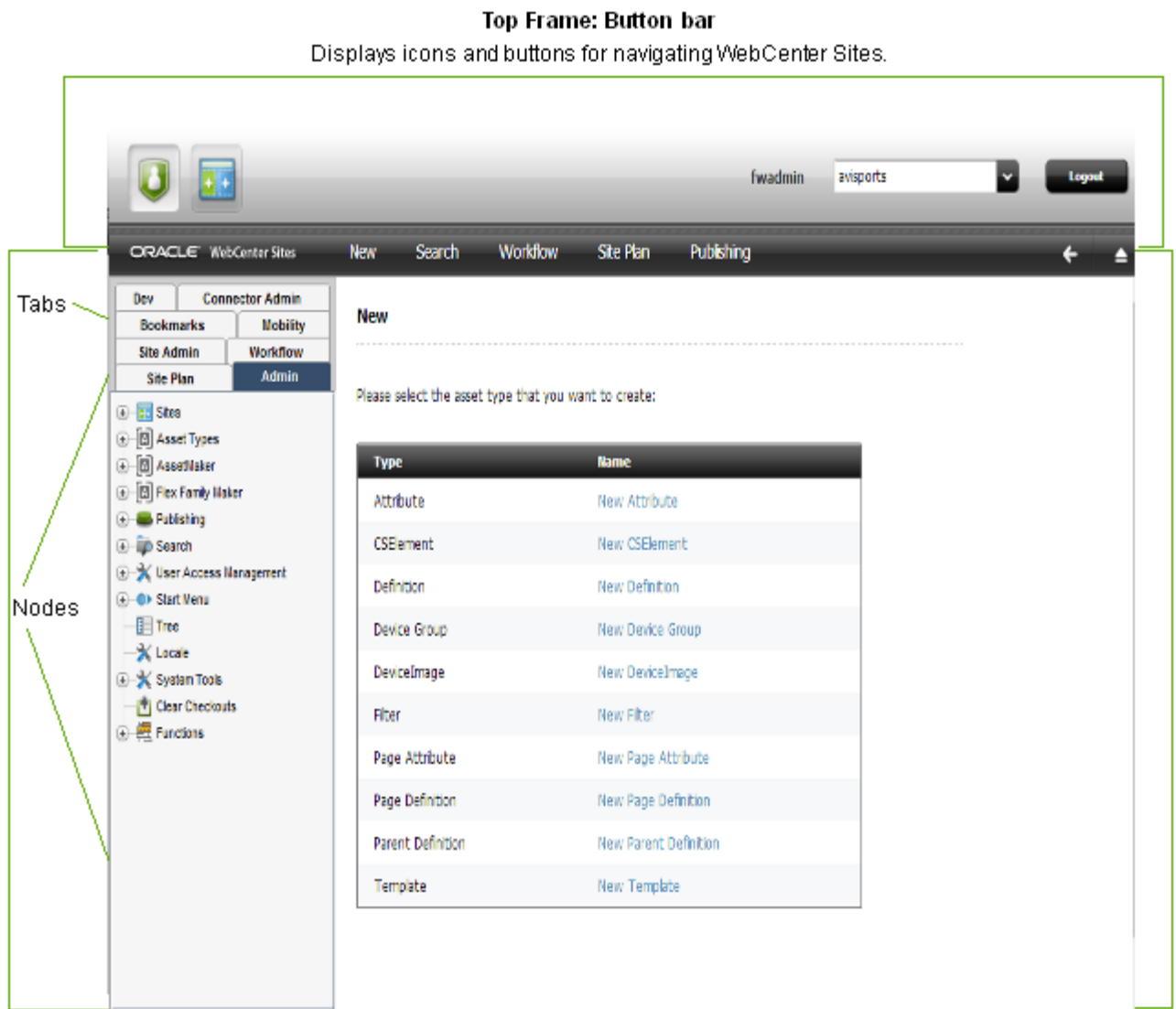


Click to enter the Admin interface.

If you select AdminSite, the icons will be slightly different.

8. At this point, you are logged in to the site of your choice. All the tasks that you complete pertain to the site, until you switch to a different site. The one exception is the set of administrative tasks that you execute from the Admin tab, which is not site specific. The Admin tab contains data for all sites in the system.
Within your selected site, Oracle WebCenter Sites displays the administrator's interface, consisting of the three frames shown in [Figure 2–3](#):

Figure 2–3 Window Components



Left Frame: WebCenter Sites Tree

The WebCenter Sites tree is a collection of tabs with nodes. Each node is a building block of the site.

Right Frame: CM Site-Building Form

This frame displays the forms that WebCenter Sites calls when you select nodes from the left frame. Each form is used to enter data into the selected node and manage the data.

Note: At this point, the first time you log in after selecting a site, you will be prompted to create a new asset type, with the New form displaying.

If you are a first time WebCenter Sites user, we recommend that you familiarize yourself with the following features:

- The administrator's interface—the Admin tab in particular. You will be using the Admin tab almost exclusively throughout this guide to create, configure, and manage sites. For more information, see [Section 2.3, "Administrator Tabs."](#)

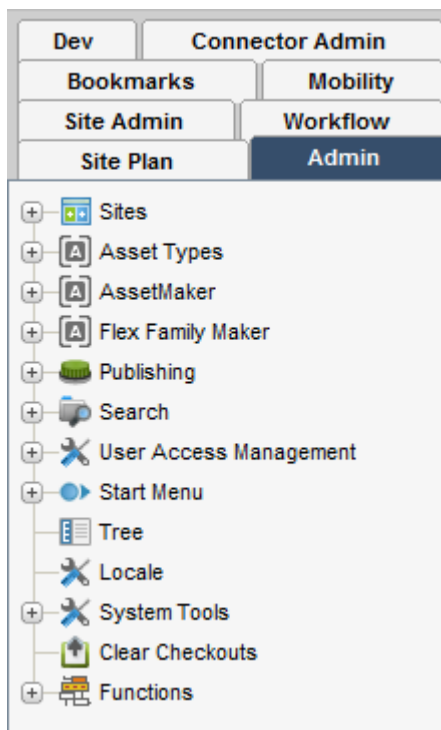
- System defaults, such as ACLs, which you will need in order to configure users, their access rights, and so on. For more information about system defaults, see [Chapter 31, "System Defaults."](#)

2.3 Administrator Tabs

Up to three administrator-specific tabs can be displayed in Oracle WebCenter Sites' tree:

- Admin
- Site Admin
- Workflow

Figure 2–4 Tabs



The tabs are permissions-based. To be exposed to the user, they require the user to have the ACLs and roles shown below:

Table 2–1 Tab Permissions and Roles

Tab	ACL	Role
Admin	xceladmin	GeneralAdmin
Site Admin	xceladmin	SiteAdmin
Workflow	xceladmin	WorkflowAdmin

The ACLs and Roles are covered in further depth in [Chapter 5, "Configuring Users, Profiles, and Attributes,"](#) and [Chapter 31, "System Defaults."](#)

The tabs are also conditional on the type of site in which you are working:

- If you log in to a Oracle WebCenter Sites installation where no sites have been created, you are required to log in to the AdminSite site. Here, you see only the Admin, Site Admin, and Workflow tabs.
- If you are logged in to a custom site, both the Admin and the Site Admin tabs are displayed in your interface. The Admin tab remains system-wide. The SiteAdmin tab is specific to the site you are logged in to. It is therefore a sub-set of the Admin tab.
- If workflow processes have been created for any site (except the default site), the Workflow tab is also displayed in your interface. The Workflow tab lists all workflow processes that have been configured for all sites in the system.

Whether administrator tabs are exposed to other users depends on whether you assign the users the associated ACLs and roles. For more information about ACLs and roles, see [Chapter 4, "Working with ACLs and Roles."](#) For more information about administering Mobility (for mobile devices), see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*

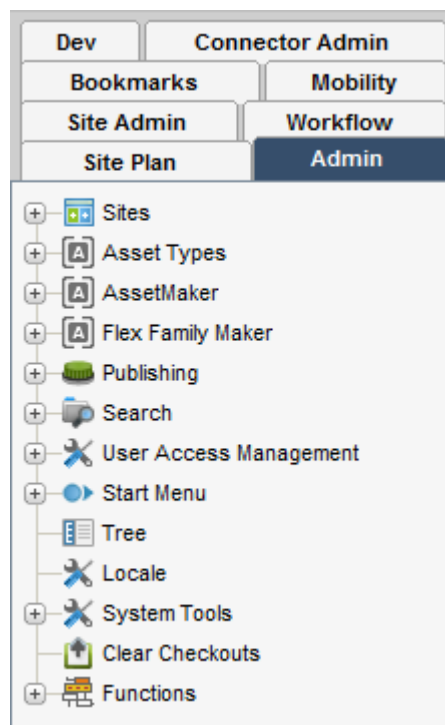
This section contains the following topics:

- [Section 2.3.1, "Admin Tab"](#)
- [Section 2.3.2, "Site Admin Tab"](#)
- [Section 2.3.3, "Workflow Tab"](#)

2.3.1 Admin Tab

The Admin tab is the main administrative tab, used to create, configure, and manage sites. The Admin tab accounts for all content management components in the Oracle WebCenter Sites installation.

Figure 2–5 Nodes in Admin Tab



The Admin tab displays the following nodes:

- **Sites**, that is, CM sites (described in [Chapter 1, "Overview"](#)).

The **Sites** node has the following functions:

- Lists sites that have been created in the system.
- Enables you to define sites and associate to them the components from the system-wide configuration pool, below the Sites node. Oracle WebCenter Sites provides several sample sites as learning tools.

The configuration pool consists of the following nodes:

- **Asset Types**, for configuring asset types:
 - To add subtypes or asset associations for basic asset types.
 - To create start menu items for asset types that appear on the **New** and **Search** menus.
 - To enable or disable revision tracking for specific asset types. See [Chapter 26, "Revision Tracking"](#) for information.
 - To enable assets for Sites Desktop. For information, see [Chapter 23, "Configuring the User Interfaces"](#).
 - To enable assets for Sites DocLink. For information, see [Chapter 23, "Configuring the User Interfaces"](#).
- **AssetMaker**, for site developers to create asset types using the basic asset data model. See the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information.
- **Flex Family Maker**, for site developers to create new flex families and flex asset types. See the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information.
- **Publishing**, for setting up the publishing system on the development and management systems. See [Chapter 12, "Publishing with Oracle WebCenter Sites"](#) for more information on publishing in general. See [Chapter 15, "The Export to Disk Publishing Process"](#) and [Chapter 16, "The Mirror to Server Publishing Process"](#) for more information on static and dynamic publishing. See [Chapter 19, "RealTime Publishing Overview"](#) for information about interactive dynamic publishing.
- **Search**, for configuring asset types so that they are indexed correctly when you are using one of the supported third-party search engine modules. See [Chapter 25, "Configuring the Lucene Search Engine"](#) for information.
- **User Access Management**, which you use to create ACLs and users, to enable or disable revision tracking for non-asset tables, and to perform various other operations on WebCenter Sites' database. This node includes the following:
 - **Users**, which you use to manage WebCenter Sites users and the roles associated with each user ID. See [Chapter 5, "Configuring Users, Profiles, and Attributes"](#) for more information.
 - **Roles**, which you use both for determining the recipients of workflow assignments and to control users' access to functions in the Oracle WebCenter Sites interfaces. For more information, see [Chapter 4, "Working with ACLs and Roles."](#)
 - **ACLs**, which are used for allowing access to individual tables and sites. For more information, see [Chapter 4, "Working with ACLs and Roles."](#)

- **User Profiles**, which you use to specify email addresses for your users. See [Chapter 5, "Configuring Users, Profiles, and Attributes"](#) for more information.
- **REST Security**, for configuring the security groups and assigning users to those groups. Adding someone to the security groups is tied to the WebCenter Sites interface. See [Chapter 6, "Setting Up External Security"](#) and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information.
- **Start Menu**, which you use to create links in the form that is displayed when users click the **New** button or **Search** button in the Oracle WebCenter Sites interface. You use the links to create new assets or search for existing assets. Start Menu items can also be created for managing access to the Sites Desktop and Sites DocLink interfaces. See [Chapter 9, "Managing Access to Content Management Site Components"](#) for information.

Note: This release of WebCenter Sites supports automatic creation of the New and Search Start Menu items.

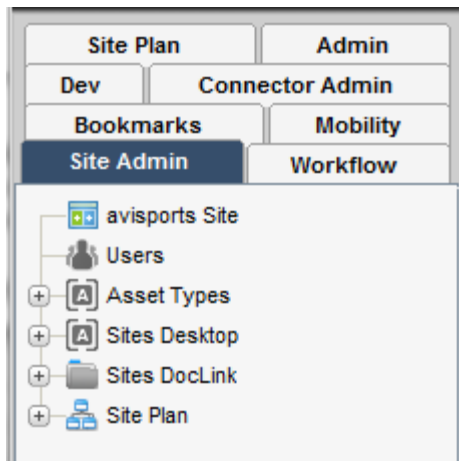
- **Tree**, which you use to control access to the existing tree tabs, create new ones, or view assets in the tree in the Contributor interface. See [Chapter 9, "Managing Access to Content Management Site Components"](#) for information.
- **Locale**, which you use to set the default locale for the WebCenter Sites system. See [Chapter 23, "Configuring the User Interfaces"](#) for information.
- **System Tools**, which you use to troubleshoot problems directly from the administrator's interface. Features include configuring log4j logging, accessing various types of system information, managing caches, searching logs, and testing the performance of the shared file system. See [Chapter 30, "System Tools"](#) for information.
- **Clear Checkouts**, which you use to unlock and check back in to the database the assets that are checked out by specific users. See [Chapter 26, "Revision Tracking"](#) for information.
- **Functions**, which administrators can enable or disable in the Oracle WebCenter Sites interfaces of all users or selected users.

2.3.2 Site Admin Tab

The Site Admin tab is site-specific. It is used to manage the site that you are logged in to.

The Site Admin tab holds a subset of the administrative functions in the Admin tab. It allows you to manage existing users of the site, as well as enable or disable asset types. It does not allow you to create users, asset types, or sites.

Figure 2–6 Nodes in Site Admin Tab

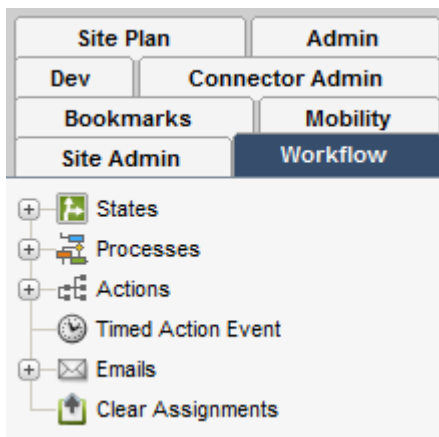


For any other user to access this tab, the user must be assigned the xceladmin ACL and the SiteAdmin role for the site(s) that the user logs in to.

2.3.3 Workflow Tab

The Workflow tab is used to configure workflow processes from the workflow building blocks that were created using the Admin tab.

Figure 2–7 Nodes in Workflow Tab



Note: While the Workflow tab itself is site-specific, its content is not.

The tab. When a user selects a site, the Workflow tab is displayed if the user is assigned the xceladmin ACL and WorkflowAdmin role *for that site*. In this respect, the Workflow tab is site-specific.

The tab's content. The Workflow tab displays system-wide information, that is, all workflow processes for all sites in the installation. Therefore, the content under the nodes in the Workflow tab is not site-specific.

- **States**, which you use to create points in the workflow process that represent asset status within the workflow. See

- **Actions**, which you use to create the workflow building blocks called actions and conditions. These items can be used by any workflow process.
- **Timed Action Event**, which you use to configure the frequency with which due dates are calculated for assets that are in workflow.
- **Email**, which you use to create the workflow email messages that can be used by any workflow process.
- **Clear Assignments**, which you use to clear workflow assignments.

For more information on workflow, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

2.4 Non-Administrative Tabs

Your interface displays default non-administrative tabs, and custom tabs if they have been configured by another administrator. You can create additional tabs as necessary.

For information about the default tabs, see [Section 31.7, "Default Tree Tabs."](#) For information about creating and configuring tabs, see [Section 9.5, "Managing Access to the Tree \(Administrator's interface only\)."](#)

2.5 System Defaults

WebCenter Sites has a number of system defaults:

- Non-administrative tabs, described in the previous section
- ACLs
- Roles
- Users
- Asset types
- System Tables

You must be familiar with the defaults in order to configure users and sites. Information about system defaults is given in [Chapter 31, "System Defaults"](#) and [Chapter 32, "System Data: WebCenter Sites Database."](#)

2.6 Sample Sites

To help you better understand WebCenter Sites, we have packaged it with sample sites named avisports and FirstSiteII. You can use the sites to learn about and experiment with WebCenter Sites without affecting your own sites. The sample sites vary in complexity and purpose. In addition to providing code samples for developers, they exemplify how sites are configured.

If your installation engineer has elected to install the sample sites, they will be displayed in your interface.

To work on the sample sites, follow the steps in [Section 2.2, "Logging in to the Administrator's Interface."](#) You can access the corresponding online sites by pointing your browser to each site's URL, available from your installation engineer.

Site Configuration Guidelines

Site configuration in WebCenter Sites is based on a relational system, where components in a system-wide configuration pool must be associated with each other and with a site name in order to function as a site. Making the correct associations creates a functional site—one that hosts the correct users, provides them with the correct asset types for authoring and managing content, supports workflow as necessary, publishes correctly, and delivers approved content to the intended audiences.

This chapter contains procedures that show first-time administrators how to configure sites. It also contains recommendations for the more experienced administrators to help them manage and maintain established sites.

This chapter contains the following sections:

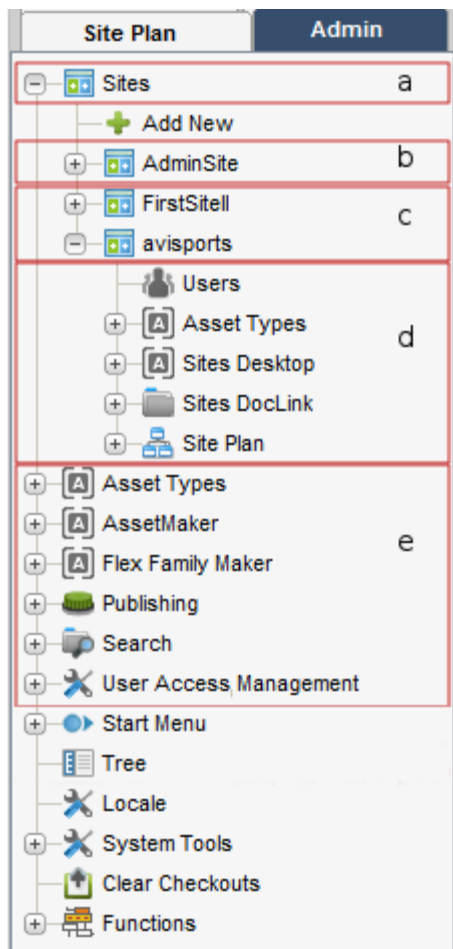
- [Section 3.1, "Overview"](#)
- [Section 3.2, "Site Replication Option"](#)
- [Section 3.3, "Site Configuration Steps"](#)

3.1 Overview

When creating and configuring sites, you will be using the Admin tab, shown in [Figure 3-1](#). The tab is logically separated into two sections. The first section is the Sites node, where sites are first established; the second is the system-wide configuration pool, where site components are created, associated to each other, and managed. The configuration pool begins with the AssetMaker node and ends with User Access Management.

Initially, when a new site is defined and added to the Sites node, it is empty and non-functional, because it is not associated in any way with the configuration pool. You make the site functional by associating with it the various components in the configuration pool. The components are sharable among sites. Any component that you associate with one site, you can associate with as many other sites as necessary. More information about the Sites node and the configuration pool is given on the next page.

Figure 3–1 Nodes in Admin Tab, Described



In [Figure 3–1](#), the marked areas are described in this way:

- Section a highlights the Sites node.
- Section b highlights the default site.
- Section c highlights the custom sites. These sites are created by the administrator. In this illustration, there are only two custom sites, which are samples provided with WebCenter Sites.
- Section d highlights the default nodes. There is one set per site. Each default node must be configured by the administrator.
- Section e highlights the system-wide configuration pool. Components in the configuration pool must be selectively associated with each other and with the sites (as listed in section b) in order to make the sites functional.

Sites are configured in four basic steps from the Admin tab, as shown below. We emphasize basic to indicate that the steps are intended to provide only an overview of the site configuration process. Detailed procedures are given in the rest of this guide.

Note: In the procedure below, steps 1 and 2 are interchangeable.

Table 3–1 Sites Procedures

Step	Description
<p>Add sites to WebCenter Sites:</p> <ol style="list-style-type: none"> 1. In the Admin tab, expand Sites and double-click Add New. 2. Enter the site name and description. 3. Select a preview method by which you will preview the site's content (when the content is created). 	<p>The Sites node (rather than the configuration pool) can be the starting point for creating sites.</p> <p>When you execute step 1, WebCenter Sites adds the site to the Sites node. The Contributor and Admin interfaces are automatically assigned to each newly added site.</p> <p>Under the newly added site, WebCenter Sites creates a set of sub-nodes that prompt you for site components:</p> <p>The Users sub-node prompts you for users.</p> <p>The Asset Types sub-node prompts you for asset types (created by developers).</p> <p>The Sites Desktop sub-node prompts you to enable Sites Desktop.</p> <p>The Sites DocLink sub-node prompts you to enable Sites DocLink.</p> <p>Each sub-node is equipped with a linking mechanism that allows you to associate the sub-node, and therefore the site, with data in the configuration pool. Because of this linking mechanism, the sub-nodes cannot be deleted or supplemented with custom sub-nodes. All data required by the sub-nodes must be entered in to the configuration pool, described next.</p>
<p>Populate the configuration pool with components, such as users, asset types, and roles.</p>	<p>The configuration pool begins with the AssetMaker node and ends with User Access Management, as Figure 3–1 shows.</p> <p>The configuration pool enables you to select and create components for the site and to associate the components with each other. Data that you establish in the configuration pool remains independent of any sites (and therefore inaccessible to users) until you associate the data with the sites.</p>
<p>Associate and enable components in the configuration pool with the site:</p> <ol style="list-style-type: none"> 1. In the Admin tab, expand Sites and expand the desired site node. 2. Add users and asset types to the site. 3. Enable the users and asset types by means of start menu items, access permissions, and permissions to tree tabs. 	<p>The Sites node is the endpoint of the site building process. In this step, you once again use the Sites node, this time to associate components in the configuration pool to the site that you added in step 1.</p> <p>Note that components in the configuration pool are reusable; they can be associated with as many sites as necessary.</p>

3.2 Site Replication Option

If you plan to create many sites that are similar in content and structure, you can shorten the site creation process. You can configure the appropriate source sites, then replicate the sites as often as necessary, and finally modify the replicates as necessary. The source sites must conform to certain replication requirements. For more information, see [Chapter 11, "Replicating Content Management Sites."](#)

3.3 Site Configuration Steps

In the site configuration process, certain components in the configuration pool are required by other components and must be created first if re-tracing of steps is to be minimized. For example, before users can be created, their ACLs must first exist.

In general, the order of steps in the site configuration process is flexible. One possible order is given in the procedure below (starting with [Section 3.3.1, "Assumptions"](#)), where the critical site components are created first, then assembled into a site, and finally enabled for use; the optional components (with the exception of workflows) are considered last, after the publishing process is configured.

If you are a first-time administrator, we suggest that you familiarize yourself with the procedure below in order to obtain a basic understanding of the configuration process and gather all the information that you will need. Once you understand how sites are configured, you can modify the procedure to suit your needs.

We also suggest that you make a copy of the procedure below and keep it handy when configuring sites. Each step refers you to sections of this guide (or the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*) where you can find detailed information about the particular step.

This section contains the following topics:

- [Section 3.3.1, "Assumptions"](#)
- [Section 3.3.2, "Pre-Configuration Decisions"](#)
- [Section 3.3.3, "Configuration Steps at the Development System"](#)
- [Section 3.3.4, "Configuration Steps at the Management System"](#)

3.3.1 Assumptions

We assume that you and developers will code the data model at the development system, and after successful testing, migrate the data model to the management system. At the management system, you will complete the site configuration tasks, test the management system, and finally deploy the system. Before beginning the configuration tasks, read [Chapter 1, "Overview"](#) for a general understanding of WebCenter Sites concepts, components, and content management models.

3.3.2 Pre-Configuration Decisions

The decisions to make before you start configuring are best handled in a certain order.

Determine the user management method

Your options are WebCenter Sites' native user manager, an LDAP plug-in, or an external manager.

While this guide is based on the native WebCenter Sites user manager, it does describe system behavior for LDAP-integrated systems. For information, see [Chapter 33, "Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems."](#) For information about configuring users in LDAP, refer to the LDAP product documentation.

Determine users' table-level permissions (ACLs)

Regardless of which user management method you will be implementing, you must have ACLs defined in WebCenter Sites. Examine the list of system ACLs (and custom ACLs, if any have been created) to determine which ACLs to assign to users, and whether you need to create additional ACLs. Typically, the system ACLs are sufficient.

For general information about ACLs, see [Chapter 4, "Working with ACLs and Roles."](#) For specifications, see [Chapter 31, "System Defaults."](#)

Determine the users' roles

WebCenter Sites defines several system roles and additional roles for the sample sites. You can reuse the roles for your sites, or create your own. Typically, you will need to create roles to cover the full range of users' responsibilities on any given site.

For information about roles, see [Chapter 1, "Overview"](#) and the section [Section 4.5, "Working with Roles"](#) for general information, and see [Chapter 31, "System Defaults,"](#) for the list of system roles and their specifications.

Determine whether you will be replicating the sites you are configuring

If you plan to replicate the sites you are configuring, make sure that template-coding and other requirements are satisfied for the source sites (the sites you are currently configuring). For an overview of site replication as well as replication options and requirements, see [Chapter 11, "Replicating Content Management Sites."](#)

Determine system security

Before developers begin designing the online site, or contemplating changes to the user interface on the management system, you must determine and implement your security protocols. Decisions that you make about security affect the way that developers code and implement the online site. For information about security, see [Chapter 6, "Setting Up External Security."](#)

3.3.3 Configuration Steps at the Development System

Configuration steps at the development system are also given in detail in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. When coding at the development system is complete, the data model is migrated to the management system, where site configuration is completed.

Complete configuration steps in this order:

Create the sites that will be used on the management system

In this step, you will create sites with the names that will be used on the management system. For information and procedures on creating sites, see [Chapter 8, "Assembling Content Management Sites."](#)

Create the data model (asset types and their definitions)

For developers to create a data model or any site used on the management system, the developers must be created and configured, preferably as an administrative user, at the development system. For more information on this, see [Section 3.3.4, "Configuration Steps at the Management System,"](#) and create administrators on the developers' system.

Create content asset types

Start by creating the assets with the broadest use to maximize reusability:

1. Create design asset types (templates), bearing in mind site-replication requirements, if any.
2. Create management asset types.
3. Create sample assets of each type.

For information about creating asset types, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Test the data model

Test the data model in accordance with your company's QA requirements.

Migrate the data model to the management system

To migrate the data model, publish it to the management system. For instructions, see the chapters on real-time publishing.

3.3.4 Configuration Steps at the Management System

Once the development system has been put into place, you can then configure the management system.

Complete configuration steps in this order:

Create ACLs, if necessary

For procedures on creating ACLs, see [Chapter 4, "Working with ACLs and Roles."](#)

Note: Under no circumstances should you modify or delete a system ACL.

Database tables. When the data model is migrated to the management system, assign ACLs to the custom database tables (asset types), as necessary. For procedures on assigning ACLs to database tables, see [Chapter 4, "Working with ACLs and Roles."](#)

WebCenter Sites page entries. Typically there is no need to assign ACLs to WebCenter Sites page entries on the management system because you use roles to control access to assets and interface functions. However, if you need instructions for assigning ACLs to WebCenter Sites page entries, see [Section 4.3.5, "Assigning ACLs to WebCenter Sites Pages \(SiteCatalog Page Entries\)."](#)

Configure the external user manager, if you are using one

For procedures on configuring an LDAP server, see the *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

Create Roles

For procedures on creating roles, see [Chapter 4, "Working with ACLs and Roles."](#)

Create the users (content providers)

For instructions on creating user accounts in WebCenter Sites, see [Section 5.4.1, "Creating a New User."](#) If you are creating user accounts in external managers, refer to the product documentation.

Create user profiles, if necessary

User profiles are required for users who will be working with:

- Sites content applications (Engage, Analytics)
- Language packs and setting a default language

- Workflow processes, in which email messages will be sent to notify workflow participants of their assignments. The user profile supports workflow actions by mapping a user name to an email address.

User profiles must be created in WebCenter Sites. For instructions, see [Section 5.5, "Working with User Profiles and User Attributes."](#)

Set user attributes, if necessary

If users require attributes beyond the locale and email attributes that are specified in the user profile, you can create the attributes. For instructions, see [Section 5.5.3, "Modifying, Adding, and Deleting User Attributes."](#)

Create workflow processes for asset types, as necessary

Implementing workflow processes is a business decision, and not a WebCenter Sites requirement. If you need to create workflow processes, do the following:

1. Create the Workflow tab for the site, if one does not already exist. For instructions, see [Section 9.5.2.1, "Creating a New Tree Tab."](#)
2. Plan your workflow process by sketching it. See [Chapter 10, "Creating and Managing Workflow Processes"](#) for help with this step. Then refer to your sketch and notes throughout this section.
3. If you have not already done so, create the roles that are needed for the workflow processes. In you need instructions, see [Section 4.5.1, "Creating a Role."](#)
4. Ensure that users are set up to participate effectively in the workflow:
 - a. Ensure that users who will participate in workflow have user profiles created for them. Otherwise, they will not receive email messages from the workflow process.
 - b. For shared assets, if you want the pool of workflow candidates to include people from all the sites that an asset is shared to, enable the cross-site assignments feature. See [Section 10.2.2.3, "Cross-Site Assignments and Participants"](#) for details.
5. Create the email objects that you need for your actions and enable the `xcelerate.emailnotification` property in the `futuretense_xcel.ini` file. For instructions, see [Chapter 1, "Overview."](#)
6. Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions that you need. For instructions, see [Section 10.3.3, "Setting Up the Actions and Conditions."](#)
7. If your states have deadlines, be sure to configure the Timed Action Event so that the deadlines of assets are calculated regularly and the appropriate timed actions (if any) are invoked in a timely way. For instructions, see [Section 10.3.4, "Setting Up the Timed Action Event."](#)
8. Create your states. For instructions, see [Section 10.3.5, "Setting Up the States."](#)
9. Create your process. While creating your workflow process, you create the steps for that process. The steps link together the states so they occur in the proper order. Additionally, while creating your process, you configure any function privileges that you need. For instructions, see [Section 10.3.6, "Setting Up the Workflow Processes."](#)
10. Test your workflow processes. For instructions, see [Section 10.3.7, "Testing Your Workflow Process."](#)

Set up your start menu shortcuts so that workflow processes are assigned automatically to assets when they are created. For help with start menu items, see [Section 9.2.2, "Creating Start Menu Items."](#)

Re-create the sites that were created on the development system

For information and procedures on creating sites, see [Chapter 8, "Assembling Content Management Sites."](#)

Migrate the data model, if you have not already done so

At this point, you will need the data model in order to complete most of the remaining steps. Real-time publish the data model from the development system to the management system by using procedures in the chapters on real-time publishing.

Assign custom ACLs (if any) to the custom tables, if you have not already done so

For procedures on assigning ACLs to database tables, see [Chapter 4, "Working with ACLs and Roles."](#)

Assemble the sites

1. Grant users access to the sites. For instructions, see [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)
2. Enable asset types for the sites by associating the asset types with the sites. For instructions, see [Section 8.3.1, "Enabling Asset Types for a Site."](#)

Note: This step provides you with the option of allowing WebCenter Sites to automatically create Start Menu items for the asset types you are enabling. The Start Menu items New and Search items allow site users to create and search for assets of those types. You can also create your own Start Menu items.

Make the site components available to users

1. If you have chosen to create your own Start Menu items for various asset types, go to [Section 9.2, "Managing Access to Asset Types via Start Menus"](#) for additional instructions.
2. Set access permissions on assets. For instructions, see [Section 9.3, "Setting Access Permissions to Assets."](#)
3. Give users access to the tabs in WebCenter Sites' tree. For instructions, see [Section 9.5, "Managing Access to the Tree \(Administrator's interface only\)."](#)

Configure the Sites publishing process

1. Determine which type of publishing your management system is to execute:
 - RealTime
 - Export to Disk
 - Mirror to Server
 - Export Assets to XML
2. Configure the publishing process so the information that content providers create can be made available to the delivery system.

For general information and instructions on configuring the publishing process, see the chapters on export and mirror publishing. For information on the RealTime publishing process, see the chapters on real-time publishing.

Configure user interfaces, as necessary

The user interface options are Contributor, Admin, WEM Admin, Sites Desktop, and Sites DocLink. For instructions on making these interfaces available to content providers, see [Chapter 23, "Configuring the User Interfaces."](#)

Enable asset types for search engines, as necessary

In some cases, such as upgrades, it may be necessary to configure the asset types to be searched so that they can be indexed correctly and found by your search engine. For instructions on configuring the asset types, see [Chapter 25, "Configuring the Lucene Search Engine."](#)

Set up revision tracking, as necessary

Revision tracking prevents assets from being edited by more than one user at a time. When you enable revision tracking for a database table, WebCenter Sites maintains multiple versions of the assets in that table. For instructions on setting up revision tracking, see [Chapter 26, "Revision Tracking."](#)

Set up the delivery system

For more information, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Test all systems

Test all systems in accordance with your company's QA policies.

Give users the URL and their log in information to the to the WebCenter Sites installation

Once users log in, they can then control and manage the content as you've configured.

Working with ACLs and Roles

Several user management components are used to control access to a WebCenter Sites system: ACLs, user accounts, user profiles, roles, and sites. Site configuration, itself, begins with ACLs, with the creation of user accounts, and with the creation of roles.

ACLs can be used on both your management and your delivery systems. However, the main focus of this chapter is user management on the management system. For information about user management on your delivery system, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

This chapter contains the following sections:

- [Section 4.1, "Overview"](#)
- [Section 4.2, "ACLs"](#)
- [Section 4.3, "Working with ACLs"](#)
- [Section 4.4, "Roles"](#)
- [Section 4.5, "Working with Roles"](#)

4.1 Overview

ACLs and roles are of paramount importance in WebCenter Sites.

ACLs are used to regulate entry in to the WebCenter Sites system. The assignment of ACLs to users, database tables, and WebCenter Sites Pages determines users' permissions to operate on WebCenter Sites' database tables. If the user's ACLs match the database ACLs, the user has certain permissions (defined by the ACLs) to operate on the database tables. ACLs therefore serve as the foundation of the security and user management model. Without ACLs user accounts cannot be created.

Roles are used to manage access to sites and their components. The assignment of roles to users and interface functions on a given site determines whether the interface functions are enabled for the users or hidden from them. If the user's roles match the roles that are assigned to the interface functions, the functions are enabled for the user. Otherwise, the functions are hidden.

Before you can establish users you must determine which ACLs and roles the users need to be assigned. If the ACLs and roles do not already exist, you must create them in WebCenter Sites, even if you intend to use LDAP plug-ins to create the user accounts.

This chapter provides you with a basic explanation of ACLs and roles, and shows you how to create, modify, and delete ACLs and roles.

4.2 ACLs

Access Control Lists, called ACLs for short, are named sets of database operation permissions such as read, write, create, and retrieve. Because just about everything in WebCenter Sites (and the WebCenter Sites content applications) is represented as one or more rows in one or more database tables, user management on any of your Sites systems starts with ACLs.

With ACLs, you can limit access to the following items:

- Individual database tables
- Individual WebCenter Sites pages

WebCenter Sites and the Sites content applications use ACLs to enforce access restrictions to the various functions of those applications by controlling the user's access rights to the database tables that represent those functions. How? By verifying that the users attempting the function have the same ACL assigned to their user accounts as are assigned to the database table.

For example, user account information is contained in the system tables named SystemUsers and SystemUserAttrs. Because certain ACLs are assigned to those system tables, only a user with the same ACLs assigned to his or her user account is able to create new users or edit existing user information.

ACLs serve as the foundation of the security and user management model in your Sites system by providing authorization functionality. Even if you are using an external user manager like LDAP to store user information, you must use WebCenter Sites ACLs.

A user must always have at least the Browser ACL in order to view WebCenter Sites pages. However, additional ACL restrictions are enforced only when the `cc.security` property in the `futuretense.ini` file is set to `true`. For information about the `cc.security` property, see `futuretense.ini` in the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*. For information about configuring security on your WebCenter Sites system, see [Chapter 6, "Setting Up External Security."](#)

ACLs are assigned to three things: user accounts, database tables, and page entries in the SiteCatalog table (that is, WebCenter Sites pages).

Note: User Management on the Delivery System. User management on your delivery system is also based on ACLs. If your online site is designed to require visitors to register or log in before they can access areas of the site, you create the ACLs that are needed on the delivery system and then assign them to the appropriate database tables.

Typically your site designers take care of assigning ACLs to WebCenter Sites pages. The *Oracle Fusion Middleware WebCenter Sites Developer's Guide* discusses how to design a user management process on the delivery system and provides code samples for pages that log visitors in to the site and verify their identities.

User Accounts

Every user must be assigned at least one ACL and the ACLs assigned to a user define that user's access to the WebCenter Sites system.

While users have one user account and one set of ACLs, no matter how many sites they have access to, they can have one set of roles for one site and a different set of

roles for another site. Therefore, users must be assigned all the ACLs necessary to provide them the permissions that they need to fulfill all of their site-specific roles.

For example, if you create a role that allows a user with the role the ability to create template assets, the user assigned that role must also be assigned the ElementEditor ACL, because creating templates writes data to the ElementCatalog table.

Database Tables

To restrict access to the data in a table, assign an ACL to it through the WebCenter Sites Database forms available through the Admin tab. Then, only those users with the same ACL have access to the data in that table.

If you assign more than one ACL to a table, a user needs only one of those ACLs to access the table. The user's access rights to that table (read, write, create, and so on) are the ones defined by the ACL.

All of the WebCenter Sites system tables (and several of the Sites content applications tables) have ACL restrictions. The SystemInfo table lists all the tables in the WebCenter Sites database and the ACLs assigned to them.

Note: Do not add ACLs to database tables through the WebCenter Sites Explorer application. Instead, use the WebCenter Sites Database form in the **Admin** tab.

With one exception—to register a foreign table in the WebCenter Sites database—never attempt to change the information in the SystemInfo table even if your user account has the ACL that allows you to do so.

For information about:

- Assigning ACLs to database tables, see [Section 4.3.4, "Assigning ACLs to Custom Tables"](#)
- Registering foreign tables, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Page Entries in the SiteCatalog Table

The SiteCatalog table holds page entries for all the pages displayed for the Sites content applications as well as the pages displayed for your online site (that is, the site that you are delivering from the delivery system.) If you want to restrict access to a page, assign an ACL to it.

Typically, site developers determine how page restrictions should be configured on the delivery system. If you are customizing the user interface on the management system, however, you might need to use ACLs to restrict access to your custom pages.

This section contains the following topics:

- [Section 4.2.1, "System ACLs"](#)
- [Section 4.2.2, "Sample ACLs"](#)
- [Section 4.2.3, "Custom ACLs"](#)

4.2.1 System ACLs

WebCenter Sites and the Sites content applications use a number of system ACLs to control user access to their features and functions. Different combinations of these ACLs must be assigned to users. Information about system ACLs and their

permissions are given in [Chapter 31, "System Defaults."](#)

4.2.2 Sample ACLs

A newly installed WebCenter Sites system contains only system ACLs. No additional ACLs have been created for any of the sample sites that are packaged with WebCenter Sites.

4.2.3 Custom ACLs

Because WebCenter Sites provides a comprehensive set of ACLs, it is unlikely that you will need to create your own. However, situations can arise where user management needs on the management or delivery systems require you to create ACLs. For example,

- If your online site requires user registration, you may need to create a set of ACLs for site visitors.
- If your developers create new functions and put them on new tabs to customize the management system, you might need to create additional ACLs (or roles) to support the new functions and tabs.

Although creating and applying ACLs is an administrative task, you must first work with your site designers and developers to determine which ACLs you need and how to apply them. Once you have determined the ACLs, create them by following procedures in the rest of this chapter.

4.3 Working with ACLs

This section shows you how to create ACLs, edit, and delete custom ACLs; apply ACLs to the database tables and WebCenter Sites pages; and customize access restricted messages

Note: When using an LDAP integration option, be aware of system response to user and site management operations. For information about system response, see [Chapter 33, "Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems."](#)

This section contains the following topics:

- [Section 4.3.1, "Creating a New ACL"](#)
- [Section 4.3.2, "Editing a Custom ACL"](#)
- [Section 4.3.3, "Deleting a Custom ACL"](#)
- [Section 4.3.4, "Assigning ACLs to Custom Tables"](#)
- [Section 4.3.5, "Assigning ACLs to WebCenter Sites Pages \(SiteCatalog Page Entries\)"](#)
- [Section 4.3.6, "Setting the ACL Restriction Error Message"](#)

4.3.1 Creating a New ACL

Note: When creating ACLs, consider the roles you will be using in order to ensure that the ACLs are commensurate with the roles. For example, if you will be creating a role that allows a user to create template assets, the user who is assigned that role must also be assigned the ElementEditor ACL, because creating templates writes data to the ElementCatalog table.

To create a new ACL

1. In the **Admin** tab, expand **User Access Management**, then double-click **ACLs**.
The ACLs form appears.

Figure 4–1 ACLs form

ACLs

Select an ACL:

Select Operation:

Modify ACL

Add ACL

Delete ACL

OK

2. Select **Add ACL** and click **OK**. The value in the Select an ACL field does not matter at this point.

The Add ACL form appears.

Figure 4–2 Add ACL Form**Add ACL**

ACL Name:

Description:

Access Privileges:

- Read
- Retrieve
- Write
- Create
- Delete
- RevisionTracking Audit
- RevisionTracking Admin

Add

3. In the ACL Name field, enter a unique name.
4. Select the access privileges you want to assign to this ACL. For information on each privilege, see [Section 31.1.1, "Permissions."](#)
5. Click **Add**.

WebCenter Sites creates the ACL and writes it to the `SystemACL` table. The new ACL appears in the drop-down list in the form described in step 1 of this procedure.

6. If you are using LDAP, create a group (on your LDAP server) that exactly matches the ACL you just created. After you create the group, assign it to the appropriate users.

4.3.2 Editing a Custom ACL

Caution: Never modify any of the system ACLs. For a list of these ACLs, see [Section 31.2, "System ACLs."](#)

To edit a custom ACL

1. In the **Admin** tab, expand **User Access Management**, then double-click **ACLs**.
The ACLs form appears.

Figure 4-3 ACLs Form**ACLs**

Select an ACL:

Select Operation:

Modify ACL

Add ACL

Delete ACL

2. In the Select an ACL field, select the ACL to edit.
3. Select **Modify ACL** and click **OK**.

If the ACL you selected is a system ACL, an alert will display. Do not modify system ACLs.

4. In the Modify ACL form, make the desired changes to the Description and Access Privileges. For information on the displayed options, see [Section 31.1.1, "Permissions."](#)
5. Click **Modify**.

WebCenter Sites writes your changes to the SystemACL table.

4.3.3 Deleting a Custom ACL

Caution: Never delete a system ACL. For a list of these ACLs, see [Section 31.2, "System ACLs"](#).

To delete a custom ACL

1. If you are using LDAP, delete (from your LDAP server) the group corresponding to the ACL you will be deleting.
2. In the **Admin** tab, expand **User Access Management**, then double-click **ACLs**.
The ACLs form appears.

Figure 4–4 ACLs Form**ACLs**

Select an ACL:

Select Operation:

Modify ACL

Add ACL

Delete ACL

3. In the Select an ACL field, select the ACL to delete.
4. In the form that appears, select the desired ACL from the drop-down list and click **OK**.

WebCenter Sites displays a warning message.

5. Click **OK**.

The ACL has been deleted.

4.3.4 Assigning ACLs to Custom Tables

If you or the site designers create new tables, you might need to restrict access to those tables by assigning ACLs to them. Typically, you assign ACLs to new tables when you create those tables. (For more information, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.)

Note: Do not assign additional ACLs (beyond the ones assigned by default) to system or core product tables.

To assign ACLs to an existing table

1. In the **Admin** tab, expand **User Access Management**, then double-click **Sites Database**.

WebCenter Sites displays the Sites Database form.

Figure 4–5 Sites Database Form

Sites Database

Enter Table Name:

(use "%" as a wild card)

Select Operation:

- Modify Table
- Add Table
- Mirror Table
- Delete Table

2. Enter the name of the table to which you want to assign ACLs. If you do not know the name of the table you want to work with, do one of the following:
 - Leave the field blank. WebCenter Sites will return a list of all tables in the database.
 - Enter a partial name, ending with the wildcard character (%). WebCenter Sites will return a list of tables named similarly to your criteria.
3. Select **Modify Table** and click **OK**.
4. In the list of tables, select the desired table.
WebCenter Sites displays the Modify Catalog form.
5. In the **ACL** field, select the ACL(s) you want to assign to the selected table. To select multiple ACLs, **Ctrl-click** each desired ACL. You can also select a range of ACLs by selecting the first and last ACL in the range with **Shift-click**.

Note: Do not change the value of the File Storage Directory field. For information about this field, look up the `defdir` property in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

6. Click **Modify**.

4.3.5 Assigning ACLs to WebCenter Sites Pages (SiteCatalog Page Entries)

There are at least two ways to assign ACLs to SiteCatalog page entries.

- When developers create SiteEntry or template assets, they can assign ACLs to the page entry created for that asset through a field in the Create or Edit form.
- For page entries that are not associated with a SiteEntry or template asset, developers can use the User Access Management.

To assign ACLs to a page entry that is not associated with a SiteEntry or template asset

1. In the **Admin** tab, expand **User Access Management**, then double-click **Site**.

WebCenter Sites displays the Site form:

Figure 4–6 Site Form

Site

Enter Page Name:
(use "%" as a wild card)

Select Operation:

- Modify Page
- Add Page
- Modify Status
- Modify ACLs
- Modify Page Cache
- Delete Pages
- View Page

OK

2. Enter the full path and name of the page to which you want assign ACLs. If you do not know the name of the page you want to work with, do one of the following:
 - Leave the field blank. WebCenter Sites will return a list of all page entries in the SiteCatalog table.
 - Enter a partial name, ending with the wildcard character (%). WebCenter Sites will return a list of pages named similarly to your criteria.
3. Select **Modify ACLs** and click **OK**.
The Modify ACLs form appears.

Figure 4–7 Modify ACLs Form

Modify ACLs

Select the ACL:

A scrolling list of ACLs with a blue header and footer. The list items are: Browser, ContentEditor, ElementEditor, ElementReader, PageEditor, PageReader, RemoteClient, SiteGod, TableEditor, and UserEditor.

Select the pages to modify:

All

None

Page Name	ACL	Apply?
AlloyUI/Tree/Util/placePage_XML		<input type="checkbox"/>
AlloyUI/Tree/loadAssetTree		<input type="checkbox"/>
AlloyUI/Tree/loadSitePlanTree		<input type="checkbox"/>
AlloyUI/Tree/placePage		<input type="checkbox"/>

- In the scrolling list at the top of the window, select the ACLs you want to assign to one or more pages. To select multiple ACLs, **Ctrl-click** each desired ACL. You can also select a range of ACLs by **Shift-clicking** the first and last ACLs in the range.
- In the list of pages, select the **Apply?** check box next to each page to which you want to assign the ACLs you selected in step 4. ACLs currently assigned to each page are shown in the **ACL** column. (To select all of the pages in the list, click **All**. To deselect all of the pages in the list, click **None**.)
- At the bottom of the form, click **Apply**.

4.3.6 Setting the ACL Restriction Error Message

When users attempt to access a page for which they do not have the appropriate permissions, WebCenter Sites displays an error message. This message is stored in the following file:

```
<cs_install_dir>/futuretense_cs/formpriv.html
```

You can customize this and other error message pages in the `futuretense_cs` directory, with the following restrictions:

- Do not rename any of the files.
- Do not alter any occurrences of the `{0}` string in any of the files. WebCenter Sites uses the `{0}` string to automatically generate these error messages.

4.4 Roles

User management includes the concept of roles. Roles complement users in the following ways:

- Whereas the user definition (account) describes an individual's access to the underlying WebCenter Sites functionality (database tables) through ACLs, roles are used to manage site-specific access to WebCenter Sites' interface functions.
- A role represents a job description or the title of individuals with similar functions: for example, content provider, editor, site designer, and administrator.
- Each WebCenter Sites user has one user definition (account) no matter how many sites have been created. However, the user's roles can vary by site.
- When you enable a user for a site, you enable that user within the context of the roles that user is to fulfill for that site.

The roles assigned to a user for a site determine the following:

- Which assets the user can create on that site.
- Which assets the user can search for on that site.
- Which tabs are displayed in the tree when the user logs in to the site.
- Whether the user is eligible for participation in any workflow processes, and, if so, for which steps in those workflow processes.
- Which functions a user can or cannot perform on an asset while it is moving through a workflow process.
- Whether the user can administer a workflow process or create or modify a workflow group on that site.

This section contains the following topics:

- [Section 4.4.1, "System Roles"](#)
- [Section 4.4.2, "Sample Roles"](#)
- [Section 4.4.3, "Custom Roles"](#)

4.4.1 System Roles

Several system roles are installed by WebCenter Sites. One role is required for the Sites content applications to function, and three are required for the WebCenter Sites administrators to function. For more information, see [Section 31.5, "System Roles."](#)

4.4.2 Sample Roles

If you installed one or more sample sites, you will have access to a number of sample roles that are included with the sites. The roles permit the sample site users to access different tree tabs. You can use the sample roles as examples of how you can configure access control on your sites.

4.4.3 Custom Roles

Unlike ACLs, roles are objects that you will most likely need to create in order to account for the full range of users' responsibilities on your sites. To create roles, follow instructions in the next section, [Section 4.5, "Working with Roles."](#)

4.5 Working with Roles

This section shows you how to create, edit, and delete roles.

Note: If you are using LDAP, be aware of system responses to user and site management operations. For information about system responses, see [Chapter 33, "Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems."](#)

This section contains the following topics:

- [Section 4.5.1, "Creating a Role"](#)
- [Section 4.5.2, "Editing a Role"](#)
- [Section 4.5.3, "Deleting a Role"](#)

4.5.1 Creating a Role

To create a new role

1. In the **Admin** tab, expand **User Access Management**, then expand **Roles**, then double-click **Add New**.

WebCenter Sites displays the Add New Role form.

Figure 4–8 Add New Role Form

Add New Role

*Name:

*Description:

Cancel

Add New Role

2. In the **Name** field, enter a unique name of up to 32 characters.
3. In the **Description** field, enter a short and informative description of no more than 255 characters.
4. Click **Add New Role**.
5. Add the role to the default tree tabs (**Workflow**, **Bookmarks**, **Site Admin**, **Admin**, **Site Plan**, and **Active List**), as appropriate. For instructions, see [Section 9.5.3, "Editing Tree Tabs."](#)

4.5.2 Editing a Role

Although you cannot change the name of a role after you have created it, you can edit the description of the role.

To edit the description of a role

1. In the **Admin** tab, expand **User Access Management**, then double-click **Roles**. A list of all roles appears in the main window.
2. In the list of roles, navigate to the role you want to edit and click its **Edit** (pencil) icon.
3. In the Edit Role form, make your changes, then click **Save**.

4.5.3 Deleting a Role

Caution: Do not delete any of the system default roles: GeneralAdmin, SiteAdmin, WorkflowAdmin, AdvancedUser, SitesUser.

To delete a role

1. In the **Admin** tab, expand **User Access Management**, then double-click **Roles**. A list of all roles appears in the main window.
2. In the list of roles, navigate to the role you want to delete and click its **Delete** (trash can) icon.

WebCenter Sites displays a confirmation message.
3. Click **Delete Role**.

The role has been deleted.

Configuring Users, Profiles, and Attributes

Users can be created through one of the following options: WebCenter Sites' native user manager (in the administrator's interface), or via LDAP. However, to configure user profiles and attributes, you must use the administrator's interface.

This chapter outlines your options for creating users. This chapter also shows you how to create users in WebCenter Sites, and configure their profiles and user attributes. For information about creating users in external sources, refer to the product documentation.

This chapter contains the following sections:

- [Section 5.1, "Overview"](#)
- [Section 5.2, "User Management Options"](#)
- [Section 5.3, "Configuring Users in WebCenter Sites"](#)
- [Section 5.4, "Working with User Accounts"](#)
- [Section 5.5, "Working with User Profiles and User Attributes"](#)

5.1 Overview

WebCenter Sites users can be created through WebCenter Sites' native user manager, through WEM Admin, or through external user managers such as LDAP.

Every WebCenter Sites user is defined by the following set of data:

- User account, which gives the user access to the WebCenter Sites system and its database tables
- User profile, which is required for users who will be working with:
 - WebCenter Sites applications
 - Language packs and setting a default language
 - Workflow processes, in which email messages will be sent to notify workflow participants of their assignments. The user profile supports workflow actions by mapping a user name to an email address.
- User attributes (in addition to the email and locale attributes in the user profile), if actions and events in addition to workflow must be supported.

Once users are created and configured, they must be associated, by means of roles, with the sites they are to work in. This chapter shows you how to create and configure users. Procedures for associating users with sites are given in [Chapter 8, "Assembling Content Management Sites."](#)

5.2 User Management Options

WebCenter Sites' Directory Services API enables your WebCenter Sites system to connect to external directory servers or user managers that contain authentication information, user information, and so on. The following connection options are available:

- Native system—The WebCenter Sites native user manager, which uses the native WebCenter Sites user management tables `SystemUsers` and `SystemUserAttrs`.
- The LDAP plug-in—With this option, user names and attributes are stored in the directory server rather than in the WebCenter Sites database.

Because WebCenter Sites security is based on ACLs, any external user management system (such as LDAP) must be configured to match the WebCenter Sites ACLs.

Information about configuring the plug-ins is given in the *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*. Properties that configure the plug-ins are located in the files `futuretense.ini` (the **Authentication** tab), `ldap.ini`, and `dir.ini` files. The files are described in the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

Note: This guide uses the native WebCenter Sites user manager throughout.

This section contains the following topics:

- [Section 5.2.1, "Native WebCenter Sites User Manager"](#)
- [Section 5.2.2, "LDAP Plug-In"](#)

5.2.1 Native WebCenter Sites User Manager

If you are using the native WebCenter Sites user manager, follow the guidelines in [Chapter 3, "Site Configuration Guidelines"](#) to create and configure users, and then grant them access to the management system.

5.2.2 LDAP Plug-In

If you are using LDAP to manage your users on either the management or the delivery system, you create user accounts with LDAP rather than with the WebCenter Sites administrator's interface. However, you must still use the administrator's interface to create ACLs and roles in the WebCenter Sites database (It is also possible to use WEM Admin to create roles). Instructions for granting users access to a Sites management system integrated with LDAP are given in our guide, the *Oracle Fusion Middleware WebCenter Sites: Installing and Configuring Supporting Software*.

5.3 Configuring Users in WebCenter Sites

Each WebCenter Sites user is completely defined by a user account, user profile, and, if necessary, user attributes.

- A user account is required for anyone who is to work with WebCenter Sites.
- A user profile is required for users who will be working with WebCenter Sites modules and products, setting a default language, and participating in workflow processes in which email messages will be sent.

- User attributes, in addition to the locale and email attributes in the user profile, may also be required for your operation. If so, the additional attributes can be created.

Once you have created the user, you must enable that user for the appropriate sites by assigning roles to the user name for each site the user will work in. For information about enabling users after you have created them, see [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)

After you have created and enabled a new user, be sure to give that user the following information:

- The user name/password combination of the user account.
- The URL to the WebCenter Sites:

```
http://<server>:<port>/<context>/login
```

where

<server> is the host name or IP address of the machine running WebCenter Sites. Depending on how the system was set up, you might also need to include the port number— `server:8080` for example; and

<context> is the name of the web application on the same server.

The rest of this chapter shows you how to create user accounts, profiles, and attributes, as well as modify and delete them.

5.4 Working with User Accounts

This section shows you how to create, edit, and delete user accounts in WebCenter Sites.

Note: This section provides procedures for creating, editing, and deleting users in the WebCenter Sites administrator's interface. If you are using LDAP, refer to the LDAP product documentation. If you are using WEM Admin, refer to [Chapter 34, "Web Experience Management Framework."](#)

Also, be sure to substitute the word "group" for the word "ACL" when reading that guide, and create users who belong to the groups with these names.

This section contains the following topics:

- [Section 5.4.1, "Creating a New User"](#)
- [Section 5.4.2, "Creating Common User Types"](#)
- [Section 5.4.3, "Editing a User"](#)
- [Section 5.4.4, "Deleting a User from the System"](#)

5.4.1 Creating a New User

This section shows you how to create users in the WebCenter Sites administrator's interface.

Note: If you are using LDAP, refer to the LDAP product documentation, if you are using WEM Admin, refer to [Chapter 34, "Web Experience Management Framework."](#)

Before creating a user

Before creating a user, determine the user's:

- Log in name
- Password
- ACLs, which regulate the user's access to WebCenter Sites' database tables.
 - To determine the user's required ACLs, see [Section 31.4, "Required ACLs for Custom Users."](#)
 - To determine the user's additional system ACLs, see [Section 31.2, "System ACLs."](#)
 - To determine which ACLs are assigned to the database tables the user must access, follow the steps in [Section 4.3.4, "Assigning ACLs to Custom Tables."](#)

To create a user

1. In the **Admin** tab, expand **User Access Management**, then double-click **User**.
2. In the form that appears, select **Add User** and click **OK**.

You can add a user name before clicking OK; it will appear auto-filled in the Login Name field on the next form.

3. Fill in the fields in the Add User form:

Figure 5–1 Add User Form

Add User

***Login Name:**

***Access Privileges:**

***Password:**

***Re-enter Password:**

Add

- a. In the Login Name field, enter a unique name. Do not include spaces or special characters, such as punctuation. The underscore character (`_`) is allowed.
- b. In the Access Privileges list, select ACLs for the user. To select multiple ACLs, **Ctrl-Click** each desired ACL; you can also select a range of ACLs by selecting the first and last ACL in the range with **Shift-Click**.
- c. Enter the same password into the **Password** and **Re-Enter Password** fields.
- d. Click **Add**.

The user has been created.

- e. You can create a general administrator, For more information about making the user an administrator or a content contributor, see [Section 5.4.2, "Creating Common User Types."](#)

To follow up with post-creation procedures

1. If the user will be implementing any of the following options:
 - Oracle WebCenter Sites products such as Oracle WebCenter Sites: Engage
 - Language packs and different languages
 - Workflow processes that send email messages

create a profile for the user. For instructions, see [Section 5.5.1, "Creating and Editing a User Profile."](#)

2. If the user requires attributes in addition to or in place of locale and email (specified in the user profile), create the attributes. For instructions, see [Section 5.5.3, "Modifying, Adding, and Deleting User Attributes."](#)
3. Once the user has been completely defined, you must associate the user with a site by means of roles.
 - a. If you have not already done so, create roles for the user, following instructions in [Section 4.5.1, "Creating a Role."](#)
 - b. To associate the user to the site, create the site and add the user to the site. For instructions, see [Section 8.1.1, "Creating a Site"](#) and [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)

5.4.2 Creating Common User Types

When creating a user, it is important to consider the roles that the user will have on separate web sites. In many cases, a user may be an administrator on one site and a content contributor on another. This section details how to make each user a specific type of user.

This section describes the following topics:

- [Section 5.4.2.1, "Making the User a General Administrator"](#)
- [Section 5.4.2.2, "Making the User a Site Administrator"](#)
- [Section 5.4.2.3, "Making the User a Workflow Administrator"](#)
- [Section 5.4.2.4, "Making the User a Content Contributor"](#)

5.4.2.1 Making the User a General Administrator

The general administrator for a site controls all aspects of the site. You should be aware that when a user is a general administrator, their administrative actions can only be controlled when they are logged in to the one site, however, they can administrate all sites when logged in.

1. For the newly created user, double-check that it has the correct ACLs assigned to it.

To determine the ACLs for a user, see [Section 31.4, "Required ACLs for Custom Users."](#) Looking at the table there, you can see that the ACLs for a general administrator are Browser, Element Reader, PageReader, UserReader, xceleditor, xceladmin, TableEditor, UserEditor, and VisitorAdmin.

2. On the Admin tab expand security, expand Assign Users to Groups, and double-click Add New. The Assign Groups to User form appears.
3. Select the user to make a general administrator, and then select RestAdmin in the Groups field.

You can use **Ctrl-Click** to select more than one user or group, but as you will likely make few general administrators, it is suggested you add them one at a time to ensure you do not accidentally add a user to the RestAdmin group.

4. Open the WebCenter Sites tree, and select the site to add the user to.
5. Click Users to open the User Role Management form.

This form is also available under Site Admin tab by double clicking Users if you are logged in to the specific site.

6. Enter the log in information for the user, and click the Edit icon to open the Edit Roles for User form. Select GeneralAdmin as well as any other roles you wish the user to have on this web site.

Note: The names of roles used in this documentation are the names of the roles used in the FirstSiteII example web site that comes installed. You might choose to use different roles or role names with your published web site.

When adding a general administrator to a site, you may want to consider if you will want the same person to have different roles on a different site. You will need to assign these roles to the user when logged in to the site or sites you want to assign these different roles for the user on.

When a user has general administrator rights, the Admin tab will only appear when they are logged into the web site (or web sites) where they have the general administrator role.

5.4.2.2 Making the User a Site Administrator

1. For the newly created user, double-check that it has the correct ACLs assigned to it.

To determine the ACLs for a user, see [Section 31.4, "Required ACLs for Custom Users."](#) Looking at the table there, you can see that the ACLs for a site administrator are Browser, Element Reader, PageReader, UserReader, xceleditor, and xceladmin.

2. Open the WebCenter Sites tree, and select the site to add the user to.
3. Click Users to open the User Role Management form.

This form is also available under the **Site Admin** tab by double-clicking **Users** if you are logged in to the specific site.

4. Enter the log in information for the user, and click the Edit icon to open the Edit Roles for User form.
5. Select the user to make a site administrator, and then select **SiteAdmin_AdminSite** in the Groups field.

You can use **Ctrl-Click** to select more than one user or group, but as you will likely make few site administrators, it is suggested you add them one at a time to ensure you do not accidentally add a user to the SiteAdmin_AdminSite group.

6. Select **SiteAdmin** and **Advanced User** as well as any other roles you wish the user to have on this web site. The SiteAdmin role grants the user access to the Site Admin tab, the AdvancedUser role grants access to the Admin interface.

Note: The names of roles used in this documentation are the names of the roles used in the FirstSiteII example web site that comes installed. You might choose to use different roles or role names with your published web site.

When adding a site administrator to a site, you may want to consider if you will want the same person to have different roles on a different site. You will need to assign these roles to the user when logged in to the site or sites you want to assign these different roles for the user on.

When a user has site administrator rights, the Site Admin tab will only appear when they are logged into the web site (or web sites) where they have the site administrator role.

5.4.2.3 Making the User a Workflow Administrator

1. For the newly created user, double-check that it has the correct ACLs assigned to it.

To determine the ACLs for a user, see [Section 31.4, "Required ACLs for Custom Users."](#) Looking at the table there, you can see that the ACLs for a workflow administrator are Browser, Element Reader, PageReader, UserReader, xceleditor, and xceladmin.

2. Open the WebCenter Sites tree, and select the site to add the user to.
3. Click Users to open the User Role Management form.

This form is also available under **Site Admin** tab by double-clicking **Users** if you are logged in to the specific site.

4. Enter the log in information for the user, and click the Edit icon to open the Edit Roles for User form.
5. Select WorkflowAdmin and AdvancedUser, as well as any other roles you wish the user to have on this web site. The WorkflowAdmin role grants the user access to the Workflow tab, the AdvancedUser role grants access to the Admin interface.

Note: The names of roles used in this documentation are the names of the roles used in the FirstSiteII example web site that comes installed. You might choose to use different roles or role names with your published web site.

When adding a workflow administrator to a site, you may want to consider if you will want the same person to have different roles on a different site. You will need to assign these roles to the user when logged in to the site or sites you want to assign these different roles for the user on.

When a user has workflow administrator rights, the Workflow tab will only appear when they are logged into the web site (or web sites) where they have the workflow role.

5.4.2.4 Making the User a Content Contributor

For the newly created user, double-check that it has the correct ACLs assigned to it.

To determine the ACLs for a user, see [Section 31.4, "Required ACLs for Custom Users."](#) Looking at the table there, you can see that the ACLs for all users are Browser, ElementReader, PageReader, UserReader, and xceleditor. You may want to add additional ACLs needed for a content contributor, depending on the type of content you want the user to contribute.

For example, the PageEditor ACL would allow the user to add pages to the site, and modify the content based on the defined elements. The ElementEditor ACL would allow the contributor to create templates. For more information on the specific ACLs and how they impact the contributor's editing access, see [Chapter 31, "System Defaults."](#)

The user does not need additional ACLs if you only want them to modify existing content on a page. Simple editing is contained in the assigned role.

1. Open the WebCenter Sites tree, and select the site to add the user to.
2. Click **Users** to open the User Role Management form.
This form is also available under **Site Admin** tab by double clicking **Users** if you are logged in to the specific site.
3. Enter the log in information for the user, and click the **Edit** icon to open the Edit Roles for User form. Select **SitesUser** for the role.

It may be necessary to add different roles depending on how the roles are connected to the assets. When assets are created, they are connected to roles. These roles are the only roles that have access to the asset.

Note: Typically you would not assign a user the SitesUser role as well as the AdvancedUser role for the same web site. The SitesUser role grants access to the Contributor interface; the AdvancedUser role grants access to the Admin interface.

The names of roles used in this documentation are the names of the roles used in the FirstSiteII example web site that comes installed. You might choose to use different roles or role names with your published web site.

For most content contributors, you will not want to assign them roles on the web site beyond SitesUser, depending on how you have set up the different assets. Content contributors will make changes to the content only through the Contributor interface, and not through the Admin interface that administrators use.

5.4.3 Editing a User

This section shows you how to edit users in the WebCenter Sites administrator's interface. If you are using LDAP, refer to the LDAP product documentation.

To edit a user

Caution: Do not change the names or ACLs of WebCenter Sites system users (DefaultReader, ContentServer, xceladmin).

1. In the **Admin** tab, expand **User Access Management**, then double-click **User**.
2. In the form that appears, enter the name of the user you want to work with. If you do not know the user name, leave the field blank; WebCenter Sites will return a list of all users in the system.
3. Select **Modify User** and click **OK**.
4. In the list of users, select the user you want to work with.
5. In the Modify User form, make the desired changes, then click **Modify**.

5.4.4 Deleting a User from the System

This section shows you how to delete WebCenter Sites users using the WebCenter Sites administrator's interface. If you are using LDAP, refer to the LDAP product documentation.

To delete a user from the system

Caution: Do not delete any of the WebCenter Sites system users (fwadmin - or the user used as admin, ContentServer, or DefaultReader).

1. Delete the user profile, as shown in [Section 5.5.2, "Deleting a User Profile."](#)
2. In the **Admin** tab, expand **User Access Management**, then double-click **User**.
3. In the form that appears, enter the name of the user you want to delete. If you do not know the user name, leave the field blank; WebCenter Sites will return a list of all users in the system.
4. Select **Delete User** and click **OK**.
5. In the list of users, select the **Delete** radio button next to the user you want to delete.
6. Click **Delete**.
WebCenter Sites displays a warning message.
7. Click **OK**.
The user has been deleted.

5.5 Working with User Profiles and User Attributes

A user profile is required for any user who will be working with the following:

- Sites modules and products
- Language packs
- Workflow processes in which email messages will be sent to notify workflow participants of their assignments. The user profile supports workflow actions by mapping a user name to an email address.

A user profile holds a set of user attributes. By default, the only user attributes a user profile holds are:

- The **email attribute**, which is used to support workflow actions and takes the user's email address as a value. (You can create workflow actions that send workflow participants email about the assets that are assigned to them.)
- The **locale attribute**, which is used to determine which language to use for a given user. This attribute takes the user's preferred location as a value.
- The **timezone attribute**, which is used to determine which time zone to use for a given user. This attribute takes the user's preferred time zone as a value.

If the user was created in the WEM Admin interface, the avatar is also an attribute. You can add more user attributes and store values for them in the WebCenter Sites user management tables if you want to. However, to use these values in the WebCenter Sites interfaces requires you to customize the elements that display the user profile forms. For information about customizing elements for the WebCenter Sites interfaces, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

This section contains the following topics:

- [Section 5.5.1, "Creating and Editing a User Profile"](#)

- [Section 5.5.2, "Deleting a User Profile"](#)
- [Section 5.5.3, "Modifying, Adding, and Deleting User Attributes"](#)

5.5.1 Creating and Editing a User Profile

If you are using LDAP, be aware of system responses to user and site management operations. For information about system responses, see [Chapter 33, "Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems."](#)

To create or edit a user profile

1. In the **Admin** tab, expand **User Access Management**, then double-click **User Profiles**.

WebCenter Sites displays the User Profile Management form.

2. In the form, enter the desired user name and click **Select**.

WebCenter Sites displays the profile of the selected user.

3. Click the **Edit** icon.

WebCenter Sites displays the Edit User Profile form:

Figure 5–2 Edit User Profile Form

The screenshot shows a web form titled "Edit User Profile". The form is enclosed in a light gray border. At the top left, the title "Edit User Profile" is displayed. Below the title is a horizontal dashed line. The form contains the following fields:

- User Name:** A text field containing the value "DefaultReader".
- Email:** An empty text input field.
- Locale Preference:** A dropdown menu with "No preference" selected and a downward arrow.
- Time Zone:** A dropdown menu with "Auto-Detected" selected and a downward arrow.

At the bottom of the form are two buttons: "Cancel" and "Save", both with a dark background and white text.

4. In the **Email** field, enter the user's email address.
5. (Optional) Select a locale preference for this user from the **Locale Preference** drop-down list. The user's local preference overrides the language preference for the WebCenter Sites system (which is set from the Locale Manager).

Note: If you do not set a locale preference for the user (**No preference** is selected in the **Locale Preference** drop-down list), the WebCenter Sites interfaces will be displayed in the language set for the WebCenter Sites system. For information about setting the locale preference for the WebCenter Sites system, see

6. (Optional) In the **Time Zone** drop-down list, select the user's time zone preference. If you do not specify a time zone preference (**Auto-Detected** is selected in the drop-down menu), the system will automatically detect the time zone of the user from the user's browser.
7. Click **Save**.
8. Enable this user for the sites the user needs to work with. For instructions, see [Chapter 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)

5.5.2 Deleting a User Profile

To delete a user profile

1. In the **Admin** tab, expand **User Access Management**, then double-click **User Profiles**.
2. In the User Profile Management form, click **Delete**.
WebCenter Sites displays a warning message.
3. Click **Delete User Profile**.
WebCenter Sites displays a message confirming the deletion.

5.5.3 Modifying, Adding, and Deleting User Attributes

By default, the only user attributes that the WebCenter Sites content applications need are an email address and locale preference. Users created in WEM will additionally have an avatar attribute. You use the user profile feature to assign these attributes to a user, as shown in [Section 5.5.1, "Creating and Editing a User Profile."](#) If you need to, you can store and use additional user attributes for your users in this table, even if you are using LDAP.

The Modify User Attributes option allows you to modify the attributes that are used in the user profile. It also allows you to add and delete attributes.

To modify a user's attributes

1. In the **Admin** tab, expand **User Access Management**, then double-click **User**.
In the form that appears, enter the name of the user you want to work with. If you do not know the user name, leave the field blank; WebCenter Sites will return a list of all users in the system.
2. Click the user name in the list that whose attributes you want to modify. The User Attributes form will display.
3. Fill in the fields of the User Attributes form:

Figure 5-3 User Attributes Form**User Attributes:DefaultReader**

Attribute Name	Attribute Values
<input type="text"/>	<input type="text"/>
Add new attribute and value .	

Modify

Do one of the following, as required:

- Change the current value (or values) assigned to an attribute by editing the contents of the **Attribute Values** field.
- Add a new attribute by entering its name and at least one value in the fields at the bottom of the form.
- Delete any unwanted attributes by deleting the associated value (in the **Attribute Values** field).

4. Click **Modify.**

Your changes are committed to the database.

Setting Up External Security

Determining what your security protocols should be and then implementing them is an important part of a WebCenter Sites administrator's duties. You and the site developers must discuss and determine how security will be configured on both the management and the delivery systems before they start coding templates and designing asset types.

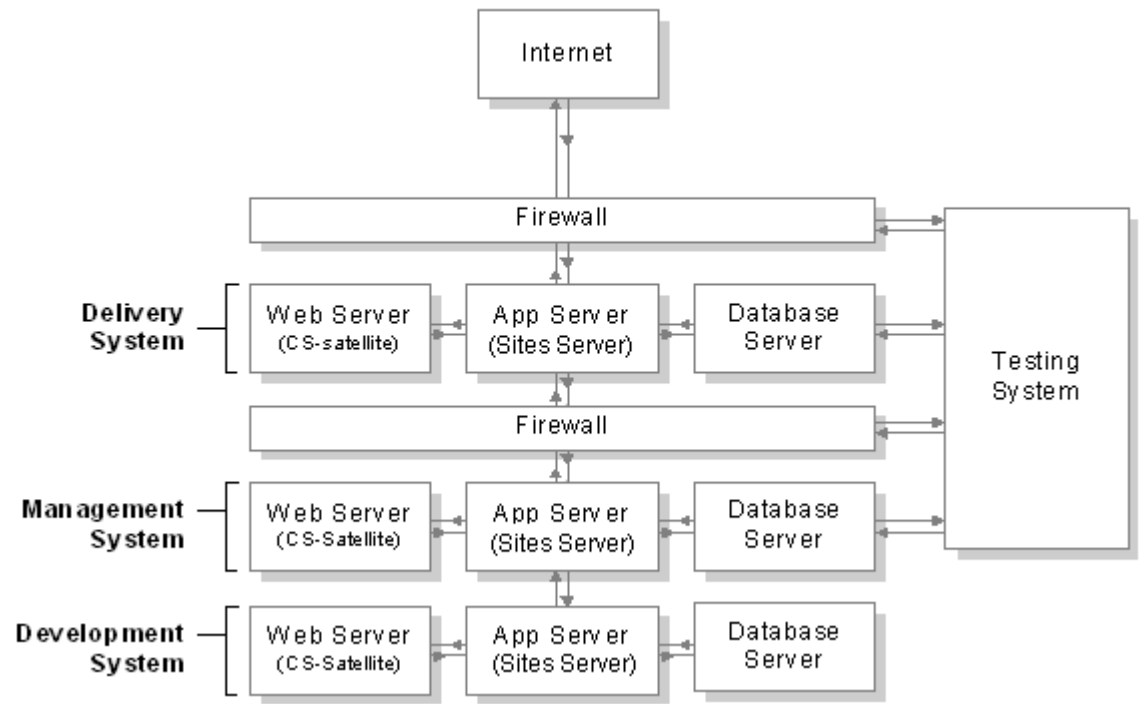
This chapter contains the following sections:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Implementing Security"](#)
- [Section 6.3, "Testing Security"](#)

6.1 Overview

Before developers begin designing the online site or contemplating changes to the user interface on the management system, you must determine and implement your security protocols. The decisions you make about security configuration affects the way that you code and implement your online site.

At regular intervals, you should review your systems to determine whether they are working as they should. [Figure 6-1](#) shows an example of a secure WebCenter Sites system:

Figure 6–1 WebCenter Sites System

This section contains the following topics:

- [Section 6.1.1, "ACLs and Security"](#)
- [Section 6.1.2, "DefaultReader, secure.CatalogManager, and secure.TreeManager"](#)
- [Section 6.1.3, "BlobServer and Security"](#)
- [Section 6.1.4, "Security Goals"](#)

6.1.1 ACLs and Security

As mentioned in [Chapter 4, "Working with ACLs and Roles,"](#) ACLs (access control lists) serve as the foundation of the security and user management model in your WebCenter Sites system. ACLs are sets of permissions that restrict access to both database tables and WebCenter Sites pages.

ACL restrictions are enforced only when the `cc.security` and the `security.checkpagelets` properties in the `futuretense.ini` file are set to true. These properties are set to true by default. If you need to disable security for any reason, open the Property Editor and set the `cc.security` property to false. However, Oracle recommends that you keep this property set to true on all systems to ensure that site development is designed to work with security enabled.

When planning security measures for your system, examine the list of default ACLs (described in [Chapter 31, "System Defaults"](#)), and determine whether you need additional ACLs. You might need to create an ACL with a different combination of permissions than any of the system ACLs provide if your developers plan to create new tables or to make additions to the user interface.

Note: Under no circumstances should you modify a system default ACL or modify the ACLs assigned to a WebCenter Sites system table or a Sites content application table.

6.1.2 DefaultReader, secure.CatalogManager, and secure.TreeManager

WebCenter Sites and the WebCenter Sites content applications are delivered with several default user accounts, one of which is named DefaultReader. This account is assigned to all unauthorized site visitors (because anyone who visits a site hosted by WebCenter Sites must have a WebCenter Sites user identity).

By default, the DefaultReader user account has the following ACLs: Browser and Visitor. Visitor enables the tracking of visitors for demographic purposes. Security issues arise because many of the database tables also have the Browser ACL assigned to them, and various Engage tables are assigned the Visitor ACL. Therefore, anyone using the DefaultReader account can examine information in the tables (although they cannot write to the tables as this user). To prevent someone from using the DefaultReader user account to view tables in your WebCenter Sites database, set the following properties in the futuretense.ini file to true:

- `secure.CatalogManager`
- `secure.TreeManager`

When these properties are set to true, the DefaultReader user cannot access the CatalogManager and TreeManager servlets—not even for read-only data.

Note: More information about Oracle WebCenter Sites: Engage ACLs (Visitor and VisitorAdmin) is available in [Chapter 31, "System Defaults."](#) See [Table 31-3, "System ACLs and Their Descriptions"](#) for permissions that are associated with each ACL and [Table 31-4, "Default Users and Their ACLs"](#) for descriptions of the ACLs.

6.1.3 BlobServer and Security

When you want to implement security for your blobs you must enable the security feature for the BlobServer servlet. You do this by setting the `bs.security` parameter in the `futuretense.ini` file to true.

When `bs.security=true`, BlobServer refuses to serve a blob unless the URL for the blob contains evidence that the person requesting it has been authenticated as a valid user. What evidence? A value in the URL from the `csblobid` parameter whose value matches a session variable named `csblobid`. Therefore, when BlobServer security is enabled, your developers must code links to blobs differently than usual.

For information about how those links should be coded, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

6.1.4 Security Goals

Typically, you have a different set of security goals for each of your WebCenter Sites systems: the development, management, and delivery systems.

This section contains the following topics:

- [Section 6.1.4.1, "Security Goals for the Development System"](#)
- [Section 6.1.4.2, "Security Goals for the Management System"](#)

- [Section 6.1.4.3, "Security Goals for the Delivery System"](#)

6.1.4.1 Security Goals for the Development System

Even though development systems are typically behind firewalls, you should implement the same security configuration on the development system that will be in place on the system the developers are designing for (management or delivery). Why? To be sure that the code will work properly on the target system, because the same conditions exist on both systems:

- When you plan to use BlobServer security on the delivery system, templates that create URLs for blobs must be coded differently. Therefore, you must enable BlobServer security on the development system as well.
- If your online site will require that visitors identify themselves, you must have the same security configuration in place on the development system that you will use on the delivery system.

6.1.4.2 Security Goals for the Management System

Security on the management system encompasses two main concepts:

- Ensuring that only valid WebCenter Sites users can access the system (described in this chapter).
- Ensuring that those valid users can access only the functions that are appropriate for them. For information, see [Chapter 4, "Working with ACLs and Roles."](#)

6.1.4.3 Security Goals for the Delivery System

The precautions that you take on the delivery system are more stringent by nature because when you deliver a site to the public, you must ensure that while visitors can access the content on your site, hackers cannot access areas that you do not want made public.

When you configure your delivery system, you disable the WebCenter Sites user interfaces to prevent anyone from adding assets or code through the interfaces or with any of the developer tools. Additionally, you restrict access to some of the WebCenter Sites servlets.

6.2 Implementing Security

This section describes the steps you must take to configure the security measures you have decided to implement: setting properties, changing passwords for default user accounts, using SSL for systems accessible outside your company's network, mapping URLs for specific WebCenter Sites servlets, and disabling certain parts of the applications on the delivery system.

Each section states which steps should be performed on which system or systems.

This section contains the following topics:

- [Section 6.2.1, "Properties That Configure Security Settings"](#)
- [Section 6.2.2, "Users and Passwords"](#)
- [Section 6.2.3, "SSL and Digital Certification"](#)
- [Section 6.2.4, "URLs and the Web Server \(Delivery System Only\)"](#)
- [Section 6.2.5, "WebCenter Sites Forms and Pages \(Delivery System Only\)"](#)

6.2.1 Properties That Configure Security Settings

This section describes how to configure the properties in the `futuretense.ini` file that implement various kinds of security.

For All Systems

Use the Property Editor to open the `futuretense.ini` file and verify that the following properties are set to true:

- `cc.security`
- `security.checkpagelets`
- `secure.CatalogManager`
- `secure.TreeManager`

If you plan to use BlobServer security, set the following property to true:

`bs.security`

To ensure that the session timeout value is appropriate for each system, set the following property, as well:

`cs.timeout`

On the development and management systems, set the timeout value for as long as you think that you safely can. On the delivery system, set the timeout value for as short a time as you can without frustrating your visitors.

For the Delivery System

In addition to the properties described in the preceding section, there is one more property to specify for the delivery system: `cs.wrapper`.

Wrappers are the HTML files located in the `futuretense_cs` directory on the web server. Also included in that directory is a subdirectory named `Dev`, which you should remove from your delivery system.

However, if you decide to remove the entire `futuretense_cs` directory from the web server on the delivery system, you must set the `cs.wrapper` property to `false`.

For more information, see [Section 6.2.5, "WebCenter Sites Forms and Pages \(Delivery System Only\)."](#)

6.2.2 Users and Passwords

Be sure that the default user accounts were made secure after WebCenter Sites was installed on all systems.

For All Systems

Complete the following steps on all systems—development, management, and delivery:

- Change the default password for the `fwadmin` user account (**User** node under **User Access Management** in the **Admin** tab in the administrator's interface).
- If the WebCenter Sites user that was created during the installation is named `ContentServer`, verify that its password is not the default password (`password`). If the password used is the default password, change it (**User** node under **User Access Management** in the **Admin** tab in the administrator's interface).
- Change the default administrator username/password for your application server.

- Change the default administrator username/password for your web server.
- If you have the sample sites installed, a mirror user was created for the Mirror to Server publishing method (name: `mirroruser`; password: `mirroruser`). Change the password for this user.
- For any user accounts that have the SiteGod ACL, change the password frequently (**User** node under **User Access Management** in the **Admin** tab in the administrator's interface). Handle any user account that has the SiteGod ACL as you would the UNIX `root` user.

Note: Do not change the default password for the `DefaultReader` user.

For the Management and Delivery Systems

If any of the sample sites except `AviSports` or `FirstSiteII` were installed on either the management or delivery systems, delete all the sample users including `editor`, but do not delete the `xceleditor` ACL and do not delete the `fwadmin` user. Additionally, change the password for the `mirroruser` user.

For the Delivery System

Do not assign the `xceleditor` ACL to any user on the delivery system other than the system administrator of that system. This ACL allows access to the WebCenter Sites content applications if they are installed on that delivery system.

6.2.3 SSL and Digital Certification

For any system accessible outside your company's network or containing proprietary data, such as a management system open to remote employees, it is recommended to use SSL to establish a secure, encrypted connection.

You must use a digital certificate approved by a trusted authority rather than a self-signed certificate. Using a self-signed certificate can have the following consequences:

- Internal calls to the system made through Java may fail.
- Some features of the user interface may not function correctly, for example, the left-hand navigation tree or publishing.

6.2.4 URLs and the Web Server (Delivery System Only)

On the web server of the delivery system, be sure to give access to the following WebCenter Sites servlets only. How? By mapping only their URLs to the application server:

- `ContentServer`
- `BlobServer`
- `Satellite`
- `CookieServer`

Do not map URLs to the application server for any of the other WebCenter Sites servlets. Instead, map the URLs for the following servlets to an error page such as the "404 Not Found" page:

- `HelloCS`

- CatalogManager
- TreeManager
- DebugServer
- CacheServer
- Inventory

6.2.5 WebCenter Sites Forms and Pages (Delivery System Only)

On the delivery system, be sure to disable or completely remove the following pieces of the WebCenter Sites applications:

- Remove the `futuretense_cs/Dev` directory from the web server. This directory holds forms that are useful for developers, which means that you do not want them on your delivery system.

Note: Do not remove this directory from the application server. Remove it from the web server only.

You can optionally remove the entire `futuretense_cs` directory from the web server. (Do not remove it from the application server.) If you do, be sure to set the `cs.wrapper` property in the `futuretense.ini` file to `false` (the wrappers are the HTML files from that folder). If you forget, visitors on the site will see "404 Page Not Found" rather than some of the system error messages.

- Go to `SiteCatalog/OpenMarket/Xcelerate/UIFramework` and rename all of the pages. At the very least, rename `LoginPage` and `LoginPost`.

6.3 Testing Security

After you have implemented your security measures, test your systems.

Security Tests for All Systems

Complete the following steps on your development, management, and delivery systems:

1. Try to log in to the database with Sites Explorer using the default user accounts:
 - `DefaultReader`
If you can log in using `SomeReader` as the password, the `secure.CatalogManager` and `secure.TreeManager` properties are set to `false`. Change them to `true`.
 - `ContentServer`
If you can log in using `password` as the password, change the password immediately.
 - `editor`
If you can log in using `xceleditor` as the password, change the password immediately.
 - `fwadmin`
If you can log in using `xceladmin` as the password, change the password immediately.

2. Verify that the sample site users do not exist on the management or delivery systems.
3. Verify that you cannot log in with ContentServer/password using a CatalogManager URL:

`http://<server>:<port>/<context>/CatalogManager?ftcmd=login&username=ContentServer&password=password`
4. Verify that you cannot flush the entire cache as ContentServer/password using a CacheServer URL:

`http://<server>:<port>/<context>/CacheServer?all=true&authusername=ContentServer&authpassword=password`
5. Verify that you cannot log in to the application server as the default administrator user.
6. Verify that you cannot log in to the database as the default administrator user.
7. Verify that you cannot log in to the web server as the default administrator user.

Implementing Security Using Cryptography and Encryption Key Storage

WebCenter Sites provides secure server access to protect your system from internal or external security attacks. All access is protected through cryptographic encryption of sensitive information whether located in files or within the server itself. It is important for all organizations to establish security standards to ensure that important information is not compromised

Audit Trail Information reports are described in this chapter. They are:

- [Section 7.1, "WebCenter Sites Security Overview"](#)
- [Section 7.2, "Considerations for External Key Stores"](#)
- [Section 7.3, "The SitesSecurityContext.xml Configuration File"](#)
- [Section 7.4, "Mixed Credential Stores"](#)
- [Section 7.5, "Sites Security Utilities"](#)
- [Section 7.6, "Using Java JCEKS Key Stores"](#)
- [Section 7.7, "ESAPI Security Validation"](#)

7.1 WebCenter Sites Security Overview

WebCenter Sites secures the system by default and provides several options to further tighten security to meet your installation's requirements.

The system comes installed with a secured key store that provides the necessary encryption keys to properly handle passwords and data exchanges across the network through encryption techniques. This ensures that sensitive information is not visible to unauthorized individuals which could compromise the security of your system.

The system has the option to provide an external key store so that the creation and administration of encryption keys can be controlled by the system administrator. An external key store uses standard Java cryptography to create and maintain symmetrical encryption keys.

The system has the option to provide an external credential store rather than a Java key store for increased security. A credential store facility uses Oracle's advanced cryptography to create and maintain symmetrical encryption keys. The credential store location is identified by a properly configured JPS (Java Platform Security) provider. When used with the WebLogic application server, access and modification of the credential store is protected by JPS permissions. Only authorized programs are permitted to access the credential store for reading, modification, or both.

7.2 Considerations for External Key Stores

The use of standard Java cryptography or Oracle's advanced cryptography functionality depends on the type of storage for external encryption keys. The system does not support inter mixing of cryptography packages, they are mutually exclusive in all cases. The external key store (type is JCEKS) must use the standard Java cryptography package. The Oracle Credential Store Facility (CSF) only allows encryption keys created by the Oracle Security Developer Tools (OSDT). Because encryption keys created by one cryptographic package will not be compatible across all types of key stores, you should choose the type of store that suits your security needs and use it in all cases.

For components that require cryptographic support across application server contexts or machines (such as required for clustered configurations) or both the same encryption keys must be available. In other words, where a data value is encrypted at one location it must be decrypted using the same key at the destination location. This means that you must use copies of the central key store each place where encryption and decryption are necessary.

The replication and sharing of encryption keys through a key store or credential store varies on your environment and is not discussed in this document.

Configuration of the cryptography package and external encryption key stores is controlled by the `SitesSecurityContext.xml` configuration file. This file must reside on the class path of the programs that require it. Customizing this file is covered later in the next section. The Sites product uses a standard naming convention for associating encryption keys to different functions within the product. There is not a single encryption key used in all cases. This permits changing the key by functional use at any time without affecting other functions. There are four associative names used, for any key store type, which are:

1. **masterkey** – reserved for future use
2. **tokenskey** – the encryption key used by the Sites token generator
3. **generalkey** – the encryption key used for all encrypted POST exchanges and sensitive information located in external files.
4. **passwordkey** – the encryption key used when the user password in the database is encoded by symmetrical encryption.

Note: For a JCEKS key store, the storage of cryptography keys provides password protection for every key. This version of Sites does not make use of this capability and expects each of the above keys to have the same password as the key store itself.

There are two types of encryption key storages that can be used:

1. JCEKS is the standard key store facility to housing symmetrical encryption keys. Keys stored in this facility must be created and used with the standard Java cryptography functions. Although this type is universal it is provider specific and care must be observed not to mix a key store created by one provider and used by another. This is tied to the Java runtime used to run your applications. Oracle or Sun Microsystems use a standard JCE provider to create and retrieve encryption keys. IBM provides their own Java runtime which has a different key store provider. Although it is possible to modify the IBM security environment to accept the standard Java JCE provider, the steps to do this are beyond the scope of this document.

2. Oracle Credential Store Facility (CSF) is a very secure key storage facility that is accessed through Java Platform Security (JPS). This facility can be used instead of JCEKS key store across all environments and is not provider specific as it is an integral part of Oracle security. Only symmetrical encryption keys created by Oracle OSDT (Oracle Security Developer Tools) can be stored in this facility. In the WebLogic environment, this credential store is protected by JPS permissions. Programs that access or manipulate contexts of the credential store must be granted permission. Those programs which do not have the proper permission are denied access. For situations requiring the highest level of security, this is the preferred option.

7.3 The SitesSecurityContext.xml Configuration File

The `SitesSecurityContext.xml` file controls all aspects of the security environment which WebCenter Sites and its related utilities operate within. It is a Spring configuration file which instantiates the implementation classes for cryptography, encryption key storage, CSRF token generation, and internal security processing within the Sites server. It is an integral part of security between the various system components.

When it is necessary to change system defaults to satisfy varied security requirements this file is modified to provide the desired overrides. There must be a copy of this configuration file, with the modifications, available to each component (that is, WebCenter Sites utility program, server component, REST client) to ensure that the proper cryptographic package is used with access to the required key store. Failure to provide the proper configuration and associated key store to all components will cause part of the system to cease functioning.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-2.5.xsd">

  <!-- Crypto support package -->
  <bean id="cryptoPackage"
    class="com.fatwire.security.common.JavaSecurityCrypto" />

  <!-- Credential storage (using JCEKS repository) support package -->
  <bean id="credentialPackage"
    class="com.fatwire.security.common.JCEKSCredentialStore" >
  <property name="keyStoreJar"
    value="com/fatwire/security/common/data/WCSitesDefault.keystore" />
  <property name="IBMStoreJar"
    value="com/fatwire/security/common/data/WCSitesDefaultIBM.keystore" />
  </bean>

  <!-- CSRF Token generation/validation package (Sites server only) -->
  <bean id="tokenPackage" class="com.fatwire.auth.SecurityTokenImpl" />

  <!-- Security Basis ... This is the root of all evil -->
  <bean id="securityBasis"
    class="com.fatwire.security.common.SecurityBasis" >
  <property name="securityCrypto" ref="cryptoPackage" />
```

```

    <property name="securityToken" ref="tokenPackage" />
    <property name="credentialStorage" ref="credentialPackage" />
  </bean>

  <!-- Security Context (Sites server only)-->
  <bean id="securityContext"
        class="COM.FutureTense.Security.Context.AdvancedSecurityContextImpl" >
    <property name="securityBasis" ref="securityBasis" />
  </bean>

</beans>

```

The SitesSecurityContext.xml file is broken down into five beans; three are mandatory (cryptoPackage, credentialPackage, and securityBasis) and two used only by the Sites server (tokenPackage and securityContent). Once modified, the revised file must be placed on the class path of all components that require it.

The beans in this file are described as:

- **cryptoPackage** – this bean specifies which cryptographic package is initialized. There are two choices, expressed as a fully qualified class name:
 - com.fatwire.security.common.JavaSecurityCrypto – using Java runtime
 - com.fatwire.security.common.OracleSecurityCrypto – using OSDT. This cryptography package must be specified if the Oracle Credential Store Facility is to be used.
- **credentialPackage** – this bean specifies which key storage is initialized. There are three choices, expressed as a fully qualified class name:
 - com.fatwire.security.common.JCEKSCredentialStore – default support using JCEKS key stores.
 - com.fatwire.security.common.DefaultCredentialStore – standard CSF implementation for all external programs or application servers other than WebLogic.
 - com.fatwire.security.common.WeblogicCredentialStore – WebLogic specific CSF implementation.
- **tokenPackage** – this bean is required for the Sites server only. The class keyword selects the proper class implementation (default is com.fatwire.auth.SecurityTokenImpl). You have the option to implement your own security token generator which implements the securityToken interface.
- **securityBasis** – this bean binds all security components together and is not to be modified. The property entries refer to other bean definitions. The securityToken property is only specified on the Sites server.
- **securityContext** – this bean is required for Sites server only. It defines the class which handles all internal security processing. This bean is not to be modified.

This file must be available on the class path for each application that requires cryptographic encryption and access to a key store facility. Copies of this file must be configured with the same cryptography and credential store packages. The following components require a properly configured copy of this file.

1. Sites Content web application server WEB-INF/classes
2. Sites CAS web application server WEB-INF/class (when CAS is the configured authenticator)

3. Sites TokenAuthority web application servlet (when OAM is the configured authenticator)
4. Sites REST client application programs
5. Sites utility programs (CatalogMover, PropertyEditor, XmlPost, CSE)

7.4 Mixed Credential Stores

It is possible to implement mixed key storage methods provided the encryption keys are created with the same salts. For example, CAS can be configured with Java cryptography and JCEKS key storage which can be used with Sites content server using Oracle cryptography and CSF key storage. Both key stores are populated with the named symmetrical encryption keys using the identical salts. Although this will work it is recommended to use a single key storage methodology for all applications for simplicity.

7.5 Sites Security Utilities

There are several utilities available with the Sites system to aide in the creation and maintenance of encryption key storage. These utilities are located in `site-security-11.1.1.8.0.jar` and are executed by invoking by the specific class names from the java command. When running, it is necessary to define the class path for all the necessary jars required by the utilities. The following is a sample shell script that will run one of the utility classes when the class name is supplied as the first argument:

```
#!/bin/sh
# Set JAVA_HOME
JAVA_HOME="/fmw/jdk/bin"
# Set CLASS_PATH
CLP=".:sites-security-11.1.1.8.0.jar:commons-logging-1.1.1.jar:spring-2.5.5.jar"
CLP="${CLP}:/fmw/middleware/oracle_common/modules/oracle.osdt_11.1.1/osdt_core.jar"
CLP="${CLP}:/fmw/middleware/oracle_common/modules/oracle.osdt_11.1.1/osdt_core_fips.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.jps-common_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.jps-api_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.jps-internal_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.jps-mbeans_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.jps-unsupported-api_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.identitystore_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.oraclepki_1.0.0.0.jar"
CLP="${CLP}:/fmw/middleware/modules/com.oracle.osdt_cert_1.0.0.0.jar"
$JAVA_HOME/java -cp ${CLP} -Doracle.security.jps.config=./config/jps-config.xml
com.fatwire.security.util.$1 $2 $3
```

Description of Java classes

com.fatwire.security.util.CreateDefaultKeyStore.class: This Sites utility will create the key store and populate it automatically with a randomly generated set of Java encryption keys. It requires one argument: key store filename. The password for the key store must be supplied when prompted for on the system console. The password that is entered at the console prompt must be provided in the `SitesSecurityContext.xml` configuration file.

Arguments (positional):

Keystore-name Keystore-password

Example:

```
java -cp sites-security-11.1.1.8.0.jar
com.fatwire.security.util.CreateDefaultKeyStore mystore.keystore
```

com.fatwire.security.util.CreateDefaultCredentialStore: This utility will create the Oracle Credential Store Facility (CSF) *cwallet.sso* file and populate it automatically with a randomly generated set of Oracle encryption keys. The current JPS configuration is used to locate and update the *cwallet.sso* file. If there is no credential store, then it will be created. The *replace* argument should be set to *yes* if the utility should replace any Sites encryption keys that already exist.

Arguments:

product=sites

replace=no | yes (default is no)

com.fatwire.security.util.ExportFromJCEKS.class will extract the salts for each of the Sites keys in the currently active key store and write the values to the *export.keys* file in the working directory. The export file is used as input to one of the import utility programs listed below.

Arguments: (optional)

export.keys

Example:

```
java -cp
sites-security-11.1.1.8.0.jar;commons-logging-1.1.1.jar;spring-2.5.5.jar
com.fatwire.security.util.ExportFromJCEKS
```

com.fatwire.security.util.ImportToJCEKS.class will take the extracted salts from the *export.keys* file, create each of the Sites encryption keys using the associated salt, and add to the output key store. It requires two arguments: key store filename and export keys file. The password for the key store must be supplied when prompted for on the system console. The password that is entered at the console prompt must be provided in the *SitesSecurityContext.xml* configuration file. If the key store does not exist, then it will be automatically.

Arguments (positional):

Keystore-name export.keys

Example:

```
java -cp
sites-security-11.1.1.8.0.jar;commons-logging-1.1.1.jar;spring-2.5.5.jar
com.fatwire.security.util.ImportToJCEKS mystore,keystore export.keys
```

com.fatwire.security.util.ImportToCSF.class will take the extracted salts from the *export.keys* file, create each of the Sites encryption keys using the associated salt, and add to the output credential store. If the credential store does not exist, then it will be automatically created. The output credential store is defined by the location property in the *jps-config.xml*.

Arguments:

product=sites

import=export.keys

replace=no | yes (default is no)

Example:

```
java -cp
sites-security-11.1.1.8.0.jar;commons-logging-1.1.1.jar;spring-2.5.5.jar
com.fatwire.security.util.ImportToCSF
```

7.6 Using Java JCEKS Key Stores

The JCEKS key store which is used to store Java.security encryption keys. It is necessary to create this file and populate it with the encryption key set required by Sites. The key store can be created manually using the Java keytool utility located in the Java JRE bin folder, or using a creation utility class that is provided in the sites-security jar (`com.fatwire.security.util.CreateDefaultKeyStore.class`). The sites utility will create the key store and populate it automatically with a randomly generated set of Java encryption keys.

The key store requires two arguments:

- `arg[0]` = key store filename
- `arg[1]` = key store password

This password must be provided in the `SitesSecurityContext.xml` configuration file. In a minimum configuration (CS and CAS), there must be a `SitesSecurityContext.xml` file in each servlet context's class path that points to the proper key store.

If you are not using the `CreateDefaultKeyStore` utility, use the Java keytool utility supplied with the Java runtime. The store type must be `JCEKS`. You must create four keys with the following names listed below. Each encryption key is assigned the same password as the key store itself.

- `masterkey` – reserved for future use
- `tokenskey` – the encryption key used by the Sites token generator
- `generalkey` – the encryption key used for all encrypted POST exchanges and sensitive information located in external files.
- `passwordkey` – the encryption key used when the user password in the database is encoded by symmetrical encryption.

For WebSphere, you must create the key store using the IBM Java runtime. You cannot mix IBM runtime processing with a key store created with a normal Java runtime.

If you are using the `CreateDefaultKeyStore` utility, specify the name and password as arguments when running the class. The utility will create a complete set of the above keys.

Depending upon how you create the key store, the `SitesSecurityContext.xml` file must be modified for CS and a copy provided to override the default setup for CAS. You must replace the properties in the `credentialPackage` bean with a property `keyStoreFile` with a value that is the fully qualified file descriptor where the key store resides in the file system. The property `keyStorePassword` must provide the key store password provided when the key store was created.

Changing key stores does not affect password encoding in the Sites database. This ensures you can always log into the system even when key stores are not working properly. However, all external passwords stored in `INI` files will have an encryption from the internal key store and no longer be able to be encrypted. All passwords in external files should be changed back to their clear text values. Then after the new key store is in place, use the property editor to force each password to be encrypted using

the new keys. Remember to include a `SitesSecurityContext.xml` file to be accessed by the `PropertyEditor` utility.

Sometimes it is necessary to transfer the same salts used to create encryption keys from one key store to another. An example is creating mixed provider key stores in an environment with Java and IBM runtimes (used within WebSphere). The utility classes included in the `sites-security` jar file are provided to help this transition.

- **`com.fatwire.security.util.ExportFromJCEKS.class`** will extract the salts for each of the keys in the currently active key store and write the values to the `export.keys` file in the working directory.
- **`com.fatwire.security.util.ImportToJCEKS.class`** will take the extracted salts from the `export.keys` file, create each of the Sites encryption keys using the associated salt, and add to the output key store. If the key store does not exist, then it will be automatically created.

Arguments:

```
product=sites
import=export.keys
replace=no | yes
```

7.7 ESAPI Security Validation

The Sites server employs the Enterprise Security API to check for and prevent security vulnerabilities that may occur from injection of malicious web data. The validation can be customized to meet tighter control over the content of HTTP headers and incoming data. The validation expressions contained in the `ESAPI.properties` file can be modified; this file is included when you install the Sites server. This file resides in the Sites `WEB-INF/classes` folder.

Further information about ESAPI is available at the following link:

https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API

Assembling Content Management Sites

Content Management sites can be assembled once their basic components are created. The basic components are the users (including their ACLs and roles) and the data model. Optionally, you can enable auxiliary user interfaces such as Sites DocLink and Sites Desktop. This chapter shows you how to work with sites (create, edit, and delete sites); how to manage site users (add, edit, and delete users); and how to manage asset types for the sites (enable and remove asset types).

This chapter contains the following sections:

- [Section 8.1, "Working with Sites"](#)
- [Section 8.2, "Assigning and Managing Site Users"](#)
- [Section 8.3, "Enabling and Managing Asset Types"](#)
- [Section 8.4, "Enabling and Managing User Interfaces"](#)

Note: The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Sites Desktop, Sites DocLink.

8.1 Working with Sites

Assembling a site requires you to assign users to the site, assign content (asset types) to the site, and optionally enable the Sites Desktop and Sites DocLink interfaces. In this chapter, you will create sites, associate users to the sites, and assign asset types to the sites. In later chapters, you will develop each site by tuning site users' permissions to content on the site.

This section contains the following topics:

- [Section 8.1.1, "Creating a Site"](#)
- [Section 8.1.2, "Obtaining Site Configuration Information"](#)
- [Section 8.1.3, "Editing a Site"](#)
- [Section 8.1.4, "Deleting a Site"](#)

8.1.1 Creating a Site

Creating a site means specifying a site name, description, and a method for previewing the content that the site will display.

To create a site

1. In the **Admin** tab, expand **Sites** and double-click **Add New**.

The Add New Site form is displayed.

Figure 8–1 Add New Site Form

Add New Site

***Name:**

***Description:**

Preview method: ▼

2. In the **Name** field, enter a unique name of up to 255 characters.

Note: Be certain that you have entered a correct name for the site. Although you can edit the description of a site, you cannot change the name of a site after you have created it.

3. In the **Description** field, enter the name of the site as you want it to appear in the WebCenter Sites interfaces (up to 64 characters). This display name will be used in site lists throughout the user interface.
4. In the **Preview method** field, select the method how you will preview the site content.
5. Click **Add Site**.

The Site form appears, where you can execute a number of operations. For example, inspect, edit, or delete the site, manage users, or configure Site Launcher to replicate the site.

Figure 8–2 Site Form

Site: LoafersBakery

 [Inspect](#)  [Edit](#)  [Delete](#)

Name: LoafersBakery

Description: Loafers Bakery

Publication ID: 1333670710396

Preview method: Use default (Standard servlet preview)

Wrapper page:

Users: [▶ Manage users for this site](#)

Enabled Asset Types: No Asset Types are enabled for this site

[Enable Asset Types...](#)

[Disable Asset Types...](#)

[List all Start Menu items for this site](#)

Workflow Processes: [▶ List all Workflow Processes for this site](#)

Sites Desktop: [▶ Configure Sites Desktop for this site](#)

Site Launcher: [▶ Configure Site Launcher for this site](#)

[▶ List all sites](#)

When the new site is added to the **Sites** node, WebCenter Sites creates, below the site name, a set of sub-nodes that prompt you for site components:

- Click **Manage users for this site** for this site to manage site users.
- Click **Enable Asset Types...** to select asset types for the website; click **Disable Asset Types...** to remove asset types. Asset types are created by developers.
- Click **List all Start Menu items for this site** to display the Start Menu items which have been created for the site.
- Click **List all Workflow Processes for this site** to manage the workflows for the site.
- Click **Configure Sites Desktop for this site** to configure and enable Sites Desktop.

- Click **Configure Site Launcher for this site** to configure and enable Site Launcher.

Each sub-node is equipped with a linking mechanism that allows you to associate the sub-node, and therefore the site, with data in the configuration pool. Because of this linking mechanism, the sub-nodes cannot be deleted or supplemented with custom sub-nodes. Selecting a sub-node allows you to create All data required by the sub-nodes must be entered in to the configuration pool, described next.

Note that the procedure above only adds new sites.

- To make a site usable, you must associate users and asset types (created by developers) with the site. If you plan to use workflow processes, you should create those processes as well.
- To copy or migrate the site to another WebCenter Sites system, use the RealTime publishing method. See the chapters on real-time publishing.

8.1.2 Obtaining Site Configuration Information

In certain situations, you will need to know how a site has been configured—for example, which asset types and workflow processes are enabled, and who are the site users.

To obtain site configuration information

1. In the **Admin** tab, expand **Sites** and double-click the desired site.
2. In the Site form, select the option that provides the information you are looking for.

8.1.3 Editing a Site

Editing a site means changing its description and/or preview method. If you need to change the name of a site, you must delete the site and create a new one with the correct name.

To edit a site description

1. In the **Admin** tab, expand **Sites** and double-click the desired site.
2. In the Site form, click **Edit**.
3. In the Edit form, do the following:
 - a. In the **Description** field, make the appropriate changes.
 - b. In the **Preview Method** field, select a different preview method, if necessary.
4. Click **Save**.

8.1.4 Deleting a Site

When you delete a site, all configuration information for that site is either removed or unshared (if the site shared data with other sites). The following configuration information is affected by the deletion of a site:

- Start Menu items that were enabled for the site are deleted or unshared
- Smartlists are deleted or unshared
- Publishing destinations are unshared
- Users are removed from the site

- Workflows that were shared are unshared or deleted
- Template assets that were shared have their SiteEntry removed
- Asset subtypes that were enabled for that site as well as tree tabs are removed or unshared

Therefore, before deleting a site, be sure that you have shared the assets that you need to keep.


To delete a site

1. In the **Admin** tab, double-click **Sites**.
2. In the list of sites, navigate to the site you want to delete and click its **Delete** icon.

A warning is displayed to inform you that site components will be deleted. The warning identifies the components and prompts you to confirm your intention to delete the site.

Figure 8–3 Delete Site Form, Showing Warning

Delete Site

 This action will permanently delete assets that belong to the site exclusively. It will delete all site configuration information, including start menus, workflow processes, workflow groups, and tree tabs which are configured solely for this site. It will permanently remove the site. Are you sure that you want to delete this site? ✕

Name: LoafersBakery

Description: Loafer's Bakery and Restaurant

Description: Loafer's Bakery and Restaurant

Publication ID: 1328442563990

3. Click **Delete Site**.

8.2 Assigning and Managing Site Users

This section shows you how to grant WebCenter Sites users access to a site, view site users, change users' role assignments on a site, and remove users from a site.

This section contains the following topics:

- [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)"](#)
- [Section 8.2.2, "Viewing Site Users and Reassigning Roles"](#)
- [Section 8.2.3, "Deleting Users from a Site"](#)

8.2.1 Granting Users Access to a Site (Assigning Roles to Users)

For a user to gain access to a site, the user must be assigned at least one role on the site. Assigned roles associate the user with the site and allow the user to log in to the site. When one of the user's assigned roles is also specified for an interface function on the site, the user gains access to the interface function.

Note: In this section, we assume that you have created at least one user, a role, and a site. If you need to create one or another, follow the steps in [Section 5.4.1, "Creating a New User,"](#) [Section 4.5.1, "Creating a Role,"](#) and [Section 8.1.1, "Creating a Site."](#)

To give a user access to a site

1. In the **Admin** tab, expand **Sites**, then expand the desired site.
2. Double-click **Users**.

The User Role Management form appears.

Figure 8–4 User Role Management Form

User Role Management

Site: ▶ **FirstSiteII**

Username:

Leave blank to select all users in FirstSiteII.

Select

3. Enter the desired user name and click **Select**.

The User Role Management form displays the user name. No roles are listed next to the user name.

Note: If you need a list of system-wide users, open the list as follows:

Expand User Access Management and double-click **User**. Leave the Enter User Name field blank, select **Modify User**, and click **OK**.

4. Click the **Edit** icon next to the user name.

WebCenter Sites displays the Edit Roles form:

Figure 8-5 Edit Roles Form**Edit Roles for User: Coco**

User Name: Coco

Site: FirstSiteII

*Roles:

- AdvancedUser
- Analyst
- Analytics
- Approver
- ArtworkAuthor
- ArtworkEditor
- Author
- Checker
- ContentAuthor
- ContentEditor

5. Select one or more roles from the list. To select multiple roles, **Ctrl-Click** each desired role. To select a range of roles **Shift-Click** the first and last role in the range.

Note: Consider the user's type. A general administrator, site administrator, and regular user require different sets of system-defined roles. For example, a regular user requires access to the content contributor's interface. In this instance, select the **SitesUser** role. For additional help with selecting system-defined roles, see [Section 31.5, "System Roles."](#)

Selecting a role (or roles) makes the user a member of the site that you chose to work with in step 1. It does not necessarily enable the user to manage the site's content. To manage content, the user must have access to interface functions (such as Search) on the site.

If you know which roles are allowed to invoke which interface functions on the user's site, select the roles in this step. Otherwise, the roles that you select here you will also specify (in later chapters) for interface functions that the user needs. The functions may be start menu items, tree tabs, workflow processes, and asset approval and publishing functions.

For more information about assigning user roles, see [Section 4.5, "Working with Roles."](#)

6. Click **Save**.
7. Give the user access to site components by following procedures in [Chapter 9, "Managing Access to Content Management Site Components"](#)

Note: Remember that the user cannot perform the functions available to her roles unless the user also has adequate ACL permissions to operate on the database tables. (While roles allow access to interface functions, ACLs activate those functions by allowing access to the database tables.) For more information about roles and their relationship to ACLs, see [Section 1.4.4, "Roles."](#)

8.2.2 Viewing Site Users and Reassigning Roles

To view the list of users and their roles on a given site, and to reassign roles

1. In the **Admin** tab, expand **Sites**, then expand the desired site.
2. Under the desired site, double-click **Users**.
3. In the User Role Management form, leave the user name field blank and click **Select**.



















WebCenter Sites displays a list of all the users currently assigned roles in the selected site:

Figure 8–6 User Role Management Form

User Role Management

Site: [▶ HelloAssetWorld](#)

Select the user to modify:

User Name	Roles
   Bobo	AdvancedUser, WorkflowAdmin, SitesUser, GeneralAdmin
   Coco	HelloDesigner, AdvancedUser, SitesUser, GeneralAdmin
   Flo	AdvancedUser, SitesUser, HelloEditor
   fwadmin	HelloDesigner, AdvancedUser, SiteAdmin, WorkflowAdmin, SitesUser, HelloEditor, HelloAuthor, GeneralAdmin
   Joe	AdvancedUser, SitesUser, HelloAuthor
   Moe	AdvancedUser, SitesUser, HelloAuthor

[▶ Manage users for this site](#)

4. (Optional) If you want to reassign the user's roles, do the following:
 - a. Navigate to the desired user name and click its **Edit** icon.
 - b. In the Edit Roles form, select (and/or deselect) one or more roles from the list, then click **Save**.
 - c. Repeat steps a and b for each additional user, as necessary.

8.2.3 Deleting Users from a Site

You can delete a user from a site without deleting that user from the system.

To delete a user from a site

1. In the **Admin** tab, expand **Sites**, then the desired site.
2. Under the desired site, double-click **Users**.
3. In the User Role Management form, enter the desired user name and click **Select**.
4. In the form that appears, click the **Delete** icon next to the user.
WebCenter Sites displays a warning message.
5. Click **Delete User from Site**.

8.3 Enabling and Managing Asset Types

This section shows you how add asset types to a site (enable the asset types), and delete them from a site.

This section contains the following topics:

- [Section 8.3.1, "Enabling Asset Types for a Site"](#)
- [Section 8.3.2, "Removing Asset Types from a Site"](#)

8.3.1 Enabling Asset Types for a Site

Enabling an asset type means associating the asset type with a chosen site or sites. Once the assignment is made, the asset type itself must be made available to users through an interface function such as a start menu item or a tree tab, as shown in [Chapter 9, "Managing Access to Content Management Site Components."](#)

Note: The following procedure describes how to enable an asset type for use. Enabling an asset type for use with Sites Desktop is described in [Section 23.2.2, "Configuring the Word Asset Types for Sites Desktop."](#)

To enable asset types for a site

1. In the **Admin** tab, expand **Sites**, then expand the desired site.

Note: If you have the SiteAdmin role, but not the GeneralAdmin role, select the **Site Admin** tab.

2. Under the desired site, expand **Asset Types** and double-click **Enable**.

WebCenter Sites displays the Enable Asset Types form, listing asset types that have not yet been enabled in this site:

Figure 8–7 Enable Asset Types Form

Enable Asset Types: LoafersBakery

Name: LoafersBakery

Description: Loafer's Bakery and Restaurant

Publication ID: 1328442563990

Enable Asset Types:

Name	Description
<input type="checkbox"/> Article	Article
<input type="checkbox"/> AArticles	Article Flex
<input type="checkbox"/> AttrTypes	Attribute Editor
<input type="checkbox"/> CSElement	CSElement
<input type="checkbox"/> Collection	Collection
<input type="checkbox"/> Content_C	Content
<input type="checkbox"/> CAttributes	Content Attribute
<input type="checkbox"/> Content_A	Content Attribute
<input type="checkbox"/> ContentTmpls	Content Definition
<input type="checkbox"/> Content_CD	Content Definition

3. Select the asset types which you want to enable for this site.

Note: Content Query assets are not supported within the Admin UI.

4. Click **Enable Asset Types**.
5. (Optional) In the form that follows, check the box next to each start menu item that you would like WebCenter Sites to create. (For more information on start menu items, see [Section 9.2, "Managing Access to Asset Types via Start Menus."](#))

Figure 8–8 Enable Type Assets Form - Start Menu Selection

Enable Asset Types: **LoafersBakery**

Start Menu selection

Asset Type	<input type="checkbox"/> Available Start Menus
Article	<input type="checkbox"/> Enable Find Article for this site?
	<input type="checkbox"/> Enable New Article for this site?
	<input type="checkbox"/> Enable New Columnist Article for this site?
Content Attribute	<input type="checkbox"/> Enable Find Content Attribute for this site?
	<input type="checkbox"/> Enable New Content Attribute for this site?
Content Attribute	<input type="checkbox"/> Enable Find Content Attribute, FirstSiteII for this site?
	<input type="checkbox"/> Create New start menu for Content Attribute

If you choose not to generate start menu items at this time, you must manually create them later. No one can create assets of the asset types you just enabled until the corresponding start menu items are created.

6. Click **Enable Asset Types**.
7. The asset types are now enabled for the site. If you need to create start menu items for them, continue to [Section 9.2, "Managing Access to Asset Types via Start Menus."](#)

Note: To view the assets in the Contributor interface's trees (Site Tree, Content Tree, and My Work tree) you also need to manually create the tree tabs for the assets.

8.3.2 Removing Asset Types from a Site

To remove asset types from a site

1. In the **Admin** tab, expand **Sites**, then the desired site.

Note: If you have the SiteAdmin role, but not the GeneralAdmin role, select the **Site Admin** tab.

2. Under the desired site, expand **Asset Types** and double-click **Disable**.

WebCenter Sites displays the Disable Asset Types form, showing asset types that have been enabled for the selected site:

Figure 8–9 Disable Asset Types Form

Disable Asset Types: LoafersBakery

Name: LoafersBakery

Description: Loafer's Bakery and Restaurant

Publication ID: 1328442563990

Disable Asset Types:

Name	Description
<input type="checkbox"/> Article	Article
<input type="checkbox"/> CAttributes	Content Attribute
<input type="checkbox"/> Content_A	Content Attribute

Cancel

Disable Asset Types

3. Select the asset types that you want to remove from the site.
4. Click **Disable Asset Types**.

Note: If start menu items or workflow processes exist for the asset type that you disabled on a site, edit the start menu items or workflow processes so that they reflect the change.

8.4 Enabling and Managing User Interfaces

If you plan to enable the Sites Desktop and Sites DocLink interfaces for users on certain sites, follow instructions in [Chapter 23, "Configuring the User Interfaces."](#)

Managing Access to Content Management Site Components

This chapter shows you how to manage access to site components. It is assumed that users have already been given the permission to access the site containing those components.

This chapter contains the following sections:

- [Section 9.1, "Overview"](#)
- [Section 9.2, "Managing Access to Asset Types via Start Menus"](#)
- [Section 9.3, "Setting Access Permissions to Assets"](#)
- [Section 9.5, "Managing Access to the Tree \(Administrator's interface only\)"](#)

Note: The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Sites Desktop, Sites DocLink.

9.1 Overview

Once users are granted permission to access a site (as shown in [Chapter 8, "Assembling Content Management Sites"](#)) their ability to work with the site's content is managed through:

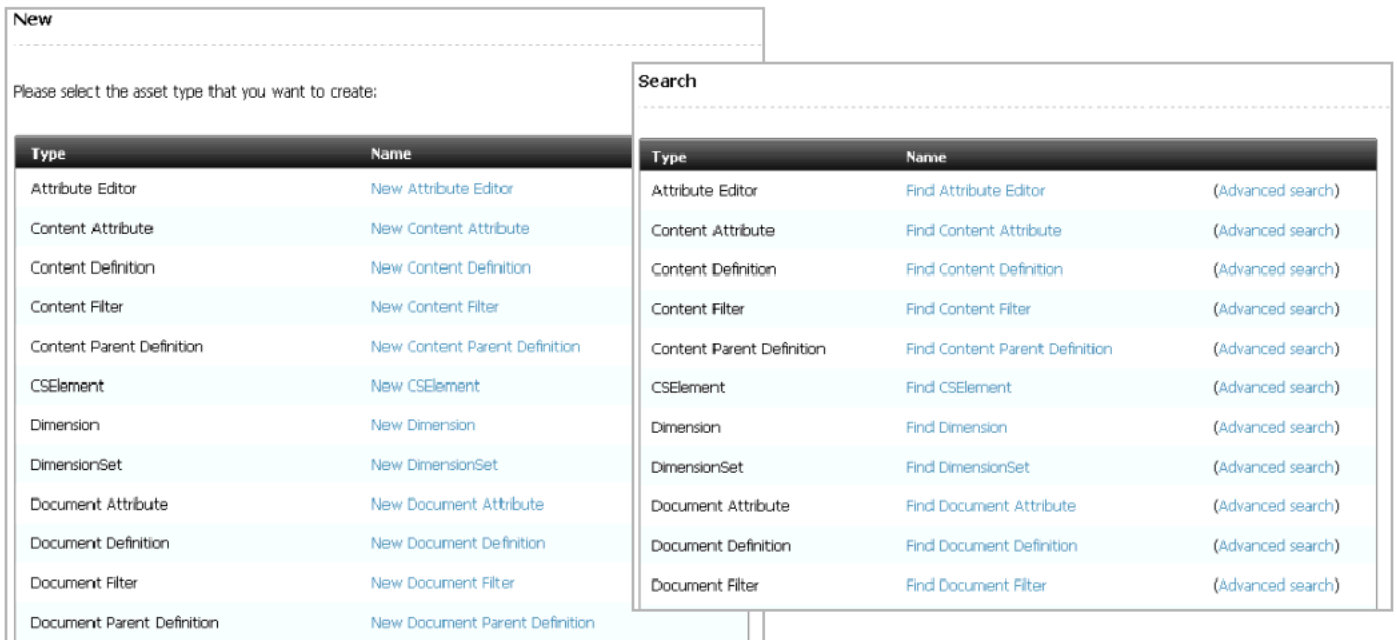
- Start menus, which determine whether the users can create, search for, and edit assets of certain types on the site to which the users are assigned.
- Permissions to assets (that are not part of a workflow process). For such assets, you can create policies that define which roles have permissions to perform which functions on assets.
- Permissions to WebCenter Sites' tree. You can control users' permissions to the tree, tabs in the tree, nodes in the tabs, and items in the nodes.

Roles are the components that you invoke in order to associate users with interface functions and the content that the users must manage.

9.2 Managing Access to Asset Types via Start Menus

A start menu is a list of hyperlinked asset types, as [Figure 9-1, "New and Search Start Menu in the Admin Interface"](#) shows:

Figure 9–1 New and Search Start Menus in the Admin Interface



Clicking an asset type name opens a form for either creating an asset of the selected type, or searching for assets of the selected type. For example, clicking an asset type in a New start menu opens a form for creating an asset of the selected type. Clicking an asset type in the Search start menu opens a form that is used to search for and edit assets of the selected type.

The start menus shown in [Figure 9–1, "New and Search Start Menus in the Admin Interface"](#) are accessible from the Admin Interface by the user clicking the New and Search buttons, at the top left of the interface. You, as the administrator, grant a user permission to a start menu item by assigning the same role to the user and the start menu item.

This section contains the following topics:

- [Section 9.2.1, "Start Menu Items"](#)
- [Section 9.2.2, "Creating Start Menu Items"](#)

9.2.1 Start Menu Items

A start menu item is an option in a start menu. A start menu item names the type of asset a user can create, copy, or search for. For example, the New start menu shown below belongs to the Admin Interface and displays 16 start menu items to the user who is authorized to work with the items. The user can create assets of the types that are listed in the menu.

Figure 9–2 New Form, Showing Start Menu Items

The screenshot shows a web form titled "New" with a dashed line separator. Below the title, it says "Please select the asset type that you want to create:". A table lists various asset types and their corresponding names. An arrow labeled "start menu item" points to the first row of the table.

Type	Name
Attribute Editor	New Attribute Editor
Content Attribute	New Content Attribute
Content Definition	New Content Definition
Content Filter	New Content Filter
Content Parent Definition	New Content Parent Definition
CSElement	New CSElement
Dimension	New Dimension
DimensionSet	New DimensionSet
Document Attribute	New Document Attribute
Document Definition	New Document Definition
Document Filter	New Document Filter
Document Parent Definition	New Document Parent Definition

In the Admin Interface, start menu items are displayed in a list when users click New or Search in the button bar.

You configure the start menu items to determine the following about the new assets that are created with them:

- Which roles on which sites have access to the start menu item. Specifying roles controls which content providers can create or search for assets of this type on the specified sites.
- Field/value pairs. You can supply values for certain fields so that those fields are automatically filled in whenever a content provider creates a new asset with the shortcut.

For example, for basic assets, it might be useful to have the start menu item set a pre-determined value for the **Dimension, Description, End Date, Start Date, Subtype, Template** or **User** fields (or some combination). If your basic assets have associations that can be different based on the subtype of the asset, be sure to create start menu items that set a subtype for new assets so that the only the appropriate associations appear in the New form.

For flex assets, it is useful to have start menu items that set the flex definition for new flex assets so the content providers have fewer steps.

- Which workflow process to assign by default to new assets created with the start menu item and the users who participate in that workflow process.

Start menu items also determine whether certain operations can be performed on an asset type that is listed in a tree tab (in the Admin interface). If an asset type is displayed on a tree tab, but there is no start menu item created for it, the only options

listed in the right-mouse menu for the asset type are Refresh, Edit, and Inspect. To have the New option included in the right-mouse menu, a New start menu item must be created for the asset type.

Start Menu Items for Different Kinds of Users

If you have different groups of content providers who create different types of content, you could create start menu items for each type.

For example, suppose that you have a group of writers who create business articles. You could create a start menu item that creates a new article, sets the Category to Business, sets the template to the correct template for articles on the business page, and places it in a workflow for business articles. Additionally, if you create a role for business writers, you can configure the start menu item so that only users with the business writer role can use this start menu item.

Start menu Items for Flex Assets

For flex assets, you can set the definition of a flex asset with the start menu item.

For example, suppose that your public site were a catalog of household goods and that your flex assets were products. You could create a start menu item named "Toaster" that sets the product definition to "toaster" for new toaster products, a start menu item named "glassware" that sets the appropriate product definition for new glassware products, and so on. A start menu item that sets the flex definition in this way eliminates a step for your content providers, thereby reducing the chance that they will select the wrong flex definition by mistake.

9.2.2 Creating Start Menu Items

Managing user's access to asset types on a site entails creating start menu items for the asset types. Start menu items determine which users can create and search for assets of specific types on a specific site. WebCenter Sites supports four kinds of start menu items:

- **New** – creates a link for the asset type on the list that is displayed when you click **New** in the button bar. If you do not create **New** items for the asset types that are enabled for a site, no one can create assets of those types. For information about creating **New** start menu items, see [Section 9.2.2.1, "Creating New Start Menu Items."](#)
- **Search** – creates a link for the asset type on the list that is displayed when you click **Search** in the button bar. If you do not create **Search** items for the assets that are enabled for a site, no one can search for and edit assets of those types in any of the WebCenter Sites interfaces. For information about creating **Search** start menu items, see [Section 9.2.2.2, "Creating Search Start Menu Items."](#)
- **Sites Desktop** – makes the asset type available to content providers who use the Sites Desktop interface instead of one of the browser-based interfaces. For information about creating Sites Desktop start menu items, see [Section 23.2.2, "Configuring the Word Asset Types for Sites Desktop."](#)
- **Sites DocLink** – makes the asset type available to content providers who use the Sites DocLink interface instead of one of the browser-based interfaces. For information about creating Sites DocLink start menu items, see [Section 23.3.2, "Configuring the Sites DocLink Asset Types."](#)

This section contains the following topics:

- [Section 9.2.2.1, "Creating New Start Menu Items"](#)

- [Section 9.2.2.2, "Creating Search Start Menu Items"](#)
- [Section 9.2.2.3, "Editing Start Menu Items"](#)
- [Section 9.2.2.4, "Creating Start Menu Items for Sites Desktop"](#)
- [Section 9.2.2.5, "Creating Start Menu Items for Sites DocLink"](#)
- [Section 9.2.2.6, "Displaying a List of All Start Menu Items"](#)

9.2.2.1 Creating New Start Menu Items

Before you begin creating start menu items, note the following:

- Create start menu items for only the asset types that are enabled for that site. If the assets are not enabled, go back to [Chapter 8, "Assembling Content Management Sites,"](#) and complete the steps in [Section 8.3.1, "Enabling Asset Types for a Site."](#)
- If you want to specify a default workflow process, create the workflow process first. For information about workflow, see [Chapter 10, "Creating and Managing Workflow Processes"](#).
- Talk with your developers to ascertain which fields are used by queries, collections, or other design elements for the online site. That way, you can specify useful default values for those fields.

Note: Content Query assets are not supported within the Admin UI.

- Talk with your developers about specific dates when the site will change so that you can create start menu items with appropriate default start and end dates for Site Preview. Also note that for published assets, the end date determines when the assets automatically expire from the web site. Speak with developers about their needs when formulating end date defaults.
- Talk with your developers to find out which templates are appropriate for the various subtypes of asset types that you are creating start menu items for.
- You **cannot** set values for the following kinds of fields:
 - An upload field or any field that writes data to a URL column.
 - Fields that accept more than one value.
 - Association fields.
 - Flex attribute fields for flex and flex parent assets.

To illustrate, the following asset types cannot have set values in the Start Menu:

- Visitor Attribute
- CS Element
- Dimension
- Dimension Set
- Flex Attribute
- Flex Child Definition
- Flex Filter
- Flex Parent Definition

- History Attribute
- History Definition
- Page Attribute
- Page Definition
- Page Filter
- SiteEntry
- Template
- Attribute Editor

To create a New start menu item

1. In the **Admin** tab, expand **Start Menu**, then double-click **Add New**.
2. In the Start Menu form, do the following:

Figure 9–3 Start Menu Form

Start Menu

Name:

*Asset Type:

- Article
- Article Category
- Attribute
- Attribute Editor
- CSElement
- Collection
- Content
- Content Attribute
- Content Definition
- Content Filter

- a. In the **Name** field, enter the name of the item, using up to 64 characters. Although it is not required, it is a good idea to start the name with the word "New" (for example, "New Article" or "New Template"). This is to help the user later realize that the start menu item is for a new asset.

The name used in the New field will be displayed on the New and Search forms if the Description field (described in step 4) is left blank.

- b. In the **Asset Type** field, select the type of asset for which you are creating the Start Menu item.
- c. Click **Continue**.

3. If you have chosen a flex asset type, the Flex Definition field will appear. If you did not select a flex asset type, skip to step 4.
 - a. In the **Flex Definition** field, select the asset subtype.
 - b. If the form does not automatically reload, click **Continue**.
4. In the next Start Menu form, fill in the fields as explained below:

Figure 9-4 Start Menu Form

Start Menu

***Name:**

Description:

Type:

Asset Type: [Article](#)

Flex Definition: Article

Default Values

Field:

Value:

***Sites :**

- Any
- FirstSite1
- AdminSite
- avisports
- LoafersBakery

***Roles:**

- Any
- AdvancedUser
- Analytics
- Approver
- ArtworkAuthor
- ArtworkEditor
- ContentAuthor
- ContentEditor

Workflow Process:

- a. In the **Description** field, enter a short description, using up to 255 characters. The name entered in the Description field will be displayed on the New and Search forms.
- b. In the **Type** field, select **New**.
- c. Note that the **Asset Type** field is an information-only field that displays your choice of asset type from the previous form.
- d. (Optional) In the **Default Values** section, complete the following steps if you want field values to be set automatically when a content provider creates an asset with this start menu item.
 - a. Select a field from the **Field** drop-down list. This list displays all the fields for assets of this type; for example, category, source, and template.
 - b. In the **Value** field, enter or select a value.

Note: All values are case-sensitive.

WebCenter Sites does not validate the value against the field that you are setting it for. Therefore, you must be sure that you have entered an appropriate value.

- c. Click the **Add** arrow to add the value to the field.
- d. Repeat the previous three steps, as necessary.

Note: When you set a field value through a start menu item, the field does not become a read-only field. That is, no matter what value you set in the Start Menu form, that value can be overridden by the content provider who creates an asset through the start menu item. The values that you specify are for convenience only.

- e. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl-Click** to select more than one site.
- f. In the **Roles** field, select the roles that can have access to the start menu item. Use **Ctrl-Click** to select more than one role.

Note: If you are assigning a default workflow with this start menu item, the roles that you select in this step must match the roles that are (or will be) authorized for the start step in the workflow process.

- g. (Optional) To configure workflow process details for assets created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the Select Workflow form, select the appropriate workflow process from the drop-down list and then click **Select Workflow**.
 - c. In the Set Participants form, select at least one user for each role that appears, and click **Set Participants**.
 - d. Click **Continue**.

The system saves the workflow information and re-displays the Start Menu form.

5. Click **Save**.

9.2.2.2 Creating Search Start Menu Items

To create a Search start menu item

1. In the **Admin** tab, expand **Start Menu**, then double-click **Add New**.
2. In the Start Menu form, do the following:

Figure 9–5 Start Menu Form

Start Menu

Name:

*Asset Type:

- Article
- Article Category
- Attribute
- Attribute Editor
- CSElement
- Collection
- Content
- Content Attribute
- Content Definition
- Content Filter

- a. In the **Name** field, enter the name of the item, using up to 64 characters. Although it is not required, it is a good idea to start the name with the word "Find" (for example, "Find Article" or "Find Template"). This is the name that is displayed on the **Search** list.
- b. In the **Asset Type** field, select the type of asset for which you are creating the Start Menu item.
- c. Click **Continue**.
3. If you have chosen a flex asset type, the Flex Definition field will appear. If you did not select a flex asset type, skip to step 4.
 - a. In the **Flex Definition** field, select the asset subtype.
 - b. If the form does not automatically reload, click **Continue**.
4. In the next Start Menu form, fill in the fields as explained below:

Figure 9–6 Start Menu Form

Start Menu

***Name:**

Description:

Type:

Asset Type: [Article](#)

Flex Definition: Article

Default Values

Field:

Value:

***Sites :**

Any

FirstSite1

AdminSite

avisports

LoafersBakery

***Roles:**

Any

AdvancedUser

Analytics

Approver

ArtworkAuthor

ArtworkEditor

ContentAuthor

ContentEditor

Workflow Process:

5. In the **Description** field, enter a short description, using up to 255 characters.
6. In the **Type** field, select **Search**.

7. Note that the **Asset Type** field is an information-only field that displays your choice of asset type.
8. (Optional) In the **Default Values** section, complete the following steps if you want field values to be set automatically when a content provider creates an asset with this start menu item.
 - a. Select a field from the **Field** drop-down list. This list displays all the fields for assets of this type; for example, category, source, and template.
 - b. In the **Value** field, enter or select a value.

Note: ■All values are case-sensitive.

WebCenter Sites does not validate the value against the field that you are setting it for. Therefore, you must be sure that you have entered an appropriate value.

- c. Click the **Add** arrow to add the value to the field.
- d. Repeat the previous three steps, as necessary.

Note: When you set a field value through a start menu item, the field does not become a read-only field. That is, no matter what value you set in the Start Menu form, that value can be overridden by the content provider who creates an asset through the start menu item. The values that you specify are for convenience only.

9. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl-Click** to select more than one site.
10. In the **Roles** field, select the roles that can have access to the start menu item. Use **Ctrl-Click** to select more than one role.

Note: If you are assigning a default workflow with this start menu item, the roles that you select in this step must match the roles that are (or will be) authorized for the start step in the workflow process.

11. (Optional) To configure workflow process details for assets created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the Select Workflow form, select the appropriate workflow process from the drop-down list and then click **Select Workflow**.
 - c. In the Set Participants form, select at least one user for each role that appears, and click **Set Participants**.
 - d. Click **Continue**.

The system saves the workflow information and re-displays the Start Menu form.

12. Click **Save**.

9.2.2.3 Editing Start Menu Items

To edit a Start Menu item

1. In the **Admin** tab, expand **Start Menu**, then expand the node the desired Start Menu item is located under. Double-click the desired Start Menu item.

The Start Menu Item form appears.

2. Click Edit.

The Start Menu form with the stored information for the item displays.

Figure 9–7 Start Menu Form

Start Menu

***Name:**

Description:

Type:

Asset Type: [Article Flex](#)

Flex Definition: [Any]

***Sites :**

***Roles:**

ID: 1330554014955

3. Make the changes to the item.
4. Click Save.

9.2.2.4 Creating Start Menu Items for Sites Desktop

For information about creating Sites Desktop start menu items, see [Section 23.2.2, "Configuring the Word Asset Types for Sites Desktop"](#).

9.2.2.5 Creating Start Menu Items for Sites DocLink

For information about creating Sites DocLink start menu items, see [Section 23.3.2, "Configuring the Sites DocLink Asset Types"](#).

9.2.2.6 Displaying a List of All Start Menu Items

In addition to the individual start menu nodes that appear for each of the asset types that exist on your WebCenter Sites system, there is a top-level Start Menu node in the Admin tab. Use this node when you want to see a list of all the start menu items in the system.

Additionally, there is a link to the list of all the start menu items from the Inspect form for any individual start menu item.

9.3 Setting Access Permissions to Assets

WebCenter Sites provides a system by which you can set access permissions to assets that are not part of a workflow process. For such assets, you can create policies and, within them, define which roles have permissions to perform which functions on assets. For example, a policy for the Copy function might specify which roles are allowed to copy an asset. A different policy for the Copy function might specify which roles are denied the ability to copy an asset. For flex assets, inheritable policies can be set on parents, meaning that the policies will be inherited by the children.

Functions that can be performed on assets, by default, are the following: checkout, copy, edit, delete, rollback, share, approve, and build. (For definitions of the functions, see [Section 10.2.6, "Determine the Function Privileges"](#)).

All of the above-listed functions are defined in the `futuretense_xcel.ini` file, as values of the property `xcelerate.authorizefunction` (located in the **Authorization** tab, if you are using the Property Editor). You can specify additional functions (which are listed in the **Authorization** tab).

Access permissions serve a three-fold purpose:

- Extend the workflow function privileges system, making it possible to set permissions for assets when they are not in workflow
- Introduce new functions (authorize, inspect, and preview) for viewing the contents of assets that are not in workflow
- Introduce a hierarchical permission inheritance model for assets that are not in workflow

Note: The access permissions feature does not apply to other entities, such as pagelets.

This section contains the following topics:

- [Section 9.3.1, "Permission Structure"](#)
- [Section 9.3.2, "Types of Authorization"](#)
- [Section 9.3.3, "Summary"](#)
- [Section 9.3.4, "Setting Access Permissions"](#)

9.3.1 Permission Structure

Permission structure determines which functions (for example, edit and view) a role can perform. A user's permissions are managed by assigning him the roles that can perform the functions. WebCenter Sites implements a three-level permission structure: Grant, Deny, and Inherited. Grant and Deny denote the explicit permissions, that is, they grant or deny the permission to perform a function. Inherited means that permissions are neither granted nor denied. Instead, the permissions are inherited from the parent asset's policy, if one exists. If the parent's policy does not exist, then the asset's permissions are inherited from the policy of the parent's parent, and so on.

If permissions are ambiguous, the following considerations may help resolve the ambiguity:

- A permission is granted if at least one of the user's roles grants the permission.
- Grant overrides Deny. If one role grants a permission, but another role denies the same permission, the permission is granted.
- A permission is denied if at least one of the user's roles denies the permission, and none of the user's roles grant the permission.

The above scenarios are summarized in the following table:

Table 9–1 Permission Structure

Role 1	Role 2	Result
Granted	N/A	Granted
Granted	Denied	Granted
Denied	N/A	Denied

- Assets can inherit permissions from parent assets.
For example, if a parent asset named "Money Market Funds" grants the delete permission to a role named `Editor`, and user Bob has that role, then Bob can delete the "Money Market Funds" asset and all of its child assets.
- Asset permissions can override the parent assets' permissions.
A "Prime Money Market semi-annual report" asset denies the delete permission to a certain role. That same role is granted the delete permission in the "Prime Money Market" parent asset. This means that Bob cannot delete the report, even though the parent asset grants the delete permission.
- If there are no applicable permissions found on a particular asset, then the administrator needs to examine the `futuretense_xcel.ini` file for this permission. In the `futuretense_xcel.ini` file, permissions for a particular function are set for all assets, rather than for a single asset.
For more information about setting permissions in the `futuretense_xcel.ini`, see ["To set access permissions from futuretense_xcel.ini"](#) on page 9-21.
- Asset permissions overrides `futuretense_xcel.ini` settings.
For example, if there is a permission `xcelerate.copy.grant` in the `futuretense_xcel.ini` file and the permission to delete asset A is revoked, then the permission to delete asset A is denied.

9.3.2 Types of Authorization

Three types of authorization can be defined on a particular function and a particular asset: workflow function privileges, access permissions, and authorization properties in `futuretense_xcel.ini`, all independent of each other. (For information about workflow function privileges, see [Chapter 10, "Creating and Managing Workflow Processes."](#))

When a user attempts to perform a function on an asset, the system performs a series of authorization checks in the following order, for that function:

For assets in workflow:

The system checks the workflow function privileges for the function in the workflow process in which the asset is entered.

1. If no function privilege is defined for the function, the system checks the access permission policy for that function, as defined in the asset's access permissions interface.
2. If no access permission is set for the function (including inheritance of the parent's policy), then the system checks the authorization properties in `futuretense_xcel.ini`.
3. If none of the above types of authorization are defined, permission to perform the function on the asset is granted, by default.

Note: The above types of authorization are checked in the order shown. If two or more types of authorization are defined, the authorization that is found first overrides all other types of authorization. For example, for a specific function such as Edit, if both workflow function privileges and access permissions are defined on an asset, then function privileges on Edit override access permissions on Edit.

For assets that are not in workflow:

1. The system checks the access permissions policy for the function, as defined in the asset's access permissions interface.
2. If no access permission is set for the function (including inheritance of the parent's policy), then the system checks the authorization properties in `futuretense_xcel.ini`.
3. If none of the above levels of authorization are defined, permission to perform the function on the asset is granted, by default.

9.3.3 Summary

The different types of authorization described above are independent of each other, which means that only one type of authorization can apply to an asset at any given time for any given function. The following guidelines can help you decide which type of authorization to use:

1. If a function does not require authorization, do not use any of the authorization types. All roles will have permission to perform that function.
2. If you need a simple authorization policy across multiple assets, set authorization properties in `futuretense_xcel.ini`.

3. If you need a simple authorization policy on a particular asset, set access permissions.
4. If you need a more complex type of authorization that allows workflow to control the permissions on an asset, set workflow function privileges.

9.3.4 Setting Access Permissions

Setting access permissions means creating a policy for an asset by specifying which roles can perform which function(s) on the asset.

There are several ways to set access permissions on assets:

- Using the administrator's interface, to set permissions on individual assets
- Using the `futuretense_xcel.ini` file, to set permissions on all assets at once

Note: If your asset is entered in to a workflow for which function privileges are defined, the function privileges override the asset's access permissions. For information about function privileges, [Chapter 10, "Creating and Managing Workflow Processes."](#)

This section contains the following topics:

- [Section 9.3.4.1, "Asset Toolbar"](#)
- [Section 9.3.4.2, "Setting Access Permissions from the Administrator's Interface"](#)
- [Section 9.3.4.3, "Setting Access Permissions from Property Files"](#)

9.3.4.1 Asset Toolbar

When an asset is viewed, a toolbar appears at the top to aid in working with the asset. This is where you can perform certain function with each asset.

Figure 9–8 Asset Toolbar



The icons in the toolbar perform the following actions, described from left to right:

- **Edit:** to open the Edit form for the asset.
- **Delete:** Click to delete the asset.
- **Approve for Publish:** Click to approve the asset for publication the next time publication for this asset occurs.
- **Share Content:** Click to share the content with other sites as enabled by the system administrator.
- **Add to My Bookmarks:** Click to add a bookmark to this asset in the **Bookmarks** tab. If the asset is already added to the **Bookmarks** tab, it will be unavailable.
- **Status:** Click to open a form detailing the workflow approval state of the asset.
- **Access Permissions:** Click to open the Access Permissions form for the asset. See [Section 9.3.4.2, "Setting Access Permissions from the Administrator's Interface"](#) for more information.

- **Drop-down menu:** this menu allows you additional ways to work with the asset, including using the asset to copy and create a new asset, as well as create a wholly new asset and search for other assets.

9.3.4.2 Setting Access Permissions from the Administrator's Interface

To set access permissions from the Administrator's interface

1. Locate the asset for which you will set access permissions.
2. In the asset's Inspect form, select **Access Permissions** from the icon menu bar.
3. Select one of the radio boxes, click **View by role** or **View by function**. In this example, **View by Function** is used.

Figure 9–9 Attribute Editor Form

Attribute Editor:CKEditor

Name: CKEditor

Description: CKEditor

Status: Created

Created: Monday, February 12, 2007 8:19:37 AM PST by firstsite

Modified: Monday, February 12, 2007 8:19:37 AM PST by firstsite

Permissions: View by role
 View by function

Functions	Roles		
	Inherited	Granted	Denied
Approve for F Build Checkout Copy Delete Edit Inspect Preview Rollback Share Assets			
		<input type="button" value="Grant"/> <input type="button" value="Deny"/>	
		<input type="button" value="Remove"/>	<input type="button" value="Remove"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>			

Note: In this procedure, we are using View by function, which allows you to select a function and then specify the roles that are granted or denied permission to perform it. If you select View by role, the order will be reversed: you will have to select the role first, then specify the functions it is granted or denied.

4. Select a function.

Roles that can perform the function are now displayed in the Inherited list box, by default. Inherited refers to the permissions policy that the asset inherits from its parent, or its parent's parent, and so on, depending on which asset in the hierarchy has a policy.

5. Select a role (or roles) to which you will deny or grant the permission to perform the function. (To select a block of roles, **Shift-Click** the extremes of the block. To select non-adjacent roles, **Ctrl-Click** each role).

6. Select either **Grant** or **Deny** to allow or forbid the role(s) to perform the function:

- **Grant** explicitly allows the selected role(s) to perform the selected functions on the specific asset. All other roles are denied permission to perform the selected function on that asset unless explicitly granted permission.
- **Deny** explicitly forbids the selected role(s) to perform the selected function on the specific asset. All other roles are denied permission to perform the selected function on that asset unless explicitly granted permission
- Selecting neither **Grant** nor **Deny** leaves the roles in the Inherited list box to indicate that access permissions are inherited.

7. Repeat steps 4–6 for each function and role that must be granted or denied or which you want to grant or deny permission.

When you have completed selecting functions and roles, your access permissions form will look similar to the one below:

Figure 9–10 Attribute Editor Form

Attribute Editor:CKEditor

Name: CKEditor

Description: CKEditor

Status: Created

Created: Monday, February 12, 2007 8:19:37 AM PST by firstsite

Modified: Monday, February 12, 2007 8:19:37 AM PST by firstsite

Permissions: View by role
 View by function

The screenshot shows the 'Attribute Editor:CKEditor' form with the 'View by function' option selected. The 'Functions' list on the left includes 'Inspect', which is highlighted. The 'Roles' section on the right is divided into three columns: 'Inherited', 'Granted', and 'Denied'. The 'Inherited' column lists roles like Analytics, Approver, Designer, DocumentAuthor, DocumentEditor, Expert, HelloAuthor, and Marketer. The 'Granted' column lists roles like ArtworkAuthor, ArtworkEditor, Author, Checker, ContentAuthor, ContentEditor, GeneralAdmin, ProductAuthor, ProductEditor, and SiteAdmin. The 'Denied' column lists roles like HelloEditor, Pricer, HelloDesigner, and WorkflowAdmin. There are 'Grant' and 'Deny' buttons between the 'Inherited' and 'Granted' columns, and 'Remove' buttons below the 'Granted' and 'Denied' columns. At the bottom of the form are 'Cancel' and 'Save' buttons.

8. Click **Save** when finished.

Users who now access this asset will be able to perform the functions that their roles allow.

9.3.4.3 Setting Access Permissions from Property Files

To set access permissions from `futuretense_xcel.ini`

In `futuretense_xcel.ini`, either grant or deny permissions by setting the following properties:

- `xcelerate.grant.functionname = rolist`
- `xcelerate.deny.functionname = rolist`

where

functionname is the name of the function, as shown in the `futuretense_xcel.ini` file, on the **Authorization** tab if you are using the Property Editor.

rolist is a comma-separated list of roles for which the permission is either denied or granted.

If the permission is granted or denied in the `futuretense_xcel.ini`, it is granted or denied for all assets. These permissions can be overridden by specifically setting access permissions on an asset-by-asset basis, using WebCenter Sites' access permissions feature.

9.4 Configuring Other Options for Asset Types

In the Admin tab, there are several other items that configure asset types. These items contribute to the design and implementation of the asset types, which means that they are typically used by the developers. Administrators typically do not manage the items.

However, after an asset type is designed and installed, you might be asked to maintain the following items:

- sources
- categories
- subtypes
- associations
- mimetypes

For information about these items and procedures for creating and editing them, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

9.5 Managing Access to the Tree (Administrator's interface only)

You have several ways to configure the tree that is displayed in the left-hand panel of the administrator's interface:

- You can configure the tree to be either displayed by default in the main window or hidden by default.
- You can control whether users who toggle a tree off are allowed to toggle the tree back on.
- You can configure tabs within the tree, adding items or functionality to them, and determining which users have access to them based on their roles.
- You can configure the tabs' nodes to specify the number of items that are displayed under any given node.

WebCenter Sites supports two general categories of tree tabs:

- System default tabs, which provide administrative functionality (**Admin**, **Site Admin**, **Workflow**) and access to individual assets (**History**, **Bookmarks**, **Site Plan**).

Assets and asset types intended for developers (for example, Template) are either available only for inspection or inaccessible in the Contributor interface.

Conversely, the following assets are available only through the Contributor interface: All flex assets and their parent assets, all basic assets, Engage assets (such as recommendations, segments and promotions), Query, Collection, and Page assets.

- Custom tabs
 - Tabs that you create to make it more convenient for users in different roles to create the most commonly used asset types.

- Tabs that developers create to offer customized behavior or new functionality.

Note: The Workflow Groups tab is a hybrid between a system default and a custom tab. Although the Workflow Groups tab is not installed by default and you must create it to use it, the element that provides its logic is installed. That is, while you must manually create this tab, your developers do not have to code a new element for you.

This section describes how to add roles to the system default tabs, how to create basic tree tabs that provide access to asset types, and how to configure whether the tree is displayed by default.

For information about creating custom tree tabs that implement new functionality, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

This section contains the following topics:

- [Section 9.5.1, "Displaying and Hiding the Tree"](#)
- [Section 9.5.2, "Creating Tree Tabs"](#)
- [Section 9.5.3, "Editing Tree Tabs"](#)
- [Section 9.5.4, "Deleting Tree Tabs"](#)
- [Section 9.5.5, "Changing Tree Tab Order"](#)
- [Section 9.5.6, "Configuring the Number of Items Displayed Under Nodes on Tabs"](#)

9.5.1 Displaying and Hiding the Tree

Any user can toggle the tree on and off whenever he needs less or more space in the main window to display the assets that he is working on.

If you want, you can configure your WebCenter Sites system so that the tree is toggled off by default when users first log in to the system. You can also control whether users are allowed to toggle the tree back on.

The WebCenter Sites features for working with assets are available in right-mouse menus on the tabs as well as through the icons at the top of the window and the action bars displayed in the New, Edit, and Inspect forms for individual assets managed solely in the Admin interface. Therefore, it is not necessary to display the tree for content providers. However, because the administrative functions are available only through the Admin, Site Plan, and Workflow tabs, administrative users must always have access to the tree.

To configure display of the tree and toggling rights

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. Select the **Preference** tab.
3. To specify that the tree should be toggled off by default when users first log in to the administrator's interface, set the value of the `xcelerate.showSiteTree` value to `false`.
4. (Optional) To configure the system so that only administrative users (that is, users who have the `xceladmin` ACL assigned to their user accounts) are allowed to toggle the tree back on, set the `xcelerate.restrictSiteTree` property to `true`.
5. Save the property file and close the Property Editor.

6. Restart the application server.

9.5.2 Creating Tree Tabs

A tree tab is a tab (such as Admin) that appears in the left-hand panel of the administrator's interface. Access to tree tabs is controlled through roles.

Several default tree tabs are core to WebCenter Sites. Before creating a new tree tab, review the default tabs to gain an understanding of WebCenter Sites' existing capabilities. The default tree tabs are:

- **Bookmarks** – displays the assets that you selected and placed in your bookmarks. All users see this tab as soon as they use the **Add to My Bookmarks** button to place an asset in their bookmarks.
- **Admin** – displays the administrative functions that affect all of the sites in the system. By default, only users with the default system role named GeneralAdmin have access to this tab.
- **Design** — serves as the source for creating pages on your site. Some of these sources are: Templates, Product Definition, Content Definition, and other sources for the creating pages.
- **History** – displays the assets that you worked with during the current session. All users see this tab as soon as they create, inspect, edit, or copy their first asset. **This tab is not configurable.**
- **Site Admin** – holds a subset of the administrative functions that apply only to the site that you are currently logged in to. By default, only users with the default system role named SiteAdmin have access to this tab. This tab is useful if you have individuals who manage which existing users have access to individual sites, but who do not need to create new users or new sites.
- **Site Plan** – displays a graphical and hierarchical representation of the pages, collections, queries, and content assets that make up the online site that you are making available on the delivery system. By default, all of the FirstSiteII sample site roles and system default roles grant access to this tab.
- **Workflow** – has the workflow configuration functions. By default, only users with the Workflow Admin role have access to this tab.

This section contains the following topics:

- [Section 9.5.2.1, "Creating a New Tree Tab"](#)
- [Section 9.5.2.2, "Creating the 'Workflow Groups' Tab"](#)

9.5.2.1 Creating a New Tree Tab

















Note: Before creating a tree tab, familiarize yourself with the default tree tabs to avoid duplicating their functions. The default tree tabs are described earlier.

To create a new tree tab

1. In the **Admin tab**, double-click **Tree**.
2. In the Tree Tabs form, click **Add New Tree Tab**.

Figure 9–11 Tree Tabs Form

Tree Tabs

	Title	Tooltip
	Admin	Administrative Functions
	Articles	
	Artwork	Artwork
	Bookmarks	Bookmarks
	Content	Content
	Design	Administrative Assets
	Dev	Developer Assets
	Documents	Documents
	Marketing	Marketing
	Media	Media
	Products	Products
	Site Admin	Site Administration
	Site Design	Site Design
	Site Plan	Site Plan Tab
	Visitors	Visitors
	Workflow	Workflow Administration

Add New Tree Tab

Order Tree Tabs

- Fill in the Add New Tree Tab form:

Figure 9–12 Add New Tree Tab Form

Add New Tree Tab

***Title:**

Tooltip:

***Sites:**
 FirstSite1
AdminSite
 avisports
 LoafersBakery

***Required Roles:**
 AdvancedUser
 Analytics
 Approver
 ArtworkAuthor

Tab Contents:

<p>Available</p> <input type="list" value="Article"/> Article Category Attribute Attribute Editor CSElement Collection Content Content Attribute	<p>Selected</p> <input type="list" value=""/> Display Order <input type="button" value="▲"/> <input type="button" value="▼"/>
<input type="button" value="Add Selected Items"/>	<input type="button" value="Remove"/>

Section Name:

Element Name:

- To create a custom section, enter the section name and element name, click "Add New Section", then Save. When the tree tab is reloaded, this element will be called.
- To edit an existing custom section, select the custom section name (in the Selected area above), make the changes, click Edit, then Save.

- a. In the **Title** field, enter a short, descriptive name of up to 64 characters.
- b. In the **Tooltip** field, enter a short, informative description of the tab, up to 255 characters. This description is displayed when the mouse hovers over the tab.
- c. In the **Site** field, select the sites that will display the tab.
- d. In the **Required Roles** field, select which roles can access the tab.
- e. In the **Tab Contents** field, select the asset types that will be displayed on the tab and then click **Add Selected Items**.

- f. (Optional) If you want to add custom functionality to this tab, use the **Section Name** and **Element Name** fields at the bottom of the form. You will need help from your developers for this step. See also the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for more information.
4. Click **Save**.
The new tab is displayed in the tree (refresh the display, if necessary).

9.5.2.2 Creating the 'Workflow Groups' Tab

The Workflow Groups tab displays the workflow groups that exist for a site and the assets that are included in those groups. For information about workflow groups, see [Section 10.1.6, "Workflow Groups."](#) This tab does not appear by default. If you want to use it, you must create it.

To create the Workflow Groups tab

1. In the **Admin** tab, double-click **Tree**.
WebCenter Sites displays the Tree Tabs form.
2. In the form, click **Add New Tree Tab**.
3. In the **Title** field, enter a short, descriptive name of up to 64 characters.
For example: **Workflow Groups**.
4. In the **Tooltip** field, enter a short, informative description of the tab, up to 255 characters.
For example: **Workflow Groups**.
5. In the **Site** field, select the sites that will display the tab.
6. In the **Required Roles** field, select which roles can access the tab.
7. In the **Tab Contents** area, scroll down to the section area. In the **Section Name** field, enter the following name exactly as it appears here: **Groups**.
8. In the **Element Name** field, enter the following name exactly as it appears here:
OpenMarket/Gator/UIFramework/LoadWorkflowGroups
9. Click **Add New Section**.
The word Groups appears in the **Selected** column.
10. Click **Save**.

9.5.3 Editing Tree Tabs

Editing a tree tab means that you can:

- Rename the tree tab
- Edit its tooltip
- Enable the tree tab for a different set of sites
- Display the tree tab to a different set of roles
- Select a different set of content for the tree tab, and/or rearrange the content's display order

Note: ■ A tree tab is displayed in the user's interface when two conditions are met:

The tree tab is enabled on the user's site, and Both the tree tab and the user have at least one role in common.

- When you create site users with new roles, you must select the new roles in the default tree tabs so that the users have access. (Default tree tabs are not automatically enabled for new roles.) The list of default tabs can be found in [Section 9.5.2, "Creating Tree Tabs"](#).
-
-

To edit a tree tab

1. In the **Admin** tab, double-click **Tree**.
2. In the Tree Tabs form, navigate to the desired tab and click its **Edit** icon.
3. In the Edit Tree Tab form, make the necessary changes.

For example:

- If you wish to enable the tree tab for a new site, select the site from the **Sites** list.
 - If your new site has users with new roles, and you wish to enable the tree tab for those users, select the users' roles from the **Roles** list.
4. Click **Save**.

9.5.4 Deleting Tree Tabs

To delete a tree tab

1. In the **Admin** tab, double-click **Tree**.
2. In the Tree Tabs form, navigate to the desired tree tab and click its **Delete** icon.
WebCenter Sites displays a warning message.
3. Click **Delete Tree Tab**.

The tree tab has been deleted.

9.5.5 Changing Tree Tab Order

You can change the order in which the tree tabs are displayed.

To change the order of tree tabs

1. In the **Admin** tab, double-click **Tree**.
2. At the bottom of the Tree Tabs form, click **Order Tree Tabs**.

WebCenter Sites displays the Order Tree Tabs form:

Figure 9–13 Order Tree Tabs Form**Order Tree Tabs**

Tab Order:

Site Plan
Admin
Site Admin
Workflow
Bookmarks
Marketing
Site Design
Products
Content
Artwork
Documents
Visitors
Hello Content
Hello Design
Product

Display Order

▲

▼

Cancel Save

The form lists the tabs in the order in which they will appear in the tree.

The tree arranges its tabs into rows. When there are more tabs than a single row can accommodate, an additional row is formed behind it. The tabs in each row are ordered from left to right. Rows are ordered front-to-back.

3. To move a tab in the tree, select the tab from the list and then use the arrow buttons to move the tab to the correct position. Repeat this step for each tab that you want to reposition.
4. Click **Save**.

The new tab order is displayed in the tree (refresh the display, if necessary).

9.5.6 Configuring the Number of Items Displayed Under Nodes on Tabs

To optimize WebCenter Sites' response time, you may want to limit the number of assets that are listed under their respective node in a tree tab. Specifically, when the WebCenter Sites database holds many assets of any one type, WebCenter Sites can take a long time to load all the assets and then list them under their node in the tab. To minimize the loading time, you can limit the number of items to be listed under the node by setting a threshold. Then, when the number of assets (or other items) for a node exceeds the threshold, and a user attempts to access the assets, the administrator's interface prompts the user to enter a search criterion that restricts the number of items to be returned.

For example, suppose that a node contains more than 300 articles, and the display threshold is set to 100. When a user expands the node icon that represents the articles, WebCenter Sites displays a small search criteria form that prompts the user to enter values which WebCenter Sites uses to limit the number of articles that it returns.

To configure the number of items under a node in a tree tab

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. Select the **Preference** tab.
3. Select the `xcelerate.treeMaxNodes` property and specify a value for it. By default, this property is set to 100, which means that up to 100 items can be displayed under a node.
4. Save and close the property file.
5. Restart the application server.

Creating and Managing Workflow Processes

Workflow is a feature provided by WebCenter Sites that you use to manage the work on an asset when more than one person participates in its creation. For example, if assets of a certain type must be reviewed by an editor or a legal representative before it can be approved for publishing, the workflow feature can route those assets to the appropriate people at the appropriate time.

This chapter contains the following sections:

- [Section 10.1, "Overview"](#)
- [Section 10.2, "Planning Your Workflow Processes"](#)
- [Section 10.3, "Configuring Your Workflow Processes"](#)
- [Section 10.4, "Moving Your Work"](#)
- [Section 10.5, "Clearing Workflow Assignments"](#)

10.1 Overview

The end goal for any content provider is to have content assets published. Before an asset can be published, it must be approved for publishing. The workflow feature routes assets through whatever series of tasks you deem necessary in a workflow process that ushers assets from creation to approval.

You can configure a workflow process with as many or as few tasks as necessary to reflect the way the work at your organization is accomplished. You can configure email messages that WebCenter Sites sends to notify people when assets are assigned to them and to remind them that a deadline is approaching or has been missed.

Because there are so many configuration possibilities, it is typical to create a separate workflow process for each asset type that you plan to use workflow with rather than attempt to create one process for more than one asset type.

This section contains the following topics:

- [Section 10.1.1, "Workflow Participants"](#)
- [Section 10.1.2, "Workflow States"](#)
- [Section 10.1.3, "Workflow Steps: Moving Assets from State to State"](#)
- [Section 10.1.4, "Multiple Paths for the Asset"](#)
- [Section 10.1.5, "Managing Deadlocks"](#)
- [Section 10.1.6, "Workflow Groups"](#)
- [Section 10.1.7, "Delegating and Clearing Assignments"](#)

- [Section 10.1.8, "Placing an Asset in Workflow"](#)
- [Section 10.1.9, "Restricting Access to Assets While They Are in Workflow"](#)
- [Section 10.1.10, "Deadlines"](#)
- [Section 10.1.11, "Scheduling a Deadline Calculation"](#)
- [Section 10.1.12, "How Does a Workflow Process End?"](#)
- [Section 10.1.13, "Roles Required to Configure Workflow Processes"](#)

10.1.1 Workflow Participants

When you begin creating a workflow process, the first general question is this: what are the job titles of the people who work on assets of this type? For example, are they authors, editors, marketers, graphic artists, product managers, lawyers?

The job titles of the people who participate in a workflow process are considered roles in the WebCenter Sites interfaces. Roles describe the function of an individual on a site. When you enable a user for a site, you assign that user the roles that he fulfills for that site.

When you create a workflow process, you determine which roles are appropriate for each task. Then, when an individual asset is going through that workflow process, only the users who have the appropriate role are allowed to complete the task. The individual user who is selected from the pool of users who have the correct role at any point is called a workflow participant.

10.1.2 Workflow States

Next, what are the tasks that are performed for assets of this type? For example, writing, pricing, editing, fact checking, legal review, and so on.

These tasks are called workflow states. A state is a point in the workflow process that represents the status of the asset at that point. For example, writing article, reviewing image, legal review, and so on. Participants complete the work that the state represents while the asset is in that state.

An asset that a participant is working on (or is supposed to be working on) is called an assignment. A user's assignment list is displayed in the My Assignments form in the administrator's interface. An asset appears on a user's assignment list as soon as it enters a state for which the user has a role to fulfill as a workflow participant.

Should an asset in a specific state remain in that state for only a specific amount of time? If so, assign a deadline to the state. You can then configure the workflow process to send email messages that remind a participant when the deadline is approaching or has been missed. These email messages are examples of timed actions.

You can create one or more timed action for each state.

10.1.3 Workflow Steps: Moving Assets from State to State

Next, how does the asset move from state to state? Does a marketing writer send a prospectus asset to a legal reviewer? Does a graphic artist send an image asset to an editor? And then what?

The movement of an asset between states is called a step. Because creating the steps in a process links together states in a specific order, creating the steps in your workflow process is what organizes your process.

When you create a step in a process, you specify which state a step moves the asset from (the From State), which state that step moves the asset to (the To State), and which roles can take the step.

This section contains the following topics:

- [Section 10.1.3.1, "How Steps Work"](#)
- [Section 10.1.3.2, "Step Actions and Conditions"](#)

10.1.3.1 How Steps Work

A step places an asset in a state and notifies participants from the appropriate roles. This operation creates the assignment. You then need a step for the participants from those roles to take that moves the asset to the next state. In other words, the roles notified by the previous step should be the roles authorized to take the next step. For example:

Figure 10–1 Steps In Editing



How does a user take a step in a workflow process? By specifying that he or she has finished the assignment (except for the start step, which is described below). When a user selects the Finish My Assignment option for an asset that is assigned to him or her, that option invokes the step, moves the asset to the next state, and assigns the asset to the next participants.

When more than one participant is assigned an asset in the same state, using the Finish My Assignment option is also referred to as voting. Each participant votes to move the asset to the next state.

The first step in a workflow process is a start step. A start step is one which has no From State. That is, the start step begins the workflow process, moving the asset to the first state in the series of states. This is the step that is invoked when the workflow process is assigned to the asset. Only users who have the roles authorized for the start step can start the workflow process for an asset.

10.1.3.2 Step Actions and Conditions

Step actions are events that occur when the step is completed. For example, when a step occurs, the asset is assigned to a participant. Should the participant be notified that an asset has been assigned to him or her? If so, you can configure a step action to send an email message to the new participant. You can specify one or more step actions for each step.

Another example of a step action is the ApproveForPublish action. It is a default action delivered with WebCenter Sites that approves the asset. Typically you use this action in the final step of a workflow process.

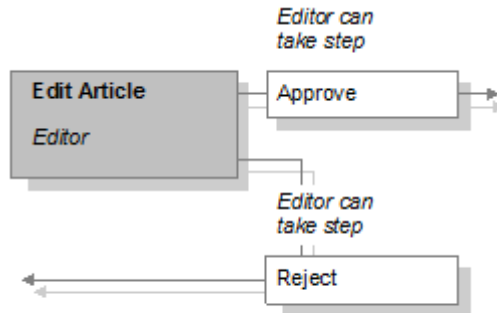
Finally, are there any requirements that must be met for an asset in a state before the step can move the asset to the next state? If so, configure and assign a step condition to the step. For example, you could configure a condition that verifies that the asset has all of its association fields filled—that an article has the associated images that it needs—before it progresses to the next state.

10.1.4 Multiple Paths for the Asset

When a workflow process includes a state in which people are reviewing an asset, typically there is more than one path possible for an asset in that workflow because the reviewer can either accept it the way it is or reject it.

In such a case, you create two steps for moving the asset from the review state. When the asset is in that state, the Finish My Assignment form lists both options and the participants select the appropriate step when they finish their assignments. For example:

Figure 10–2 Steps in Workflow



What happens if more than one participant are reviewing the same asset in the same state and they choose different steps (that is, vote differently)? That depends on how you configured each step. There are several possibilities:

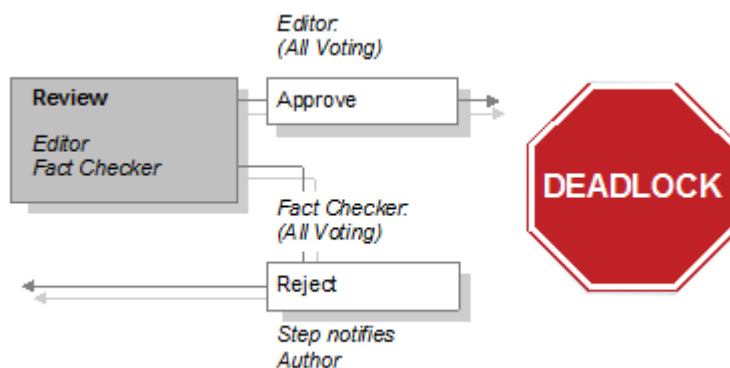
- Configure the steps so that the first participant to finish the assignment (vote) determines the direction the asset takes.
- Configure the steps so that all participants have to select the same step (vote the same way) in order for the asset to progress in either direction. A step that all the participants must select before that step can be completed is called an **all-voting step**. Note that this option can result in **deadlocks**, described in the next section.
- Use a combination of the preceding possibilities: configure one all-voting step (all participants must agree), and configure the other step(s) so that if any participant selects it, the step completes and the asset takes that path.

10.1.5 Managing Deadlocks

A deadlock occurs when the following conditions are true:

- There is more than one step from a state.
- Two or more of the steps require the participants to perform that step. That is, they are all-voting steps.
- An asset in that state is assigned to more than one participant.
- The participants select different all-voting steps when they finish the assignment.

This diagram illustrates a deadlock:

Figure 10–3 Deadlocked Workflow

Note that if even one of the steps is not all-voting and a participant selects that step, the asset will not become deadlocked.

This section contains the following topics:

- [Section 10.1.5.1, "Resolving Deadlocks"](#)
- [Section 10.1.5.2, "Preventing Deadlocks"](#)
- [Section 10.1.5.3, "Notifying Participants When Deadlocks Occur"](#)

10.1.5.1 Resolving Deadlocks

An asset that is in a deadlocked state cannot progress through the workflow process until the deadlock is resolved. To resolve the deadlock, the participants must confer with each other and agree on that path that the asset should take. Then, the participants who must change their selection can do one of the following to resolve the deadlock:

- Finish the assignment and select the step that they all agreed to take.
- Select the **Abstain from Voting** option from the **Workflow Commands** drop-down list on the asset's Status form.

10.1.5.2 Preventing Deadlocks

Before configuring a workflow process that can result in a deadlock, be sure that it is absolutely necessary to have complete agreement on all the possible steps from the state. As you can see from this description, deadlocks cause additional work for all the participants so be sure that you use this feature only when you need to.

For example, consider a review state with two possible steps: "return for revisions" and "approve for publish." If you configure the steps so that "return for revisions" does not require a unanimous vote but "approve for publish" does, you have created a desirable control—all the reviewers must agree before the asset can be published and any rejection stops the asset from being published—without risking a deadlock.

10.1.5.3 Notifying Participants When Deadlocks Occur

If you do need to create a workflow process that can result in a deadlock, be sure to configure and assign a deadlock action that notifies the participants of the deadlock for each of the steps that can cause the deadlock. The default deadlock action is an email message that WebCenter Sites sends to the appropriate participants when a step causes a deadlock.

10.1.6 Workflow Groups

Is there ever a situation in which several assets are so closely connected that they need to be thought of as one unit of work or they need to be approved at the same time? In such a case, you can use the workflow group feature.

This section contains the following topics:

- [Section 10.1.6.1, "Using Workflow Groups"](#)
- [Section 10.1.6.2, "Adding a Synchronize Step"](#)
- [Section 10.1.6.3, "Managing Group Deadlocks"](#)

10.1.6.1 Using Workflow Groups

Workflow groups enable content providers to send a defined set of assets though the workflow together. While it is the content providers who create workflow groups and select the assets that are assigned to the group, you, the administrator, still need to know what kind of assets will be included in workflow groups. Why? So that you can configure the workflow processes appropriately.

For example, you can configure workflow steps that allow each asset in the group to progress to the next state when it is finished or you can configure a step that requires all the assets in the group to reach that point before any of them can progress. (This second example, called a synchronize step, is described next.)

10.1.6.2 Adding a Synchronize Step

When creating a workflow process that will be used with workflow groups, it is usually best to configure the process so that it has only one synchronize step. Multiple synchronize steps can slow down the work on those assets unnecessarily. Assess the business process that is reflected by the workflow process and determine which steps must truly be synchronized: perhaps all the assets should go to legal review in one batch, or perhaps all the assets should be approved for publishing at the same time, for example.

10.1.6.3 Managing Group Deadlocks

If you create a workflow process to be used with workflow groups and any of the steps can result in a deadlock, be sure to configure and assign a group deadlock action that notifies participants when there is a group deadlock and assign it to the process.

10.1.7 Delegating and Clearing Assignments

People go on vacation, get reassigned to new work groups, and move on to different jobs. What happens to the assets that they are working on? They can delegate their assignments to other participants who have the appropriate roles.

Additionally, each workflow process can have a workflow administrator. The administrator of a workflow process can delegate assignments on behalf of the other participants.

When an asset is delegated to a new participant, should that person receive an email notice? If so, configure a delegate action to send an email message to new assignees when assets are delegated to them. You can specify one or more delegate actions for each workflow process.

If an asset no longer needs to be assigned or if it is easiest to clear the assignment and then start over, you can use the Clear Assignments feature in the Admin tab.

10.1.8 Placing an Asset in Workflow

An asset begins its participation in a workflow process in one of the following ways:

- A user selects a workflow process from the **Workflow Commands** field on the Status form for the asset.
Selecting the workflow process invokes the start step for the process, which places the asset into the first state.
- A user creates an asset and the start menu **New** item for assets of that type is configured such that there is a default workflow process.
In this case, saving the asset invokes the start step for the process, which automatically places the asset into the first state.

Because steps are enabled for specific roles, only the users who have a role that is assigned to the start step of a process can select that process. This means that if you are using start menu items to place assets in workflow, you must be sure that the roles assigned to the start menu item are the same roles that are assigned to the start step of the default workflow.

10.1.9 Restricting Access to Assets While They Are in Workflow

Although workflow routes an asset through a business process, sending it to the appropriate users at the appropriate times, the fact that the asset is assigned to a specific user does not stop other users from modifying or even deleting that asset.

If you want to restrict who has access to an asset while it is in workflow, use the workflow feature called function privileges. These are restrictions set on functions such as edit, copy, approve, delete, show versions, and so on in the context of workflow states and workflow roles.

There are three parts to a function privilege:

- The function being restricted.
- The roles allowed or not allowed to perform the function.
- The state during which users with those roles are allowed or not allowed to perform the function.

When a function privilege is in effect, it means that a user can perform that function only when the following conditions are true:

- The user has an appropriate role.
- The asset is in the correct state.
- The asset is **assigned** to the user.

This means that even if the user has the correct role and the asset is in the correct state, the user cannot perform that function on that asset unless the asset is assigned to that user.

This section contains the following topics:

- [Section 10.1.9.1, "Function Privileges and Step Actions"](#)
- [Section 10.1.9.2, "Function Privileges: All or None"](#)

10.1.9.1 Function Privileges and Step Actions

Function privileges restrict access to a function from the user interface only. This means that you can program step actions that invoke a function when a step is taken regardless of what the function privilege is set to at that moment.

The ApproveForPublish step action is an example. Even if you specify function privileges that restrict users from using the Approve option in the user interface, those same users can approve an asset with a workflow step if the workflow step invokes the Approve For Publish action and the user has the correct role to take the step.

In other words, you can use function privileges to prevent users from selecting and changing assets by mistake and use actions to invoke those functions in a highly controlled way.

10.1.9.2 Function Privileges: All or None

If you create even one function privilege that allows or restricts access to a function for a given role, you must create function privileges that cover all the other roles for that function.

For example, suppose that the only function that you want to restrict is the Delete function—you want to allow only the editors to delete article assets and only then if the article is in the Review state. If you create a function privilege that allows editors to delete article assets that are in the Review state but you do not create any other function privileges, nobody else can delete article assets that are in the Review state until you explicitly grant other roles access to the Delete function in the Review state.

Additionally, if you create a function privilege that denies editors access to the Delete function for assets in the Review state, no user can delete article assets that are in the Review state until you explicitly grant other roles access to the Delete function in the Review state.

10.1.10 Deadlines

There are two kinds of workflow deadlines:

- **Assignment deadlines** – which specify how long an asset should remain in an individual state. When you set a value for the **Estimated Time** field of a state, that creates a deadline for the assets that are assigned to workflow participants when the assets are in that state.
- **Process deadlines** – which specify how long it should take for an asset to go through the entire process.

Note that these different types of deadlines do not interact with each other—they are calculated separately and are mutually exclusive. Most likely you will use either one kind or the other, but not both for the same workflow process.

This section contains the following topics:

- [Section 10.1.10.1, "Setting and Overriding State Deadlines"](#)
- [Section 10.1.10.2, "Setting Process Deadlines"](#)

10.1.10.1 Setting and Overriding State Deadlines

When you create a workflow state, you can set an Estimated Time for it and determine whether reminder emails should be sent when assignments miss the deadline that is calculated from the Estimated Time.

When you create a workflow process that uses the state, you specify whether the Estimated Time for the state can be changed (overridden) when a user takes the step that moves the asset into that state.

If the deadline can be changed, any participant who takes the step can override the deadline unless you configure function privileges that restrict their ability to do so. Note that any participant who has a role that was designated as an administrator role for the workflow process can always override an assignment deadline if the deadline can be changed.

10.1.10.2 Setting Process Deadlines

When you create a workflow process, you determine whether a process deadline can be set. A process deadline is set when an asset is first placed in workflow, in the Select Workflow form. Unlike an assignment deadline, however, you cannot configure the process to send reminder emails when a process deadline is approaching.

If a process deadline can be set, any participant who places the asset in the workflow can set a process deadline for that asset—unless you configure function privileges that restrict their ability to do so. Additionally, any participant who has a role that was designated as an administrator role for the workflow process can always set a process deadline, if a deadline can be set.

10.1.11 Scheduling a Deadline Calculation

There are two kinds of actions:

- Actions that are invoked by a step. These actions are events that WebCenter Sites completes when a step is taken.
- Actions that are triggered by a deadline. These actions are queued and are triggered only after WebCenter Sites calculates the deadlines for the assets and determines which timed actions (if any) should be invoked.

You specify how often the deadlines are calculated by configuring the Timed Action Event. This is an event that invokes a background calculation process of all deadlines. It is similar to the publishing event that invokes the background approval calculation process.

Just as the publishing process calculates approvals to determine which assets should be published, the deadline calculation process that is invoked by the Timed Action Event calculates the deadlines for all assets that are participating in workflow processes—to determine whether any reminder messages should be sent for assignment deadlines and to determine the times that should be displayed for assets in the Due and Process Deadline columns of assignment lists.

You can configure the Timed Action Event on your management system to run as often as you find it necessary.

10.1.12 How Does a Workflow Process End?

A workflow process ends when there are no more states for the asset to progress through. This occurs when the final participant takes the end step for the workflow process.

An end step is the opposite of a start step—it has a From State but no To State. When a user takes the end step, it is moved to a stateless state, which means the asset is no longer in workflow and any function privileges set for that workflow process no longer apply.

10.1.13 Roles Required to Configure Workflow Processes

The workflow building blocks are located on two tree tabs in the administrator's interface:

- **Admin** – holds email objects, actions, conditions, and the **Timed Action** Event.
- **Workflow** –holds workflow states and processes.

For access to the Admin tab, you must be assigned the GeneralAdmin role and have the xceladmin ACL assigned to your user account. For access to the Workflow tab, you must be assigned the WorkflowAdmin role for the site you want to create workflow processes for.

For more information about access rights in the WebCenter Sites interfaces, see [Chapter 4, "Working with ACLs and Roles"](#) and [Chapter 5, "Configuring Users, Profiles, and Attributes."](#)

10.2 Planning Your Workflow Processes

When you create a workflow process, you create steps that link together the states for that process. This means that you or someone else must create the workflow components—roles, email messages, the various kinds of actions, step conditions, and states—before you can create a workflow process.

This section provides more details about the configuration of each of the workflow components so that you can plan and implement your workflow processes. When planning a workflow, you will do the following:

- [Section 10.2.1, "Start with a Sketch"](#)
- [Section 10.2.2, "Determine Roles and Participants"](#)
- [Section 10.2.3, "Determine the Email Objects, Actions, and Conditions"](#)
- [Section 10.2.4, "Determine the States"](#)
- [Section 10.2.5, "Determine the Steps"](#)
- [Section 10.2.6, "Determine the Function Privileges"](#)
- [Section 10.2.7, "Determine Additional Workflow Process Details"](#)

10.2.1 Start with a Sketch

Where do you begin? With sketches of the business processes that you need to implement as workflow processes:

- Use boxes to represent the states.
- Use arrows to represent the steps that connect the states.

As you read through the descriptions in this section, write on your sketches the details about which roles, actions, conditions, deadlines, and so on are appropriate for each state or step in the process.

Then, refer to your sketches as you use the administrator's interface to create your workflow processes, described in [Section 10.3, "Configuring Your Workflow Processes."](#)

10.2.2 Determine Roles and Participants

When you start planning your workflow processes, begin with the roles. What kind of functional groups participate in workflow? Then, determine how the individual users from those roles will become the participants in a workflow process for a specific asset.

This section contains the following topics:

- [Section 10.2.2.1, "Planning Roles"](#)
- [Section 10.2.2.2, "Selecting Individual Participants"](#)
- [Section 10.2.2.3, "Cross-Site Assignments and Participants"](#)

10.2.2.1 Planning Roles

To determine the roles that you should create for your processes, ask yourself the following questions:

- What are the job titles or roles of the content providers who are using your Sites management system?
- What do the people in each role do? (You can map the tasks that they complete to your states.)
- How are the roles organized? For example, is there one group of reviewers who review everything but several groups of content providers who are organized by subject (for example, sports writers, financial writers, marketing writers, and so on)
- Do certain groups of people work with specific asset types but not with others? If so, which roles should have access to each asset type that you plan to create a workflow process for?
- Would you ever need to notify someone who is not participating in a workflow about the status of an asset that is in workflow? If so, that person will need a role so a step or timed action can send an email message to that person.

On your sketches of your workflow processes, write the names of the roles that will have the asset assigned to them in each state.

For information about creating roles, see [Chapter 4, "Working with ACLs and Roles."](#)

10.2.2.2 Selecting Individual Participants

When you configure a workflow process, there are several ways to determine which specific users participate in the workflow for each individual asset that goes through the workflow:

- When the workflow is assigned.
 - You can configure the process so that the person who first assigns the workflow to the asset must select all the users who will participate. For each state, they select users from a list. The list includes all the users who have the correct role for that state.
- When a participant finishes the assignment.
 - You can configure the process so that each participant determines who the next participant is when he or she finishes the assignment. That participant selects a user name from a list that includes all the users who have the correct role for the next state.

You specify how participants will be selected when you configure the steps.

10.2.2.3 Cross-Site Assignments and Participants

Assets can be shared between sites. If a shared asset is entered into a workflow process, should users from all the sites that have access to the asset be considered as candidates for participating in the workflow? If the answer is yes, enable the cross-site assignments feature.

When you use the cross-site assignments feature, users see all of their assignments from all the sites that they have access to in their assignment list no matter which site they are currently logged into. Having one, consolidated assignment list is very convenient for your users.

Keep in mind, however, that if your users have different roles in different sites and you are using function privileges in your workflow processes, they might not be able to work on an asset that they can see in their assignment lists. For example, say that some of your users have the author role in one site and the editor role in another. If you are using function privileges to restrict editing to editors in a certain state, they can see an asset that they aren't allowed to edit in their assignment list when they are logged in to the site where they function as authors.

To enable this feature, you set the value of the `xcelerate.crosssiteassign` property to `true`. This property is in the `futuretense_xcel.ini` property file. For information about the property, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

10.2.3 Determine the Email Objects, Actions, and Conditions

An action is an event that is triggered in one of three ways:

- A step invokes it (step action).
- A deadline triggers it (timed action).
- A workflow situation triggers it (deadlock, group deadlock, and delegate actions)

A condition is an event that is assessed when a step is attempted. If the condition is not met, the step cannot be completed.

When you create an action or a condition, you identify an element. Elements are named pieces of code that are stored in the `ElementCatalog` table. It is the element that invokes the function represented by the action. If the element is coded to expect variables or arguments, you identify values for them when you create the action and those variables are then passed to the element when the action is triggered.

As an administrator, you are not responsible for coding elements. If your workflow needs cannot be met by the default workflow elements that are provided, your developers can code the functionality that you need. See the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information about customizing workflow.

WebCenter Sites provides the following default workflow elements:

Table 10–1 Workflow Element Names and Expected Variables

Element Name	Variables It Expects
OpenMarket/Xcelerate/Actions/Workflow/StepActions/ApproveForPublish	target The name of the publishing destination that the asset is to be approved for.
OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees	emailname The name of the email object to send.

Table 10–1 (Cont.) Workflow Element Names and Expected Variables

Element Name	Variables It Expects
OpenMarket/Xcelerate/Actions/Workflow/StepConditions/ExampleStepCondition	
OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail	emailname The name of the email object to send.
OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees	emailname The name of the email object to send.
OpenMarket/Xcelerate/Actions/Workflow/GroupActions/SendEmailToAssignees	emailname The name of the email object to send.

These elements are described in detail in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

With the exception of ApproveForPublish and ExampleStepCondition, these elements take a variable called `emailname` and send the email message that is identified by that variable.

How do you determine the value of the `emailname` variable? By creating workflow email objects. The names of the email objects that you create in the Admin tab in the WebCenter Sites administrator's interface are the names that you specify as the arguments with the `emailname` variable when you create actions that send email messages.

This section contains the following topics:

- [Section 10.2.3.1, "About Email Objects"](#)
- [Section 10.2.3.2, "About Timed Actions"](#)
- [Section 10.2.3.3, "About Delegate Actions"](#)
- [Section 10.2.3.4, "About Step Actions"](#)
- [Section 10.2.3.5, "About Step Conditions"](#)
- [Section 10.2.3.6, "About Deadlock Actions"](#)
- [Section 10.2.3.7, "About Group Deadlock Actions"](#)

10.2.3.1 About Email Objects

Email objects are building blocks separate from the actions so that you can use them with more than one action. For example, the default deadlock action and the default group deadlock action use the same email message (named Deadlock Message).

You create email objects by giving them a name, a description, a subject line, and text for the body. When the message is sent, the text provided for the subject is placed in the subject line and the text provided for the body is placed in the body of the message.

Workflow Email Variables

There are several variables that you can use in the subject line and body text which makes it easier for you to write email messages that are personalized for each recipient.

The following table lists the default workflow variables that you can use with any email message:

Table 10–2 Default Workflow Variables

Variable Name	Description
Variables.assetname	The name of the asset. Use this variable in every email message so the recipient knows which asset is being referred to.
Variables.assigner	The user name of the participant who assigned the asset to the person receiving the email. Use this variable in email messages for step actions that notify participants that a new asset has been assigned to them.
Variables.time	The time specified in the Estimated Time field for a state. Use this variable for timed actions.
Variables.instruction	The text that the previous participant entered in the Action to Take field of the Finish My Assignment form when he or she finished the assignment. Use this variable in email messages for step actions that notify participants that a new asset has been assigned to them.

For examples of custom email variables, see the email object named Deadlock Message and the corresponding deadlock action element named OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees.

Default Workflow Email Objects

The default workflow email objects are these:

- Assignment Due Reminder – specifies the asset and the time that it is due.
- Assignment Message – specifies the asset, the person who assigned it, and in the message from the **Action To Take** box on the Finish My Assignment form.
- Deadlock Message – describes how the deadlock occurred by listing the users and the steps that they took.
- Rejection Message – is similar to the Assignment Message, but states that the asset was rejected by the previous participant (the assigner).

Planning Your Email Objects

To determine the email objects that you need to create, you must also determine the actions that will send the email messages held in the objects. By using the email variables in the subject and body, it is likely that you can use the same email object with more than one timed action or step action.

Typically, you need to compose email messages that specify the following kinds of things:

- An asset has been assigned to a participant. (For step actions.)
- An asset has been delegated to a new participant. (For delegate actions.)

- A deadline is approaching for an asset. (For timed actions.)
- A deadline for an asset has been missed. (Also for timed actions.)
- An asset is in a deadlock. (For a deadlock action.)
- A group of assets are in a deadlock. (For a group deadlock action.)

In the **Admin** tab, double-click **Email**. Examine the default email objects to determine whether you can use them. You can modify the default messages or create your own email messages.

In one corner of your sketches of the workflow processes, list the email messages that you will need for that workflow.

10.2.3.2 About Timed Actions

Timed actions are actions that are based on the deadline of a state. If you specify a deadline for a state, you can specify a timed action to remind the participants about the deadline.

You create a timed action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

To create a timed action that sends email notices about a deadline, specify the `OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail` element and use the `Argument` field to specify which email message to send.

When you create a state, you specify which timed actions to use and when to trigger them, relative to the deadline specified for the state. You can specify more than one timed action for each state.

When the Timed Action Event runs and calculates the deadlines for all the assets currently participating in workflow, the appropriate timed actions are triggered.

Default Timed Actions

There is one default timed action: `Send Email`. It uses the `OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail` element to send the `Assignment Due Reminder` email message.

Planning Timed Actions

When planning your timed actions, you must also consider the following workflow components:

- The states that you will create, because these actions are triggered in terms of the deadlines that you set for your states.
- The email objects that you need to create, because it is likely that your timed actions will send email messages

Because the time that a timed action is triggered is determined outside of the action itself (you do this in the form for the state), you can probably create a small number of generic timed actions—whose only difference is the email message they send—and use them repeatedly with your states.

In the **Workflow** tab, expand **Actions**, then **Timed Actions**. Examine the `Send Email` action to determine whether you can use it. You can modify the default timed action or create your own.

On each of your sketches of your workflow processes, list the timed actions that you need for that process near the list of the email messages.

10.2.3.3 About Delegate Actions

When you assign a delegate action to a workflow process, the action is triggered whenever a participant (or the workflow administrator) delegates an assignment to another participant.

You create a delegate action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

You can specify one delegate action per workflow process.

Default Delegate Action

There are no default delegate actions provided.

Planning Delegate Actions

It is likely that your delegate action will send an email message to the participant to whom an assignment is delegated. If this is the case, you can create a delegate action that uses any of the default workflow elements that send email and create a new email object for that element to send.

On each of your sketches of your workflow processes, write the name of the delegate action that you will use for that process. (You can assign one for each process.)

10.2.3.4 About Step Actions

When you assign a step action to a step, the action is triggered when a participant takes the step.

You create a step action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

There are two general categories of step actions: those that complete functions and those that send email messages to the participants who are being assigned the asset by the step.

Step actions are completed regardless of function privileges. That is, if you create a step action that performs a WebCenter Sites function, the action does not check the function privileges of the participant taking the step. This means that you can restrict access to a function from the user interface and require participants to use a step in a workflow in order to complete a specific function (approve for publish, for example).

You can specify one or more step actions for each step.

Default Step Actions

The default step actions are these:

- Approve for Publish, which uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/ApproveForPublish` element to approve the asset. Then, the next time a publishing process runs, the asset is published (as long as all of its dependencies are also approved).

To use this action for your workflow processes, you must specify the publishing destination(s) that the asset is to be approved for by using the `targets` argument.
- Send Assignment Email, which uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees` element to send the Assignment Message email object to all the participants assigned the asset (that is, who are specified in the **Assignment Method** for the step).

- Send Rejection Email, which uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees` element to send the Rejection Message email message to the new assignee (the participants specified in the **Assignment Method** for the step).

Planning Step Actions

To determine what kind of step actions you need to create, you must consider the following workflow components:

- The steps that you will create for the process. Do people need to be notified that a step has assigned an asset to them? If so, you need a step action that sends an email message.
- The states that the steps move the assets into. Does the state represent the asset after a function has been completed? If so, you need a step action that implements that function.

In the **Workflow** tab, select **Actions**, then **Step Actions**. Examine these actions to determine whether you can use them. You can both modify the default step actions and create your own.

On each of your sketches of your workflow processes, write the name of the step actions next to the appropriate steps. You can assign one or more step actions for each step.

10.2.3.5 About Step Conditions

When you assign a step condition to a step, the condition is assessed when the step is taken. The condition determines whether the step can be completed or not.

You create a step condition by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary. If you want to use conditions in your workflow processes, talk to your developers about the conditions you need them to implement through elements.

You can specify one or more conditions for each step.

Default Step Conditions

There is one default step condition: Example Step Condition. It is a "hello world"-style example that illustrates how to code an element to check for a condition. You and your developers should examine it to learn how it works and then your developers should create their own condition elements.

Planning Step Conditions

When thinking about step conditions, you must consider the steps and the state. If you determine that conditions are necessary for your workflow processes, ask your developers to create them.

On each of your sketches of your workflow processes, write the name of any step conditions that you need next to the appropriate steps.

10.2.3.6 About Deadlock Actions

When you assign a deadlock action to a step that can result in a deadlock, the action is triggered whenever an asset becomes deadlocked during the step.

You create a deadlock action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

You can assign one or more deadlock actions per step.

Default Deadlock Action

There is one default deadlock action: Send Deadlock Email. It uses the OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees element to send the Deadlock Message email message.

Planning Deadlock Actions

When thinking about possible deadlock actions, you need to consider the steps that you will create for your workflow processes. If you plan to design your steps so that deadlocks cannot occur, there is no need to create any deadlock actions.

In the **Workflow** tab, expand **Actions**, then **Deadlock Actions**. Examine this action to determine whether you can use it. You can modify the default deadlock action or create your own.

In your sketches of your workflow processes, write the name of the appropriate deadlock action next to any step that can result in a deadlock.

10.2.3.7 About Group Deadlock Actions

A group deadlock action is triggered when workflow group becomes deadlocked. Whoever creates the workflow group selects the group deadlock action. That person can select one or more for each group.

The group deadlock actions are invoked in addition to any other actions that are associated with the states or the steps in the workflow process.

Default Group Deadlock Action

There is one default group deadlock action: Send Deadlock Email. It is identical to the default deadlock action.

Planning Group Deadlock Actions

The same considerations apply for group deadlock actions as for deadlock actions. However, because it is a content provider who selects the group deadlock action for a workflow group, be sure that you give a group deadlock action a meaningful, unambiguous name.

10.2.4 Determine the States

When you create a state, you specify the following kinds of information:

- Name and description. The name of a state should be meaningful and should represent the kind of work that is being done while the asset is in that state.
- Amount of time an asset should remain in this state (the **Estimated Time**). The deadline is calculated in terms of the number of hours or days since the asset entered the state.

Note that you set the estimated time (the deadline) in terms of hours or days rather than as a specific date because a state deadline is calculated for each asset as it passes through the state.

- The timed actions that should occur and when they should occur, in terms of days or hours before or after the deadline.

There are no default states, although there are example states provided with all of the sample sites. On a system where the sample sites are installed, select **Workflow**, then select **States**, and then select a state to examine to learn about how the sample site states are configured.

Planning Your States

To determine the states that you need to create, ask yourself the following kinds of questions:

- Which roles participate in each state? On your sketches, write the names of the roles next to the states.
- Do any of these states apply to more than one asset type? If so, it's possible that you can reuse the same state in more than one workflow process.
- How long should it take to complete each state?
- Does this amount of time differ by asset type or by site? If yes, you cannot reuse the same state in more than one workflow process or share the workflow process with another site.
- If there is a deadline, should WebCenter Sites notify the participant when the deadline is approaching? If yes, when? The day before? Several hours before? And which timed action (which determines which email message) should be used?
- Should there be a notice when a deadline is missed? If yes, when? And which timed action should be used?

On the sketches of your workflow processes, list all of this information next to each state.

Depending on your goals, you might consider creating a workflow process that does not really end. For example, the Hello Asset World sample site workflow process ends after a HelloArticle asset has been approved. However, in a case like this, someone could open and save the asset by mistake, which would mean that it is no longer approved and will not get published.

To keep assets from being edited after they have been approved and before they have been published, you could create a final state that holds assets after they have been approved with a function privilege on the state that restricts anyone from editing the asset.

Be sure that if you do create a state like this that you create a step that can move the asset back into an editing state in the workflow so someone can edit it on purpose.

10.2.5 Determine the Steps

You create steps within a process. You use the steps to link together the states. This is how the process is actually created.

You specify the following kinds of information for each step in your process:

- The name of the step. The name should be meaningful and should describe the path being taken. For example, Send for Review, Send for Approval, and so on.
- The **From State**, which is the state that the step is moving the asset from. If the step has no **From State**, it is the start step that begins the workflow process. A **From State** is required for each step other than the start step.
- The **To State**, which is the state that the step is moving the asset to. If the step has no **To State**, it is the end step the ends the workflow process.
- Which roles are authorized for the step.

If this is the very first step, the roles you specify determine which users can select this workflow for an asset. If you are using a Start Menu item to assign the workflow process to an asset, be sure that the roles assigned to this first (start) step match the roles assigned to the Start Menu item.

For subsequent steps after the first (start) step, the roles that you select must either match or be a subset of the roles that were selected to be notified by the previous step. Otherwise, the asset cannot leave the state because no one who is assigned the asset is allowed to move it to the next state.

- Who gets the asset next? (That is, when this step is completed). You specify the assignee for the To State by selecting one of the following assignment methods:
 - Retain *From State* assignees.

This option assigns the asset to its current assignees—that is, to the user who completes the step.

This option is useful when you are creating a step that returns to the same state (which creates an iterative state) or when it is the start step and you want the asset to be assigned to the person who selects the workflow process (whether through a start menu item or by selecting it on the Status form).
 - No assignments; control actions with function `privs`.

This option keeps the asset in the workflow process, which means that function privileges are enforced, but the asset is not actually assigned to anyone.

Note: When an asset is assigned to the current assignees (Retain *From State* assignees option) or assigned to no one (No assignments option), the workflow system records workflow history for that step.

 - Assign from a list of participants.

With this option, you select roles. When a user assigns the workflow process to an asset, a list of users with the roles that you selected is displayed. The user selects one or more users from the list, and those users are assigned the asset when it is in the state that this step moves the asset to.
 - Choose assignees when the step is taken.

You also select roles with this option. This option means that the person who takes this step (by completing the assignment) during the workflow process selects the user who is assigned the asset next from a list of users with the roles selected for the step.
- Whether the estimated time for the state that this step moves the asset to can be changed or not. (Called an Assignment Deadline.)
- Any step actions that should occur.
- Any step conditions that should be assessed.
- If the step can be taken by more than one user (assignee), whether all the assignees must take this step before the asset can move to the next state. (The **All assignees must vote** field.)
- If the step can result in a deadlock, what the deadlock action should be.
- If the process will be used for a workflow group, whether all the assets in the group must complete this step before any of the assets can move to the next state. (The **Step is group synchronized** field.)

There are example steps in all of the sample site workflow processes. On a system where the sample sites are installed, select **Workflow**, then select **Processes** and examine a process to learn about how the sample workflow steps are configured.

Planning Your Steps

When determining what your steps should be, consider the following questions:

- Every process needs a start step. Before you create a start step, you must first determine how the asset is created and then placed into workflow.

Does the first participant create the asset, place it into workflow, and then keep working on the asset? If so, configure the step so that it is assigned to the person taking the start step (choose the **Retain "From State" assignees** option) and remember to create a start menu item that assigns the workflow to the asset by default.

Or does a supervisor create the asset and then assign to it a participant? If so, configure the start step so that the person taking the start step has to select the assignees.

- What is the order of the states?
- How many paths do you need between each state? This answer determines how many steps you need.
- Do you need to create an iterative state? If so, configure the step that takes the asset back to that state so that it is assigned to the person taking the step (choose the **Retain "From State" assignees** option).
- For steps that move an asset from a state in which more than one person is working on that asset, must all the participants complete their assignment before the step can be taken? If so, configure the step to be an all-voting step.
- Does the asset have more than one possible path from a state? (That is, there needs to be more than one step with the same From State.) If yes, do any of the steps from the state require that all participants vote the same way before the asset can progress? If yes, try to design the process so only one of the steps that lead from that state require that all the participants vote the same way. If you have two all-voting steps from the same state, deadlocks can occur.
- For each step, should WebCenter Sites notify the affected participants when the step is completed and the asset progresses to the next state? If yes, which step action (which determines the email message) should the step use?
- Do groups of related assets ever need to be worked on at the same time? If so, can you configure your steps to work appropriately for both a single asset and a group of assets? If not, create separate workflow processes for the workflow groups.
- For a workflow group, are there any states that all the assets in the group must enter at the same time? (For example, all the assets should be approved at the same time.) If so, configure that step to be a synchronize step. Note that it's best to have only one synchronize step per workflow process.
- What should happen to the asset if you decide not to publish it, after all? Should you have a cancel step? How many cancel steps do you need?

On your sketches of your workflow processes, right next to the box that represents a step, list the step actions, the roles who can take the step, the roles who should be notified when the step is taken, whether the step is an all-voting step, and whether the step is a synchronize step.

10.2.6 Determine the Function Privileges

Administrators grant or deny roles the privilege to use functions (such as create and edit) on a given site. When the privilege is granted, the functions are displayed to site users with those roles. Users can invoke the functions when an asset is in the state

specified by the privilege. (By default, function privileges are granted to all roles on a given site.)

You can set function privileges for each function in a workflow process, as necessary. Instructions for setting function privileges can be found in ["Step E: \(Optional\) Configure Function Privileges"](#) on page 10-43.

The following table lists the functions that you can allow or disallow users to perform during a workflow process.

Table 10-3 Functions and User Action Allowances

Function	If the Function Privilege is Granted...
Abstain from Voting	Users have the option to not vote in a workflow step. Displayed as an option in the Workflow Commands drop-down menu on the asset's Status form.
Approve for Publish	Users can mark an asset as approved for publishing. Displayed as an option in the more... drop-down menu on the asset's Inspect and Status form.
Authorize	Users can configure permissions to an asset.
Build	Users can build a collection asset. Displayed as an option in the more... drop-down menu on an asset's Inspect and Status forms.
Checkout	Users can check out an asset, as long as revision tracking is enabled for the asset type. Displayed as the Check Out button at the top of an asset's Edit and Status forms.
Copy	Users can copy an asset. Displayed as an option in the more... drop-down menu on an asset's Inspect and Status forms.
Delegate Assignment	Users can delegate an asset (assigned to themselves) to another user (one who has the appropriate role for the asset while it is in that state). Displayed as an option in the Workflow Commands drop-down list on the asset's Status form.
Delete	Users can delete an asset. Displayed as an icon in the action bar on an asset's Inspect and Status forms. Also displayed as an icon next to the asset in lists.
Edit	Users can open an asset in its Edit form. Displayed as an icon in the action bar on an asset's Inspect and Status forms. Also displayed as an icon next to the asset in lists.
Inspect	Users can open an asset in its Inspect form. Note: Users without permission to inspect an asset cannot view any data specific to the asset type. Instead, they see a limited version of the Inspect form, which displays only standard, unrestricted fields (such as name, description, and ID).
Make Root of Translation	Users can set an asset as the master asset of translations. (Used by multilingual assets.) Displayed as the Make Master link in the Translations field of a multi-lingual asset's Inspect form.

Table 10-3 (Cont.) Functions and User Action Allowances

Function	If the Function Privilege is Granted...
Place Page	Users can place a page on the Site Plan tab. Displayed as an option in the right-mouse menu on the Site Plan tab when the user right-clicks Placed Pages.
Preview	Users can view an asset in the context of its page. Displayed as the Preview icon in the asset's toolbar.
Remove from Group	Users can remove the asset from a workflow group. Displayed as an option in the Workflow Commands drop-down list on the asset's Status form.
Remove from Workflow	Users can remove the asset from a workflow process. Displayed as an option in the Workflow Commands drop-down list on the asset's Status form.
Rollback	Users can return an asset to one of its previous versions (all of which are stored in the revision tracking system). Note: This function is displayed as the Rollback button at the top of an asset's Edit and Status forms when revision tracking is enabled and at least two versions of the asset exist.
Set Export Destination Path/Filename	Users can fill in the Path and Filename fields on an asset's New or Edit forms.
Set Participants	Users can set participants for a workflow process. Displayed as an option in the Workflow Commands drop-down list on the asset's Status form. Additionally, this function prompts the user to select participants when the workflow process is configured. In this manner, the user finishing an assignment must select the next participant (person being assigned the asset).
Set Process Deadline	Users can set a deadline indicating the date and time by which the workflow process must be completed. Note: This function is displayed only if the workflow is configured to allow its users to set a process deadline. This function then appears as an option in the Select Workflow form for assets, and when a workflow group is created or edited.
Set Assignment Deadline	Users can set a deadline indicating the date and time by which the next state must be completed. A deadline set with this option overrides any deadlines set for the state in the workflow process. Note: This function is displayed only if the workflow is configured to allow its users to set an assignment deadline. This function then appears as an option in the asset's Select Workflow form and in the Finish My Assignment form.
Share Assets	Users can share the asset with another CM site. Displayed as the Share Asset option in the more... drop-down menu on an asset's Edit, Inspect, and Status forms when the asset type is enabled for more than one CM site.
Show Participants	Users can view a list of the participants in the workflow process that is currently assigned to the asset. Displayed as an option in the Workflow Commands drop-down list on the asset's Status form.
Show Status	Users can view the Status form for an asset. Displayed as an option in the more... drop-down menu on an asset's Inspect and Status forms.

Table 10–3 (Cont.) Functions and User Action Allowances

Function	If the Function Privilege is Granted...
Show Version	Users can view information about each version of the asset that the revision tracking system is storing. Displayed as the Show Versions button at the top of an asset's Edit and Status forms when revision tracking is enabled for this asset type.
Set Nested Workflow	Users can create a workflow within an existing workflow. Note: The administrator's interface does not support nested workflow, but functionality can be obtained with the use of tags.
Translate	Users can create a translation of the master asset. (Used by multilingual assets.) Displayed as an option in a multi-lingual asset's more... drop-down menu, on the Edit, Inspect, and Status forms.

This section contains the following topics:

- [Section 10.2.6.1, "Planning Function Privileges"](#)
- [Section 10.2.6.2, "Implementing Simplified Access Control"](#)

10.2.6.1 Planning Function Privileges

For each workflow process, determine which functions you want to restrict access to, during which states you want to restrict those functions, and which roles should or should not have access to those functions when assets that are assigned to them are in those states.

List any restrictions that you want to enforce next to the appropriate states on your sketches of your workflow processes.

Remember that in order for your function privileges to enforce the restrictions that you intend, the roles of the users specified for the state in the function privilege must match the roles of the users associated with those states in the workflow process itself. If the roles don't match, the result can be a condition in which no one can move the asset out of a state because the users who are allowed to work with the asset by the privilege are not allowed to by the state.

Also, remember that if you create even one function privilege that allows or restricts access to a function for a given role, you must create function privileges that cover all the other roles for that function.

10.2.6.2 Implementing Simplified Access Control

Because function privileges are associated with workflow processes, you can restrict individual users' access to specific functions only when an asset is participating in a workflow process.

What can you do if you do not want to design and implement workflow processes but you still want to control access to specific functions? Create a simplified workflow process with one state and use the No assignments; control actions with function `privs` assignment method.

When a step moves an asset into a state using the No assignments assignment method, the asset does not appear on anyone's assignment list, it just stays in the state which means that any function privileges assigned to that state are enforced.

To implement access control in this way, follow these general steps:

1. Create a state. It needs a name and a description. It does not need a deadline or any timed actions.
2. Create a new workflow process. Select all the roles you want to enforce restrictions for and all the asset types you want to use this workflow process for.
3. Create one step—the start step that puts assets of those types into the state. For the step, select the **No assignments** option. Enable the step for the roles that you want to be able to create assets of this type and assign this workflow to.
4. Configure the function privileges that you want to enforce for assets in the state that you created.
5. Create start menu items for the asset types that automatically assign this workflow process. Be sure that the roles who can use the start menu item match the roles who are assigned to the start step in the workflow process.

Now when users select New, then select asset type, the new asset is automatically placed into your single-state workflow. Because the workflow has only the start step which places all assets of that type into the single state, those assets do not leave the state and the function privileges are always enforced.

10.2.7 Determine Additional Workflow Process Details

If you have been sketching your workflow processes, filling in all the details about actions, conditions, email messages, states, steps, and function privileges, by now you have planned nearly the entire design of your workflow processes.

Although the main task when creating a process is to create the steps that link the states (which is how your business process is represented), you must also specify the following kinds of information for each workflow process:

- Its name and description. The name should be descriptive so that users select the correct workflow process for the correct asset types.
- Which sites can use the process.
- Which asset types can use the process.
- Which roles can participate. This is a superset of all the roles that are designated in the steps.
- Which role will serve as the administrator of the workflow. A workflow administrator can delegate assignments on behalf of other participants.
- Whether a process deadline can be set for assets that participate in this workflow process.
- What the delegate action is.
- What the steps are. For each step, you specify the information described in the section named [Section 10.2.5, "Determine the Steps."](#)
- Whether any of the WebCenter Sites content application functions should be restricted to users in certain roles while they are working on assets in specific states. For function privileges, you specify the information described in [Section 10.2.6, "Determine the Function Privileges."](#)

There are example workflow processes delivered with all of the sample sites. They are described in detail in the next section.

10.3 Configuring Your Workflow Processes

This section presents the procedures for creating all the components for a workflow process and then stitching those components together into your workflow processes.

Remember that to create email objects, actions, and conditions or to schedule the timed action event, you need access to the Admin tab, which means that you need the GeneralAdmin role and the xceladmin ACL. To create workflow states and workflow processes, you need access to the Workflow tab, which means that you need the WorkflowAdmin role.

10.3.1 Overview

Before you create a workflow process, you must create the individual workflow components that you need for that process. Here are the general steps that you must take presented in the order you perform them:

1. Plan your workflow process by drawing it. See section [Section 10.2, "Planning Your Workflow Processes"](#) for help with this step. Then refer to your sketch and notes throughout this section.
2. Create the roles that you need for your workflow processes. See [Chapter 4, "Working with ACLs and Roles"](#) for help with this step. Be sure that any users who will participate in workflow have user profiles created for them. Otherwise, they will not receive email messages from the workflow process.

Note: If you want the pool of users who are candidates to be participants in a workflow to include folks from all the sites that an asset is shared to, be sure to enable the cross-site assignments feature. See [Section 10.2.2.3, "Cross-Site Assignments and Participants"](#) for details.

3. Create the email objects that you need for your actions and enable the `xcelerate.emailnotification` property in the `futuretense_xcel.ini` file. See [Section 10.3.2, "Setting Up Email Objects."](#)
4. Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions that you need. See [Section 10.3.3, "Setting Up the Actions and Conditions."](#)
5. Create your states. See [Section 10.3.5, "Setting Up the States."](#)
6. Create your process. While creating your workflow process, you create the steps for that process. The steps link together the states so they occur in the proper order. Additionally, while creating your process, you configure any function privileges that you need. [Section 10.3.6, "Setting Up the Workflow Processes."](#)
7. If your states have deadlines, be sure to configure the Timed Action Event so that the deadlines of assets are calculated regularly and the appropriate timed actions (if any) are invoked in a timely way. [Section 10.3.4, "Setting Up the Timed Action Event."](#)
8. Test your workflow processes. See [Section 10.3.7, "Testing Your Workflow Process."](#)
9. Set up your start menu shortcuts so that workflow processes are assigned automatically to assets when they are created. For help with start menu items, see [Section 9.2.2, "Creating Start Menu Items."](#)

10.3.2 Setting Up Email Objects

You create email objects so that your step and timed actions can send them to the appropriate participants at the appropriate times. Examine the sketches of your workflow processes, determine the email messages you need, and then use the procedures in this section to create and edit them.

Your user account must give you access to the Admin tab in order for you to create email objects.

This section contains the following topics:

- [Section 10.3.2.1, "Enabling the Email Feature"](#)
- [Section 10.3.2.2, "Creating Email Objects"](#)
- [Section 10.3.2.3, "Editing Email Objects"](#)
- [Section 10.3.2.4, "Deleting Email Objects"](#)

10.3.2.1 Enabling the Email Feature

To ensure that your workflow process can successfully send email messages, the following conditions must be true:

- In the `futuretense.ini` file, the properties on the **Email** tab must be configured to provide information about your email server.
- In the `futuretense_xcel.ini` file, the `xcelerate.emailnotification` property must be set to `true`.

For information about the properties cited above and using the Property Editor, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

- The workflow participants must have email addresses specified in their user profiles. For information about creating user profiles, see [Section 5.5, "Working with User Profiles and User Attributes."](#)

10.3.2.2 Creating Email Objects

To create email objects

1. In the **Admin** tab, expand **Email** and double-click **Add New**.

WebCenter Sites displays the Add New Workflow Email form:

Figure 10–4 Add New Workflow Email Form**Add New Workflow Email**

***Name:** ***Description:** ***Subject:** ***Body:**

Cancel

Add New Email

2. In the **Name** field, enter a unique name of up to 36 characters. This is the name that you provide to the `emailname` variable when you use this email object with an action.
3. In the **Description** field, enter a short, informative description of up to 36 characters.
4. In the **Subject** field, enter a short, informative subject for the email message. See [Section 10.2.3.1, "About Email Objects"](#) for a list of available variables.
5. In the **Body** field, enter the text for the message. See [Section 10.2.3.1, "About Email Objects"](#) for a list of available variables.
6. Click **Save**.

10.3.2.3 Editing Email Objects**To edit email objects**

1. In the **Admin** tab, expand **Email**, then double-click the desired email object.
2. In the action bar, click **Edit**.
3. In the Edit Workflow Email form, make the necessary changes. See [Section 10.2.3.1, "About Email Objects"](#) for a list of available variables.
4. Click **Save**.

10.3.2.4 Deleting Email Objects

To delete email objects

1. In the **Admin** tab, expand **Email**, and double-click the desired email object.
2. In the action bar, click **Delete**.
WebCenter Sites displays a warning message.
3. Click **Delete Email**.

10.3.3 Setting Up the Actions and Conditions

When you create any kind of action or a step condition, you identify an element and you supply values for the variables that the element expects. If you are creating an action that sends an email message, you identify which email object to send with the `emailname` variable.

Before you begin creating actions or step conditions, be sure that you have the elements and email objects that you need. WebCenter Sites provides several default action elements and email objects. Use the administrator's interface to examine the email objects and use WebCenter Sites Explorer to examine the `ElementCatalog` table. Determine which elements you plan to use and then write down the entire name of those elements. If you need additional email messages, consult the previous section, [Section 10.3.2, "Setting Up Email Objects"](#) and create the email messages that you need.

The following procedures describe how to create, edit, and delete the workflow actions—step, timed, delegate, deadlock, and group deadlock action—and step conditions. Your user account must give you access to the Workflow tab in order for you to create actions or conditions.

This section contains the following topics:

- [Section 10.3.3.1, "Creating Actions and Conditions"](#)
- [Section 10.3.3.2, "Configuring Approve for Publish Step Actions"](#)
- [Section 10.3.3.3, "Editing Actions and Conditions"](#)
- [Section 10.3.3.4, "Deleting Actions and Conditions"](#)

10.3.3.1 Creating Actions and Conditions

To create actions and conditions

1. In the **Workflow** tab, expand **Actions**.
2. Under **Actions**, expand the category describing the action you are creating, then double-click **Add New**.

The available categories are: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.

WebCenter Sites displays the Add New form corresponding to the type of action you selected. The example below shows the Add New Step Action form:

Figure 10–5 Add New Step Action Form**Add New Step Action**

***Name:**

***Description:**

***Element Name:**

Arguments:

3. In the **Name** field, enter a unique name of up to 40 characters.
4. In the **Description** field, enter a short, informative description of up to 40 characters.
5. In the **Element Name** field, enter the name of the element in its entirety.
For example, to use the default workflow element `SendEmailToAssignees`, enter:
`OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees`
6. In the **Arguments** field, use the following convention to supply values for the arguments or variables that the element needs to function correctly:

```
name=value
```

For the `SendEmailToAssignees` element, for example, you must provide a value for the `emailname` variable (that is, the name of an email object). For example:

```
emailname=AssignmentDueReminder
```

If an element takes more than one variable, separate the name/value pairs with the ampersand (&) character. For example:

```
name1=value1&name2=value2
```

7. Click **Add New Action**.

10.3.3.2 Configuring Approve for Publish Step Actions

The approval process approves an asset to a specific publishing destination. To use the default Approve for Publish step action with your workflow processes, you specify the publishing destination for the assets that are approved by the action.

If all the assets for all of your workflow processes are published to the same destination, you can simply configure the existing Approve For Publish action. However, if some types of assets are published to a different destination than the others, you must create an additional Approve for Publish action for each publishing destination, or combination of publishing destinations.

The Approve for Publish step action uses the OpenMarket/Xcelerate/Actions/Workflow/StepActions/ApproveForPublish element which takes the targets variable. You can use this same element with as many additional approval step actions as you need.

For the default Approve for Publish step action, provide a value for the targets variable. Possible values for the targets variable are the names of any of the publishing destinations that have been created on this WebCenter Sites system.

For example:

```
targets=serverX.
```

To specify more than one publishing destination, separate each with a comma. For example:

```
targets=serverX,serverY
```

Note: If the name of a publishing destination that is referenced by the targets variable is changed, you must also change the value of the targets variable in your Approve for Publish steps.

10.3.3.3 Editing Actions and Conditions

To edit actions and conditions

1. In the **Workflow** tab, expand **Actions**.
2. Under **Actions**, expand the category describing the action you want to edit.
The available categories are: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.
3. Under the selected category, double-click the action you want to edit.
4. In the action bar, click **Edit**.
5. Make your changes, then click **Save**.

10.3.3.4 Deleting Actions and Conditions

To delete actions and conditions

1. In the **Workflow** tab, expand **Actions**.
2. Under **Actions**, expand the category describing the action you want to delete.
The available categories are: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.
3. Under the selected category, double-click the action you want to delete.
4. In the action bar, click **Delete**.
WebCenter Sites displays a warning message.
5. Click **Delete Action**.

10.3.4 Setting Up the Timed Action Event

Before any of your timed actions can be triggered, you must configure the timed action event so that it calculates deadlines at regularly occurring intervals. Note that there can be only one timed action event per WebCenter Sites system.

To set up timed action events

1. In the **Workflow** tab, double-click **Timed Action Event**.
2. In the action bar, click **Edit**.

WebCenter Sites displays the Edit Workflow Timed Action Event form:

Figure 10–6 Edit Workflow Timed Action Event

Edit Workflow Timed Action Event

Enabled Dates and Times

<p>Hours: All</p> <p>Minutes: Every 5 Minutes of each selected hour</p> <p>Dates: All</p> <p>Days: All</p> <p>Months: Monthly</p>	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled
---	---

Recurrence Pattern

Days of the week
Days of the month
Months

<p>Select: All None</p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>Sunday</td><td>Monday</td><td>Tuesday</td><td>Wednesday</td><td>Thursday</td><td>Friday</td><td>Saturday</td> </tr> </table>	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	<p>Information:</p> <p>Event will run only on the selected days of the selected months.</p> <p>Note: Selecting none and selecting all both have the same effect on the schedule.</p>
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday		

Times of Recurrence

<p>Hours</p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>12</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td> <td style="padding-left: 5px;">AM</td> </tr> <tr> <td>12</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td> <td style="padding-left: 5px;">PM</td> </tr> </table>	12	1	2	3	4	5	6	7	8	9	10	11	AM	12	1	2	3	4	5	6	7	8	9	10	11	PM	<p>Minutes</p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td>0</td><td>5</td><td>10</td> </tr> <tr> <td>15</td><td>20</td><td>25</td> </tr> <tr> <td>30</td><td>35</td><td>40</td> </tr> <tr> <td>45</td><td>50</td><td>55</td> </tr> </table>	0	5	10	15	20	25	30	35	40	45	50	55	<p>Information:</p> <p>Event will run only on the selected minutes during the selected hours.</p>
12	1	2	3	4	5	6	7	8	9	10	11	AM																												
12	1	2	3	4	5	6	7	8	9	10	11	PM																												
0	5	10																																						
15	20	25																																						
30	35	40																																						
45	50	55																																						

Cancel
Save

3. Set the schedule for how often the state deadlines should be calculated in terms of months, days, hours and minutes, depending on the selected interval.

Selected values will be highlighted in blue. All values are toggled separately, so it is not necessary to use Ctrl to select multiple values.

The method of setting a timed action event is similar to the method for setting a publication event. See "[Example Schedule](#)" on page 18-8 for more information.

4. In the **Enabled Times and Dates** field, select **Enabled**.
5. Click **Save**.

A summary of the schedule is displayed.

WebCenter Sites uses the same abbreviations and codes to summarize the schedule for the timed action event that it does for your publishing events. For information about how to read the schedule, see [Section 18.8, "Reading the Schedule Abbreviations."](#)

10.3.5 Setting Up the States

When you create a workflow state, you specify a deadline, select a timed action, and configure when the timed action should run. Therefore, before you begin creating your workflow states, be sure that you have created the timed actions that you need.

To work with workflow states, your user name must be assigned the WorkflowAdmin role for the site that you are working with.

This section contains the following topics:

- [Section 10.3.5.1, "Creating Workflow States."](#)
- [Section 10.3.5.2, "Editing Workflow States."](#)
- [Section 10.3.5.3, "Deleting Workflow States."](#)

10.3.5.1 Creating Workflow States

To create workflow states

1. In the **Workflow** tab, expand **States** and double-click **Add New**.

WebCenter Sites displays the Add New Workflow State form:

Figure 10–7 Add New Workflow State Form

Add New Workflow State

2. In the **Name** field, enter a unique, meaningful name of up to 40 characters.
3. In the **Description** field, enter a short, informative description of up to 40 characters.
4. (Optional) In the **Estimated Time** fields, configure the deadline for assets in this state. You can specify a deadline in terms of days, hours, or a combination of the two.
5. (Optional) Click **Add Timed Actions** and complete the following steps:

Figure 10–8 Add Timed Action to Workflow State Form

Add Timed Action to Workflow State: Workflow for Flex Assets

Name:	Flex Workflow		
Description:	Workflow for Flex Assets		
Estimated Time:	10days:0hours		
*Add Timed Action:	Action to Take	Days	Hours
	SendEmail	00	0
			Offset
			Before Deadline
			After Deadline

Cancel

Add Timed Action

- a. In the **Action to Take** list, select a timed action.
 - b. In the **Offset** field, specify whether the action should be triggered before the deadline or after the deadline.
 - c. In the **Days** field and/or the **Hours** field, specify how many hours or days before or after the deadline (that you specified in step 5) that the action is to be triggered.
 - d. Click **Add Timed Action**.
 - e. Repeat this entire step for each timed action that you want to set up for this state.
6. Click **Add New State**.

10.3.5.2 Editing Workflow States

To edit workflow states

1. In the **Workflow** tab, expand **States** and double-click the state to edit.
2. In the action bar, click **Edit**.
3. In the Edit form, make the desired changes.
4. (Optional) To change the timed action or the time that it is scheduled to run, click **Add Timed Actions**, make the appropriate changes, and click **Add Timed Actions** again.
5. Click **Save**.

10.3.5.3 Deleting Workflow States

To delete workflow states

1. In the **Workflow** tab, expand **States** and double-click the state you want to delete.
2. In the action bar, click **Delete**.
3. WebCenter Sites displays a warning message.
4. Click **Delete State**.

10.3.6 Setting Up the Workflow Processes

When you create a workflow process, you specify global process information and then you create steps, assigning step actions to them as needed. The steps in the process create the flow of the process by linking the states in a specific order.

Before you can begin creating a workflow process, you must have already created your step actions, step conditions (if necessary), and states.

To work with workflow processes, your user name must be assigned the Workflow Admin role for the site that you are working with.

Note: If your content providers will use workflow groups, be sure to enable the Workflow Groups tab for them. For information, see [Section 9.5.2.2, "Creating the 'Workflow Groups' Tab"](#).

This section contains the following topics:

- [Section 10.3.6.1, "Creating Workflow Processes."](#)
- [Section 10.3.6.2, "Editing Workflow Processes."](#)
- [Section 10.3.6.3, "Editing Steps."](#)
- [Section 10.3.6.4, "Editing Function Privileges."](#)
- [Section 10.3.6.5, "Copying Workflow Processes."](#)
- [Section 10.3.6.6, "Deleting Workflow Processes."](#)
- [Section 10.3.6.7, "Deleting Steps."](#)
- [Section 10.3.6.8, "Deleting Function Privileges."](#)

10.3.6.1 Creating Workflow Processes

When you create a workflow process, you configure three categories of information: global process settings, steps, and function privileges.

Step A: Name and Set Global Settings for the Process

1. In the **Workflow** tab, expand **Processes** and double-click **Add New**.
WebCenter Sites displays the Workflow Process: (new) form:

Figure 10–9 Workflow Process Form

Workflow Process: (new)

***Process Name:**

***Description:**

***Site:**
 Any
 GE Lighting
 BurlingtonFinancial
 HelloAssetWorld

***Asset Type:**
 Any
 Article Flex
 Image Flex
 Recommendation
 Article

***Roles:**
 AdvancedUser
 Analyst
 Analytics
 Approver
 ArtworkAuthor

***Start Step:** No process steps are currently defined.

Administration Roles:
 Any
 AdvancedUser
 Analyst
 Analytics
 Approver

Process Deadline: allowed
 not allowed

Delegate Actions: No delegate actions are currently defined.

ID: (new)

2. In the **Name** field and enter a unique name of up to 25 characters.
3. In the **Description** field, enter a short, informative description of up to 64 characters.
4. In the **Sites** list, select the sites that can use this workflow process.
5. In the **Asset Type** list, select the asset type(s) that this workflow process is for.
6. In the **Roles** list, select the roles that will participate.

7. In the **Administration Roles** list, select the roles that can act as the administrator for this workflow process when an asset is using it.
8. In the **Process Deadline** section, specify whether the workflow administrator (or other user if you configure function privileges that allow it) can set a process deadline for the assets that participate in this workflow process.
9. If you have configured one or more delegate actions, select the appropriate actions in the **Delegate Actions** list.
10. In the **Start Step** section, click **Add New Step**.

WebCenter Sites displays the Add New Workflow Process Step form.

Step B: Create the Start Step

Figure 10–10 Add New Workflow Process Step Form

Add New Workflow Process Step

Process Name: sales_workflow

*Step Name:

*States: From State To State

none - (Start of Workflow) FSII: Approved and Locked FSII: Ready for Approval FSII: Ready to Edit	none - (End of Workflow) FSII: Approved and Locked FSII: Ready for Approval FSII: Ready to Edit
--	--

*Authorized Roles:

- AdvancedUser
- Analytics
- Approver
- ArtworkAuthor

*Assignment Method:

Retain "From State" assignees

No assignments; control actions with function privileges

Assign from list of participants (Select Roles)

Choose assignees when step is taken

Assign to Everyone

Assignment Deadline:

Can change

Use default

Step Actions:

Available Action(s)	Selected Action(s)
ApproveForPublish NotifyAllParticipants SendAssignmentEmail SendRejectionEmail SendRevisionNoticeEmail	

Step Conditions:

Deadlock Actions: Please add Deadlock Actions to choose from

Voting: All assignees must vote

Workflow Groups: Step is group synchronized

1. In the **Step Name** field, enter a unique, meaningful name of up to 64 characters.
2. Configure the states as follows:
 - a. In the **From State** list, select **none (start of workflow)**.
 - b. In the **To State** list, select the name of the first state in the workflow process.
3. In the **Authorized Roles** list, select the roles that are allowed to take this step by assigning this workflow to an asset.
4. In the **Assignment Method** section, specify which roles will be assigned the asset by selecting one of the following options:
 - **Retain "From State" assignees**, which, for a start step, means that the user who assigns the workflow to the asset is assigned the asset
 - **No assignments; control access with function privs**
 - **Assign from list of participants**
 - **Choose assignees when step is taken**

If you select either **Assign from list of participants** or **Choose assignees when step is taken**, select the appropriate roles from the list that is to the right of those options. (For definitions of these options, see [Section 10.2.5, "Determine the Steps"](#).)
 - **Assign to everyone**
5. In the **Assignment Deadline** section, determine whether the workflow administrator (or another user holding the appropriate function privileges) can override the Estimated Time (deadline) set for the state that this step moves the asset to.
6. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
7. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
8. (Optional) Check Voting to select that all assignees must vote to approve.
9. (Optional) Check Workflow Groups to have the step group synchronized.
10. Click **Save**.

The start step is added and the Steps for Workflow Process form is displayed.
11. Click **Save**.

WebCenter Sites saves the process and displays it in the Inspect form. Note that this step appears as the **Start Step** in the form.
12. Continue to the next procedure.

Step C: Create the Subsequent Steps

Use the following procedure to create the rest of the subsequent steps, including the end step:

1. In the Inspect form of the workflow process, click **Edit** in the action bar.
2. At the bottom of the Edit form, click **Add/Edit Steps**.
3. In the Steps for Workflow Process form, click **Add New Step**.

WebCenter Sites displays the Add New Workflow Process Step form.
4. In the **Step Name** field, enter a unique, meaningful name of up to 64 characters.

5. Configure the states as follows:
 - a. In the **From State** list, select the name of the state that you selected as the **To State** for the previous step.
 - b. In the **To State** list, select the name of the next state in the workflow process. (Note that if you want to create an iterative step, select the same state in the **From State** and the **To State** lists.)
6. In the **Authorized Roles** list, select the roles that are allowed to take this step by finishing the assignment. The roles that you select from this list should either match or contain a subset of the roles that you selected for the **Assignment Method** in the workflow step that immediately precedes this workflow step.
7. In the **Assignment Method** section, specify which roles will be assigned the asset by selecting one of the following options:
 - **Retain "From Step" assignees**

If the **From State** and the **To State** are different, the first person who takes this step (finishes the assignment) is assigned the asset, even if the previous step assigned the asset to more than one participant.

If the **From State** and the **To State** are the same, every user who was assigned the asset by the last step is assigned the asset again with this step.
 - **No assignments; control access with function privs**
 - **Assign from list of participants**
 - **Choose assignees when step is taken**

If you select either of the last two options above, select the appropriate roles from the list that is to the right of those options. (For definitions of these options, see [Section 10.2.5, "Determine the Steps"](#))
 - **Assign to everyone**
8. In the **Assignment Deadline** section, determine whether the workflow administrator (or other user if you configure the appropriate function privileges) can override the Estimated Time (deadline) set for the state that this step moves the asset to.
9. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
10. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
11. If the step moves the asset from a state in which more than one participant was working on the asset and you want all participants to finish their assignments before the step can complete, select the **All Assignees must vote** option.
12. If there are any other steps in this process with the same **From State** and any of those steps are also all-voting steps (**All Assignees must vote** is selected), select a **Deadlock Action**. For information about deadlocks and avoiding them, see [Section 10.1.5, "Managing Deadlocks"](#) and [Section 10.2.3.3, "About Delegate Actions."](#)
13. If this workflow process will be used for workflow groups and you want all the assets to progress at the same time to the **To State** that you selected in step 4 of this procedure, select the **Step is group synchronized** option. It is recommended that you create only one synchronized step in the process.
14. Click **Save**.

The step is saved and the Steps for Workflow Process form is displayed.

15. Click Save.

WebCenter Sites saves the process and displays it in the Inspect form.

16. Repeat this procedure for each step—except the end step—that you want to create for this process.

To create the end step, continue with ["Step D: \(Optional\) Create the End Step"](#) on page 10-42.

To configure function privileges, continue to ["Step E: \(Optional\) Configure Function Privileges"](#) on page 10-43.

Step D: (Optional) Create the End Step

An end step ends the workflow which means that the asset is no longer in a process and no longer has function privileges controlling access to it, if you are using function privileges. Use the following procedure to create an end step for the workflow process:

1. In the Inspect form of the workflow process, click **Edit** in the action bar.
2. In the Edit Process form, click **Add/Edit Steps** (a button at the bottom of the form).
3. In the Steps for Workflow Process form, click **Add New Step**.

WebCenter Sites displays the New Workflow Process Step form.

4. In the **Step Name** field, enter a unique, meaningful name of up to 64 characters.
5. Configure the states as follows:
 - a. In the **From State** list, select the name of the state that you selected as the **To State** for the previous step.
 - b. In the **To State** list, select none (**end of workflow**).
6. In the **Authorized Roles** list, select the roles who are allowed to take this step by finishing an assignment. The roles that you select from this list should either match or contain a subset of the roles that you selected for the **Assignment Method** of the workflow step that immediately precedes this workflow step in the process.
7. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
8. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
9. If the step moves the asset from a state in which more than one participant was working on the asset and you want all participants to finish their assignments before the step can complete, select the **All Assignees must vote** option.
10. If there are any other steps in this process with the same **From State** and any of those steps are also all-voting steps (**All Assignees must vote** is selected), select a **Deadlock Action**. For information about deadlocks and avoiding them, see [Section 10.1.5, "Managing Deadlocks"](#) and [Section 10.2.3.3, "About Delegate Actions."](#)
11. If this workflow process will be used for workflow groups and you want this step to be performed on all the assets in the group at the same time, select the **Step is group synchronized** option. It is recommended that you create only one synchronized step in the process.
12. Click **Save**.

The start step is added and the Steps for Workflow Process form is displayed.

13. Click Save.

WebCenter Sites saves the process and displays it in the Inspect form.

If you do not need to configure function privileges, this workflow process is completed. If you do need to configure function privileges, continue to the next procedures.

Step E: (Optional) Configure Function Privileges

To configure function privileges

1. Find the desired workflow process and open its Inspect form.
2. In the action bar, click **Edit**.
3. Click **Add/Edit Function Privileges** at the bottom of the form.
4. In the Functions for Workflow Process form, scroll down to the function for which you want to set a privilege.
5. Click **New** next to the desired function.

WebCenter Sites displays the Add Function Privilege form:

Figure 10–11 Add Function Privilege Form**Add Function Privilege**

Process Name: FSII: Approval for Content

Function: Authorize

***State:** FSII: Approved and Locked
FSII: Ready for Approval
FSII: Ready to Edit

***Role:** Approver
ContentAuthor
ContentEditor

Allowed?

Cancel **Add New**

6. Select the appropriate state from the **State** list.
7. In the **Roles** list, select the roles that are allowed or not allowed to perform this function when an asset is in this state.
8. Do one of the following:
 - To allow users with the selected roles to perform the function, select the **Allowed** check box.
 - To restrict users with the selected roles from performing the function, clear the **Allowed** check box.
9. Click **Add New**.
WebCenter Sites saves the privilege and re-displays the Functions for Workflow Process form.
10. Repeat steps 3–8 for each function privilege that you need to configure. For more information about function privileges, see [Section 10.2.6, "Determine the Function Privileges."](#)
11. After you have configured all of your function privileges, click **Save**.
WebCenter Sites saves the workflow process and re-displays the Workflow Process form.

The workflow process is complete.

10.3.6.2 Editing Workflow Processes

To edit workflow processes

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to edit.
2. In the action bar, click **Edit**.
3. Make your changes as follows:
 - To modify the name, description, or any of the other global settings, make your changes directly in the form.
 - To edit the steps, see "[Editing Steps](#)" on page 10-45.
 - To edit the function privileges, see [Section 10.3.6.4, "Editing Function Privileges."](#)
4. When you are finished, click **Save**.

10.3.6.3 Editing Steps

To edit a workflow step

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to work with.
2. In the action bar, click **Edit**.
3. In the Workflow Process form, click **Add/Edit Steps** (at the bottom).
4. In the Steps for Workflow Process form, click **Edit** next to the step you want to edit.
5. In the Edit Workflow Process Step form, make your changes, then click **Save**.
WebCenter Sites saves the process and displays it in the Inspect form.

10.3.6.4 Editing Function Privileges

To edit a function privilege, you must delete it and then re-create it.

To edit a function privilege

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to work with.
2. In the action bar, click **Edit**.
3. In the Workflow Process form, click **Add/Edit FunctionPrivs** (at the bottom).
4. In the Functions for Workflow Process form, click **Remove** next to the function that you want to change.
5. In the dialog box that appears, click **OK**.
6. Click **New** next to the function you just removed.
7. In the Add Function Privilege form, make your selections and click **Save**.
8. In the Functions for Workflow Process form, click **Save**.

WebCenter Sites saves the workflow process and displays it in the Inspect form.

10.3.6.5 Copying Workflow Processes

You can copy a workflow process, which can save you some steps in configuring additional processes.

To copy a workflow process

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to copy.
2. In the action bar, select **Copy Process** from the drop-down list.
WebCenter Sites displays the Copy Workflow Process form:

Figure 10–12 Copy Workflow Process Form

Copy Workflow Process: FSII: Approval for Content

*Process Name:

*Description:

3. In the **Process Name** field, enter a unique name for the process.
4. In the **Description** field, enter a short, informative description of the process.
5. Click **Save**.
6. Edit the process as necessary. See the following procedures for help:
 - [Editing Workflow Processes](#).
 - [Editing Steps](#).
 - [Editing Function Privileges](#).

10.3.6.6 Deleting Workflow Processes

To delete a workflow process

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to delete.
2. In the action bar, click **Delete**.
WebCenter Sites displays a warning message.
3. Click **Delete Process**.
The process has been deleted.

10.3.6.7 Deleting Steps

To delete workflow steps

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to work with.
2. In the action bar, click **Edit**.
3. In the Edit Process form, click **Add/Edit Steps** (at the bottom of the form).
4. In the Steps for Workflow Process form, click **Remove** next to the step you want to delete.
5. In the dialog box that appears, click **OK**.
6. Click **Save**.

WebCenter Sites saves the workflow process and displays it in the Inspect form.

10.3.6.8 Deleting Function Privileges

To delete function privileges

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to edit.
2. In the action bar, click **Edit**.
3. In the Workflow Process form, click **Add/Edit FunctionPrivs** (at the bottom of the form).
4. In the Functions for Workflow Process form, click **Remove** next to the function privilege you want to delete.
5. Click **Save**.

WebCenter Sites saves the workflow process and displays it in the Inspect form.

10.3.7 Testing Your Workflow Process

Before you move your workflow process to the management system and implement it there, test it on your development system.

To test your workflow process

- Log in as a user with administrator access and configure start menu items that assign workflow processes to assets, if necessary.
- Log in as a user who has a role that has been authorized to take the start step of the workflow process. Create an asset, select the workflow process (only if the start menu item does not assign it), and then finish the assignment.
- Log in as the participant who has the assignment now. Finish the assignment and then log in as the next participant. Continue through the entire workflow in this manner.
- Verify that the email messages that should be sent are being sent.
- Verify that your workflow sends the asset through the process correctly.

10.4 Moving Your Work

Typically you create and fine-tune a workflow process on a development system to ensure that it functions exactly as you need it to before you introduce that workflow process to the management system.

When your workflow is ready to be used by content providers, use the Initialize Mirror Target feature to move your workflow components to the management system.

For information about this feature, see [Section 18.1, "Migrating a Site from One System to Another."](#)

10.5 Clearing Workflow Assignments

People go on vacation, get reassigned to new work groups, and move on to different jobs. What should happen to their workflow assignments in these situations? One option for handling work assignments that cannot be finished by the original assignee is to delegate the assignments to someone else. However, even with the delegate feature, you, the administrator, might still be called on to clear assignments from some user's assignment list.

To clear assignments for a user

1. In the **Workflow** tab, double-click **Clear Assignments**.
2. In the Search for Assignments form, enter the desired user name. For example:

Figure 10–13 Search for Assignments Form

Search for Assignments

*User Name:

Show Assignments

3. Click **Show Assignments**.

WebCenter Sites displays a list of assets that are currently assigned to that user.

For example:

Figure 10–14 Clear Assignments Form**Clear Assignments**

The following assets have been assigned to: **firstsite**.

To clear an assignment, select the asset's check box and click Clear Assignments.

Type	Name	Description	Assigned to	Task Status	Clear
Page	FSIIHome	Home	firstsite(Designer)	active	<input type="checkbox"/>
Page	Home (fr)	Page Principale	firstsite(Designer)	active	<input type="checkbox"/>
Page	Home (de)	Home (de)	firstsite(Designer)	active	<input type="checkbox"/>
Page	Home (es)	Home (es)	firstsite(Designer)	active	<input type="checkbox"/>

Clear assignment comment:

Remove the asset from workflow if there are no other assignees for the asset:

Yes

No

Clear Assignments

- In the Clear Assignments form, select the **Clear** check box next to each assignment you want to clear.
- (Optional) In the **Clear assignment comment** field, enter a brief explanation of the reason for which you are clearing the selected assignments. The explanation you enter in this field appears in the **Action Taken** field in the **Workflow History** section of the asset's Status form.
- Do one of the following:
 - If you want the asset to be removed from workflow if it is not assigned to anyone else, select **Yes**.
 - If you want the asset to remain on the assignment list of any other user who is also assigned the asset, select **No**.
- Click **Clear Assignments**.

WebCenter Sites displays a Clear Assignments Report summarizing your changes.

Replicating Content Management Sites

To speed up the deployment of online sites, Oracle WebCenter Sites provides Site Launcher. This new feature enables you to replicate a suitably-prepared Content Management (CM) site and, in the process, either share or copy its components to the new site. You can then modify the new site as necessary, and deploy it. This chapter provides an overview of Site Launcher, replication guidelines, requirements and options, and procedures for enabling and using Site Launcher.

This chapter contains the following sections:

- [Section 11.1, "Site Launcher Overview"](#)
- [Section 11.2, "Preparing for Replication"](#)
- [Section 11.3, "Site Replication Steps"](#)
- [Section 11.4, "Post-Replication Tasks and Guidelines"](#)

11.1 Site Launcher Overview

To minimize your effort in creating new sites, WebCenter Sites provides a site-replication utility called Site Launcher. This utility is designed not for backing up CM sites, but for spinning them off.

For example, your management system hosts a dedicated CM site for a department named "Products." A new department named "Services" has been recently established and must be quickly introduced to the public. "Services" is similar to "Products" in size and structure, although its members are public relations specialists rather than advertising staff. Instead of creating a "Services" CM site from scratch, you can replicate the "Products" CM site directly on the management system, and modify the replicate to accommodate the newly established "Services" department.

Site Launcher replicates source sites directly on their native WebCenter Sites system, re-using the existing database schema. The sites are replicated quickly and easily, without the need for coding. However, while replication itself is a quick and straightforward procedure, it does require preparation and follow-up on the part of the administrator. How much, depends on the content management needs.

The rest of this chapter provides the steps that you need to follow in order to successfully replicate sites for use in any content management model—1:1, 1:n, or x:n, described in [Section 1.6, "Content Management Models"](#).

Note: Site replication can be carried out only by the general administrator. Site and workflow administrators cannot replicate the sites they manage, as they have no access to Site Launcher.

11.2 Preparing for Replication

Before attempting to replicate a site, you need to consider several recommendations regarding the nature of the site and its replicate. You must also make decisions about which components to copy or share, and finally ensure that the source site meets system requirements for replication. This section outlines our recommendations, your options, and the system requirements.

This section contains the following topics:

- [Section 11.2.1, "Ensuring the Source Site Meets Replication Requirements"](#)
- [Section 11.2.2, "Planning the New Site"](#)

11.2.1 Ensuring the Source Site Meets Replication Requirements

Site Launcher can be used to replicate almost any CM site: small, large, functional, incomplete, independent of other sites, and overlapping other sites by the sharing of components.

However, to use Site Launcher most effectively, it is best for you to start with a site that is small, functional, and similar to the site that you need to have. In general, the source site should be a skeletal one, meaning that it has structure and design, but little content. These characteristics, beyond ensuring a tractable replicate, will help you conserve resources and minimize replication time.

In addition, make sure that the source site resides on the management system. (If you need to first create a replicable source site, follow the steps in [Section 3.3, "Site Configuration Steps"](#).)

11.2.2 Planning the New Site

This section outlines your site replication options and the system requirements.

This section contains the following topics:

- [Section 11.2.2.1, "Copying vs. Sharing"](#)
- [Section 11.2.2.2, "Naming Assets"](#)
- [Section 11.2.2.3, "Planning Users"](#)
- [Section 11.2.2.4, "Choosing the Replication Time"](#)

11.2.2.1 Copying vs. Sharing

When designating a source site, you must decide whether to copy or share certain site components, and therefore, determine from the start, how the new site will function in relation to its source site—as a duplicate or a subset; as an independent site or an overlapping site. (On a bigger scale, you will also need to determine how the new site will function in relation to other sites in your content management model.)

The table below shows that when the source site is being replicated, most of its components are shared by default. Other components are shared or copied at your discretion. Users are neither copied nor shared.

Table 11–1 Components and Replication

Source Site Component	Replication method
asset subtypes	Shared
asset types	Shared

Table 11–1 (Cont.) Components and Replication

Source Site Component	Replication method
assets	Either copied or shared (admin decides)
associations	Shared
Sites Desktop and Sites DocLink configuration	Copied
publishing destinations	Shared
roles	Shared
start menu items	Shared
templates (design assets)	Either copied or shared (admin decides)
tree tabs	Shared
users	Neither shared nor copied
workflow processes	Shared

Note that copying and sharing of assets is selective at the asset type level, but not at the asset level—when you select an asset type to be copied (or shared), all assets of that type are copied (or shared). The same holds for templates (design assets).

Concerning assets and templates, your specific tasks are to determine:

- Which asset types must have their assets copied or shared.
- Whether template assets must be copied or shared.
- Whether assets will be previewed on the new site as formatted content.

To determine whether assets and templates must be copied or shared, use the following guidelines:

- Sharing an asset does not require you to share the template that renders the asset. The template can be copied.

Whether you share or copy an asset depends on how the rendering logic in the template treats asset names. For example, template logic can be written in such a way as to assume either shared templates (constructed with explicit template names in the `render:satellitepage` and `render:calleelement` tags) or copied templates (constructed with a template name using `Variables.site` prepended to the name in the `render:satellitepage` and `calleelement` tags).

The same reasoning applies to other asset types. If in your rendering logic you use asset names such as flex attribute name, or image name, then the template logic will treat the asset type as either shared (explicit name) or copied (constructed name), and you need to designate the source site accordingly.

If you decide to copy template assets, make sure that the template assets do not contain hard-coded template names, and a template is not named in the Default Values list of any start menu item.

- When assets are shared to a new site by Site Launcher, their data is not changed. The single exception is Template assets. A Template asset maintains a SiteCatalog entry for each site to which it is shared. Therefore, when a template is shared to a new site, the template's list of SiteCatalog entries is updated (the new site is added to the list of SiteCatalog entries).
- Because data in a shared asset is not changed, asset references of that asset also remain unchanged. Therefore, if assets with asset references are to be shared, the

asset references must also be shared. For example, if you have Article assets that have references to ImageFile assets, you may

- share both asset types,
- copy both asset types, and
- copy the Articles and share the ImageFiles.

However, you must not share the Article and copy the ImageFiles. Doing so will cause errors to be displayed when a copy of the site is launched.

- Start menus are shared among the source site and its replicates. Therefore, if a workflow process is set in the source site's start menu, that workflow will be set in the replicate sites. When users in the new site invoke the start menu item to create an asset, the workflow will take effect.

Your decisions in the planning stage are governed by the specifics of your installation as well as your own methods of managing the WebCenter Sites environment. The site that you create by replication must ultimately fit in to the content management model that is being (or has been) established on the management system. For example, if you have central sites and centrally managed sites, you need to determine how your new site will function in relation to them. For information about site modeling, see [Section 1.6, "Content Management Models"](#).

11.2.2.2 Naming Assets

Assets that are copied to the new site are named algorithmically, by use of prefixes as follows:

- The prefix for the source is specified when you designate a site as a site launcher source site and is, by default, the name of the site.
- The prefix for the new site is specified in Site Launcher itself and is, by default, the name of the new site. When specifying a prefix, make sure it is short.
- If the source asset has a prefix, or the asset type requires a unique name, then a prefix will be used in the name of the new asset. If the source asset has a prefix, the new asset will have the new prefix in place of the old prefix.
- If the source asset does not have a prefix, then the prefix is prepended to the asset name to make the new asset name.

11.2.2.3 Planning Users

Given that users are not copied or shared to the new site during replication, you will have to manually add them to the new site.

If the users will be different from those on the source site, you will need to create new user accounts. In the process, consider the users' functions on the new site, the roles they will need, their access to start menu items, workflows, tree tabs, and so on.

Bear in mind that the original associations among components are preserved on the new site. If you redefine users' functions, you may have to redefine their associations with other components on the site. Accounting for new associations in the planning stages will help you to assess your post-replication workload and create a functional site that also meets your expectations.

11.2.2.4 Choosing the Replication Time

Because normal operations affect site replication, it is best to run Site Launcher when the source site is not being actively edited. Otherwise, the replicate site might not match the source site. Also, make sure that prior to replication, all assets in the source

site are checked in to the database. Otherwise, if revision tracking is enabled and an asset to be shared is checked out, the sharing of the asset will fail.

11.3 Site Replication Steps

This section shows you how to enable and use Site Launcher. The procedure consists of three basic steps:

1. [Section 11.3.1, "Ensure that Replication Requirements Are Satisfied"](#)
2. [Section 11.3.2, "Enable the Source Site for Site Launcher"](#)
3. [Section 11.3.3, "Replicate the Source Site"](#)

The steps are given in detail in the rest of this chapter.

11.3.1 Ensure that Replication Requirements Are Satisfied

1. Prepare for replication by ensuring that source sites and planned sites meet the requirements. For more information, see [Section 11.2, "Preparing for Replication."](#)
2. Go to [Section 11.3.2, "Enable the Source Site for Site Launcher"](#) to enable the source site.

11.3.2 Enable the Source Site for Site Launcher

In order to replicate a source site, you must first configure Site Launcher.

To configure Site Launcher

1. Make sure that you have completed the steps shown above ([Section 11.3.1, "Ensure that Replication Requirements Are Satisfied"](#)).
2. When you complete the steps in this section, be prepared to follow up immediately with the steps in the next section, [Section 11.3.3, "Replicate the Source Site."](#) Immediate follow-up ensures that the condition of the source site has not changed.
3. In the **Admin** tab, expand **Sites** and double-click the site you want to replicate. This is your source site.
4. In the site's **Inspect** form, click **Configure Site Launcher for this site** (at the bottom of the form).
5. In the **Configure Site Launcher** form, do the following:
 - a. If you want to copy assets, go to the **Asset prefix** field and either enter a prefix for the assets, or accept the default (the name of the source site).
 - b. In the **Enabled Asset Types** section, locate the asset types whose assets you want to copy or share, and click their **Copy** or **Share** radio buttons.

Note: If you want to copy or share all assets of all types, click the **Copy All** or **Share All** button.

- c. Click the **Save** button to save your configuration options.

Figure 11–1 Configure Site Launcher Form

Configure Site Launcher: FirstSiteII

*Asset prefix: [Disable Site Launcher](#)

i All assets will be shared or copied to new site. Copied assets will be renamed using the asset prefix, if a unique name is required. All start menu items will be shared with new site. All Sites Desktop and Sites DocLink configuration information will be copied to the new site.

Enabled Asset Types:

Description	Asset Type	Skip Data	Copy	Share
Attribute Editor	AttrTypes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
CSElement	CSElement	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Content	Content_C	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Content Attribute	Content_A	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Segment	Segments	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site Visitor	FSIIVisitor	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site Visitor Attribute	FSIIVisitorAttr	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site Visitor Definition	FSIIVisitorDef	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site Visitor Parent	FSIIVisitorParent	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Site Visitor Parent Definition	FSIIVisitorPDef	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
SiteEntry	SiteEntry	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Template	Template	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
UI Configuration	FW_UIConfiguration	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Visitor Attribute	ScalarVals	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Workflow Process: **i** All workflow processes will be shared with new site.

Workflow

- FSII: Approval for Artwork
- FSII: Approval for Content
- FSII: Approval for Documents
- FSII: Approval for Products
- FSII: Approval for Promotions
- FSII: Approval for Structure

- Go to the next section, [Section 11.3.3, "Replicate the Source Site,"](#) to complete the site replication procedure.

11.3.3 Replicate the Source Site

Once you have enabled the source site for replication, you can use Site Launcher to create as many copies of the source site as you wish.

To replicate the source site

- Make sure that you have completed the steps above ([Section 11.3.1, "Ensure that Replication Requirements Are Satisfied"](#), and [Section 11.3.2, "Enable the Source Site for Site Launcher"](#)).
- At the bottom of the Inspect form, click **Launch Copy**.

3. In the Site Launcher form, do the following:
 - a. In the **Name** field, enter the name for the new site.
 - b. In the **Description** field, enter the description for the new site.
 - c. In the Publish Destinations area:
 - a. Initialize the destinations by selecting the checkboxes, as necessary.
 - b. Select the publish destinations you would like to be available for the new site.
 - d. Click **Add Site** to copy the site.

Figure 11–2 Site Launcher Form

Site Launcher: New Site from FirstSiteII (FirstSite II)

***Name:**

***Description:**

***Asset prefix:**

Copied assets with names beginning with 'FSII' will be renamed with this value.

Enabled Asset Types:

Description	Asset Type	Skip Data	Copy	Share
Attribute Editor	AttrTypes		✓	
CSElement	CSElement		✓	
Content	Content_C		✓	
Content Attribute	Content_A		✓	
Content Definition	Content_CD		✓	
Content Filter	Content_F		✓	
Content Parent	Content_P		✓	
Content Parent Definition	Content_PD		✓	
Dimension	Dimension			✓
DimensionSet	DimensionSet			✓
Document	Document_C		✓	
Document Attribute	Document_A		✓	
Document Definition	Document_CD		✓	
Document Filter	Document_F		✓	
Document Parent	Document_P		✓	
Site Visitor Parent Definition	FSIIVisitorPDef		✓	
SiteEntry	SiteEntry		✓	
Template	Template		✓	
UI Configuration	FW_UIConfiguration		✓	
Visitor Attribute	ScalarVals		✓	

Workflow Process:

Workflow

- FSII: Approval for Artwork
- FSII: Approval for Content
- FSII: Approval for Documents
- FSII: Approval for Products
- FSII: Approval for Promotions
- FSII: Approval for Structure

Publish Destinations:

Enable?	Initialize
<input type="checkbox"/> Destination 1 (static)	<input type="checkbox"/> Set publish start point using 'FSII-Home'.
<input type="checkbox"/> Destination 2 (dynamic)	<input type="checkbox"/> As a production site.
	<input type="checkbox"/> As a management site.
<input type="checkbox"/> FSII Destination (RealTime)	<input type="checkbox"/> As a production site.
	<input type="checkbox"/> As a management site.
<input type="checkbox"/> FSII Destination (dynamic)	<input type="checkbox"/> As a production site.
	<input type="checkbox"/> As a management site.

Cancel **Add Site**

After the new site is created, WebCenter Sites displays a summary that indicates:

- Which assets were copied or shared. For each asset that is shared, WebCenter Sites adds a row in the `AssetPublication` table, indicating the site id (publication id) of the new site. For each asset that is copied, WebCenter Sites enters the copy into the table where the original asset is stored.
 - The number of start menu items, workflow processes, and publish destinations that were shared with the new site.
4. At this point, you need to add users and otherwise ensure that the new site is properly configured. For instructions and guidelines, see the next section, [Section 11.4, "Post-Replication Tasks and Guidelines."](#)

11.4 Post-Replication Tasks and Guidelines

When the new site is established, you need to complete its configuration by completing the following steps:

1. Add existing users to the new site, or establish new users, depending on how you planned the new site.
 - To add existing users, follow instructions in [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)
 - To establish new users, follow the guidelines in [Chapter 3, "Site Configuration Guidelines."](#)
2. If workflow processes exist, make sure that the correct asset types and roles are associated with the processes through Start Menu items.
3. Test the new site.
4. If necessary, create a site administrator. Be sure to assign the `xceladmin` ACL to the user account and give the user the `SiteAdmin` role for the new site.

Publishing with Oracle WebCenter Sites

Publishing with Oracle WebCenter Sites makes content available to the visitors of your online site by copying from the management system to the delivery system. Copying content from one system to another is called publishing.

The WebCenter Sites publishing system supports three publishing methods: Export to Disk, Mirror to Server, and Export to XML. WebCenter Sites also provides an approval system that determines which content gets published, a scheduling function that enables you to set the publication time, and a utility for configuring site-specific publishing destinations.

This chapter describes the publishing and approval systems. It contains the following sections:

- [Section 12.1, "Overview"](#)
- [Section 12.2, "Publishing Methods"](#)
- [Section 12.3, "Publishing Destinations"](#)
- [Section 12.4, "The Approval System"](#)
- [Section 12.5, "The Publishing Schedule"](#)
- [Section 12.6, "What Happens During a Publishing Session?"](#)
- [Section 12.7, "Obtaining Information About a Publishing Session"](#)

Note: The following feature is deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Mirror publishing. This feature is replaced by RealTime publishing.

12.1 Overview

Before assets can be published, either you or WebCenter Sites must determine the following information:

- Which publishing method WebCenter Sites should use: Export to Disk, Mirror to Server, or Export Assets to XML.

As an administrator, you specify the publishing method for the destination when you configure the publishing destination. For more information, see [Section 12.2, "Publishing Methods."](#)

- The publishing destination WebCenter Sites should publish to during a given session.

As an administrator, you configure the publishing destinations for your system. For more information, see [Section 12.3, "Publishing Destinations."](#)

- Which assets have been approved and are ready to be published to the current destination. The approval system determines this information.

When an asset is deemed ready for publication, a content provider marks it as approved for a specific publishing destination. The approval system validates the approved label to determine whether publishing the asset is likely to create broken links on the live site. If the potential for broken links exists, the asset is held back from the publishing session until its dependencies on other assets are resolved by the user approving those assets, as well. If the approved asset either has no dependencies or its dependencies are satisfied, the approval system releases the asset to the publishing system.

The approval system's process is complex and varies from one publishing method to another. For detailed information about the workings of the approval system, see [Chapter 13, "The Approval System"](#) and [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

- When assets should be published.

As an administrator, you set up the publishing schedule. The publishing process runs as a batch process that you schedule to occur at regularly-occurring intervals. On an as-needed basis, you can also override the schedule and publish on demand. For more information about publishing schedules, see [Section 12.5, "The Publishing Schedule."](#)

During a publishing session, the session information is recorded in log files. You can monitor the session through the Publish Console to determine both the publishing history and the status of currently running publishing sessions. You can also use your browser to complete other administrative tasks, as the publishing process runs in the background. For information about the events that occur during a publishing session, see [Section 12.6, "What Happens During a Publishing Session?"](#) For information on how to obtain publishing information, see [Section 12.7, "Obtaining Information About a Publishing Session."](#)

12.2 Publishing Methods

This section describes the publishing methods that WebCenter Sites supports. [Figure 12-1](#), [Figure 12-2](#), and [Figure 12-3](#) summarize the publishing methods.

- For **Export to Disk publishing**, the delivery system is a web server. Publishing to this type of system is done as follows:

Approved assets in the CM system database are rendered by templates into HTML files. The files are saved to a file system and subsequently published to the web server by an administrator using a transfer protocol, such as FTP.

When published content is requested by site visitors, the HTML files are served as pages to the browser.

- For **Mirror to Server publishing**, the delivery system is a WebCenter Sites system. Publishing to this type of system is done as follows:

Approved assets and their database tables are mirrored from the CM system database to the delivery system database. Throughout the publishing session, the publishing system communicates and cooperates with the **CacheManager** on the delivery system. The CacheManager is a WebCenter Sites servlet that manages a system's page cache. CacheManager ensures caching of the pagelets or pages that refer to the assets which

will be mirrored. After the publishing session concludes, CacheManager generates those pages again to display the updated content, and caches the new pages and pagelets.

When published content is requested by site visitors, it is provided on-the-fly. The content is drawn from the delivery system database by templates and served to the browser, where it is finally rendered as pages.

- For **Export Assets to XML**, the delivery system is a database or an external system running an application other than WebCenter Sites. The publishing method is a data transformation method that outputs XML files. Rather than creating pages that are ready to be displayed by a web server, this publishing method uses the Export API to create one XML file for each approved asset.

When published content is requested by site visitors, it is provided on-the-fly. The content is drawn from the delivery system database by templates and served to the browser, where it is finally rendered as pages.

These chapters describe each publishing method in detail:

- [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process"](#)
- [Chapter 15, "The Export to Disk Publishing Process"](#)
- [Chapter 16, "The Mirror to Server Publishing Process"](#)
- [Chapter 17, "The Export Assets to XML Publishing Process"](#)

Figure 12–1 *Export to Disk Publishing*

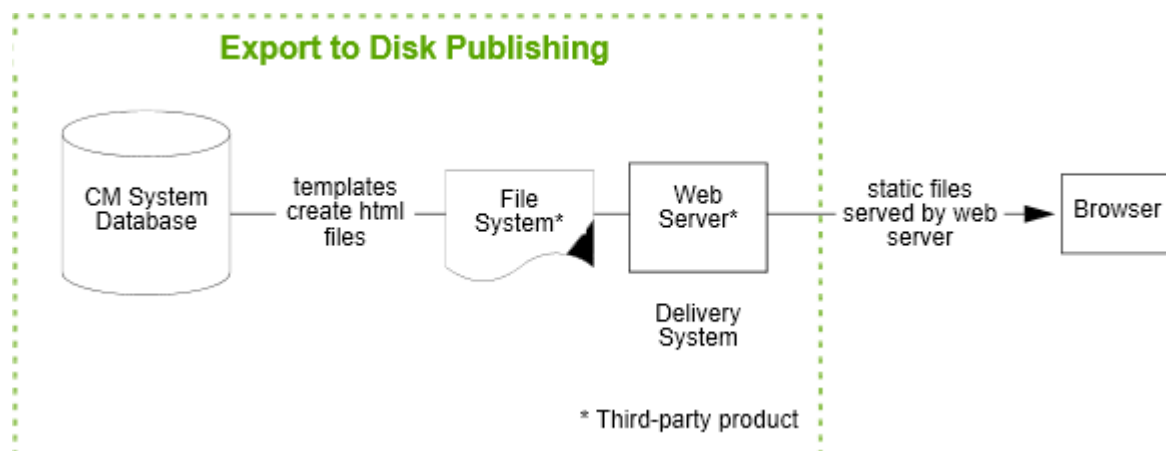


Figure 12–2 Mirror to Server Publishing

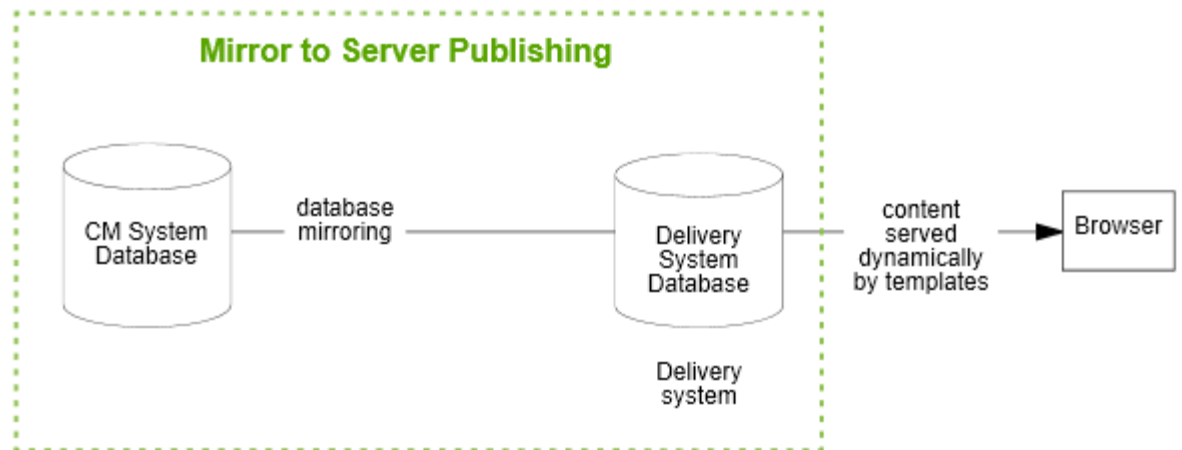
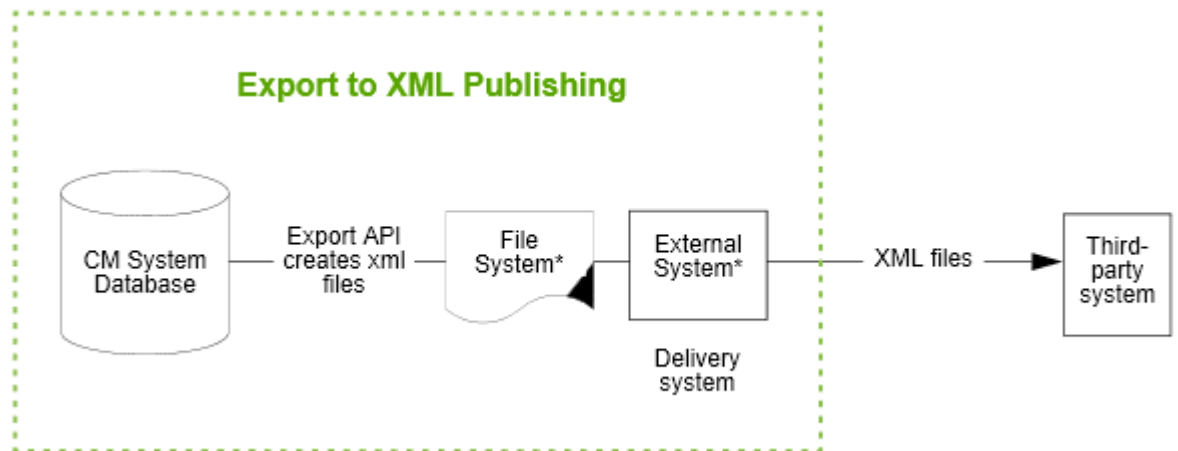


Figure 12–3 Export to XML Publishing



12.3 Publishing Destinations

A publishing destination is either the WebCenter Sites database to which you are mirroring content, or a directory to which you are exporting. A publishing destination is expressed as a named object that defines the following parameters:

- A publishing method (Export to Disk, Mirror to Server, or Export Assets to XML)
- A location
 - For Export to Disk publishing and Export Assets to XML, the publishing system converts the approved assets into HTML or XML files. The location is the root directory to which the files are saved. You set this directory with the `cs.pgexportfolder` property in the `futuretense.ini` file.
 - For Mirror to Server publishing, the destination is the **server name in URL format** (including the port number) of the server that is supposed to deliver the content. (You set the server name in the **Destination address** field of the Destination form.)
- The names of CM sites whose assets can be published to the destination.

For any publishing method, you can configure more than one publishing destination for your Sites system. Depending on how your online sites are designed by the developers, you may need to publish both static content (HTML files, using Export to Disk) and dynamic content (pages generated on the fly by the Mirror to Server or Export to XML publishing method). For this kind of dual implementation, consult with your developers to determine how to configure your publishing destinations.

In this document, the following terms are used when discussing publishing destinations and configuration:

- **Source.** The WebCenter Sites database that serves as the source for a publishing session. Because you can mirror assets from any Sites system to any other Sites system, the source is not always the content management system.
- **Destination.** Either the WebCenter Sites database that you are mirroring to or the directory that you are exporting to.

For information about creating publishing destinations, see [Section 15.4.3, "Configure an Export Destination"](#) and [Section 16.3.5, "Create a Mirror Destination."](#)

12.4 The Approval System

When assets are approved for publication on a specific destination, WebCenter Sites verifies that each asset is ready to be published by invoking the approval system. The approval system runs a pre-publication process that protects the online site against the possibility of broken links and outdated content. In this process, the approval system determines answers to the following questions:

1. Which assets are deemed to be ready for publishing—that is, which assets are marked by the user as approved?
2. Do the approved assets have dependencies? That is, must any other assets also accompany those approved assets? or, must any other assets also exist on the destination to ensure that all the links on the site will work correctly?

For example, you may have a site set up that has article assets that refer to image files. An approved article with an unapproved associated image file cannot be published until the image file is also approved.

Depending on the answers to these questions, the approval system either allows the publishing system to publish approved assets at the next publishing session, or it stalls the publication of assets that fail the dependency test.

Note: In Export to Disk publishing, the choice of templates can cause assets to bypass the approval system. For more information about the approval system in Export to Disk publishing, see [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

It is typical to build an approval step into your workflow processes. For information about workflow, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

For information about how to approve assets, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Detailed information about the workings of the approval system is given in [Chapter 13, "The Approval System"](#) and [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

12.5 The Publishing Schedule

A publishing session (no matter what the publishing method) runs as a background, batch process called a publish event. As the administrator, you configure the schedule for publishing events.

You can schedule publishing events to occur daily, weekly, hourly, every 15 minutes, or in various combinations of these time increments. When scheduling multiple publishing sessions, bear in mind several rules:

- Publishing to a given destination must be done serially.

Because each publishing session is a background event, a given destination can support only a single publishing session at a time. If publishing to a specific destination is still in progress when the next event for that destination is scheduled, the second event attempts to run and then fails, reporting that a publishing session to that destination is already underway.

Similarly, multiple sources must not be configured to publish simultaneously to the same destination. Doing so will cause publishing errors and problems with data integrity.

- Never schedule a publishing session to a destination for a time when the destination system could be publishing to another destination.

For example, you publish from the development system to the management system, and from the management system to the delivery system. Do not schedule a publishing session from the development system to the management system while the management system is publishing to the delivery system.

- Publishing to multiple destinations can be done simultaneously. You can publish to multiple destinations at the same time by setting up events for the destinations and selecting the same time for them.

Because a publishing event completes database transactions, the publishing feature must have a user account specified for it. This user is called the batch user and you use the `xcelerate.batchuser` and `xcelerate.batchpass` properties in the `futuretense_xcel.ini` file to identify the batch user account for your Sites system.

Note: Publishing also requires a mirror user, different from the batch user. For more information, see [Section 16.2.1, "Users and Mirror to Server Publishing"](#).

12.6 What Happens During a Publishing Session?

When the publishing system begins publishing approved assets, WebCenter Sites does the following:

- Creates a publishing session by adding a row to the `PubSession` table and assigns the session a unique ID, called the `PubSession ID`.
- Runs a query to gather all the assets that are approved and are ready to be published to the destination that the session is publishing to.
- Locks the assets that are returned by the query, and also notifies the `CacheManager` about these assets, so that they cannot be edited while the publishing session is underway.
- Invokes the appropriate element for the publishing method, passing it the list of assets that should be published.

Then, those assets are either rendered into files or mirrored to a destination database, depending on the publishing method in use. For information about what the Export to Disk, Mirror to Server, and Export Assets to XML publishing methods do with the list of approved assets, see the following chapters:

- [Chapter 15, "The Export to Disk Publishing Process"](#)
- [Chapter 16, "The Mirror to Server Publishing Process"](#)
- [Chapter 17, "The Export Assets to XML Publishing Process"](#)

When the publishing session concludes, the publishing system notifies the approval system of which assets were published and then it updates the page cache.

12.7 Obtaining Information About a Publishing Session

WebCenter Sites writes information about each publishing session to log files and to several tables in the WebCenter Sites database. Much of this information is displayed in the Publishing Console as the publishing history for a session. To display the Publishing Console, click the Publish button located in the top button bar in the main window. See also [Section 18.16, "Troubleshooting."](#)

Note: You must have the `xcelpublish` ACL to view the **Publish Console**.

The Approval System

This chapter explains the basic workings of WebCenter Sites' approval system. It does not contain procedures on performing specific operations. Rather, its purpose is to help the administrator gain a basic understanding of the approval system's process, which is both complex and highly interactive, especially in the case of Export to Disk publishing.

Each section of this chapter is written to be as self-contained as possible, with many examples to illustrate the concepts. For continuity, users can read the sections sequentially, and for reference, turn to the section of interest. As this chapter progresses, it builds on the introductory material (especially approval-system terms and definitions), and ends with a comprehensive picture of how the approval system performs its function.

Note: Readers who would like a summary of the approval process can skip to the section [Section 13.13, "Putting It All Together."](#)

This chapter contains the following sections:

- [Section 13.1, "Overview"](#)
- [Section 13.2, "A Brief Look at Dependency Analysis"](#)
- [Section 13.3, "Terms and Definitions"](#)
- [Section 13.4, "Rules for Dependency Analysis"](#)
- [Section 13.5, "Approval Dependencies and Parent-Child Relationships"](#)^x
- [Section 13.6, "Export to Disk Publishing"](#)
- [Section 13.7, "Mirror to Server Publishing and Export to XML"](#)
- [Section 13.8, "Ensuring Version-Matched Content"](#)
- [Section 13.9, "Exists-Exact Dependencies in Mirror to Server and Export to XML Publishing"](#)
- [Section 13.10, "Exists-Exact Dependencies in Export to Disk Publishing"](#)
- [Section 13.11, "Exists/Exact Dependency Rules"](#)
- [Section 13.12, "Evaluating Published Content"](#)
- [Section 13.13, "Putting It All Together"](#)
- [Section 13.14, "Reference: Approval States"](#)

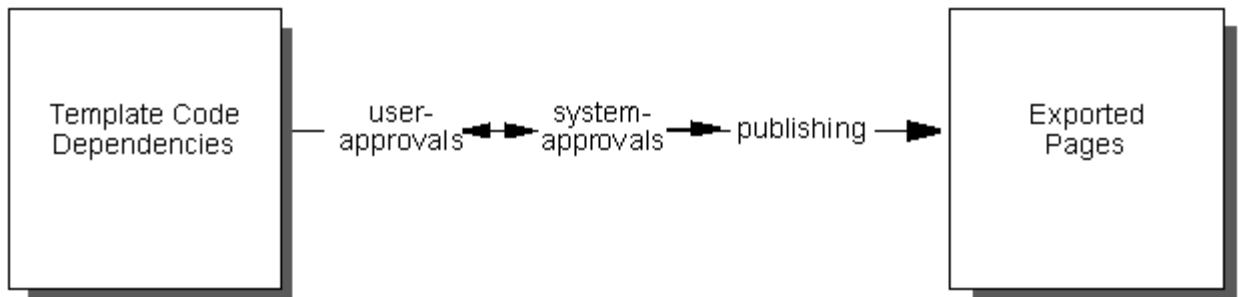
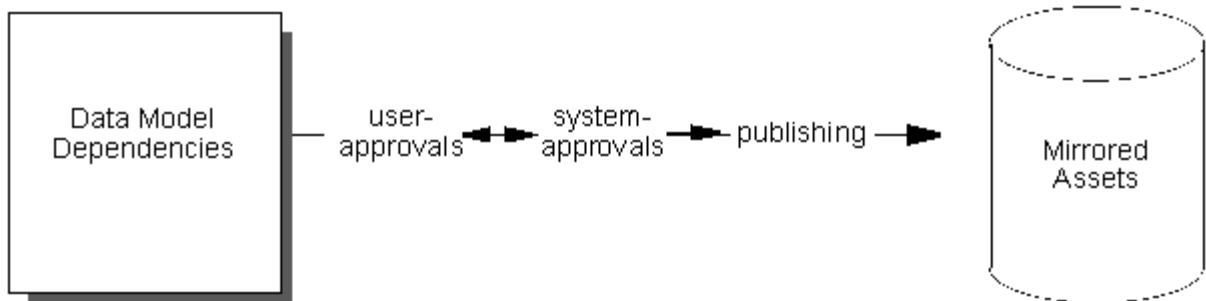
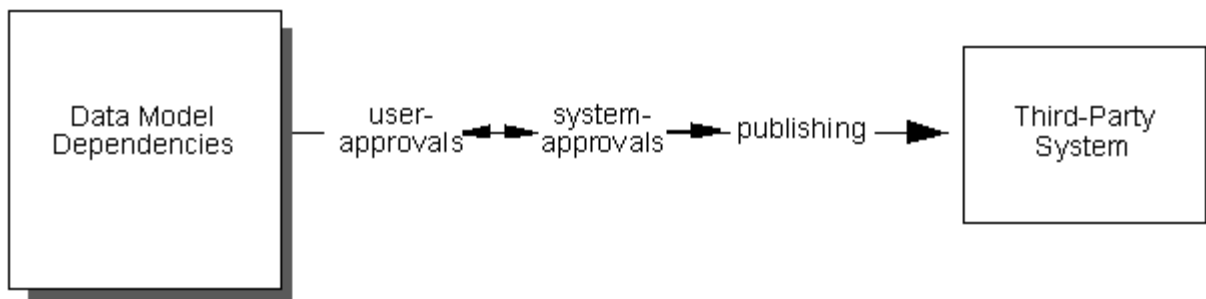
13.1 Overview

Before a publishing session can begin, the user is required to approve assets for the publishing destination. This task is far more intricate than it appears to be. Enterprise sites host enormous numbers of assets, making it impossible for users to track which assets are bound to each other by dependencies and therefore must be approved together in order to be published. As a result, users typically approve only some of the assets that require approval. The unapproved assets compromise the integrity of the approved assets by causing broken links among them. This is where the approval system takes over.

The ultimate purpose of the approval system is to protect the integrity of published content. To this end, the approval system does not automatically allow the publication of user-approved assets. Instead, it evaluates the condition of the user-approved assets in order to determine whether they are indeed ready for publication. If necessary, the approval system then assists the user in approving the supporting assets.

During its operation, the approval system ensures that asset dependencies, encoded in templates or the data model, are kept intact among approved assets, so the dependencies can be reproduced on the publishing destination when the assets are published. If an approved set of assets is free of broken links, it is likely to be free of broken links when published. If an approved set of assets upholds version matching, it is likely to uphold version-matching when published.

Figure 13–1 Approval System in Publishing Methods

Export to Disk Publishing**Mirror to Server Publishing****Export to XML**

13.2 A Brief Look at Dependency Analysis

When analyzing approved assets for broken links and inconsistent versions the approval system performs a dependency test:

1. The approval system determines whether dependencies among *user-approved* assets match dependencies that are specified in either the template code or the data model, taking into consideration the publishing method. As shown in [Figure 13–2](#) and [Figure 13–3](#):
 - In Export to Disk publishing, the approval system tests dependencies (among approved assets) against template code. The dependencies are determined as the code in the template is evaluated. If a default template has been assigned to assets of a given type, the approval system uses that template to determine the dependencies. If there is no default template, the approval system uses the template that is specified for the given asset. For detailed information about

approvals and Export to Disk publishing, see [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

- In Mirror to Server publishing and Export to XML, the approval system tests dependencies (among approved assets) against the data model.
2. If the approval system finds that dependencies among user-approved assets are satisfied, it allows the assets to be published and releases them to the publishing system for inclusion in the next publishing session.

However, if it finds the dependencies to be breached, the approval system looks for published assets that will satisfy those dependencies at publication time and therefore turns to published content. The approval system scans the `PublishedAssets` table to determine what has been published to the target destination:

- If an asset requiring approval does not exist on the target destination (and in the correct version, if applicable), the approval system prompts the user to approve the asset.
- If the asset requiring approval already exists on the target destination (and in the correct version, if applicable), the asset will satisfy the dependency at publication time and therefore does not require approval (and re-publication).

Note: Version matching can be imposed by the WebCenter Sites administrator, in which case user-approved assets must match the versions of published assets in order to satisfy the dependency criteria (which are called `exists` and `exact`).

For Mirror to Server publishing, version matching is specified in the `mwb.conservativedependencies` property, in `gator.ini`. For Export to Disk publishing it is specified in the template code. For more information see [Section 13.8, "Ensuring Version-Matched Content."](#)

Figure 13–2 Export to Disk Publishing Methods and Approvals

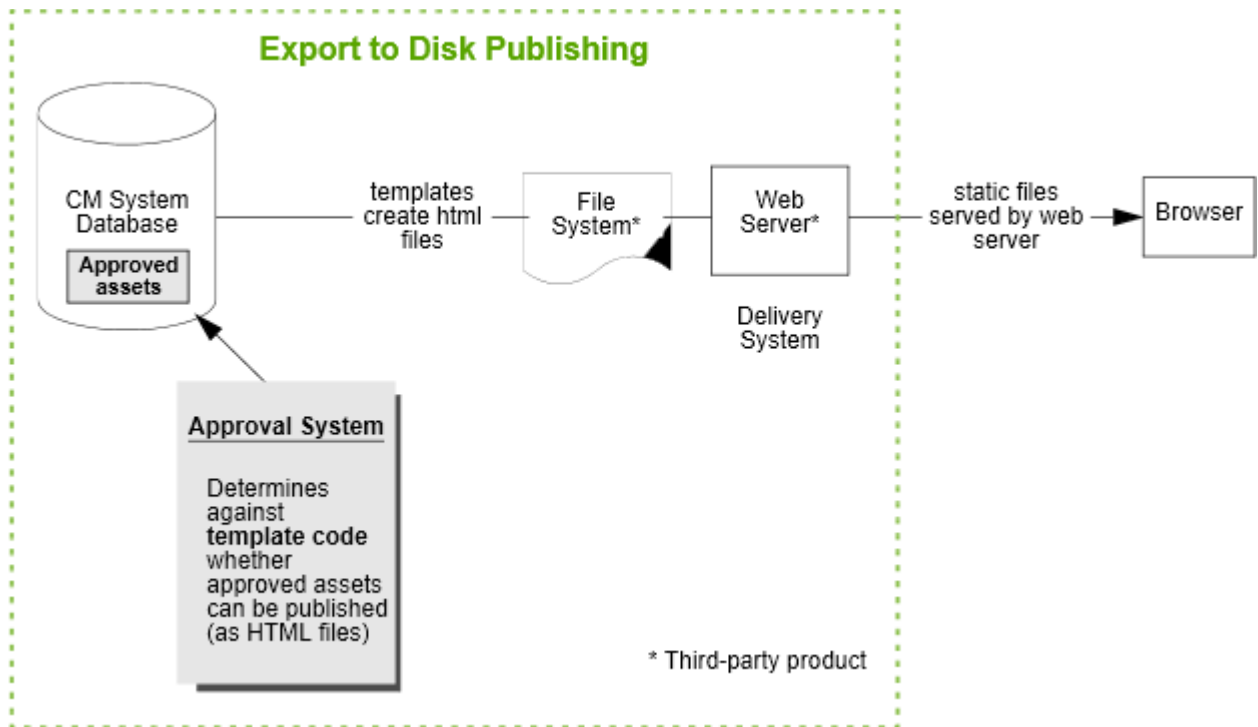
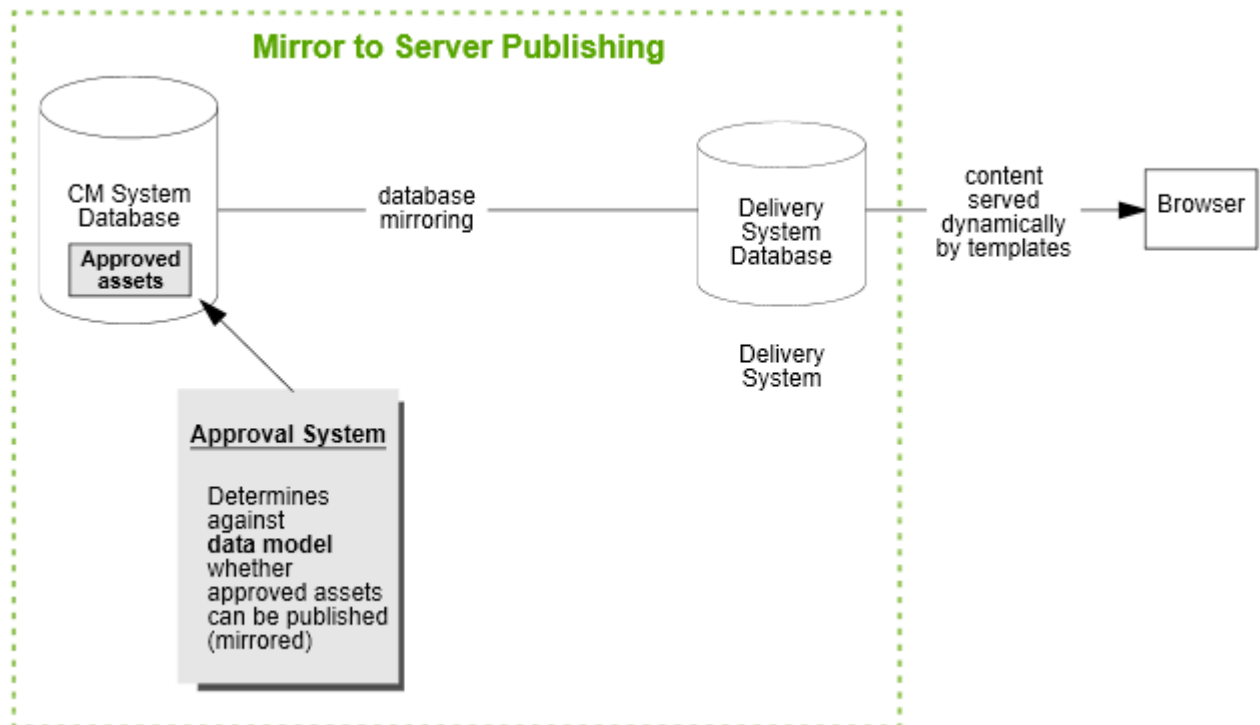


Figure 13–3 Mirror to Server Publishing Methods and Approvals



3. On completing its analysis of all the user-approved assets, the approval system either approves or disallows the publishing of certain assets:

The approval system supports fractional publishing rather than "all or none" publishing. Assets that pass the dependency test are released to the publishing system for inclusion in the next publishing session. Assets that fail the dependency test are held back from the publishing system, and the user is informed as to which additional assets also require approval. Dependencies among approved assets are hence the cause of lighter-than-expected publishing sessions.

Dependencies among approved assets have a specific treatment and a name ("approval dependencies"), both of which are critical to the understanding of how the approval system works. Approval dependencies are the main subject of the rest of this chapter. They are described and explained in the section [Section 13.3, "Terms and Definitions,"](#) along with the equally important terms, "approvals" and "approval states." A summary of the approval process can be found in the section [Section 13.13, "Putting It All Together."](#)

13.3 Terms and Definitions

Three especially important terms in the WebCenter Sites publishing model are "approval," "approval dependencies," and "approval states." These terms deal with dependencies that are critical to the understanding of how the approval system works. They are also used throughout this guide, as well as the WebCenter Sites publishing interface. We recommend that you familiarize yourself with the terms, their definitions, and the supporting examples.

This section contains the following topics:

- [Section 13.3.1, "Approval: Intent to Publish vs. Permission to Publish"](#)
- [Section 13.3.2, "Approval Dependencies"](#)
- [Section 13.3.3, "Approval States"](#)

13.3.1 Approval: Intent to Publish vs. Permission to Publish

For an asset to be published, it requires approval first from the user, then from the approval system. To distinguish one type of approval from the other, we use the terms user-approval and system-approval.

- The difference between user-approval of assets and system-approval of assets is the following:

Whereas the user's approval of an asset signals the user's intent to publish the asset, the system's approval of the asset grants the publishing system permission to publish the asset.

The user's approval of an asset is equivalent to submitting the asset to the approval system for a dependency test. The system's approval of a user-approved asset is equivalent to (1) confirming that the user-approved assets pass the dependency test, and (2) releasing the asset to the publishing system for publication in the next publishing session.

- The approval system never approves assets before the user has had a chance to do so:

When we say that an asset is system-approved, it is understood that the user has already approved the asset.

- When a user approves an asset, it is always to a given publishing destination.

When we say that an asset is user-approved, the publishing destination is understood (and explicitly identified, if necessary).

- System-approvals are conditional:

When approving assets, the user can assign the Approved state. The approval system responds by treating the approval as tentative and requiring verification. Only when the approval system completes its dependency test and confirms that the Approved state is valid does the system approve (release) the assets for publication.

If, however, the approval system cannot confirm the Approved state, it rejects the state, assigns its own approval state to the assets, informs the user as to which supporting assets also require approval, and holds the current user-approved assets from publication until the user can approve the supporting assets. The system's approval of a user-approved asset is thus conditional on the user also approving the supporting assets. Conditional approvals are caused by approval dependencies, which are explained in the next section.

- The approval system can unapprove assets:

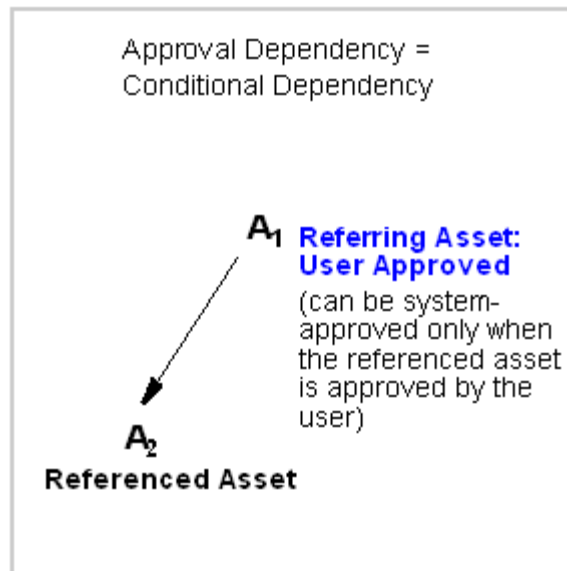
The approval system supports an approval queue to handle asset modification events as a way of keeping approval tables up to date. For example, if an asset is approved for publication, but then changed, the approval queue handles this change by unapproving the asset—that is, rejecting it from the publishing queue.

The approval queue is not accessible to the user. It runs, by default, every 5 minutes. However, when you invoke a feature that uses approval functionality (such as the **Publishing** tab), the approval queue is forced to run before the screen is rendered in order to keep approval information up to date.

13.3.2 Approval Dependencies

Note: Before reading this section, make sure you understand the distinction between the terms user-approval and system-approval. For more information, refer to the section [Section 13.3.1, "Approval: Intent to Publish vs. Permission to Publish."](#)

An approval dependency is a conditional dependency and results in a conditional approval:

Figure 13–4 Conditional Approval

Condition 1. An approval dependency is created if the user approves an asset that refers to another asset for content.

Condition 2. When an approval dependency is created, the system's approval of the referring asset is conditional on the approval of the referenced asset. In other words, the approval system can approve a referring asset only if the referenced asset is also approved. The referring asset is said to have an approval dependency on the referenced asset.

Note: From this point forward, we define "referring asset" to be the parent asset. The "referenced asset" is the child asset. For simplicity, we also assume that parent assets are solely parents, and child assets are solely child assets.

An important implication of approval dependencies is the following:

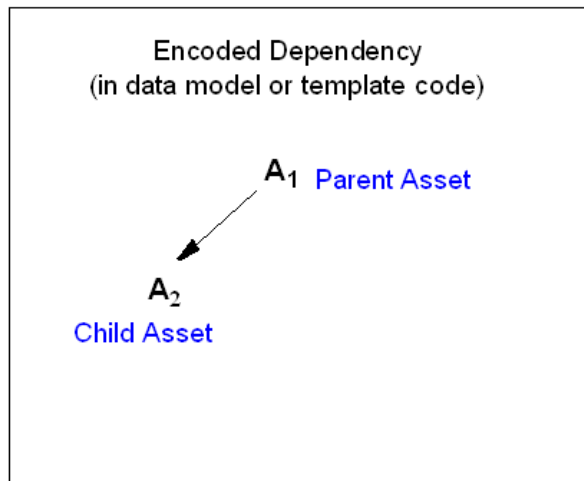
While approval dependencies originate in one of two sources (the data model or template code), the number of approval dependencies is not necessarily equal to the number of dependencies in the source. The existence of a dependency in the source can lead to an unsatisfied approval dependency, a satisfied approval dependency, or no approval dependency at approval time. Which outcome is observed depends on two factors:

- Which asset(s) the user has approved.
- How the approval system treats approval dependencies. The approval system does not simply determine that a dependency exists for a given pair of assets. It explicitly accounts for the direction of the dependency by testing whether the approved asset is a parent or a child. If it determines that a parent asset has been approved, it logs an approval dependency. If it determines that a child asset has been approved, it does not log an approval dependency.

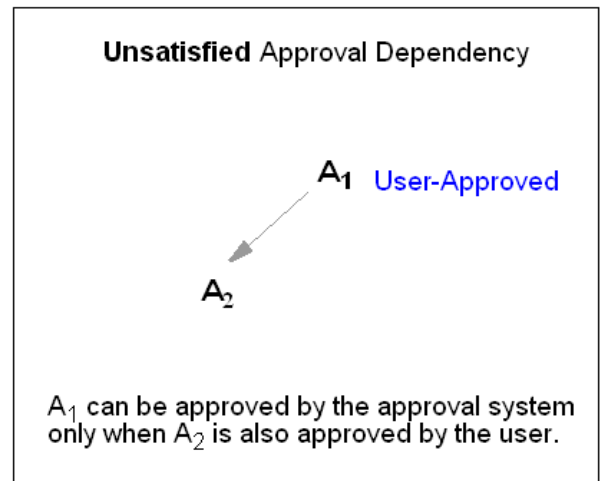
For detailed examples of how approval dependencies are determined, see [Section 13.3.2.1, "Examples of Approval Dependencies."](#)

Figure 13–5 Encoded Dependencies and Approval Dependencies

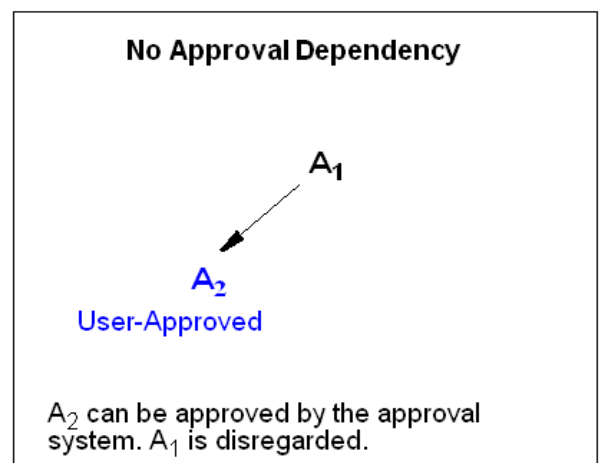
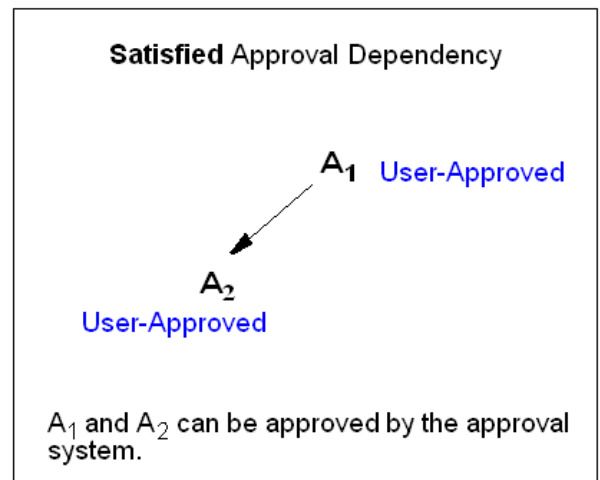
Encoded Dependency



Possible Approval Dependencies



For detailed examples of how approval dependencies are determined, see [“Examples of Approval Dependencies,”](#) on page 229.



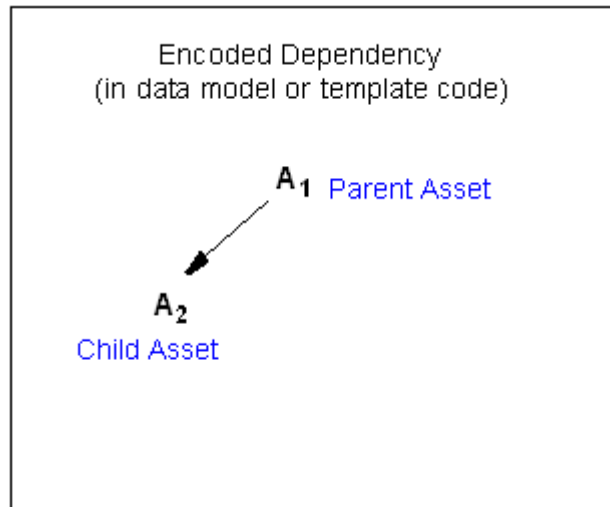
This section contains the following topics:

- [Section 13.3.2.1, "Examples of Approval Dependencies"](#)
- [Section 13.3.2.2, "Summary"](#)

13.3.2.1 Examples of Approval Dependencies

This example supplements the previous section by illustrating in greater depth how approval dependencies are determined.

Figure 13–6 Encoded Dependencies

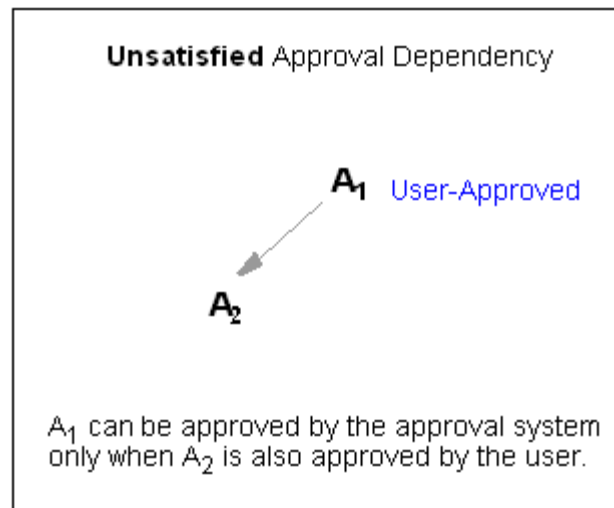


In this example, we have a two-asset data model. Asset A1 refers to asset A2 for content. While the data model specifies one dependency, the user has three different ways to approve the assets and, therefore, create different approval dependencies:

- ["Unsatisfied approval dependency"](#) on page 13-10
- ["Satisfied approval dependency"](#) on page 13-11
- ["No approval dependency"](#) on page 13-12

Unsatisfied approval dependency

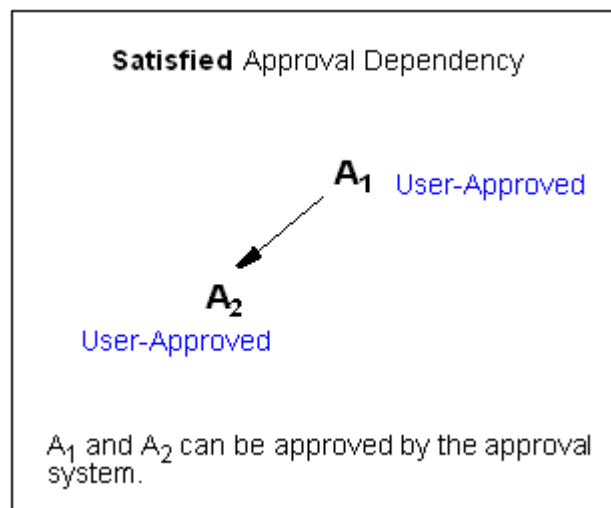
In this instance, the user approves the parent asset (A1), but not its child (A2). The approval system crawls the data model (using the approved asset as a seed), discovers the A1 to A2 dependency, determines the dependency to be unsatisfied (by the user's approval of A1 alone), classifies the dependency to be an unsatisfied approval dependency, rejects the Approved state for A1, disallows the publication of A1 until A2 is also approved, and informs the user that A2 needs approval.

Figure 13–7 Unsatisfied Dependencies

In this instance, the system's approval of A1 is conditional on the user approving A2. A1 is said to have an approval dependency on A2. (Because the approval dependency is unsatisfied, the approval system notifies the user.)

Satisfied approval dependency

In this instance, the user approves both the parent asset and its child. The approval system finds the data model dependency to be satisfied (by the user's approval of both A1 and A2), classifies the dependency to be a satisfied approval dependency, confirms the Approved states, and allows publication of both A1 and A2 when the next publishing session begins.

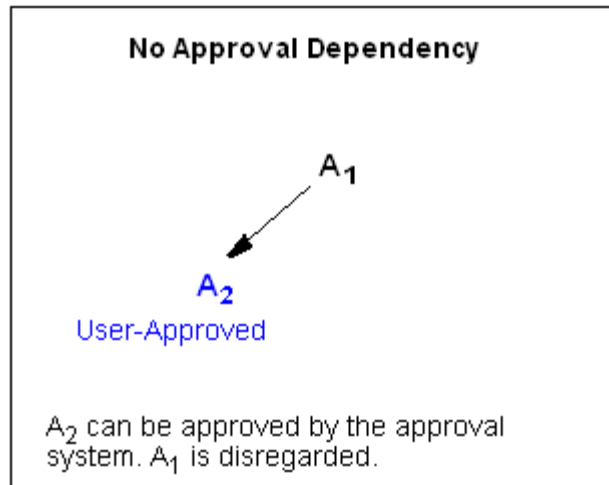
Figure 13–8 Approval Dependencies

As in the previous instance, the system's approval of A1 is conditional on the user approving A2. This time, however, the approval dependency is satisfied. No notice is returned to the user. The asset will be published in the next publishing session.

No approval dependency

In this instance, the user approves the child asset (A2), but not its parent (A1). Here, the approval system does not recognize an approval dependency (because A2 has no dependency on A1). The approval system confirms the user's Approved state. As a result, A2 will be published (but A1 will not).

Figure 13–9 No Approval Dependencies

**13.3.2.2 Summary**

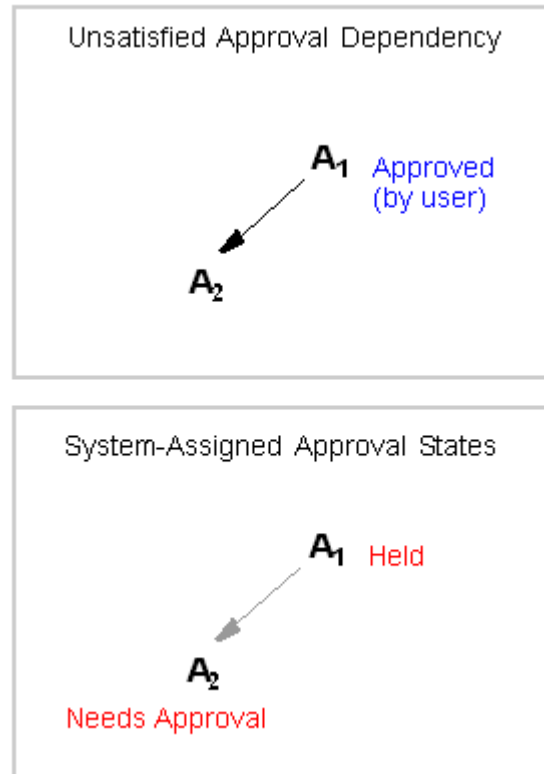
- Whereas data models and template code distinguish between the types of dependencies they specify (for example, associations and recommendation associations), the approval system simplifies these dependencies by treating them as parent-child approval dependencies. The parent is the referring asset and depends on its child for content.
More information about this rule (and other rules regarding dependency analysis) is given later in this chapter. See [Section 13.4, "Rules for Dependency Analysis."](#)
- Whereas data model dependencies and template dependencies are created by the developer to specify how and which assets refer to each other for content, approval dependencies are created by users who approve assets. Approval dependencies are analyzed by the approval system and used to flag which user-approved assets can be released for publication and which must be held back from publication; which assets have yet to be approved by the user, which assets have been published, and so on.
- Data model and template code dependencies are fixed for the given data model and template. Approval dependencies can vary, even for a fixed set of assets and a given publishing destination. This is so because the combination of assets approved by the varies from one approval attempt to another.

13.3.3 Approval States

When the approval system detects an approval dependency, it must either confirm the user's Approved label for each of the assets, or flag the assets with approval states of its own. An approval state is a label assigned to user-approved assets by the approval system in order to identify their approval status: why an approved asset cannot be published, whether the asset has been approved and published, whether an asset is eligible for publishing (that is, approved but not yet published because the publishing

session has not yet been started), and so on. A complete listing of approval states is given in [Table 13-4, "Approval States Assigned by the Approval System"](#).

Figure 13-10 Approved and Unapproved Approval States

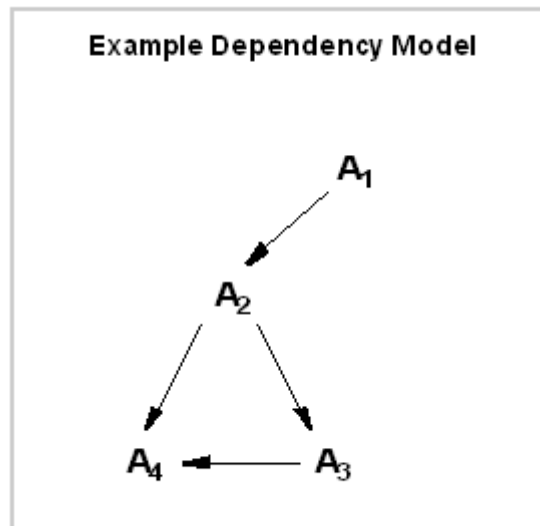


For example, the user marked asset A1 as Approved. The system, however, treats the approval as tentative. Following a dependency analysis, the system determines that A1—even though approved—has an unsatisfied approval dependency on A2 and must be held from publication (until A2 is also approved).

The system overrides the user's approval of A1 by assigning A1 the Held approval state. It assigns A2 the Needs Approval state.

Approval State Scenarios

Even for a fixed set of assets and a given publishing destination, the list of approval states can (and typically does) vary from one approval attempt to another. This is so, because the combination of assets approved by the user tends to vary from one approval attempt to another. To illustrate, we use a simple dependency model and three approval scenarios, all involving one and the same user:

Figure 13–11 Simple Dependency Model

Each arrow represents an approval dependency. For all four assets to be published, four approval dependencies must be satisfied at publication time:

A1 to A2

A2 to A3

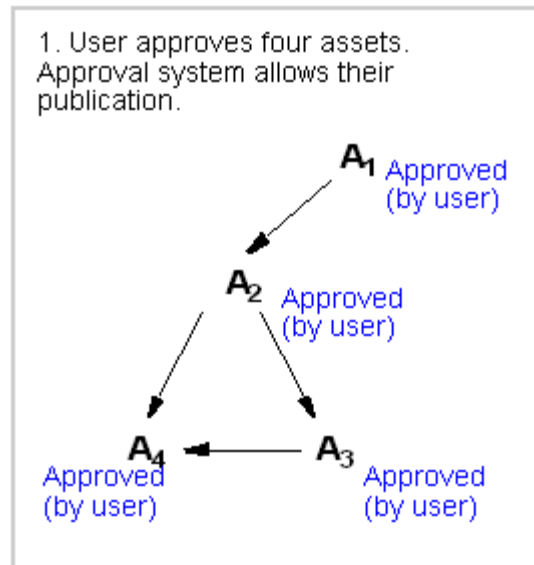
A2 to A4

A3 to A4

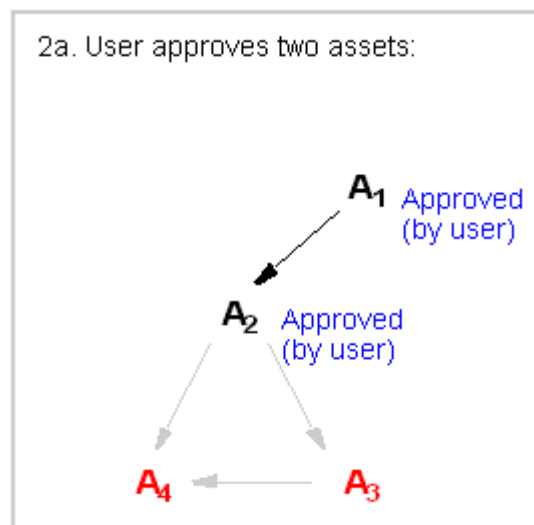
(None of the assets have been previously published.)

Now, consider three scenarios in which the user approves different combinations of assets:

1. In the first (and ideal) attempt, the user approves all four assets, which satisfies all four approval dependencies at once. In this scenario, the system logs four approval dependencies, determines they are satisfied, confirms the assets as approved, and allows their publication when the next publishing session begins.

Figure 13–12 Approval System - All Approved

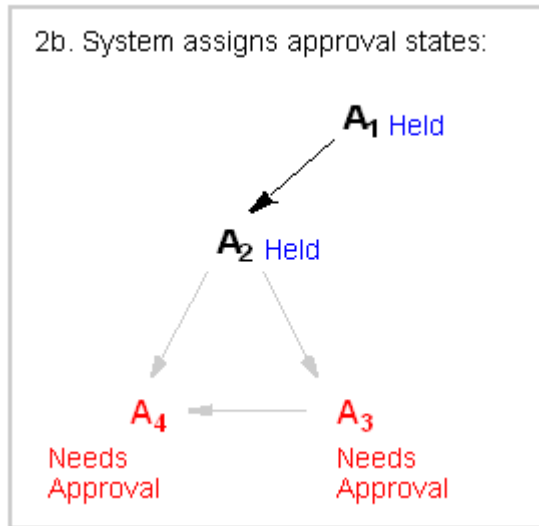
2. In the second scenario, the user approves two assets—A1 and A2—which satisfies one of the four approval dependencies. Three approval dependencies remain to be satisfied:
- A2 to A3
 - A2 to A4
 - A3 to A4

Figure 13–13 Dependency Model

3. The approval system logs three unsatisfied approval dependencies, determines that none of the assets are ready for publication, assigns its own approval states to all the assets, and returns to the user the following list:
- A3 and A4, in the Needs Approval state

- A1 and A2, in the Held state; these assets cannot be published until A3 and A4 are approved

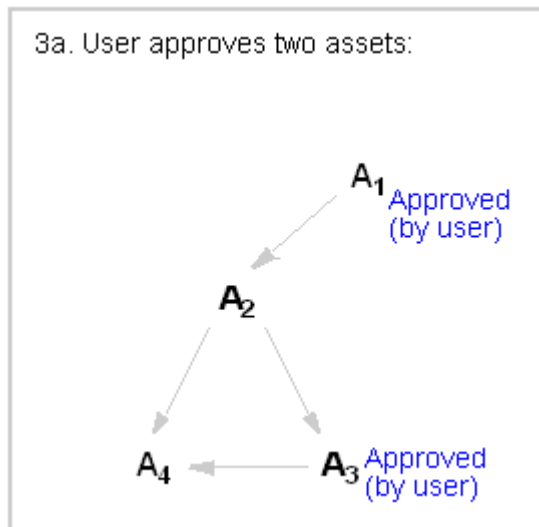
Figure 13–14 Assets in Held State Waiting for Approval



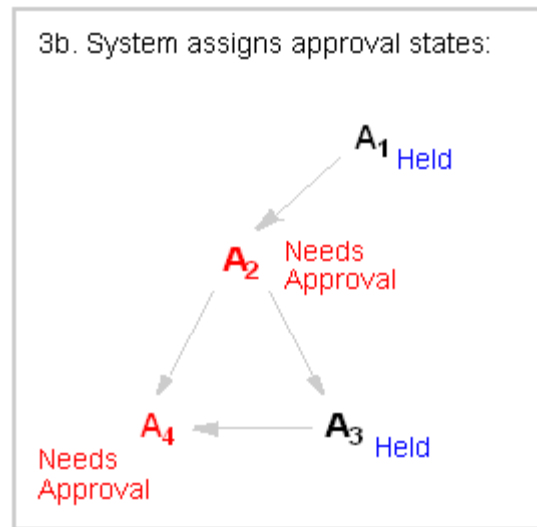
4. In the third scenario, the user also approves two assets—A1 and A3—but this time, satisfies none of the four approval dependencies:

- A1 to A2
- A2 to A3
- A2 to A4
- A3 to A4

Figure 13–15 Partial Approval State



5. The approval system logs four unsatisfied approval dependencies, and determines that none of the assets are ready for publication. The approval system returns a list of assets and approval states different from the list in scenario 2.

Figure 13-16 Held Dependencies, Waiting for Approval

The list identifies:

- A2 and A4 in the Needs Approval state
- A1 and A3 in the Held state

Note: If the user had approved a different set of assets, the system would have determined different approval dependencies, and therefore different approval states for the participating assets:

- If the user had approved A2, the approval system would have determined only three approval dependencies (A2 to A3, A2 to A4, and A3 to A4), even though the data model in our example specifies four dependencies.
- If A3 were approved, one approval dependency would have been determined (A3 to A4).
- If A4 were approved, no approval dependencies would have been determined.

For more information as to how approval dependencies are determined, see [Section 13.3.2, "Approval Dependencies"](#).

13.4 Rules for Dependency Analysis

To perform its function, the approval system follows several rules, summarized below and covered in more detail in the sections that follow:

- The approval system simplifies all approval dependencies by treating them as parent-child relationships. For more information, see [Section 13.5, "Approval Dependencies and Parent-Child Relationships."](#)
- In Export to Disk publishing, the approval system tests approval dependencies against template code. For more information, see [Section 13.6, "Export to Disk Publishing."](#)
- In Mirror to Server publishing and Export to XML, the approval system tests approval dependencies against the data model. For more information, see

[Section 13.7, "Mirror to Server Publishing and Export to XML."](#)

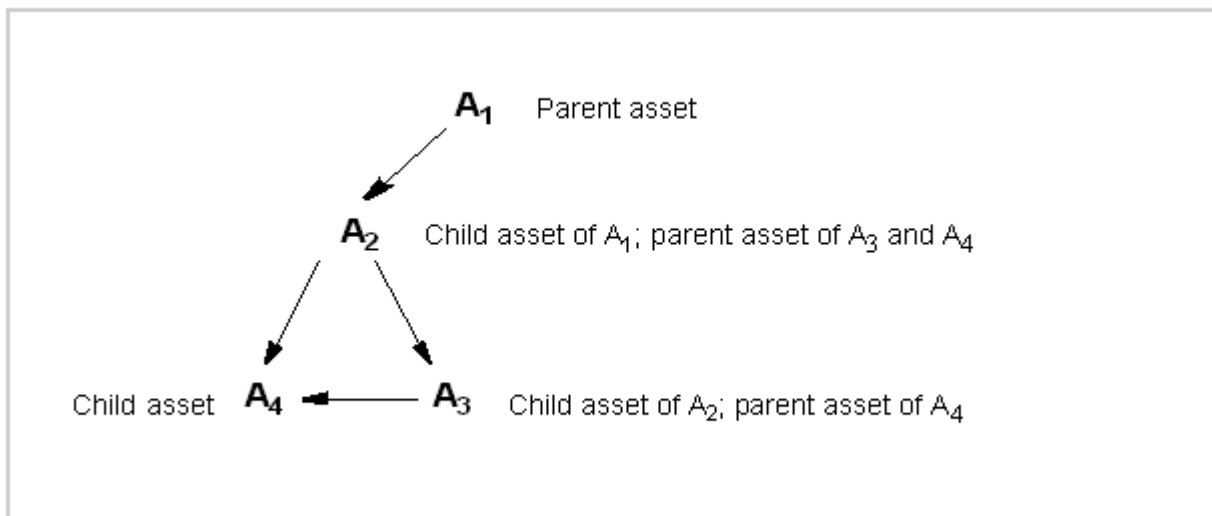
- If version matching needs to be controlled, the administrator can set an `exists`, `exact`, or `none` dependency to impose or cancel version matching. For more information, see [Section 13.8, "Ensuring Version-Matched Content."](#)
- If the approval system determines the existence of an unsatisfied approval dependency, it searches for published content to avoid redundant approvals. For more information, see [Section 13.12, "Evaluating Published Content."](#)

13.5 Approval Dependencies and Parent-Child Relationships

The approval system simplifies the many types of dependencies that are specified in data models and template code (for example, associations and recommendation associations). Regardless of their origin and type, the approval system classifies all approval dependencies as parent-child relationships and adheres to the following rules.

- An asset that refers to another asset is always the parent asset; the referenced asset is the child asset. [Figure 13–17, "Parent-Child Relationships"](#) shows an example of parent-child relationships, in which one asset is solely a parent asset, another is solely a child asset, and the rest are a mixture.

Figure 13–17 Parent-Child Relationships

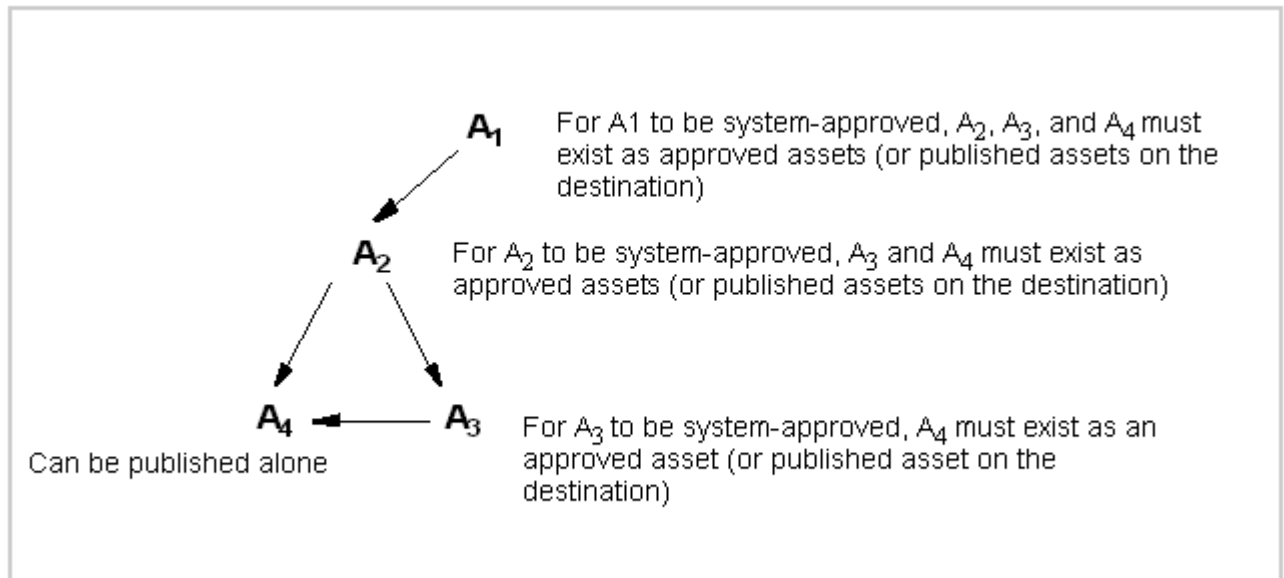


- A parent asset is dependent on the child asset for content.
- The approval system approves parent assets only if the child assets are either user-approved or they already exist on the publishing destination.
 - If the child assets are also parents, then their child assets must be approved (or must exist on the destination), and so on.
 - If the publishing method is Mirror to Server or Export Assets to XML, version matching can be a requirement (at the WebCenter Sites administrator's discretion), in which case the approval system checks assets for version. For more information about version matching, see [Section 13.11, "Exists/Exact Dependency Rules"](#).
- The approval system approves child assets for publication independently of their parent assets. On the live site, the link from parent to child will not appear to be

broken; it simply will not appear. (In this respect, parent data can be inadvertently dropped from the live site.)

Figure 13–18, "Approval Dependencies" displays a hierarchical dependency among four assets to illustrate which assets must be approved concurrently (as a result of approval dependencies), and which can be approved alone.

Figure 13–18 Approval Dependencies



13.6 Export to Disk Publishing

For Export to Disk publishing, the approval system tests approval dependencies against template code.

Note: Approval templates represent the developer's approval strategy for Export to Disk publishing. Because approval for Export to Disk publishing is complex, it is given a separate treatment in [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

This section contains the following topics:

- [Section 13.6.1, "Evaluating Approval Dependencies"](#)
- [Section 13.6.2, "Test Rendering"](#)

13.6.1 Evaluating Approval Dependencies

For both flex and basic assets that are published using Export to Disk, there are two types of approval dependencies: template and reference.

Template dependencies are strictly independent of the data model, regardless of how (or whether) the assets are associated in the data model. For example, your data model can specify an article-image association. In Mirror to Server publishing, the article depends on the image for content, and cannot be published without the image. In

Export to Disk publishing, however, the template chosen to render the asset defines its own dependencies. Some possibilities are:

- The approval template code re-creates the article-image association as a dependency, thereby creating on the published page a relationship that already exists in the data model.
- The approval template code creates dependencies on yet other assets (such as audio files), thereby creating relationships that do not exist in the data model, but will exist on the published page.
- The approval template code does not create any dependency between the two assets, thereby creating no relationship at all on the published page. The assets are treated as stand-alone content and published independently of each other, even though the data model specifies an association.

Tags that generate template dependencies are:

- `<asset:load>`
- `<asset:loadall>`
- `<assetset:setasset>`
- `<assetset:setlistedassets>`
- `<render:logdep>`

Reference dependencies are generated when a link is created from one page to another. They are registered as reference dependencies between the primary assets of the two pages. For example, if we create a hyperlink from the approval template of asset A to a page where asset B is the primary asset, the approval system will register this as asset A's reference dependency on B. Tags that generate this kind of dependency are:

- `<render:getpageurl>`
- `<render:gettemplateurl>`
- `<render:gettemplateurlparameters>`

Approval dependencies for Export to Disk publishing are a complex topic. Additional information about approval dependencies can be found [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

13.6.2 Test Rendering

When an asset is user-approved to a Export to Disk publishing destination, the approval system determines approval dependencies for that asset by evaluating the code in the template that is assigned to the asset. The approval system also performs a test-render in which it evaluates the template code for compositional dependencies, which are manifested when the content is published.

Compositional dependencies are the dependencies of a generated page on the assets that were used to generate that page. They are determined by the logic in that page's template. The same tags that create template and approval dependencies at approval time also create compositional dependencies at publication time.

If a default template has been assigned to assets of that type, the approval system uses it to determine the dependencies. If there is no default template, the approval system uses the template that is specified for the given asset.

Your developers create default templates for asset types for the following reason: when the Export to Disk publishing method actually publishes an asset, it does not necessarily use the template that is assigned to the asset—the code in another element

could determine that a different template renders that asset in certain cases. If this is the case for your online site, it is likely that the developers who created the templates also designed default templates for the approval system to use when determining approval dependencies.

You or your site designers can set default approval templates for each asset type and for each publishing destination. See [Section 18.11, "Assigning Approval or Preview Templates"](#).

13.7 Mirror to Server Publishing and Export to XML

For Mirror to Server and Export to XML publishing methods, the approval system tests approval dependencies against the data model.

Note: **Mirror to Server** publishing and **Export to XML** allow for the possibility of unsatisfied **compositional dependencies** on the live site, because templates that render the page are not evaluated during Mirror to Server and Export to XML publishing.

Compositional dependencies appear when a dynamic page is assembled. They involve a set of assets that are used to generate the page. For more information about compositional dependencies, see [Section 13.6.2, "Test Rendering"](#) and [Chapter 14, "Various Topics in Export to Disk Publishing and the Approval Process."](#)

13.8 Ensuring Version-Matched Content

Any publishing session (such as the one in [Figure 13–18, "Approval Dependencies"](#)) can be made more stringent by the administrator requiring parent assets to match child assets in terms of version, thereby ensuring self-consistent content. Dependency on version is not encoded in either the data model or the template code:

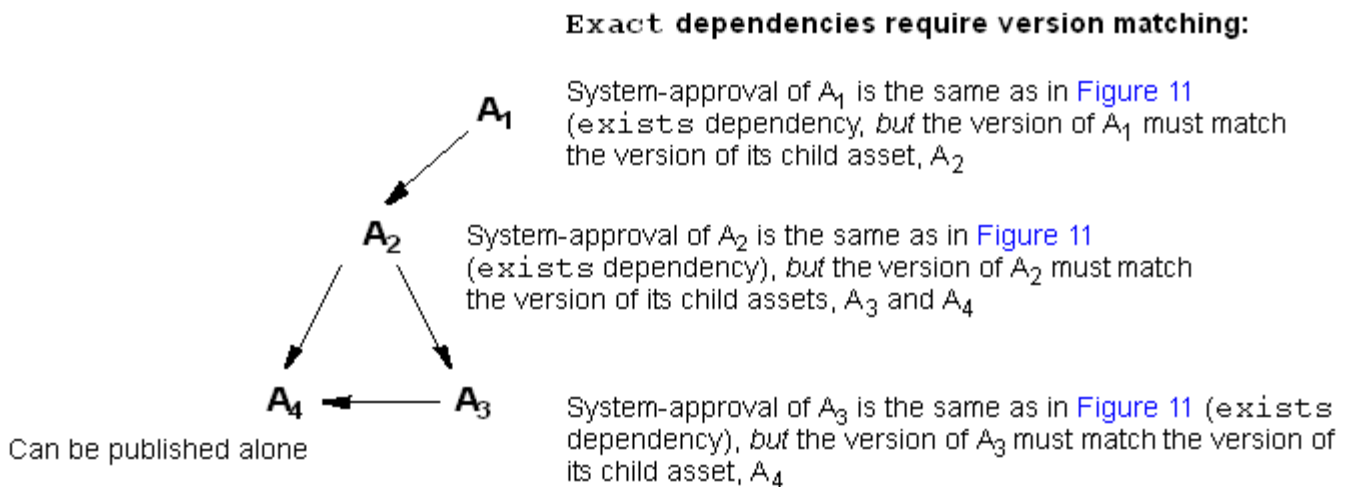
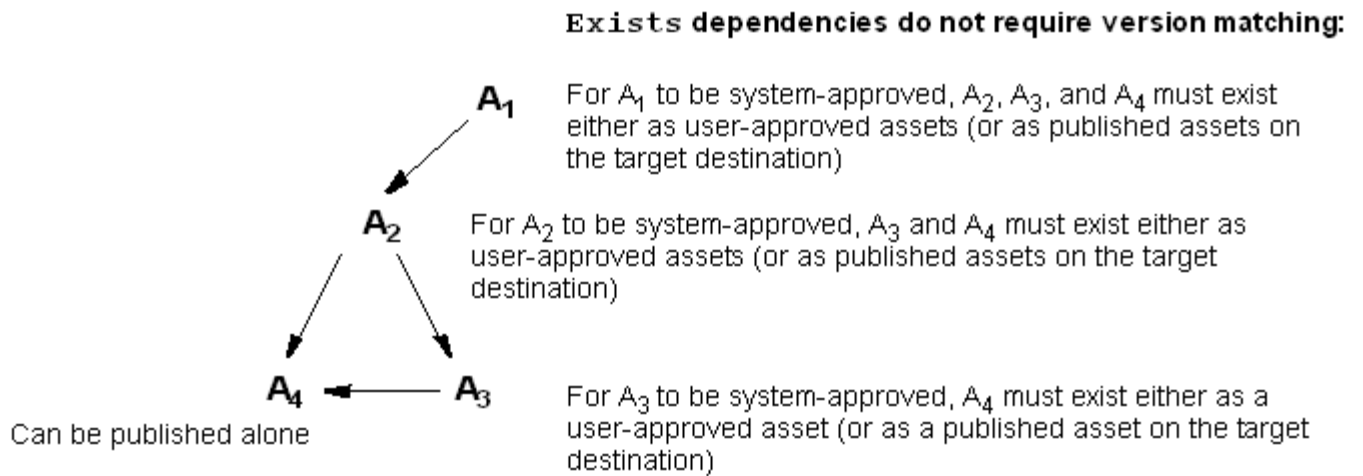
- For Mirror to Server and Export to XML publishing, it is specified in one of WebCenter Sites' properties—`mwb.conservativedependencies` (in the `gator.ini` file)—by setting the value of the property to `exists` or `exact`.
- For Export to Disk publishing, it is specified by setting the `deptype` attribute in the relevant tags to `exists`, `exact`, or `none` (the tags are listed in [Section 13.6.1, "Evaluating Approval Dependencies"](#)).

Dependency on version is qualified by the terms `exists`, `exact`, or `none`.

- An **exists dependency** does not require version matching. For a parent to be published, its child assets must simply exist either as approved assets or as published assets on the publishing destination. The version of the parent asset *need not* match any versions of its child assets.
- An **exact dependency** requires version matching. This dependency is identical to the `exists` dependency, except that the version of the parent must match the versions of its child assets (which means that all assets in the dependency must match each other's version).
- A **none dependency** causes the tag in which it is used to specify no approval dependency, at all.

[Figure 13–19](#) summarizes and illustrates `exists` and `exact` dependencies.

Figure 13–19 *Exists/Exact Dependencies*



13.9 Exists-Exact Dependencies in Mirror to Server and Export to XML Publishing

For Mirror to Server and Export to XML publishing of flex assets, whether the exists or exact dependency is in effect depends on the value of the property `mwb.conservativedependencies`, in the `gator.ini` property file.

- The default value (`false`) sets exists dependencies.
- A value of `true` sets exact dependencies.

Note: When `mwb.conservativedependencies` is set to false (exists) and an attribute is in use, the following fields of the attribute are locked and cannot be changed: Value Type, Storage Style, External ID, External Table, and External Column.

If you change the value of `mwb.conservativedependencies`, you must re-approve the assets that were affected by the change.

13.10 Exists-Exact Dependencies in Export to Disk Publishing

For Export to Disk publishing, the developer sets exists, exact, or none dependencies using the `deptype` attribute in applicable tags, such as `getpageurl` and `assetload`. (For a listing of the tags, see [Section 13.6.2, "Test Rendering"](#)).

Template dependencies are by default exact dependencies—if you wish to approve asset A you must approve B if B has been changed. Reference dependencies are always exists dependencies—if you approved and published B once, you are not required to approve it again in order to re-publish A.

The exception is when you set `deptype="none"` on any of the tags. As a result, no approval dependency at all is created by the tag. This means no record is created for the dependency during approval. In all other contexts, such as Export to Disk publishing and live sites, the `deptype` attribute is ignored.

13.11 Exists/Exact Dependency Rules

When an asset is approved to a Mirror to Server or Export Assets to XML destination, the approval system determines approval dependencies against the data model.

Note: When reading this section, bear in mind that in the context of an approval dependency, the term "dependent asset" is equivalent to the "parent asset." For rules governing parent-child relationships in approval dependencies, see [Section 13.5, "Approval Dependencies and Parent-Child Relationships."](#)

This section contains the following topics:

- [Section 13.11.1, "Basic Asset Types"](#)
- [Section 13.11.2, "CSElement and SiteEntry Assets"](#)
- [Section 13.11.3, "Flex Families"](#)
- [Section 13.11.4, "Oracle WebCenter Sites: Engage Visitor Data Assets"](#)

13.11.1 Basic Asset Types

For basic assets, the approval system follows these dependency rules:

Table 13–1 *Asset Relationships and Dependencies*

Relationship to Approved Asset	Dependency - Exists	Dependency - Exact
Association	Yes by default, unless configured to be exact	N/A
For page asset only: another page asset at a lower level in site plan	Yes	N/A

Table 13–1 (Cont.) Asset Relationships and Dependencies

Relationship to Approved Asset	Dependency - Exists	Dependency - Exact
Embedded link	Yes	N/A
Embedded pagelet	N/A	Yes

- Rules for associations:
 - Depending on how your asset associations are designed, an approved asset has either an `exists` or `exact` dependency on assets that it references through named asset associations.
- Rules for page assets:
 - An approved page asset has an `exists` dependency on page assets that are lower than itself in the hierarchy of the site plan (which is reflected in the `SitePlanTree` table).
- Rules for embedded links:
 - An approved asset has an `exists` dependency on assets that it references by an embedded link. For information about embedded links, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
- Rules for embedded pagelets:
 - An approved asset has an `exact` dependency on assets that it references as embedded pagelets. For information about embedded pagelets, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

13.11.2 CSElement and SiteEntry Assets

The root element for a `SiteEntry` asset is represented by a `CSElement` asset. The following rule applies:

An approved `SiteEntry` asset has an `exists` dependency on the `CSElement` that it references.

13.11.3 Flex Families

For flex family members, the approval system follows these dependency rules:

Table 13–2 Flex Family Dependency Rules

Approved Asset	Flex Parent Definition	Flex Parent	Flex Definition	Flex Attributes	Flex Filter	Attribute Editor	Template
Flex Parent Definition	<code>exists</code>	N/A	N/A	<code>exists</code>	N/A	N/A	N/A
Flex Parent	<code>exists</code>	<code>exists</code>	N/A	<code>exists</code>	N/A	N/A	N/A
Flex Definition	<code>exists</code>	N/A	N/A	<code>exists</code>	N/A	N/A	N/A
Flex Asset	N/A	<code>exists</code>	<code>exists</code>	<code>exists</code>	N/A	N/A	<code>exists</code>
Flex Attribute	N/A	N/A	N/A	N/A	<code>exists</code> (by default)	<code>exists</code>	N/A

- Rules for flex parent definitions:

An approved flex parent definition has an `exists` dependency on its flex parent definition(s) and flex attributes.

- Rules for flex parents:

An approved flex parent has an `exists` dependency on its flex parent definition, flex parent, and flex attributes.

- Rules for flex definitions:

An approved flex definition has an `exists` dependency on its flex parent definition, flex parent, and flex attributes.

- Rules for flex assets:

An approved flex asset has an `exists` dependency on its flex parents, flex asset definitions, flex attributes, and template.

- Rules for flex attributes:

An approved flex attribute has either an `exists` or an `exact` dependency on its flex filter, depending on how the flex filter is coded. The default flex filter is coded for an `exists` dependency. If the attribute is of type asset, the user can decide whether there is an `exists` or `exact` dependency on that asset.

An approved flex attribute has an `exists` dependency on its attribute editor (if one is assigned).

For information regarding `exists` and `exact` dependencies among flex family members, see [Section 13.11.3, "Flex Families."](#) For more information about the functions of the flex family members, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

13.11.4 Oracle WebCenter Sites: Engage Visitor Data Assets

The approval system uses the following rules to determine approval dependencies for the Engage visitor data asset types:

Table 13-3 Engage Visitor Data Assert Approval Dependencies

Approved Asset	History Definition	History Attribute	Visitor Attribute	Recommendation	Related Flex Asset
History definition	N/A	exact	N/A	N/A	N/A
Segment	exists	N/A	exists	N/A	N/A
Recommendation	N/A	N/A	N/A	N/A	N/A
Promotion	N/A	N/A	N/A	exists	N/A
Flex Asset	N/A	N/A	N/A	N/A	exists

- History definitions have `exact` dependencies on their history attributes.
- Segments have `exists` dependencies on the history definitions and visitor attributes used in them.
- Promotions have `exists` dependencies on the recommendation assets that they override.
- Flex assets have `exists` dependencies on any assets that are designated as related items (for Related Items recommendations)

13.12 Evaluating Published Content

If the approval system determines the existence of an unsatisfied approval dependency, it searches for published content to avoid redundant approvals.

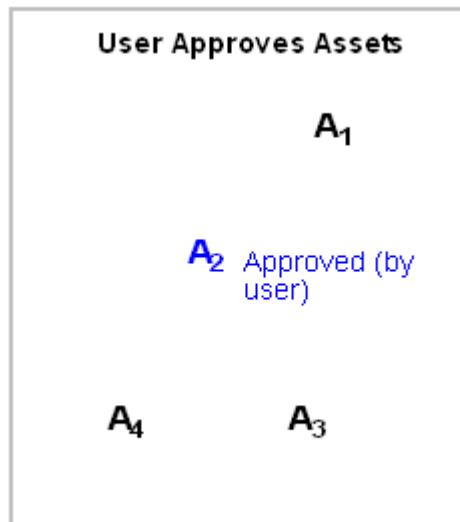
Specifically, the approval system reads the `PublishedAssets` table. If assets currently requiring approval are listed as published to the given destination, the approval system considers it unnecessary to approve (and re-publish) the same assets, assuming version-matching is not a requirement. If an exact dependency is specified in the data model or template code, the approval system checks the versions of the published assets. Information about enforcing version-matched content is given on [Section 13.8, "Ensuring Version-Matched Content."](#)

13.13 Putting It All Together

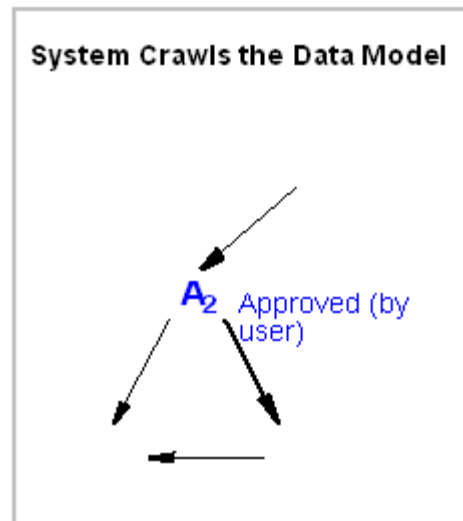
This section contains a scenario that provides a comprehensive illustration as to how the approval system works. The scenario begins with the user approving an asset, continues with the approval system's dependency analysis, and ends with the approval system's response. This scenario touches on all the concepts that were discussed throughout this chapter: approvals, approval dependencies, and approval states, using the Mirror to Server publishing example and an `exists` dependency.

1. In this scenario, a user edits and approves a single asset — `A2` (out of a possible four) — to be published dynamically under an `exists` dependency. Only `A4` has been previously published to the target destination.

Figure 13–20 *User-Approved Asset*

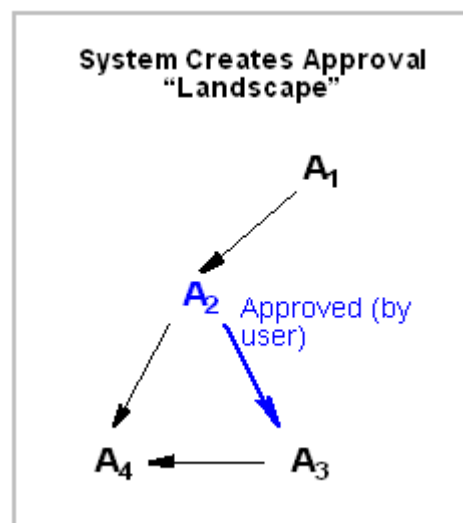


2. When the user attempts to publish, the approval system crawls the data model to determine the encoded dependencies (represented by arrows in the figure).

Figure 13–21 System Crawling Data Model

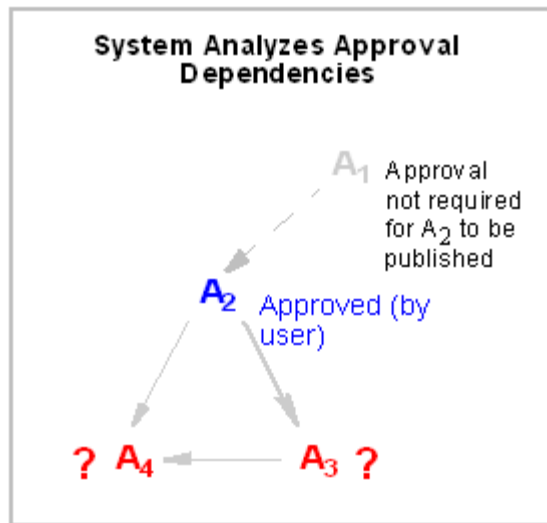
Using the approved asset as a seed, the approval system follows the asset's links to and from other assets in the data model, follows the other assets' links to and from yet other assets, and so on until the system determines the boundaries of the seed's dependency network and no more dependencies remain to be detected.

3. From the results of step 1, the approval system constructs an approval landscape, which identifies all the interdependent assets and relationships among them.

Figure 13–22 System Creating Approval

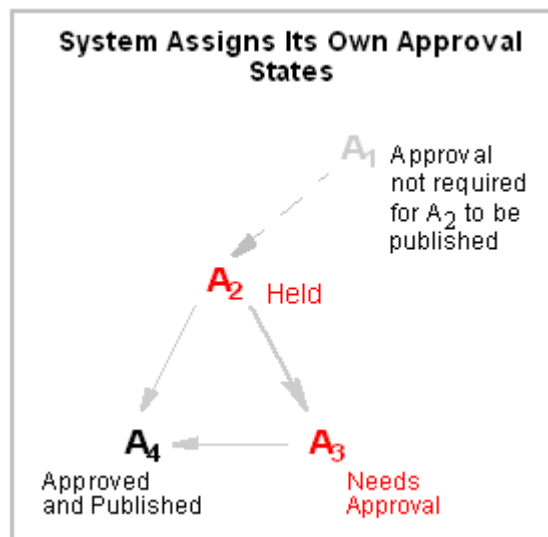
4. The approval system tests for approval dependencies and their impact on the publishing session. In this example, the approval system determines the following:

Figure 13–23 System Analyzing Approval Dependencies



- A1 does not require approval because the approved asset A2, as its child, can be published independently. However, A2 is also a parent asset and cannot be published until the status of its child assets (A3 and A4) is determined. The approval system now searches for previously published assets. Referring to the `PublishedAsset` table, the approval system determines:
 - A3 has never been published to the target destination. A3 requires approval to satisfy the A2 to A3 dependency.
 - Because A4 has been previously published to the target destination, its re-approval (and re-publication) is unnecessary. The dependency of A2 and A3 on A4 will be satisfied at publication time.
5. From step 4, the approval system concludes that three approval dependencies must be satisfied in order for the approved asset to be published, and, accordingly, assigns its own approval states to the assets:

Figure 13–24 System Assigning Approval States

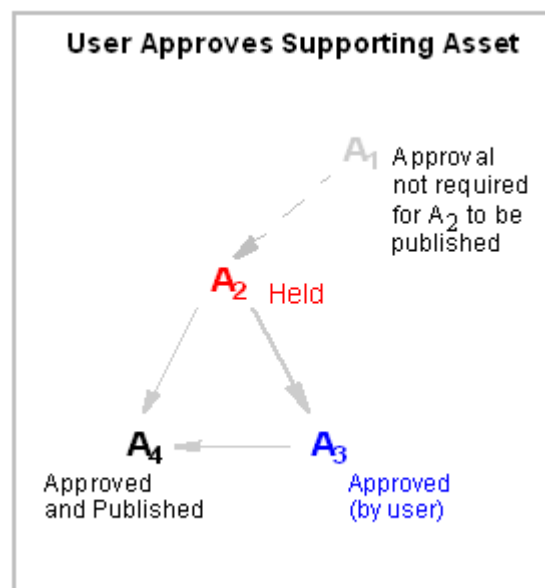


- The A2 to A3 approval dependency needs to be satisfied (by the user approving A3). The system assigns A3 the **Needs Approval** state and A2 the **Held** state.
- A3 to A4 needs to be satisfied (by the user approving A3, which is already assigned the **Needs Approval** state).
- A2 to A4 is automatically satisfied by the pre-publication of A4, which is now assigned the **Approved and Published** state.

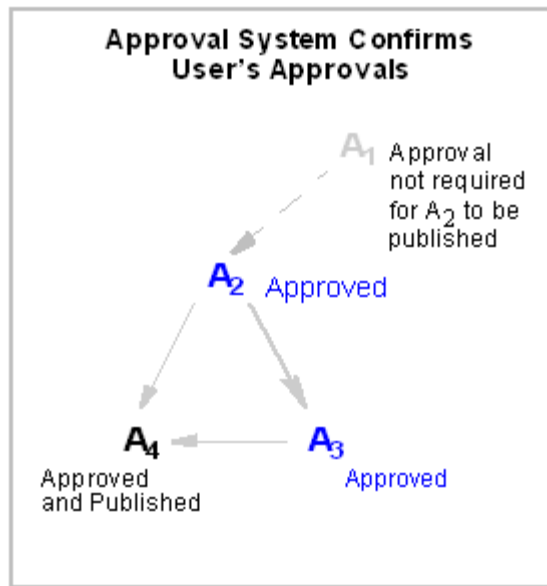
(A complete list of approval states is available in [Table 13-4, "Approval States Assigned by the Approval System"](#).)

6. Finally, the approval system displays the list of assets and their approval states (shown in [Figure 13-24](#)). In response, the user approves the asset that was assigned the **Needs Approval** state (A3).

Figure 13-25 User Approves Supporting Asset



7. To complete the approval process, the approval system updates the approval state of A2 to **Approved** and releases both A2 and A3 to the publishing system for publication in the next publishing session.

Figure 13–26 System Confirmation of User Approval

13.14 Reference: Approval States

Table 13–4, "Approval States Assigned by the Approval System" lists the possible approval states that can be assigned by the approval system to assets in an approval dependency and displayed to the user

Note: When referring to Table 13–4, bear in mind that in the context of an approval dependency, the term "dependent asset" is equivalent to the "parent asset." For rules governing parent-child relationships in approval dependencies, see Section 13.5, "Approval Dependencies and Parent-Child Relationships."

Table 13–4 Approval States Assigned by the Approval System

Approval State	Description
Approved. Approved and ready to publish to destination.	(Informational) This asset will be published at the next publishing event to this destination, unless the asset is changed, or an exact dependency changes.
Approved and published. Asset version is the same as that on destination.	(Informational) An asset with an exact dependency has been published to this destination.
Currently checked out. Will not be published to destination.	(Action may be required) The asset is checked out under revision tracking. Although approved, it cannot be published until revision tracking relinquishes control: <ul style="list-style-type: none"> ▪ Check in – The asset must be re-approved. ▪ Undo Checkout – The asset remains approved and can be published. ▪ Rollback – The asset must be re-approved.
Approved for inclusion as a link in pages exported to destination.	(Informational) This asset is approved for Export to Disk publishing, if it is linked to from the page that is being exported.

Table 13–4 (Cont.) Approval States Assigned by the Approval System

Approval State	Description
Asset has been modified since approved for publish to destination.	(Action required) The asset must be re-approved.
Approved, but approval for publish to destination was based on versions of the child assets that no longer exist.	(Action required) The asset must be re-approved so that its version matches the version of its child assets.
Held. Approved, but child assets have not been approved for publish to destination.	(Action required) The asset is held from a publishing session when any one of the following conditions is true: <ul style="list-style-type: none"> ■ The asset was approved, but was then re-edited and has not yet been re-approved. ■ The asset has a child asset with an exact dependency, and that child asset is not approved. ■ The asset has a child asset with an exists dependency and that child asset has never been published. ■ The asset has a child asset with an exact dependency. The child asset has been published, but it has since been edited and is not yet approved. ■ The asset has an exact dependency on a child of another asset that was previously approved, but has since been edited and is not yet re-approved. ■ The asset is checked out (revision tracking).
Needs Approval. Not yet approved for publish to destination.	(Action required) The asset must be approved.
This asset cannot be published until the assets it references have been approved.	(Action required) A referenced asset must be approved before this asset can be published. Related assets that are held are also listed and may require approval.

Various Topics in Export to Disk Publishing and the Approval Process

This chapter presents the main concepts and features of Export to Disk publishing, including the approval process as it relates to Export to Disk publishing. When reading this chapter, keep in mind the approval and publishing contexts. While they tend to be discrete—for example, approval templates, pubkeys, and starting points apply to only one context, but not the other—approval and publishing systems depend on each other and therefore interact to some extent.

This chapter contains the following sections:

- [Section 14.1, "Export to Disk Publishing Terminology"](#)
- [Section 14.2, "Approvals and Export to Disk Publishing"](#)
- [Section 14.3, "Export to Disk Publishing"](#)
- [Section 14.4, "Frequently Asked Questions"](#)
- [Section 14.5, "Behavior and Functionality"](#)
- [Section 14.6, "Approval and Publish-Time Templates: What Happens When They Differ?"](#)
- [Section 14.7, "Sample Templates"](#)

14.1 Export to Disk Publishing Terminology

Compared to mirror publishing, in which dependencies in both the approval and publishing stages are determined by the asset model, Export to Disk publishing is an entirely template-driven process, in both the approval and publishing stages. If the approval template and publishing templates differ with respect to the dependencies they specify—a condition we do not recommend—the likelihood of publishing unapproved content increases.

Coding templates is the responsibility of the developer. However, the WebCenter Sites administrator (and any other users who publish) must understand which templates to invoke and what outcome to expect in order to prevent adverse effects on publishing sessions.

This chapter begins by explaining the most commonly used terms in Export to Disk publishing, then delves into the workings of the approval system. It concludes with a series of examples illustrating how various relationships among assets and templates determine approval and publishing results.

Note: The terminology presented here refers only to Export to Disk publishing. Terms such as **publish key**, **dependencies**, and **references**, have completely different meanings in mirror publishing and are beyond the scope of this document.

This section contains the following topics:

- [Section 14.1.1, "Page"](#)
- [Section 14.1.2, "Publish Key"](#)
- [Section 14.1.3, "Publish Queue"](#)
- [Section 14.1.4, "Primary Asset"](#)

14.1.1 Page

In the Export to Disk publishing context, a page is typically an HTML file that is exported to a disk. (The file is not to be confused with a Page asset type.) Each exported page is represented by a [Section 14.1.2, "Publish Key."](#)

14.1.2 Publish Key

A publish key (or pubkey) is strictly a publish-time term. A pubkey represents an atomic publishing unit, which in the case of Export to Disk publishing is an exported file (equivalent to a [Section 14.1.1, "Page,"](#) not to be confused with a Page asset type).

A pubkey is defined by two constructs: an asset, and a publish-time template (not to be confused with an [Section 14.2.1, "Approval Template"](#)).

The initial pubkeys in Export to Disk publishing are the [Section 14.3.2, "Starting Points,"](#) defined by the user who initiates the publishing session. All other pubkeys are discovered and logged during publishing.

14.1.3 Publish Queue

A publish queue is a list of pubkeys awaiting publication. For a pubkey to be added to the publish queue, its [Section 14.1.4, "Primary Asset"](#) must be approved.

14.1.4 Primary Asset

The primary asset participates in the definition of a pubkey. In layman's terms, its ID and type become the cid and c parameters for the template. A primary asset must be approved before its page or pages can be exported.

Note: If it is determined that a page's primary asset is not approved, the page will not be exported. This can lead to the creation of a broken link on the publishing destination, depending on which approval and publishing templates were used.

Ultimately, it is the developer's responsibility to code approval templates such that they can validate the dependencies that are expected to exist on the target destination.

14.2 Approvals and Export to Disk Publishing

Exporting a page first requires approval of the page's primary asset. Approval of the primary asset is contingent on the approval states of its dependent assets, which are dictated by the primary asset's approval template. The dependent assets' dependencies are further dictated by their own approval templates.

Note: Because the approval template is not necessarily the one used to render the asset, it is possible to publish assets without passing them through the approval system. This implication and others are discussed and illustrated in [Section 14.5, "Behavior and Functionality"](#) and [Section 14.6, "Approval and Publish-Time Templates: What Happens When They Differ?"](#)

Also note that compositional dependencies (involving non-primary assets of a page, as explained on [Section 14.3, "Export to Disk Publishing"](#)) need not be approved in order to appear on an exported page (to filter out unapproved assets, see the `<render:filter>` tag in the *Oracle Fusion Middleware WebCenter Sites: Tag Reference* for more information).

This section contains the following topics:

- [Section 14.2.1, "Approval Template"](#)
- [Section 14.2.2, "Approval Queue"](#)
- [Section 14.2.3, "Approval Dependencies"](#)

14.2.1 Approval Template

The purpose of approval in Export to Disk publishing is to allow the developer to define dependencies in a way as to ensure that the correct content is published and it is published intact.

Compared to mirror publishing, in which asset dependencies in both the approval stage and publishing stage are determined by the asset model, Export to Disk publishing is entirely template-oriented. As a result, approval behavior (as well as publish-time behavior) is very much defined by the developer who writes the approval templates (as well as the publish-time templates). WebCenter Sites then uses the approval template to discover an asset's dependencies. If an asset is published against the approval template, its approval dependencies are likely to be the same as its [Section 14.3.3, "Compositional Dependencies."](#)

Approval templates are assigned by use of the Set Default Templates feature in the publishing destination form.

- If a default template is not explicitly chosen, the approval system will choose the asset's default template.
- If neither the approval nor default template is specified, the asset will be approved without an approval template (i.e., it will have no approval dependencies).

If the approval template is shared among sites, the approval system will choose the site entry corresponding to the current site.

14.2.2 Approval Queue

The approval queue handles asset modification events to keep approval tables up to date. For example, if an asset is approved for publication, but you then change it, the approval queue will handle this by unapproving the asset—rejecting it from the publish queue.

You never work directly with the approval queue. The queue runs every five minutes by default. However, if you invoke a feature that uses approval functionality (such as the Publishing tab), the approval queue is forced to run before the screen is rendered to keep approval information up to date.

14.2.3 Approval Dependencies

When you approve an asset, its [Section 14.2.1, "Approval Template"](#) is used to determine its approval dependencies. The following tags create approval dependencies:

- `<asset:load>`
- `<asset:loadall>`
- `<assetset:setasset>`
- `<assetset:setlistedassets>`
- `<render:logdep>`
- `<render:getpageurl>`
- `<render:gettemplateurl>`
- `<render:gettemplateurlparameters>`

If you approve asset A whose template (or any element called by the template) references asset B using one of the tags above, an approval dependency is created from A to B. This generally means that when you want to approve A, you must also approve B.

Types of Approval Dependencies

Approval dependencies are created and used at approval-time, not to be confused with publish-time dependencies (even though they can be the same). Approval for Export to Disk publishing involves two types of approval dependencies:

- Template dependencies
- Reference dependencies

Template dependencies are created when an asset's approval template uses another asset to define the content. For example, if asset A's approval template loads asset B, then A has a template dependency on B. In more practical terms, if you have an approval template for a Page asset that shows an Article asset, the Article asset is used on the Page, so the dependency is of a template kind. The following tags generate template dependencies:

- `<asset:load>`
- `<asset:loadall>`
- `<assetset:setasset>`
- `<assetset:setlistedassets>`
- `<render:logdep>`

Reference dependencies are generated when a link is created from one page to another. They are registered as reference dependencies between the primary assets of the two pages. For example, if we create a hyperlink from the approval template of asset A to a page where asset B is the primary asset, the approval system will register this as asset A's reference dependency on B. Tags that generate this kind of dependency are:

- `<render:getpageurl>`
- `<render:gettemplateurl>`
- `<render:gettemplateurlparameters>`

Template dependencies (in Export to Disk publishing) are by default exact dependencies — if you want to approve A you must approve B if B has been changed. Reference dependencies are always exists dependencies—if you approved and published B once, you are not required to approve it again in order to re-publish A.

The exception is when you set `deptype="none"` on any of the tags. In that case, no approval dependency at all is created by that tag. This means no record is created for it during approval; in all other contexts, such as Export to Disk publishing and live sites, the `deptype` attribute is ignored.

14.3 Export to Disk Publishing

Three main components play a role in the Export to Disk publishing process:

- [Section 14.3.1, "Publish-Time Template"](#)
- [Section 14.3.2, "Starting Points"](#)
- [Section 14.3.3, "Compositional Dependencies"](#)

14.3.1 Publish-Time Template

The purpose of the publish-time template is to render content as files. Typically, publish-time templates are identical to the approval templates. However, when they differ, content can be published without first being approved. Bypassing the approval system requires the developer to provide a means by which publishable content can be validated.

14.3.2 Starting Points

The starting point is the `pubkey` (or `pubkeys`) where export begins, which is at publish-time. Typically, starting points are selected to link to most, if not all the pages on your site. (The page may not necessarily be the home page; for example, it could be a sidebar with many links. The choice is yours.) The Export to Disk publishing system will crawl your site, beginning with the starting points, and will keep logging new `pubkeys` as it discovers them.

14.3.3 Compositional Dependencies

Compositional dependencies are the dependencies of a generated page on the assets that were used to generate that page. They are determined by the logic in that page's [Section 14.3.1, "Publish-Time Template."](#) Compositional dependencies dictate what is rendered on the exported page as prescribed by the template, completely ignoring any `deptype` attributes. The same tags that created template and reference dependencies in the approval template also create compositional dependencies at publish-time. The tags are listed in [Section 14.2.3, "Approval Dependencies."](#)

14.4 Frequently Asked Questions

How Do I Select an Approval Template?

The approval template (if specified) is the only template ever used by the approval system to validate a given asset. This template may or may not be used at publish-time, depending on what the starting point is set to. The best advice is to set approval templates that most closely represent the asset's intended dependencies.

What if the template that contains the most representative set of dependencies is not the template that you want to publish the asset with? Set it as the approval template for assets of that type, and use any desired template(s) as starting point(s).

Are Data Model Dependencies Accounted For in Any Way?

Associations, attributes, and so on, are not used in Export to Disk publishing. The only dependencies that matter are those established by templates.

Why Do We Track Publish-Time Compositional Dependencies?

After you export a page, its content is frozen. However, the assets used to generate that page may evolve, making the affected pages obsolete.

Because it records dependencies, WebCenter Sites is able to remind you which pages need updating, assuming you will want to republish those pages. WebCenter Sites gives you two republishing options: automatic refresh and manual refresh.

- With automatic refresh, pages are queued automatically for the next publishing session, without your having to approving compositional dependencies that have changed.

If you choose this option, you will not have to approve the modified assets in order for the page to be queued for publishing; in other words, the article page will be republished as soon as you change the image asset. To set up automatic refresh, specify `exists` or `no` dependencies between the article and the image; this disables the approval system from prompting you to approve the assets.

- With manual refresh, you review the updated pages before publishing them.

If you choose this option, the affected pages will be `held` until approved; in other words, you must reapprove the article page every time the image changes. For this to work, you need to set up `exact` dependencies.

14.5 Behavior and Functionality

Following is a compilation of simple scenarios which demonstrate Export to Disk publishing behavior:

- [Section 14.5.1, "Example 1: Template Dependencies"](#)
- [Section 14.5.2, "Example 2: Reference Dependencies"](#)

The participating asset types are `Page` and `HelloArticle` (from `HelloAssetWorld`). The templates are `Ttemplate`, `Rtemplate`, and `dummyTemplate`.

For clarity, the asset IDs have been hard-coded into the templates. Normally, assets would be loaded somehow instead of being hard-coded, so developers would need to leverage the `deptype= 'none'` attribute in their tags if they do not wish to set up additional dependencies. See [Section 14.7, "Sample Templates"](#) for the template code.

14.5.1 Example 1: Template Dependencies

Setup: Page P asset uses Ttemplate for both approval and publishing. Ttemplate loads HelloArticle A, therefore establishing a template dependency.

Starting Point: P+Ttemplate

Pubkeys (exported files): P+Ttemplate

Approval Dependencies: Page P has template dependencies on Ttemplate because of `<render:logdep>`, and on HelloArticle A because of `<asset:load>`. Because Ttemplate does not create any links, there are no reference dependencies.

Publish Dependencies: Starting Point is the only pubkey; therefore only one file will be exported.

Note that the exported page does not have any publish-time compositional dependencies because it is never actually used by the template. There is only reason why P had to be approved—P is the exported page's primary asset. So, the only compositional dependencies determined at publish-time are the exported page's dependencies on A and Ttemplate.

Table 14–1 *Template Dependencies*

Action	Approval Status	Publish Status	Comments
Initial	Approved:P, A, Ttemplate	Published:A and Ttemplate	P was not published because it was not referenced by the template.
Edit P	P status becomes Changed	Not affected by change.	Since there are no compositional dependencies on P, the change had no effect.
Edit A	P status becomes Held. A status becomes Changed – need to approve P again.	After approval, page is published again	Template dependency on A causes P to be held, and subsequent approval makes the page publishable again.

14.5.2 Example 2: Reference Dependencies

Setup: Page P asset uses Rtemplate for both approval and publishing. Rtemplate references HelloArticle A, which is in turn rendered by dummyTemplate.

Starting Point: P+Rtemplate

Pubkeys (exported files): P+Rtemplate, A+dummyTemplate

Approval Dependencies: Page P has the routine `<render:logdep>` template dependency on its Rtemplate. In addition, it establishes a reference dependency on A because of `<render:getpageurl>` (but not on dummyTemplate, because dependencies between pages are logged only between their primary assets). Note that dummyTemplate does not participate in an approval dependency because it is not referenced by any tags.

Publish Dependencies: The relevant tags are `<render:logdep>` and `<render:getpageurl>`. These dictate the publish-time compositional dependencies as Rtemplate and A. As before, note that Page P does not have a compositional dependency because it is not used by any of the templates. When export encounters `<render:getpageurl>`, it detects that another page is involved, so it creates the A+dummyTemplate pubkey on-the-fly, and runs dummyTemplate to generate its contents. Note that dummyTemplate does not officially have a compositional dependency because it is not used by any tags—it participates only as a pubkey

member; so, if `dummyTemplate` is changed, the approval/publishing cycle will not recognize the change. For this reason, we advise not altering `<render:logdep>`, which appears on every newly created template.

Table 14–2 Reference Dependencies

Action	Approval Status	Publish Status	Comments
Initial	Approved: P, A, Rtemplate	Published:A, Rtemplate	P was not published because it was not referenced by any template. The result is two HTML files: P+Ttemplate and A+dummyTemplate
Edit P	P status becomes Changed – approve it to go back to status quo	Not affected by change.	Since there are no compositional dependencies on P, the change had no effect.
Edit A	A status becomes Changed; P is still Approved, not Held because it has only an exists dependency on A (no reason to hold P) .	P+Rtemplate was automatically queued for re-publishing because <code><render:getpageurl></code> generated a compositional dependency. (If you are not sure as to why, keep reading); A+dummyTemplate was not affected because it does not use A.	A+dummyTemplate was not affected because it does not have tags that use A.

14.6 Approval and Publish-Time Templates: What Happens When They Differ?

Generally, using the same template for approval and for publishing does not create problems, because approval dependencies match compositional dependencies. Problems start to occur when you use a different template for approval than for publishing. This can happen in several situations. For template code, see [Section 14.7, "Sample Templates."](#)

This section contains the following topics:

- [Section 14.6.1, "Example 1"](#)
- [Section 14.6.2, "Example 2"](#)
- [Section 14.6.3, "Example 3"](#)

14.6.1 Example 1

Asset A uses T1 for approval. T1 does not reference any assets. Asset A uses T2 for publishing. T2 references asset B using `<asset:load>`. T2 loads asset B, which was never approved, but B will nevertheless be on the exported A+T2 page.

Does this mean you can publish assets that were never approved?

Yes! Some users prefer this, only to avoid approving assets before publishing. Approval can be a taxing operation because exact dependencies cause primary assets to be given Held status if the dependent assets change. As a result, pages of those primary assets cannot be exported. However, a better way to handle such situations is to make good use of the `deptype` attribute with `exists` or `none` values to avoid unwanted approval dependencies.

Now, what happens after asset B is published and then we change B? In this case, there is no approval dependency between asset A and asset B, but there is a compositional dependency. This means that whenever asset B is changed, the affected pages will be automatically re-published. (Because there are no dependencies, the automatic refresh option is chosen). So, A+T2 will be placed in the publish queue automatically as soon as asset B is saved—not approved. For information about republication options, see ["Why Do We Track Publish-Time Compositional Dependencies?"](#) on page 14-6.

14.6.2 Example 2

Asset A uses T1 for approval. T1 references asset B using `<asset:load>` under an exact dependency. Asset A uses T2 for publishing. T2 does not reference asset B.

In this case, there is an approval dependency between asset A and asset B, but there is no compositional dependency. This means that whenever asset B is changed, asset B has to be approved before asset A can be approved, even though the published page does not make any use of asset B.

14.6.3 Example 3

Asset A uses T1 for approval. T1 does not reference any assets. Asset A uses T2 for publishing. T2 references asset B using `<asset:getpageurl>`.

In this case, there are no approval dependencies, so A is the only approved asset. During publishing of A+T2, we realize there is another page to be created, with B as primary asset. Now, remember that primary assets must be approved before their page is exported. However, B is not approved, so its page will not be exported. A+T2 is exported, but has a broken link.

Unknown dependencies and Export to Disk publishing. Situations become complicated when you start using `<render:unknowndeps>`. This tag removes compositional dependencies, and it records unknown dependencies to the `pubkey`. Assets thus remain in the publish queue permanently (meaning they are refreshed with every publish cycle). This behavior is expected, because if unknown dependencies exist (that is, dependencies may change at any moment, such as query results), the system cannot determine which assets a given asset depends on. To be on the safe side, the system always republishes the assets.

Many exact dependencies. While this is not always avoidable, keep in mind that when a primary asset has exact dependencies on other assets, a change to any of the dependent assets will cause the approval system to change the primary asset's status to Held. The primary asset's page is not publishable unless approved again. If the modified asset is used by many pages, they will all need re-approval. You may consider setting the `deptype` to `exists`, and make the dependent pages automatically publishable—but the modified asset must be ready when the publishing session begins.

Note: The preferred way to customize export URLs (without using `simplename`) is to use `PREFERREDFILE` and `PREFERREDDIR` as parameters to `<render:getbloburl>` and `<render:getpageurl>` to specify arbitrary names. For example:

```
<render:getbloburl blobtable='MungoBlobs'
  blobcol='urldata' blobkey='id'
  blobwhere='1088466917821' outstr='pagelogoURL'
  csblobid='1088466917821'>

<render:argument name='PREFERREDFILE'
  value='myBlob.out' />
<render:argument name='PREFERREDDIR' value='myDir' />
</render:getbloburl>
```

This code exports the blob whose id is 1088466917821 to `myDir/myBlob.out`.

14.7 Sample Templates

This section lists the template implementations used in the examples in [Section 14.5, "Behavior and Functionality"](#) and [Section 14.6, "Approval and Publish-Time Templates: What Happens When They Differ?"](#) Taglib and import definitions have been skipped.

Ttemplate:

```
<!-- Record dependencies for the Template -->
<ics:if
  condition='<%=ics.GetVar("tid")!=null%>'><ics:then><render:logdep
  cid='<%=ics.GetVar("tid")%>' c="Template"/></ics:then></ics:if>
<asset:load name='myArticle' type='HelloArticle'
  objectid='1156878442427' />
```

Rtemplate:

```
<!-- Record dependencies for the Template -->
<ics:if
  condition='<%=ics.GetVar("tid")!=null%>'><ics:then><render:logdep
  cid='<%=ics.GetVar("tid")%>' c="Template"/></ics:then></ics:if>
<render:getpageurl outstr="myURL" pagename='HelloAssetWorld/
  dummyTemplate' cid='1156878442427' c='HelloArticle' />
Got URL: <a href='<%=ics.GetVar("myURL")%>'> Click here</a><br />
```

dummyTemplate:

completely blank, no logDep.

The Export to Disk Publishing Process

This chapter explains how Export to Disk publishing works and how it is configured.

This chapter contains the following sections:

- [Section 15.1, "Rendering Export to Disk Pages"](#)
- [Section 15.2, "How Export to Disk Publishing Works"](#)
- [Section 15.3, "Path Naming Convention"](#)
- [Section 15.4, "Configuring Your System for Export to Disk Publishing"](#)

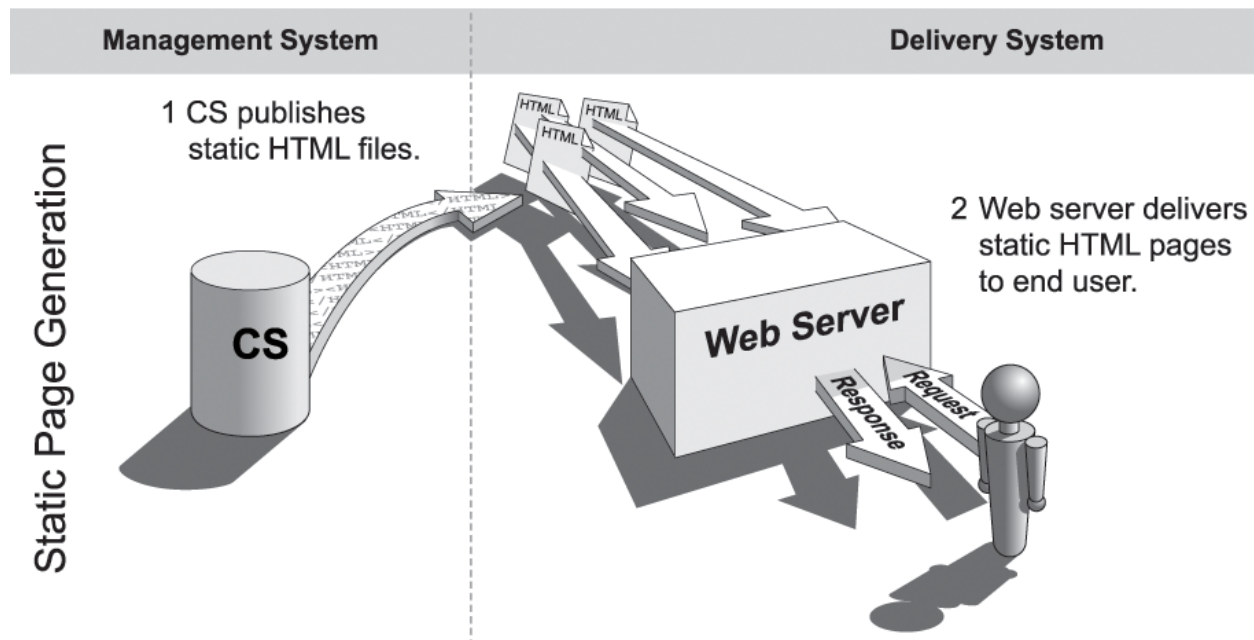
15.1 Rendering Export to Disk Pages

In extremely simple terms, WebCenter Sites renders a Export to Disk page as follows:

1. A page name is submitted by the client browser to WebCenter Sites (through HTTP or another protocol).
2. WebCenter Sites locates the page in the SiteCatalog table, invokes the root element, and renders the page to an html file.
3. You publish the file (either through FTP or another file transfer protocol) to the web server that is hosting your online site.

Typically, administrative users of the Export to Disk publishing method set up a quality assurance process to test the rendered files before moving them to the web server of the delivery system.

Figure 15–1 Export to Disk Rendering



15.2 How Export to Disk Publishing Works

When content is published using Export to Disk, the approval system, the publishing schedule, and your destination configurations all contribute to the process of exporting your approved assets into static HTML files.

When an Export to Disk publishing session runs, this is what happens:

1. The publishing system notifies the CacheManager servlet that a session is about to begin for a specific destination. CacheManager clears all the pages that were previously exported to this destination from the page cache on the source system.

During an export publishing session, exported files are cached to the page cache. If publishing sessions occur more frequently than the cache is configured to be cleared (expiration time), there could be exported pages in the cache that have the same names as the newly exported pages. To ensure that old files are not used, the CacheManager servlet clears the page cache of any pages that were exported to during previous publishing sessions before a new publishing session starts.

2. WebCenter Sites creates an export queue and adds all references that can be published to it. A reference that can be published is one of the following:

- A pubkey that has been designated as an **export starting point**. See [Section 15.3.5, "Export Starting Points."](#)

There must be a starting point for the site and its asset must be approved, or the export session cannot begin. Every site that is exported must have at least one export starting point designated for it.

- Any previously exported page or pagelet (reference) whose assets have changed and been approved again since the last publishing session to this destination. (After a site has been published, the publishing system knows which references have been published. It reads the data in the `PublishedAssets` table and then adds any references whose assets have been changed and approved again for this destination to the export queue.)

- Any asset whose template uses `RENDER.UNKNOWNDEPS` tag.
This tag alerts the approval system that the dependencies for the asset being rendered by the template cannot be calculated because they are unknown. It is typically used for queries. When the dependencies are unknown, the system cannot determine whether changed dependents would require the asset to be published. Therefore, the presence of this tag means that the asset must be published during every publishing session.
 - Any page or pagelet (reference) that has not yet been published and is connected to another page or pagelet through a `RENDER.GETPAGEURL` tag. (That is, it is referenced from another page and has not yet been published.)
 - Any page or pagelet (reference) identified by a `RENDER.SATELLITEPAGE` or `satellite.page` tag.
3. For the first export starting point, WebCenter Sites determines whether its asset has been approved:
 - If the asset has not been approved, the publishing session ends.
 - If the asset is approved, WebCenter Sites determines the page name of the template assigned to the asset.
 4. WebCenter Sites renders the export starting point by passing the following information to WebCenter Sites:
 - The page name of the template for the starting point.
 - The `rendermode` variable set to `export [destinationID]`.
 - The name and location of the export directory where the rendered files should be saved.
 5. WebCenter Sites invokes the root element of the page name, and begins rendering every approved asset that is connected to that export starting point as a reference. WebCenter Sites writes the resulting rendered pages to files rather than posting them to the browser.
 6. After each file is rendered, WebCenter Sites writes a message about that reference to the publish log for the session.
 7. If there is more than one export starting point, the process cycles back to step 3 in this description.
 8. When the publishing session is successful, WebCenter Sites concludes the session by writing information about the published references to the `PubKey` and `PublishedAssets` tables.
 9. WebCenter Sites also writes information about the assets that were published to the `ApprovedAssets` and `ApprovedAssetDeps` tables. It logs the date from the assets' `updated` field at the time that the assets were published so the approval system can calculate dependencies correctly the next time an asset is approved.

15.3 Path Naming Convention

How WebCenter Sites names paths to exported files depends on the type of asset that is published and path information that might have been set for the asset. The export path can be as simple as a single directory or it can be a path of several directories, limited only by the number of characters that are supported by the operating system.

The export path naming convention is:

<cs.pgexportfolder>/<DIR>/<file_dir>

where <cs.pgexportfolder> is a required directory. The subdirectories are conditional. Whether <file_dir> is used depends on a publishing argument named SIMPLEDIR.

Before summarizing the possible export paths, we define the export path variables and the SIMPLEDIR publishing argument.

This section contains the following topics:

- [Section 15.3.1, "Export Path Variables and SIMPLEDIR"](#)
- [Section 15.3.2, "Export Path Construction"](#)
- [Section 15.3.3, "Paths for Links Within Exported Files"](#)
- [Section 15.3.4, "File Naming Conventions"](#)
- [Section 15.3.5, "Export Starting Points"](#)

15.3.1 Export Path Variables and SIMPLEDIR

Export path variables are defined as follows:

- <cs.pgexportfolder> defines the root directory for all exported files. <cs.pgexportfolder> is the value of the cs.pgexportfolder property in the futuretense.ini file. It is a required value and applies to all Export to Disk publishing destinations.

Typically, <cs.pgexportfolder> names a testing location (a file system where the exported files are verified) rather than a location directly on the delivery system.

- <DIR> is the value of the publishing argument DIR, which is used to create subdirectories of the root directory (specified by cs.pageportfolder). When <DIR> is specified, WebCenter Sites creates <cs.pgexportfolder>/<DIR>.

Note: The DIR publishing argument is available to administrators as the Base Directory field in the Add New Destination form (step 3). Throughout this section, we use <DIR> to mean Base Directory.

This argument is typically used to organize the contents of the root export directory when multiple export destinations exist for your sites.

Note: The directory specified by <cs.pgexportfolder> or <cs.pgexportfolder>/<DIR> is called the destination directory.

- <file_dir> takes one of the values listed below, depending on whether the exported asset is a page asset and whether path information is available. <file_dir> is determined hierarchically, as WebCenter Sites looks for the values listed below, in the order shown. It is assumed that SIMPLEDIR=false.

Table 15–1 Possible File_Dir values

Possible <file_dir> Values	Description
<ForDestination:Path>	<p>Value of the For Destination: Path field, shown on step 8 (the field is displayed in the Specify Path/Filename for destination form for a specific destination).</p> <p>Setting a value in the ForDestination: Path field forces the asset to take the specified path when the asset is exported to the specified destination.</p> <p>If <ForDestination:Path> is not set, WebCenter Sites uses <parent_page's_path></p>
<parent_page's_path>	<p>Value of the Path field for the exported asset's parent page.</p> <p>Note for Page Assets: When exported, a page asset takes the path that is set for its parent page, not the one set for itself. (As a rule, a page asset's Path is applied only to child assets.)</p> <p>If <parent_page's_path> is not set, WebCenter Sites uses <ID_of_parent_page>.</p>
<asset's_path>	<p>Value of the exported asset's Path field.</p> <p>Note: <asset's_path> is used only for assets that are not page assets. If <asset's_path> is not set, WebCenter Sites uses <ID_of_parent_page> as the value of <file_dir>. If both <parent_page's_path> and <asset's_path> are not set, WebCenter Sites uses <ID_of_parent_page>.</p>
<ID_of_parent_page>	<p>ID of the asset's parent page. This value is used if <parent_page's_path> and <asset's_path> are unknown.</p>

- **SIMPLEDIR.** Although this publishing argument is not explicitly part of the export path, it determines whether <file_dir> is used when path information is unavailable. That is:
 - If SIMPLEDIR is set to **true** **and** path information is unavailable, <file_dir> is omitted from the export path. Setting a path overrides SIMPLEDIR.
 - If SIMPLEDIR is set to **false** **and** path information is unavailable, <file_dir> takes <ID_of_parent_page> as its value

Note: The SIMPLEDIR publishing argument is available to administrators as the Use simple directory naming checkbox in the Add New Destination form (see step 3).

15.3.2 Export Path Construction

This section summarizes the export paths that WebCenter Sites can construct using the convention:

```
<cs.pgexportfolder>/<DIR>/<file_dir>
```

as explained in [Section 15.3, "Path Naming Convention."](#) Note that only <file_dir> varies in all the examples below

This section contains the following topics:

- [Section 15.3.2.1, "Forced Export Path"](#)
- [Section 15.3.2.2, "Export Path for Page Assets"](#)
- [Section 15.3.2.3, "Export Path for Assets Other than Page"](#)

- [Section 15.3.2.4, "Sample Export Paths"](#)

15.3.2.1 Forced Export Path

When `<ForDestination:Path>` (defined in [Section 15.3.1, "Export Path Variables and SIMPLEDIR"](#)) is specified, the asset being published to the named destination is forced to take the specified path, even if other path information is set. The asset's full export path is:

```
<cs.pgexportfolder>/<DIR>/<ForDestination:Path>
```

15.3.2.2 Export Path for Page Assets

If `<ForDestination:Path>` is not set, WebCenter Sites constructs the export path as shown below (for examples, see [Table 15–2, "Sample Export Paths for Page Assets when SIMPLEDIR=false"](#)).

- When `<parent_page's_path>` is specified, the page asset's export path is:

```
<cs.pgexportfolder>/<DIR>/<parent_page's_path>
```

(even if `<asset's_path>` is specified).

Note: When exported, a page asset takes the path that is set for its parent page, not the one set for itself (that is, `<file_dir>` takes the value `<parent_page's_path>`).

As a rule, a page asset's Path information is applied only to child assets.

- When `<parent_page's_path>` is not specified:
 - If `SIMPLEDIR=false`, WebCenter Sites uses `<ID_of_parent_page>`:

```
<cs.pgexportfolder>/<DIR>/<ID_of_parent_page>
```
 - If `SIMPLEDIR=true`, WebCenter Sites omits `<file_dir>`:

```
<cs.pgexportfolder>/<DIR>
```

15.3.2.3 Export Path for Assets Other than Page

For an asset that is not a page, WebCenter Sites constructs one of the following export paths depending on which information is available (For examples, see [Table 15–3, "Sample Export Paths for Assets Other Than Page when SIMPLEDIR=false"](#)).

- When `<asset's_path>` is specified, the export path is:

```
<cs.pgexportfolder>/<DIR>/<asset's_path>
```
- When `<asset's_path>` is not specified, WebCenter Sites uses `<parent_page's_path>`:

```
<cs.pgexportfolder>/<DIR>/<parent_page's_path>
```
- When both `<asset's_path>` and `<parent_page's_path>` are unspecified:
 - If `SIMPLEDIR=false`, WebCenter Sites uses `<ID_of_parent_page>`:

```
<cs.pgexportfolder>/<DIR>/<ID_of_parent_page>
```


- If `SIMPLEDIR=true`, WebCenter Sites omits `<file_dir>`:
`<cs.pgexportfolder>/<DIR>`

15.3.2.4 Sample Export Paths

The following tables describe the sample export paths for assets.

Table 15–2 Sample Export Paths for Page Assets when `SIMPLEDIR=false`

Page Asset	<code>cs.pgexportfolder</code>	<code>DIR</code>	<code>file_dir</code>			Export path
N/A	N/A	N/A	<code>asset's_path</code>	<code>parent_</code> <code>page's_path</code>	<code>ID_of_</code> <code>parent_page</code>	N/A
Home	/export	N/A	/abstracts	N/A	N/A	/export
Home	/export	/Japan	/abstracts	N/A	N/A	/export /Japan
World News	/export	/Japan	/abstracts	/news/	998877665	export/ Japan/news
World News	/export	N/A	/abstracts	/news/	998877665	export/news
World News	/export	/Japan	/abstracts	N/A	998877665	export/ Japan/99887 7665
World News	/export	N/A	/abstracts	N/A	998877665	export/ 998877665

Table 15–3 Sample Export Paths for Assets Other Than Page when `SIMPLEDIR=false`

Not Page Asset	<code>cs.pgexportf</code> <code>older</code>	<code>DIR</code>	<code>file_dir</code>			Export path
N/A	N/A	N/A	<code>asset's_path</code>	<code>parent_</code> <code>page's_path</code>	<code>ID_of_</code> <code>parent_page</code>	N/A
Oil Price	/export	/Japan	/energy	N/A	997766554	/export/Japan/ energy
Oil Price	/export	/Japan	/energy	/news/	997766554	/export/Japan/ energy
Oil Price	/export	N/A	/energy	/news/	997766554	/export/energy
Oil Price	/export	/Japan	N/A	/news/	997766554	/export/Japan/ news
Oil Price	/export	/Japan	N/A	N/A	997766554	/export/Japan/ 997766554
Oil Price	/export	N/A	N/A	N/A	997766554	/export/997766 554

15.3.3 Paths for Links Within Exported Files

URLs in links for HREFs within the exported files do not include the values that specify the destination directory (`<cs.pgexportfolder>/<DIR>`). Instead, they begin within the destination directory and are relative to that directory.

The **URL Prefix** field on the Destination Configuration form is used to resolve URLs by specifying the location of those files on the delivery system. That is, URLs for links within the exported files are created like this:

```
<URLPREFIX>/<path_of_parent_page>
```

This means that the value for **URL Prefix** must match the value of the web alias that identifies the directory where the files are found. If you are using only `cs.pgexportfolder` to create the destination directory, the web alias that the URL Prefix represents must point to that location. And if you have set the Base Directory field on the destination configuration form to add a subdirectory to the root directory specified by `cs.pgexportfolder`, be sure that the web root or the web alias represented by *URL Prefix* points to this directory.

15.3.4 File Naming Conventions

How WebCenter Sites names an exported file depends on whether a file name is provided by the asset (in the Inspect form) and whether **Use Simple File Naming** is selected (on the Destination Configuration form; for more information on this setting see [Section 15.4, "Configuring Your System for Export to Disk Publishing"](#)). Possible file names are shown in [Table 15–4, "File Naming Conventions"](#). The naming convention is explained below the table.

Table 15–4 File Naming Conventions

File Name	Description
<pagename>_ <packedargs (if any)>_ <filename>.<SUFFIX>	Used for assets with a populated Filename field and Use Simple File Naming is not selected.
<packedargs (if any)>_ <filename>.<SUFFIX>	Used for assets with a populated Filename field and Use Simple File Naming is selected.
<pagename>_ <packedargs (if any)>_ <assetID>.<SUFFIX>	Used for assets with an empty Filename field and Use Simple File Naming is not selected.
<packedargs (if any)>_ <assetID>.<SUFFIX>	Used for assets with an empty Filename field and Use Simple File Naming is selected.

Note: When export files are named, the following syntax changes are made:

- Slash characters (/) in page names are converted to hyphens (-) in file names.
- Equal signs (=) in packed arguments are converted to hyphens (-) in file names.
- Ampersand characters (&) in packed arguments are converted to underscores (_) in file names.

- <pagename> is the name of the SiteCatalog page entry of the template that is used to render the asset. (All template assets have SiteCatalog page entries.)

Note: <pagename> is dropped when the Use Simple File Naming parameter is selected in the Destination Configuration form. Use the Use Simple File Naming option selectively. If your assets are rendered by more than one template, do not select Use Simple File Naming. For more information about Use Simple File Naming, see [Section 15.4, "Configuring Your System for Export to Disk Publishing."](#)

- <packedargs (if any)> are passed in from the page entry's root element. If any packed arguments are passed in from the root element of the asset's rendering template, the values of those arguments are also included in the name of the file generated for the asset.

For information about packed arguments and how they relate to URLs, see the Oracle Fusion Middleware WebCenter Sites Developer's Guide.

- <filename> is the value of the asset's Filename field. If the Filename field is empty, WebCenter Sites uses the asset_ID specified in the asset.

Note: A file name set for an asset for a specific destination in the Specify Path/Filename for Destination form supersedes the file name provided in the Filename field on the asset's New or Edit form when the asset is exported to that destination.

- <asset_ID> is the ID that is specified in the asset. <asset_ID> is used if the asset's Filename field is empty or if the following note applies.

Note: WebCenter Sites needs both the asset's ID and type to determine whether a file name is provided for the asset. If the identity of the asset's type is not provided, the file name cannot be used. In such a case, WebCenter Sites uses the asset's object ID in place of its file name.

- <SUFFIX> is a destination configuration option that specifies the extension of the file name. For more information about the `suffix` parameter, see [Section 15.4, "Configuring Your System for Export to Disk Publishing."](#)

[Table 15-5, "Export File Naming Conventions"](#) provides examples of how file names are created during Export to Disk publishing. In the examples given, Use Simple File Names is not selected in the destination configuration.

Table 15-5 Export File Naming Conventions

Asset Name	SiteCatalog		filename	asset_ID	SUFFIX	Exported File Name
	Pagename	packedargs				
Home	BF/Page/Home	cid=123 c=Page		123		BF-Page-Home_123.html
Home	BF/Page/Home	cid=123 c=Page		123	.htm	BF-Page-Home_123.htm
Home	BF/Page/Home	cid=123 c=Page	home.html	123		BF-Page-Home_home.html

Table 15–5 (Cont.) Export File Naming Conventions

Asset Name	SiteCatalog		filename	asset_ID	SUFFIX	Exported File Name
	Pagename	packedargs				
Home	BF/Page/Home	cid=123 c=Page	home.html	123	.htm	BF-Page-Home_ home.html
Home	BF/Page/Home	cid=123 c=Page PACKEDAR GS="topicw ord=oil"	home.html	123		BF-Page-Home_ topicword_oil_ home.htm

15.3.5 Export Starting Points

The Export to Disk publishing method cannot begin rendering files unless it knows where to start. You tell it where to start by designating at least one starting point: that is, a page asset and the template that should be used to render it. WebCenter Sites invokes the root element of the template's page name, and begins rendering every approved asset that is connected to that export starting point—assets connected through an association, a hyperlink, a navigation bar, a query, and so on.

Note: There must be at least one export starting point for an exported site. Typically it is your home page. However, depending on how your online site is designed, you may need to designate more than one export starting point. Why? Because if there is a section on your site that isn't connected to the rest of the site, it will not be rendered. For example:

- If your online site has more than one top-level page asset, you need an export starting point for each one.
 - If your site uses a combination of static pages and dynamic pages and there is a hard-coded static URL in an HREF on a dynamically-generated page, the asset that the URL points to should be designated as an export starting point.
-

15.4 Configuring Your System for Export to Disk Publishing

The main steps when configuring a system for Export to Disk publishing are these:

- [Section 15.4.1, "Create the Batch User Account \(If One Does Not Exist\)"](#)
- [Section 15.4.2, "Specify the Root Export Directory"](#)
- [Section 15.4.3, "Configure an Export Destination"](#)
- [Section 15.4.4, "Map a URL Prefix for Your Web Server"](#)
- [Section 15.4.5, "Create the Export Starting Points"](#)
- [Section 15.4.6, "Approve Your Assets"](#)
- [Section 15.4.7, "Publish and Test the Results"](#)
- [Section 15.4.8, "Set Up the Schedule"](#)

Note: Configuring a system for Export to Disk publishing requires you to provide information that is specific to your WebCenter Sites installation and your business practices. Before starting the procedures in this section, it is best to read them in order to gather and confirm the information they prompt you to provide.

15.4.1 Create the Batch User Account (If One Does Not Exist)

Note: The steps in this section must be completed only once regardless of the number and types of publishing methods you are setting up. If an account already exists for your installation, skip to [Section 15.4.2, "Specify the Root Export Directory."](#)

The purpose of creating a batch user account is two-fold:

- Configure a batch user account for the publishing system on the source to use.
- Specify where the publish logs should be stored.

The procedures in this section require you to set several properties in the `futuretense.ini`, `batch.ini`, and `futuretense_xcel.ini` files. Additional properties are also available to you for fine-tuning your system configuration. For a complete list of properties, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

To create the batch user account

1. Log in to the administrator's interface on your source system.
2. In the **Admin** tab, expand **User Access Management** and double-click **User**.
3. Create a user account for the publishing system on the source system to use (that is, the batch user) and assign it the following ACLs:
 - Browser
 - ElementEditor
 - PageEditor
 - TableEditor
 - UserReader
 - Visitor
 - VisitorAdmin
 - xceladmin
 - xceleditor

If you need help with this step, see [Section 5.4.1, "Creating a New User"](#).

4. Start the Property Editor:

In Windows:`propeditor.bat`, typically located in `<sites_install_dir>/`

In UNIX:`propeditor.sh`, typically located in `$HOME/<sites_install_dir>`

- a. Open the `futuretense_xcel.ini` file for the source system.
- b. In the **Publishing** tab, set values for the following properties:

`xcelerate.batchhost`

Set this property to the host name of the application server that is hosting the source system. The source system is the batch host. If the port number of the web server is anything other than 80, you must also include the port number. For example, `myserver:7001`. In a clustered WebCenter Sites environment, only one batch host is supported. The `xcelerate.batchhost` property must be set on each cluster member to point to the dedicated host.

`xcelerate.batchuser`

Set this property to the name of the user that you created in step 3 of this procedure.

`xcelerate.batchpass`

Set this property to the password of the user that you created in step 3 of this procedure.

- c. Save and close the property file.
- d. Open the `batch.ini` file for the source system.

In the **Results** tab, set the `request.folder` property to point to the directory that you want to hold your publish log files. By default, this property is set to the `dispatcher` subdirectory in the application server installation directory.

Note: Be sure to set this property to a directory that can be written to. Otherwise, no session information will be displayed in the Publish History tab of the Publish Console.

- e. Save the property file and close the Property Editor.
5. Restart the application server.
6. Continue with the configuration procedure for Export to Disk publishing. Go to [Section 15.4, "Configuring Your System for Export to Disk Publishing"](#) the next procedure in this section.

15.4.2 Specify the Root Export Directory

As mentioned previously in this chapter, the directory that WebCenter Sites exports files to is determined by the `cs.pgexportfolder` property in the `futuretense.ini` file. (If you wish to add subdirectories to this directory, you enter a value in the Base Directory field in the Destination Configuration form.)

To set the root directory for the exported files

1. Start the Property Editor on the source system. If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.
2. Open the `futuretense.ini` file.
3. In the **Export/Mirror** tab, select the `cs.pgexportfolder` property and set the value to the file directory (location) to which all files should be exported.

This is a global setting for the system. If you have multiple destinations and want the files published to those destinations to be stored in separate subdirectories within this directory, enter a value in the Base Directory field when you configure those destinations (in [Section 15.4.3, "Configure an Export Destination"](#)).

For more information about export directories, see [Section 15.3, "Path Naming Convention"](#).

4. Save the property file and close the Property Editor.
5. Restart the application server.

Note: Before you run an export publishing session, make sure that this root export directory exists and that it has enough space for the HTML pages that will be created there.

15.4.3 Configure an Export Destination

To create an export destination

1. Log in to the administrator's interface on the source Sites system.
2. In the **Admin** tab, expand **Publishing**, then **Destinations**, and double-click **Add New**.

WebCenter Sites displays the Add New Destination form:

Figure 15–2 Add New Destination Form

Add New Destination

***Name:**

Delivery Type:

Select the desired publishing option

Complete publish

Delayed publish

***Destination address:**

***Remote User:**

***Remote Password:**

Send Email on Failure:

Verbose Output:

More Arguments:

***Sites:**

***Roles (Approve for Publish):**

***Roles (Publish):**

3. Fill in the fields on the Add New Destination form as follows:
 - In the **Name** field, enter a unique name for the destination.
 - In the **Delivery Type** drop-down list, select **Export to Disk: Export Web Files to Disk**. The available fields on the form will change slightly.

- **Base Directory:** Enter a subdirectory name for the files published to this export destination. The base directory is created as a child directory of the root directory that is specified by the `cs.pgexportfolder` property.

When you enter a value in this field, the export path for the publishing destination is as follows:

```
<cs.pgexportfolder>/<DIR>
```

where <DIR> is the base directory.

Note: URLs that are contained in exported files do not include this destination directory path in the URL. Exported files have URLs that begin below this level, inside the destination directory. Therefore, if you enter a value in this field and you enter a value in the URL Prefix field as well, be sure that the web root or the web alias represented by the URL Prefix field points to this directory.

This parameter is typically used to organize the contents of the root export directory when there are multiple export destinations for your sites.

- **URL Prefix:** Enter the prefix that the export process will add to the beginning of the URLs that are used as links (HREFs) within the exported files.

Specify the web server alias of your delivery system in the URLs so that the links in the HTML files can be resolved when the files are moved to that system. If you leave this field blank, URLs will be relative.

The names of the generated files do not include this prefix—only the links within the files use it.

Because you should always test the HTML files before you move them to the delivery system, be sure that the name of the web alias on the testing system is the same as the name of the web alias on the delivery system.

- **Suffix:** In this text box, specify the file suffix to use for the generated files when the **Filename** field for the asset is not used or it does not specify a suffix. The default is **html**.
- **Use simple file naming:** If this parameter is selected, WebCenter Sites overrides the normal file-naming conventions by ignoring the `SiteCatalog` page entry of the template that is used to render the asset and uses only the value entered in the asset's **Filename** field as the file name. If the asset does not have a filename, WebCenter Sites uses the asset's ID, instead. (For a description of the file-naming conventions, see [Section 15.3.4, "File Naming Conventions"](#).)

Note: There is one exception to this rule: assets with upload fields (assets that are blobs) will always receive a standard (long) file name, even when this value is set to true.

Caution: The **Use simple file naming** parameter must be used with caution. If your site is designed such that individual assets are rendered by multiple templates, you must not use this parameter. Without the SiteCatalog page entries used in the file name, file names cannot be guaranteed to be unique when an asset is rendered by more than one template.

- **Use simple directory naming:** When selected, this parameter overrides the normal directory-naming convention. For a description of the file-naming conventions, see [Section 15.3, "Path Naming Convention."](#)

When **Use simple directory naming** is selected, WebCenter Sites puts the rendered HTML file directly into the default export directory when there is no path information available from the parent asset. (Normally, WebCenter Sites uses the page asset's ID when there is no path information.) Note that if there is a value in the Path parameter of the asset's parent page asset, this selection is ignored.

- **Old Template:** Select this parameter to use the rendering methodology that was standard in 3.5 and earlier versions of the product.

Note: If the online site that you present on your delivery system was designed for the rendering model used in Sites versions prior to 3.6 and it has not yet been redesigned for the 3.6 rendering model, you must select this parameter.

If you select this parameter, do not set an export starting point for your site. The export starting point is determined by the templates themselves.

- **Send Email on Failure:** If publishing fails and email notices to that effect are required, select this option. Additional fields appear.
 - **Email Addresses:** Enter the recipient's email address, and click Add. (This field is displayed only when **Send Email on Failure** is selected). All email addresses that will receive an email on failure are displayed.
- **Verbose Output:** Activates detailed error logging during the publishing process. When selected, messages in addition to error messages are written to the PubMessage table. Because additional information lengthens the publishing process, be sure to select this option only for troubleshooting.
- **More Arguments:** No additional arguments may be specified at this time.
- **Sites:** Select the sites whose assets can be approved for and published to this destination.
- **Roles (Approve for Publish):** Select the roles to which you are assigning approval privileges. All users assigned these roles can approve assets.
- **Roles (Publish):** Select the roles that have publish privileges. All users assigned these roles can publish.

4. Click **Add New Destination**.

The Publish Destination form appears showing the new destination.

Figure 15-3 Publish Destination Form

Publish destination: FirstSiteII_E2D

 Inspect

 Edit

 Delete

Approve Multiple Assets

Set Default Templates

Name: FirstSiteII_E2D

Delivery Type: [Export to Disk](#): Export Web files to disk

Base Directory:

URL Prefix:

Suffix:

Use simple file naming: false

Use simple directory naming: false

Old Template:

Send Email on Failure: false

Verbose Output: false

More Arguments:

Sites: FirstSite II

Roles (Approve for Publish): Any

Roles (Publish): Any

Publish Event: No existing publish event

Set Publish Event...

ID: 1328442570750

Force Approve Published Assets

[» List all Publish Destinations](#)

5. Repeat steps 2–4 for each additional export destination you need to configure.

15.4.4 Map a URL Prefix for Your Web Server

To make your exported content available to the public at its final destination (your delivery system), you must configure your web server to map a URL path prefix to the place where the files will reside so that URLs can be resolved.

Refer to the vendor documentation for your web server for information about the appropriate procedure for mapping URL prefixes on your web server. Remember that the name you specify for this prefix must match the name that you specified with the URL Prefix destination configuration parameter for the destination. If you enter a URL Prefix, be sure that the web root or alias includes the directory you entered.

15.4.5 Create the Export Starting Points

To create an export starting point

1. Using the administrator's interface on the source system, find the asset that you want to specify as an export starting point.
2. Open the asset in its Status form and scroll to the Publishing Destination section.
3. Do one of the following:
 - If the asset is not yet approved for the destination you are currently configuring, select **Approve for Publish** from the toolbar (The green check mark icon) and go to step 4.
 - If this asset is already approved for the destination you are currently configuring, go to step 6.
4. In the publishing approval form, select the destination for which you are approving the asset and click **Approve**.

WebCenter Sites calculates dependencies for all the assets linked to this asset and displays the results.
5. When WebCenter Sites displays the approval results, click the **Status** icon (a clock face with a green check mark) from the toolbar.
6. Scroll down to the section of the Status form that displays the selected destination.
7. In the **File/Path** field, click **Specify Path/Filename, Start points** to configure the export starting point:

Figure 15–4 Content Attribute Form

Content Attribute:FSIIBody

Name: FSIIBody

Description: Body

Status: Edited

Created: Tuesday, October 19, 2004 9:36:13 AM PDT by admin

Modified: Wednesday, November 3, 2004 10:19:37 AM PST by admin

Workflow commands:

Workflow process: This asset is not in Workflow.

Workflow state: There are no current assignments. Workflow is inactive.


Workflow history: [Click here to see Workflow History](#)

FSII Destination (RealTime): **Approval State:** Needs Approval. Not yet approved for publish to FSII Destination (RealTime).

FSII Destination (dynamic): **Approval State:** Needs Approval. Not yet approved for publish to FSII Destination (dynamic).

FirstSiteII_E2D: **Approval State:** Approved. Approved and ready to publish to FirstSiteII_E2D.

Template: OpenMarket/TemplateDefault

File/Path: Specify Path/Filename, Start points 

Start points: Not an export starting point.

Approve Dependents: No dependent assets

LoafersBakery: **Approval State:** Needs Approval. Not yet approved for publish to LoafersBakery.

click to
configure
export starting
point

8. In the next form, fill in the fields as instructed below:

Figure 15–5 Content Attribute Form

Content Attribute: FSIIBody

Name: FSIIBody

Description: Body

Template: *No template specified*

Status: Edited

ID: 1112192431363

Created: Tuesday, October 19, 2004 9:36:13 AM PDT by admin

Modified: Wednesday, November 3, 2004 10:19:37 AM PST by admin

Path:

Filename:

For Destination: FirstSiteII_E2D

Path:

Filename:

Is this asset an export starting point? Yes No

Using templates:

Template	Wrapper page	Force specified path	Force specified filename
<input type="checkbox"/> FSIIDetail	<input type="text" value="-None-"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> FSIIlink	<input type="text" value="-None-"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> FSIIsummary	<input type="text" value="-None-"/>	<input type="checkbox"/>	<input type="checkbox"/>

a. (Optional) For Destination section:

- a. Path:** If you want to specify unique path information for the asset being published to this destination (the path information is not provided elsewhere for this asset), enter the path in the **Path** field.
- b. Filename:** If you want to specify file name information that is different for this destination, enter it in the **Filename** field. A file name entered in this form for this destination overrides any file name information provided elsewhere for the asset.

- b. Select **Yes** for **Is this asset an export starting point?**
- c. (Optional) **Using Templates** section:
 - a. Select a template.
 - b. Select a wrapper page.
 - c. Select **Force specified path** if you would like to force this asset to be exported to the path that is named in the **For Destination** section.
 - d. Select **Force specified filename** if you would like to force the use of the file name that is specified in the **For Destination** section.
9. Click **Save**.
10. Repeat steps 1–9 for each top-level page asset in the site.

15.4.6 Approve Your Assets

If you want to perform a simple test of your configuration, select the export starting point with the fewest number of dependent assets and approve each of those assets.

If you want to complete a test publish of your entire site, approve all the assets for the site. See [Section 18.2, "Approving Multiple Assets"](#) for help with approving many assets at once.

15.4.7 Publish and Test the Results

Once the assets you want to publish to the selected destination are approved, run a test publishing session:

1. In the administrator's interface on the source system, click **Publishing** in the button bar.
2. In the **Publish Console**, select your destination from the drop-down list and click **Select Destination**.

WebCenter Sites displays information about the assets that are ready to be published to this destination. (If you have not yet created an export starting point for this destination, the form states this fact. You must create at least one export starting point before the publishing session can begin.)

3. Click **Publish**.

A confirmation message appears. Click **Yes** to proceed with the publish.

The publishing system exports all the approved assets for this destination into files.

4. In the **Publish Console Active** tab you can observe that the session is running.
5. When the session completes, click the **History** tab to view the session summary for this session.

Mouse over the session information to view summary information;

6. In the Publish Console **History** tab, select the link to the publishing destination.

A session summary displays a list of all the files that were created during the export process.

7. Click the **Preview** (binoculars) icon next to the file representing the top-level page in your site.

WebCenter Sites opens the page in a new browser window.

8. Scan the page for errors and test all the links. Make sure to test links to all of the other files that were generated.
9. Test the results by directly entering the URL of the rendered home page into your browser and navigating the entire site. To do this, you must first set up a web server root on the Sites management system that points to the export directory.
10. If you decide that the files are ready for delivery, copy them to your delivery system.

15.4.8 Set Up the Schedule

After you have ensured that:

- Your destination is configured correctly,
- The file names and directory names are created according to your needs, and
- Your web server is delivering the files correctly,

you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see [Section 18.7, "Scheduling Publish Events."](#)
- Plan how you will copy the generated files to your web servers, for both the testing area and the delivery system. For example, you can set up a script that automatically copies the files via FTP to a testing area after a publishing session completes.

The Mirror to Server Publishing Process

This chapter explains how Mirror to Server publishing works and how it is configured.

This chapter contains the following sections:

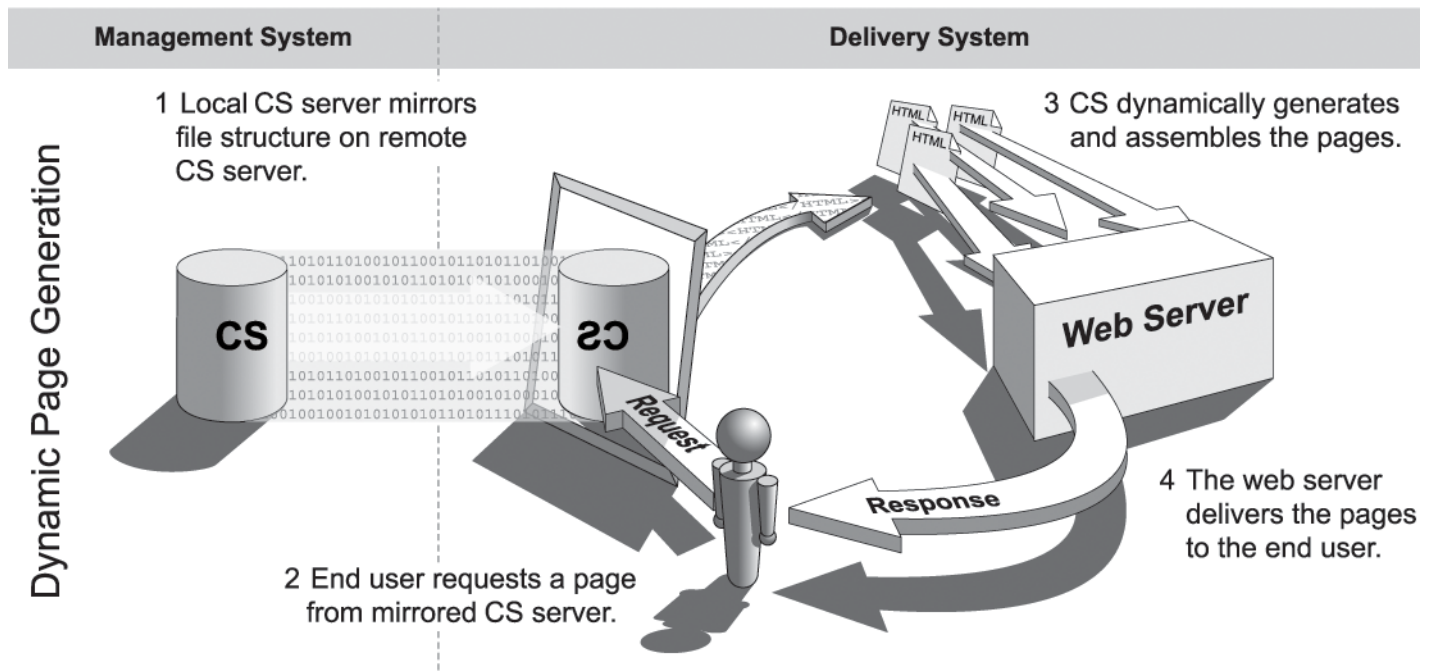
- [Section 16.1, "How Mirror to Server Publishing Works"](#)
- [Section 16.2, "Before Configuring Mirror to Server Publishing"](#)
- [Section 16.3, "Configuring Your System for Mirror to Server Publishing"](#)
- [Section 16.4, "Approval Query for Performance Enhancement"](#)
- [Section 16.5, "Retrieving Logs From Delivery WebCenter Sites Systems"](#)

Note: The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Mirror publishing, static publishing. Mirror publishing is replaced by RealTime publishing.

16.1 How Mirror to Server Publishing Works

The Mirror to Server publishing method gathers information from the approval system, the publishing schedule, and the destination configurations, copies data to the destination, unpacks that data on the destination system, and then invokes the CacheManager servlet to refresh any pages that should be regenerated to take advantage of the new content.

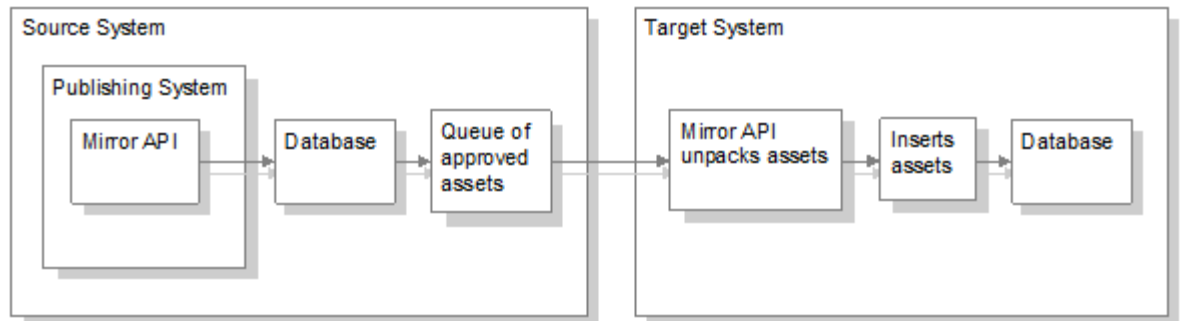
Figure 16–1 Mirror to Server Publishing



Whether or not your Sites delivery system delivers content dynamically, you probably use the Mirror to Server publishing method to move data from your development systems to your management system. For information about that process, see [Section 18.16, "Troubleshooting"](#).

When a Mirror to Server publishing session runs, this is what occurs:

Figure 16–2 Mirror to Publishing Process



1. On the source system, Mirror to Server publishing uses the list of approved assets that the publishing system passed to it to create a mirror queue for the destination.

Mirror Queue for Basic Assets

For basic assets, the following information is added to the queue:

- The asset's main table row. For example, for page assets, the asset's row in the Page table.
- The appropriate rows in the AssetPublication table. These rows list sites and which assets belong to them.

- Rows in the `AssetRelationTree` table that refer to any assets that are associated with the asset being published.
- The associated assets referenced by those rows in the `AssetRelationTree` table. The asset table rows, `AssetPublication` rows, and associated assets of any dependent assets are also mirrored, if they are approved and have not yet been published.

Mirror Queue for Flex and Complex Assets

For flex asset types and the other multi-table asset types like `template` and `CSElement`, the information from the appropriate rows from all of their tables are serialized into an object and stored in the `_Publish` table for that asset type.

For example, when a template is to be published, the appropriate rows from the `Template`, `SiteCatalog`, and `ElementCatalog` tables are serialized into an object and stored in the `Template_Publish` table.

Each item in a `_Publish` table is added to the mirror queue.

2. WebCenter Sites uses the `AssetPublishList` table to create a list of all the assets that are in the mirror queue.

3. The mirror operation starts.

First, the `AssetPublishList` is mirrored from the source to the destination.

4. The mirror queue is delivered and the publishing system unpacks the assets in the queue.

For flex assets, Mirror to Server publishing de-serializes the objects in the `_Publish` tables, and inserts the results in the appropriate tables.

For basic assets, each row in the queue is copied.

5. When the items in the mirror queue are unpacked, the publishing system sends messages that the mirror publish concluded successfully. The destination system responds as follows:
 - The newly-published assets are marked as changed on the destination system, which means that before they can be published from that system to another destination, they must be approved. Note that this feature is enabled by default. You can turn it off if you need to. See [Section 16.3.12, "Turn Off Asset Invalidation on the Delivery System"](#) for details.
 - The `CacheManager` servlet on the destination regenerates the appropriate pages in the cache so that all pages that refer to the assets that were just published are updated. It also rebuilds any pages that have unknown compositional dependencies.
 - `CacheManager` then communicates a message about which pages must be refreshed to each Satellite Server identified by the `satellite.ini` file on the destination system. It communicates with the co-resident version and any remote instances that are identified as belonging to this Sites system. The Satellite Server applications then use that information to refresh the Satellite Server page caches.
 - The `AssetPublishList` table is cleared, making it ready for the next publishing session.
6. On the source system, the publishing system updates the publish log file. Unlike the Export to Disk publishing method, which writes to the publish log after each asset is exported, Mirror to Server publishing waits until the entire mirror queue has been successfully mirrored before writing the results to the publish log.

7. When the publishing session is successful, WebCenter Sites concludes the session by writing information about the published references to the PubKey and PublishedAssets tables and by clearing the AssetPublishList table on the source system.
8. WebCenter Sites also writes information about the assets that were published to the ApprovedAssets and ApprovedAssetDeps tables so the approval system can calculate dependencies correctly the next time an asset is approved.

16.2 Before Configuring Mirror to Server Publishing

Before you configure the Mirror to Server publishing process, take into the consideration the topics that are presented in this section. Prior knowledge of the information you will be providing will help to ensure a smooth configuration process:

- [Section 16.2.1, "Users and Mirror to Server Publishing"](#)
- [Section 16.2.2, "CacheManager"](#)
- [Section 16.2.3, "Configuring a Mirror Destination"](#)
- [Section 16.2.4, "When to Use Mirror Destination Configuration"](#)

16.2.1 Users and Mirror to Server Publishing

In addition to the batch user account on the source system that a publish event uses to process a publishing session, the Mirror to Server publishing method requires another user account: the mirror user account, which is located on the destination server. This is the user account that the publishing system uses to unpack the mirror queue on the destination system.

When you set up a Mirror to Server publishing destination, you must create the mirror user account on the destination system that the destination represents.

16.2.2 CacheManager

The CacheManager is a WebCenter Sites servlet that maintains the page cache on any dynamic Sites system, including the management system.

CacheManager is important to the publishing system because it locks the appropriate assets on the destination during a mirror publish and ensures the integrity of the page cache both before and after a mirror publishing session.

For more information about the CacheManager, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

16.2.3 Configuring a Mirror Destination

The Mirror to Server publishing method copies information for approved assets from one Sites database to another. This information must be configured for the mirror destination. Configuring a mirror destination involves two steps: initializing the destination, and configuring the data for the destination.

Whenever you create a new mirror destination, you must initialize it before you can publish to it. To initialize a mirror destination, you specify the following information:

- The sites. Based on the sites that you select, the appropriate rows from the Publication, SitePlanTree, and PublicationTree tables are mirrored to the destination.

- Any custom table that supports or is directly related to your asset types. In other words, a table for assets that was not created by either AssetMaker or Flex Family Maker. Examples of these are `Source` and `Mimetype`. These tables are called auxiliary tables.

Mirror destination configuration is the process of indicating which data (for example, assets types, associations, and start menu items) will be mirrored to the target destination. Mirror destination configuration moves configuration data and rows from auxiliary tables—which are non-asset database tables that are used for the dynamic display of the assets—from one Sites database to another.

16.2.4 When to Use Mirror Destination Configuration

You use the Mirror Destination Configuration feature in several situations:

- To set up any new mirror destination.
- To move configuration items that support asset types (start menu items, associations, and so on) from a development system to a management system or to a delivery system.
- To move workflow configuration data from a development system to a management system.
- If you or your developers add any additional sites, asset types, or auxiliary tables to your system.
- If you or your developers add any additional categories or subtypes for existing asset types.
- When you need to troubleshoot your configuration. If you can successfully initialize a mirror destination, that means that the source and the destination systems are communicating.

16.3 Configuring Your System for Mirror to Server Publishing

The main steps when configuring a system for Mirror to Server publishing are as follows:

- [Section 16.3.1, "Create the Batch User Account \(If One Does Not Exist\)"](#)
- [Section 16.3.2, "Set Up the Destination System"](#)
- [Section 16.3.3, "Identify the Mirror User to the Source System"](#)
- [Section 16.3.4, "Identify the Local Proxy Server to the Source System \(If One Exists\)"](#)
- [Section 16.3.5, "Create a Mirror Destination"](#)
- [Section 16.3.6, "Initialize the Destination"](#)
- [Section 16.3.7, "Configure the Mirror Destination"](#)
- [Section 16.3.8, "Approve Your Assets"](#)
- [Section 16.3.9, "Publish the Assets"](#)
- [Section 16.3.10, "Test the Results"](#)
- [Section 16.3.11, "Set Up the Schedule"](#)
- [Section 16.3.12, "Turn Off Asset Invalidation on the Delivery System"](#)

Note: Configuring a system for Mirror to Server publishing requires you to provide information that is specific to your WebCenter Sites installation and your business practices. Before starting the procedures in this section, it is best to read them in order to gather and confirm the information they prompt you to provide.

16.3.1 Create the Batch User Account (If One Does Not Exist)

Note: This procedure needs to be performed only once on each source system, regardless of the number and types of publishing destinations you are configuring.

If a batch user account already exists on your source system, skip to [Section 16.3.2, "Set Up the Destination System."](#)

- If you have not already done so, create the batch user account. For instructions, see [Section 15.4.1, "Create the Batch User Account \(If One Does Not Exist\)."](#)
- If a batch user account already exists for your installation, skip to [Section 16.3.2, "Set Up the Destination System."](#)

16.3.2 Set Up the Destination System

To mirror publish assets from one Sites system to another, you must ensure that the sites and asset types are the same on both the source and the destination system. You must also create an additional user on the destination system, called the mirror user (as opposed to the batch user, which exists on the source). This user completes the mirror publish database transactions on the destination system.

Note: Database properties on the source and destination systems, if not an exact match, must be compatible, particularly database schema options (set in `futuretense.ini`). Be sure to restart the application server if you make changes to `futuretense.ini`.

To set up your destination system

1. If you have any custom support tables—a lookup table for a field in an asset type, for example—create those tables on the destination system.
2. On the destination system, create the mirror user and note the following:

Note: The mirror user has the same privileges as the RealTime user configured for RealTime publishing. If a RealTime user is configured for RealTime publishing, the same user may serve as the mirror user.

- This user must have the following ACLs:
 - Browser
 - ElementEditor
 - PageEditor
 - TableEditor

- Visitor
- VisitorAdmin
- xceladmin
- xceleditor
- Because the mirror user account is used by the CacheManager to regenerate the page cache after a publishing session, the mirror user must have sufficient privileges to regenerate all the pages in the cache. Therefore, the mirror user account must be given to all ACLs which are assigned to page entries in the SiteCatalog table or database tables that are holding data to be rendered.

Note: If any of the sample sites are installed on the destination system, a user named "mirroruser" already exists. For security, if you decide to keep this user as your mirror user, be sure to change the password for this user; or if you decide to create a different mirror user, be sure to delete the sample site "mirroruser."

- If you need help with creating the mirror user, see [Section 5.4.1, "Creating a New User."](#)

16.3.3 Identify the Mirror User to the Source System

Next, you identify the name and password of the mirror user on the destination system to the source system by setting property values in the futuretense.ini file on the source system.

Note: Because the proxy server is used by both Mirror to Server and RealTime publishing, it may have been set up for RealTime publishing. In such a case, review the properties in the steps below to ensure they are set correctly.

To identify the mirror user to the source system

1. Start the Property Editor and open the futuretense.ini file on the source WebCenter Sites system. If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.
2. In the **Export/Mirror** tab specify values for the following properties:
 - cs.mirroruser
Set this property to the name of the user that you created on the destination system in the preceding procedure.
 - cs.mirrorpassword
Set this property to the password of the user that you created on the destination system in the preceding procedure.
3. Save the property file.
4. Do one of the following:
 - If there is a firewall separating your source system from the destination system, go to [Section 16.3.4, "Identify the Local Proxy Server to the Source System \(If One Exists\)."](#)

- If there is no firewall separating your source and destination systems, close the Property Editor. Stop and restart the application server. Then go to [Section 16.3.5, "Create a Mirror Destination."](#)

16.3.4 Identify the Local Proxy Server to the Source System (If One Exists)

Note: This procedure needs to be performed only once on each source system, regardless of the number and types of publishing destinations you are configuring. Skip to [Section 16.3.5, "Create a Mirror Destination"](#) if the local proxy has already been identified to your source system, or a proxy server is not used.

To identify the local proxy to the source system:

1. Open `futuretense.ini` in the Property Editor on the source system, select the **Export/Mirror** tab, and specify values for the following properties:
 - `cs.mirrorproxyserver`
Set this property to either the name or the IP address of the local proxy server.
 - `cs.mirrorproxyserverport`
Set this property to the port number of the local proxy server.
2. Save the property file and close the Property Editor.
3. Restart the application server.

16.3.5 Create a Mirror Destination

To create a mirror destination

1. Log in to the administrator's interface on the source WebCenter Sites system.
2. In the **Admin** tab, expand **Publishing**, then **Destinations**.
3. Under **Destinations**, double-click **Add New**.
The Add New Destination form appears.

Figure 16–3 Add New Destination Form

Add New Destination

***Name:**

Delivery Type:

***Destination address:**

***Remote User:**

***Remote Password:**

Send Email on Failure:

Verbose Output:

More Arguments:

***Sites:**
 AdminSite
 FirstSite II
 Loafers Bakery
 avisports

***Roles (Approve for Publish):**
 AdvancedUser
 Analytics
 Approver
 ArtworkAuthor
 ArtworkEditor

***Roles (Publish):**
 AdvancedUser
 Analytics
 Approver
 ArtworkAuthor
 ArtworkEditor

4. Configure the fields on this form as follows:
 - **Name:** Enter a unique name for the destination.

- **Delivery Type:** Select Mirror to Server: Copy database rows to remote server.
 - **Destination Address:** Enter the URL of the remote server in the format shown. For <targetserver:port>, enter the hostname or IP address of the target system and the port to be used by the destination. A slash is required at the end of the URL because this URL is appended dynamically.
 - **Remote User:** Enter the name of the mirror user created in [Section 16.3.2, "Set Up the Destination System."](#) This user will be invoked by the publishing system to unpack the mirror queue on the target system.
 - **Remote Password:** Enter the password of the mirror user.
 - **Send Email on Failure:** If publishing fails and email notices to that effect are required, select the checkbox.
 - **Email Addresses:** Enter the recipient's email address. (This field is available only when Send Email on Failure is selected).
 - **Verbose Output:** Select this option to activate detailed error logging during the publishing process. When selected, messages in addition to error messages are written to the PubMessage table. Because additional information lengthens the publishing process, select this parameter only for troubleshooting.
 - **More Arguments:** This parameter is reserved for future use; no additional arguments may be specified at this time.
 - **Sites:** Select the sites whose assets can be approved for and published to this destination.
 - **Roles (Approve for Publish):** Select the roles to which you are assigning approval privileges. All users who are assigned these roles will be allowed to approve assets.
 - **Roles (Publish):** Select the roles to which you are assigning publish privileges. All users assigned these roles can publish.
5. Click **Add New Destination**.
- WebCenter Sites displays the Publish Destination form.

Figure 16–4 Publish Destination Form

Publish destination: Dynamic

 Inspect Edit Delete

Approve Multiple Assets

Name: Dynamic

Delivery Type: Mirror to Server: Copy database rows to remote dynamic server

Initialize Mirror Destination...

Destination address: http://[targetserver:port]/cs/

Remote User: fwadmin

Remote Password: *****

Send Email on Failure: false

Verbose Output: false

More Arguments:

Sites: FirstSite II

Roles (Approve for Publish): Any

Roles (Publish): Any

Publish Event: No existing publish event

Set Publish Event...

ID: 1328442570755

Force Approve Published Assets

[» List all Publish Destinations](#)

- If you are ready to initialize this destination system (you must initialize the destination system before you can publish to it), click the **Initialize Mirror Destination** button in the Publish Destination form, then proceed to [Section 16.3.6, "Initialize the Destination"](#) for detailed instructions (begin at step 4 of that procedure).

If you need to create more mirror destinations, repeat steps 1 through 5 for each additional mirror destination that you need to create for your source system.

16.3.6 Initialize the Destination

You must initialize the destination system before you can publish to it. This creates the basic site information on the destination system. Specifically, the Publication and PublicationTree tables are updated with the site names and asset types published to the target.

To initialize the destination system

1. In the **Admin** tab, expand **Publishing**, then **Destinations**.
2. Under **Destinations**, double-click the destination you want to initialize.
3. In the Publish Destination form, click **Initialize Mirror Destination**.

The Initialize Mirror Destination form appears

Figure 16–5 Initialize Mirror Destination Form

Initialize Mirror Destination: Dynamic

Sites: Select all sites that will be supported on this destination:

2FirstSite2
AdminSite
Burlington Financial
FirstSite II
GE Lighting

Auxiliary Tables: Add any other non-Asset tables whose data is referenced or displayed. These tables must exist on the destination before data can be mirrored over.

Source
MimeType
Category

4. Select the sites whose assets you want to publish to this destination.
5. In the **Auxiliary tables** fields, enter the names of the following tables:
 - *Source*, if you are using the source feature for any of your asset types
 - *Mimetype*, if you are using Sites DocLink, flex filter assets, the ImageFile asset type, or if you are using this table for any of your custom asset types.
 - *Filters*, if you are using flex filters on your site. (The *Filters* table lists the classes that are used by the flex filters. You will need to copy jar files and classes manually.)

- Any other auxiliary tables (such as lookup tables) for your asset types.

Note: Only five fields are provided for table names. If you have to enter more than five table names, complete this procedure through step 6 for the first five tables, then repeat steps 3–6 for the remaining tables.

6. Click **Mirror**.

If the initialization is successful, WebCenter Sites displays a confirmation message; otherwise, an error message is displayed.

Based on the sites that you selected in step 4, the corresponding rows from the `Publication`, `SitePlanTree`, and `AssetType` tables are copied to the destination.

Additionally, all the rows in the tables that you specified as auxiliary tables are copied to the destination.

Note: You can also initialize the target destination from the Publish Console: In the button bar, click **Publishing**.

1. In the **Publish Console**, select your mirror destination from the drop-down list and then click **Select Destination**.
 2. Click the **Create Site on Target** button. When the confirmation form appears, click **Yes**. This creates the basic site information on the destination system.
-

16.3.7 Configure the Mirror Destination

Now you need to configure the data that will be on your target destination. This configuration will vary depending on the purpose of the target destination. For example, if the target destination is strictly a delivery machine, it makes sense to only include asset types when mirroring. Before proceeding with configuring the destination, you may find it helpful to refer to [Section 18.1, "Migrating a Site from One System to Another."](#)

Note: If Revision Tracking is turned on at the publishing target destination for either `ElementCatalog` or `SiteCatalog`, then the publishing of template will fail and the template may become corrupted.

It is not recommended to enable Revision Tracking on the target destination.

1. In the **Admin** tab on the source system, expand **Sites** and double-click the site whose data you want to publish on the target destination.
2. In **Publish Destinations** (near the bottom of the form), click **Mirror site configuration for *destination*** (dynamic), where *destination* is the target destination.

WebCenter Sites displays the Mirror Site Configuration form:

Figure 16–6 Mirror Site Configuration Form

Mirror site configuration for: **FirstSiteII**

Publish destination: FSII Destination (dynamic)

Asset Types: enabled in FirstSiteII

- Attribute Editor
- CSElement
- Content
- Content Attribute
- Content Definition

Asset Subtypes for selected asset types

Asset Associations for selected asset types

Categories for selected asset types

Sites Desktop configuration for selected asset types

Sites DocLink configuration for selected asset types

Start Menu Items: available in FirstSiteII

- CS-Desktop Content_C
- CS-DocLink Document
- CS-DocLink Media
- Find Attribute Editor, FirstSiteII
- Find Content Attribute, FirstSiteII

Workflow Processes: available in FirstSiteII

- FSII: Approval for Artwork
- FSII: Approval for Content
- FSII: Approval for Documents
- FSII: Approval for Products
- FSII: Approval for Promotions

Workflow States used by selected workflow processes

All Workflow Actions

All Workflow Conditions

Workflow Groups: available in FirstSiteII

- FSII Home Page Translations

Workflow Group Actions

Tree Tabs: available in FirstSiteII

- Site Plan
- Admin
- Site Admin
- Workflow
- Bookmarks

Saved Searches: available in FirstSiteII

-

Roles: available in all sites

- AdvancedUser
- Analyst
- Analytics
- Approver
- ArtworkAuthor

Select the data you wish to make available on the target destination:

- **Asset Types:** Select the asset types that you want to make available on the target destination.

If your asset types have subtypes and categories, choose the appropriate options from the list in the **Asset Types** section and select the corresponding check boxes. For example, the `AssocNamed` table contains information about associations for asset types. The table will be mirrored only if you select **Asset associations for selected asset types** and then the relevant asset types.

Select the Sites Desktop / Sites DocLink configuration check boxes to configure the selected asset types for Sites Desktop and Sites DocLink.

- **Start Menu Items:** Select any start menu items that you or the developers designed for your content providers to use.
- **Workflow Processes:** If there are workflow processes for your asset types, select these processes and the appropriate workflow items.
- **Workflow Groups:** Select any workflow groups that you or the developers designed for your content providers to use.
- **Tree Tabs:** Select the tree tabs that you want to make available on the target destination.

Note: If any of the following tree tabs already exist on the target destination, mirroring the tree tabs will fail and potentially corrupt the tree tabs on the destination: Site Plan, Admin, Site Admin, Workflow, Bookmarks.

It is not recommended to mirror tree tabs that already exist on the target destination.

- **Saved Searches:** Select any saved searches that you or the developers designed for your content providers to use.
- **Roles:** Select the roles you want to exist on the target destination.

3. Click **Mirror**.

Based on which asset type configuration options you selected in step 1, appropriate rows from the `AssocNamed`, `AssocNamed_Subtypes`, and `Category` tables are copied to the destination.

If you selected **Start Menu Items** or **Saved Searches**, appropriate rows from the tables that implement those features are copied to the destination.

Based on which workflow configuration options you selected in step 1, appropriate rows from the workflow tables are copied to the destination.

4. Repeat steps 1–3 for each site whose assets you want to publish on the target destination.

16.3.8 Approve Your Assets

To truly test your published site, you must approve and publish all the assets for the site. See [Section 18.2, "Approving Multiple Assets"](#) for help with approving many assets at once.

If you want to perform a simple test of your configuration, temporarily create a home page asset with only a few dependents.

16.3.9 Publish the Assets

After you have approved assets that can be published to your destination, you can run a test publishing session:

1. On the source system, click **Publishing** in the button bar.
2. In the **Publish Console**, select your mirror destination from the drop-down list and click **Select Destination**.

WebCenter Sites displays information about the assets that are ready to be published to this destination.

3. If for some reason the target destination has not yet been initialized, it must be initialized now or the publishing session will fail. To initialize the destination, click **Create Site on Target** in the **Publish Console**. This creates the basic site information on the destination system.
4. Click **Publish**.

A confirmation message appears. Click **Yes** to proceed with the publish.

The publishing system mirrors all the approved assets for this destination to the WebCenter Sites database on the destination system.

5. In the **Publish Console** Active tab you can observe that the session is running.
6. When the session completes, click the **History** tab to view the session summary for this session.

Mouse over the session information to view summary information; click **View Log** to view the session log files. For information on configuring the logs to view destination information, see [Section 16.5, "Retrieving Logs From Delivery WebCenter Sites Systems."](#)

16.3.10 Test the Results

To test the results, point your browser to the home page asset on the destination system and examine the site.

If you have not already done so, you must determine the asset's URL. A WebCenter Sites URL is constructed by concatenating the following values:

- The hostname or IP address (and sometimes the port number) of the destination system.
- The CGI path, which the system obtains from the `ft.cgipath` property in the `futuretense.ini` file. For example, for WebLogic and other application servers with servlet architectures, this default path is `/servlet/`.
- The string `ContentServer?pagename=`
- A page name from a SiteCatalog entry.
- Additional information that is passed in with the WebCenter Sites page criteria variables, `c`, `cid`, `tid`, and `p` (see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information about these variables).

Complete the following steps to determine the URL of your home page and test the site:

1. Start the Property Editor and open the `futuretense.ini` file for the destination system. (If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.)

2. In the **App Server** tab, locate the `ft.cgi.path` property and write down its value.
3. In the **Compatibility** tab, locate the `ft.approot` property and write down its value.
4. Save the property file and close the Property Editor.
5. Open a text editor. Enter the server name, a slash (/), and then the `cgipath`. Precede the server name with the proper protocol—`http://` or `https://`—as in the following example:

WebLogic and WebSphere:

```
http://bigfatsun.example.com:8080/servlet/
```

6. At the end of the string, type a slash, and then add the following text:

```
ContentServer?pagename=your_home_page
```

Now the URL should look similar to the following examples:

Example for WebLogic and WebSphere:

```
http://bigfatsun.example.com:8080/servlet/
ContentServer?pagename=ExampleSite/Page/Home
```

7. Point your browser to the URL you assembled.
8. Scan the page for errors and test all links to ensure they work.

Note: It is recommended that you conduct a complete test of the system under peak load conditions after you have mirrored the entire site for the first time, and at regular points thereafter.

16.3.11 Set Up the Schedule

After you have ensured that your destination is configured correctly, you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see [Section 18.7, "Scheduling Publish Events."](#)
- If you are using images that are not assets in the design of your site—that is, your site designers want to store all images on the web server rather than manage them as assets—plan how you will move the image files from the web server for the management system to the web server for the delivery system. For example, you can set up a regular FTP transfer.
- If you are using elements and SiteCatalog page entries that are not CSElement and SiteEntry assets, you must use the CatalogMover tool to mirror them to the destination system. For help with CatalogMover, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

16.3.12 Turn Off Asset Invalidation on the Delivery System

By default, the publishing system is configured to mark an asset as changed on the destination system when you publish an asset from one system to another (source to destination). Then, the newly-published asset must be approved on the destination system before it can be published to another destination.

The default configuration is appropriate for development and management systems. However, when you are publishing to the delivery system, there is no need for the publishing system to take the time to mark the change—assets are published to the delivery system but not from it.

Therefore, on the delivery system, turn off this publishing feature by completing the following steps:

1. Start the Property Editor and open the `futuretense_xcel.ini` file. If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.
2. In the **Publishing** tab, locate the `xcelerate.publishinvalidate` property and set its value to `false`.
3. Save the file and close the Property Editor.

16.4 Approval Query for Performance Enhancement

Dynamically publishing thousands of assets takes a long time because of the time required to compute which assets are publishable for the given destination. The query is executed twice: When the user selects the publishing destination and when the user clicks the Publish button to publish the assets. The current query is also complex, given that it is written to work for both mirror and static publishing. To improve publishing performance, we simplified the query by considering only mirror publishing. The same query is used to improve the performance of RealTime publishing. The simplified query, with comments, looks as follows:

```
select pkt.localkey, pkt.urlkey from PubKeyTable pkt JOIN
  ApprovedAssets aa ON pkt.assetid=aa.assetid -- match asset id's
  WHERE aa.targetid=<targetid>
  AND pkt.targetid=<targetid> -- match target id's
  AND aa.tstate='A' -- asset must be approved
  AND (aa.lastassetdate IS NULL OR aa.assetdate > aa.lastassetdate) -- compare
  assetdate to lastassetdate
```

To implement the approval query

1. On the source WebCenter Sites system, create the file `Shared/Storage/pubquery.txt` and paste in the following query (same as above, but without comments):

```
select pkt.localkey, pkt.urlkey from PubKeyTable pkt JOIN
  ApprovedAssets aa ON pkt.assetid=aa.assetid
  WHERE aa.targetid=__TARGETID__
  AND pkt.targetid=__TARGETID__
  AND aa.tstate='A'
  AND (aa.lastassetdate IS NULL OR aa.assetdate > aa.lastassetdate)
```

2. Restart your source system and enable database debugging to verify that the query runs during mirror and RealTime publishing sessions. (To enable database debugging, go to the Admin tab, expand System Tools, and double-click **Configure log4j**. After making changes there, system restart is not necessary.)
3. Inspect the logs to verify that the query ran during publishing and that all publishable assets were published correctly.
4. Resume normal operations (disable database debugging by setting `com.fatwire.logging.cs.db` to its normal value and then restart the system).

16.5 Retrieving Logs From Delivery WebCenter Sites Systems

WebCenter Sites provides a way to retrieve publishing-related logging information from the delivery system and display it in the Publishing console. To achieve this, WebCenter Sites inserts the publishing session ID into each relevant log entry. The management interface has the ability to retrieve the relevant entries for a publishing session and display them in the Log form for that session.

To enable remote publishing logging

1. Activate the logger by setting the following property in `commons-logging.properties` on both source and target:

```
org.apache.commons.logging.LogFactory=com.fatwire.cs.core.logging.ContextAwareLogFactory
```

2. Add the following lines to your `web.xml` on the target system:

```
<filter>
  <filter-name>ContextHeaderFilter</filter-name>
  <filter-class>
    com.fatwire.cs.core.logging.context.filter.ContextHeaderFilter
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>ContextHeaderFilter</filter-name>
  <servlet-name>ContentServer</servlet-name>
</filter-mapping>
```

3. On the target system, set the following `futuretense.ini` property:

```
log.file.location=<path to Sites log file>
```

The Export Assets to XML Publishing Process

This chapter explains how Export Assets to XML works and how it must be configured.

This chapter contains the following sections:

- [Section 17.1, "The Export Assets to XML Publishing Method"](#)
- [Section 17.2, "Configuring Your System for Export Assets to XML"](#)

17.1 The Export Assets to XML Publishing Method

The Export Assets to XML publishing method is a hybrid between the Export to Disk and dynamic publishing methods. Assets are rendered into files, but this publishing method uses the dynamic dependency calculation method rather than the method of Export to Disk publishing.

The Export Assets to XML publishing method differs from the other two in that it is really a data transformation method. While Export to Disk publishing creates one HTML file per page, which could mean that several assets are rendered into one file, Export Assets to XML creates one XML file for each asset that is approved to a destination configured to use this publishing method.

Export to Disk publishing creates Export to Disk files that are to be delivered from a Export to Disk Sites delivery system. In contrast, Export Assets to XML creates XML files for delivery to a database or non-Sites system.

When an Export Assets to XML publishing session runs, this is what happens:

1. On the source system, the approval system provides a list of the assets approved for this destination to Export Assets to XML.
2. Export Assets to XML renders each asset in the list to an XML file. The output file describes the asset, stating all the values for all of the asset's fields.

When you use the Export Assets to XML publishing method, typically you set up a quality assurance process to test the generated files before you move them to the external (non-Sites) system.

This section contains the following topics:

- [Section 17.1.1, "The XML Output"](#)
- [Section 17.1.2, "File Naming Conventions for Export Assets to XML"](#)

17.1.1 The XML Output

The output from Export Assets to XML is well-formed XML that describes an asset object in terms of its field values. Fields are described as attributes.

For each of the asset's fields, there is a name/value pair. The statement of the value includes the data type of the field. For example, the Name field holds characters. So the name/value pair for an asset's Name field would appear as follows in the output:

```
<attribute name="Name">
  <string value="nameOfAsset"/>
</attribute>
```

Flex attributes serve as the fields for flex assets. In the output XML for a flex asset, the flex attribute values are prepended with `Attribute_`. Here's an example of an attribute value for the price of a sample site product asset:

```
<attribute name="Attribute_price">
  <decimal value="5.9"/>
</attribute>
```

Note that the XML output uses the column names rather than the field name of each field/attribute and that column names and field names can be different.

The following is a description of the XML file created by this publishing method, presented as a pseudo-dtd file

```
<!-- ASSET:
-- ASSET defines an asset object
-- an asset is made up of attributes
-->
<!ELEMENT ASSET (ATTRIBUTE)*>
<!ATTLIST ASSET TYPE CDATA #REQUIRED>
<!ATTLIST ASSET ID CDATA #IMPLIED>
<!ATTLIST ASSET SUBTYPE CDATA #IMPLIED>

<!ELEMENT ATTRIBUTE (STRING | DATE | INTEGER | DECIMAL | BINARY | FILE |
ASSETREFERENCE | ARRAY | STRUCT | LIST)>
<!ATTLIST ATTRIBUTE NAME CDATA #REQUIRED>

<!ELEMENT STRING>
<!ATTLIST STRING VALUE CDATA #REQUIRED>

<!ELEMENT DATE>
<!ATTLIST DATE VALUE CDATA #REQUIRED>

<!ELEMENT INTEGER>
<!ATTLIST INTEGER VALUE CDATA #REQUIRED>

<!ELEMENT DECIMAL>
<!ATTLIST DECIMAL VALUE CDATA #REQUIRED>

<!ELEMENT BINARY>
<!ATTLIST BINARY VALUE CDATA #REQUIRED>

<!ELEMENT FILE (CDATA) >
<!ATTLIST FILE NAME CDATA #REQUIRED>

<!ELEMENT FILE>
<!ATTLIST FILE NAME CDATA #REQUIRED>

<!ELEMENT ASSETREFERENCE>
```

```

<!ATTLIST ASSETREFERENCE TYPE CDATA #REQUIRED>
<!ATTLIST ASSETREFERENCE VALUE CDATA #REQUIRED>

<!ELEMENT ARRAY (STRING | DATE | INTEGER | DECIMAL | BINARY | FILE |
ASSETREFERENCE | ARRAY | STRUCT | LIST)+>

<!ELEMENT STRUCT (FIELD)+>

<!ELEMENT FIELD (STRING | DATE | INTEGER | DECIMAL | BINARY | FILE |
ASSETREFERENCE | ARRAY | STRUCT | LIST)+>
<!ATTLIST FIELD NAME CDATA #REQUIRED>

<!ELEMENT LIST (ROW)+>

<!ELEMENT ROW (COLUMN)+>

<!ELEMENT COLUMN ( STRING | INTEGER | DECIMAL | DATE | ASSETREFERENCE)>
<!ATTLIST COLUMN NAME CDATA #REQUIRED>

```

17.1.2 File Naming Conventions for Export Assets to XML

Exported XML files are named according to the following convention:

```
assetID.xml
```

For example, if the asset ID is 3344556677, the file name for the asset's XML file is:
3344556677.xml

The Export Assets to XML publishing method does not use any information that is entered in an asset's Path or Filename field.

17.2 Configuring Your System for Export Assets to XML

The main steps when configuring for the Export Assets to XML delivery type are as follows:

- [Section 17.2.1, "Create the Batch User Account \(If One Does Not Exist\)"](#)
- [Section 17.2.2, "Specify the Root Export Directory"](#)
- [Section 17.2.3, "Configure an Export Assets to XML Destination"](#)
- [Section 17.2.4, "Approve Your Assets"](#)
- [Section 17.2.5, "Publish and Test the Results"](#)
- [Section 17.2.6, "Set Up the Schedule"](#)

17.2.1 Create the Batch User Account (If One Does Not Exist)

Creating a batch user account is required before any type of publishing can take place. This procedure needs to be performed only once, regardless of the number and types of publishing destinations you are creating.

- If you have not already done so, create the batch user account. For instructions, see [Section 15.4.1, "Create the Batch User Account \(If One Does Not Exist\)."](#)
- If a batch user account already exists for your installation, skip to [Section 17.2.2, "Specify the Root Export Directory."](#)

17.2.2 Specify the Root Export Directory

As mentioned earlier in this chapter, the directory to which WebCenter Sites exports files is determined by the `cs.pgexportfolder` property in the `futuretense.ini` file.

To set the root directory for exported files

1. Start the Property Editor. If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.
2. Open the `futuretense.ini` file on the source WebCenter Sites system.
3. On the **Export/Mirror** tab select the `cs.pgexportfolder` property and set its value to the full path of the directory to which files should be exported. This is a global setting for the system.

Save the file and close the Property Editor.

4. Restart the application server.

Note: Before you run an Export to XML publishing session, make sure that this destination directory exists and that it has enough space for the XML files that will be created there.

17.2.3 Configure an Export Assets to XML Destination

Next, create the Export to XML destination by completing the following steps:

1. In the **Admin** tab on the source WebCenter Sites system, expand **Publishing**, then **Destinations**.
2. Double-click **Add New**.

The Add New Destination form appears:

Figure 17-1 Add New Destination Form

Add New Destination

*Name:

Delivery Type:

Send Email on Failure:

Verbose Output:

More Arguments:

*Sites:
 2FirstSite2
 AdminSite
 Burlington Financial
 FirstSite II
 GE Lighting

*Roles (Approve for Publish):
 AdvancedUser
 Analyst
 Analytics
 Approver
 ArtworkAuthor

*Roles (Publish):
 AdvancedUser
 Analyst
 Analytics
 Approver
 ArtworkAuthor

Fill in the fields as described below:

- **Name:** Enter a unique name for the destination.
 - **Delivery Type:** Select Export Assets to XML: Export XML file for each asset.
 - **Sites:** Select the sites whose assets can be approved for and published to this destination.
 - **Roles (Approve for Publish):** Select the roles to which you are assigning asset approval privileges. All users who are assigned these roles can approve assets.
 - **Roles (Publish):** Select the roles to which you are assigning publish privileges. All users assigned these roles can publish.
3. Repeat step 1 for each additional export destination that you need to configure for your source system.

17.2.4 Approve Your Assets

If you want to perform a simple test of your configuration, select an asset and approve it and any of its dependent assets.

If you want to complete a test publish of your entire site, approve all the assets for the site. See [Section 18.2, "Approving Multiple Assets"](#) for help with approving many assets at once.

17.2.5 Publish and Test the Results

After you have approved assets that can be published to your destination, you can run a test publishing session:

1. On the source WebCenter Sites system, click **Publishing** in the button bar.
2. In the **Publish Console**, select your destination from the drop-down list and then click **Select Destination**.

WebCenter Sites displays information about the assets that are ready to be published to this destination.

3. Click **Publish**.

A confirmation message appears. Click **Yes** to proceed with the publish.

WebCenter Sites exports all assets approved for this destination to files. The files are stored in the directory you specified earlier.

4. In the **Publish Console** Active tab you can observe that the session is running.
5. When the session completes, click the **History** tab to view the session summary for this session.

Mouse over the session information to view summary information; click **Log** to view the session log files.

6. The log file displays a list of all the resulting XML files, verify that the files are correct.

17.2.6 Set Up the Schedule

After you have ensured that your destination is configured correctly, that the directory names are created according to your needs, you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see [Section 18.7, "Scheduling Publish Events"](#).
- Plan how you will move the generated files to your external, non-Sites systems. For example, you can set up a regular FTP transfer that automatically moves files to a testing area after a publishing session completes.

Additional Publishing Procedures

In addition to the configuration steps described in the preceding sections of this chapter, administrators and developers also perform, whenever necessary, the publishing procedures described in this chapter.

This chapter contains the following sections:

- [Section 18.1, "Migrating a Site from One System to Another"](#)
- [Section 18.2, "Approving Multiple Assets"](#)
- [Section 18.3, "Creating Destinations"](#)
- [Section 18.4, "Editing Destinations"](#)
- [Section 18.5, "Deleting Destinations"](#)
- [Section 18.6, "Creating Export Starting Points"](#)
- [Section 18.7, "Scheduling Publish Events"](#)
- [Section 18.8, "Reading the Schedule Abbreviations"](#)
- [Section 18.9, "Editing Publish Events"](#)
- [Section 18.10, "Overriding the Schedule"](#)
- [Section 18.11, "Assigning Approval or Preview Templates"](#)
- [Section 18.12, "Monitoring Sessions in the Publishing Console"](#)
- [Section 18.13, "Verifying Publishing Readiness"](#)
- [Section 18.14, "Managing Publishing History Information"](#)
- [Section 18.15, "Publishing All Approved Assets"](#)
- [Section 18.16, "Troubleshooting"](#)

Note: The following feature is deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Mirror publishing. This feature is replaced by RealTime publishing.

18.1 Migrating a Site from One System to Another

During the development phase for your sites, your site designers and developers develop sites and asset types with all their supporting configuration (subtypes, associations, start menu items, and so on), code templates, and assist the administrators in customizing the user interface, when necessary, by writing elements

for workflow and so on. Your site designers and developers do this work on the development system.

When the site is ready for the content providers, you can use the Mirror Destination Configuration feature to migrate it to the management system and the delivery system.

This section contains the following topics:

- [Section 18.1.1, "Moving a Site from a Development System to a Management System"](#)
- [Section 18.1.2, "Moving a Site to a Delivery System"](#)

18.1.1 Moving a Site from a Development System to a Management System

Complete the following procedure once:

1. Log in to the WebCenter Sites interface on the management (destination) system as the **fwadmin** user.
2. If you have any custom support tables—a lookup table for a field in an asset type, for example—create those tables on the management (destination) system.
3. On the management system, create all the users.
4. On the source system, log in to the WebCenter Sites interface.
5. Create a **Publishing Destination** for the destination system that uses the RealTime publishing method.

For help with this step and the following two steps, see [Section 20.2, "Configuring Your System for RealTime Publishing."](#)

6. Initialize the publishing destination.
7. Configure the publishing destination.
8. On the destination (management) system, configure the users for the site. For help with this step, see [Section 8.2.1, "Granting Users Access to a Site \(Assigning Roles to Users\)."](#)
9. If you created a flex family, log back in to the source system. Approve and then publish all of the data structure flex assets to the management (destination) system.

You can either approve each asset individually or use the **Approve Multiple Assets** feature on the **Admin** tab. For information about this feature, see [Section 18.2, "Approving Multiple Assets."](#)

10. Approve and then publish all of the rest of the appropriate assets for the destination system: template, page, collection, CSElement, and SiteEntry assets, and so on.
11. If there are any elements or page entries in the SiteCatalog table that are not CSElement or SiteEntry assets, use CatalogMover to mirror those items to the management system. For help with CatalogMover, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

18.1.2 Moving a Site to a Delivery System

When you move a site to a delivery system, there is no need to mirror start menu items, workflow, saved searches, and so on.

Complete the following steps:

1. Log in to the WebCenter Sites interface on the delivery (destination) system as the **fwadmin** user.
2. If you have any custom support tables—a lookup table for a field in an asset type, for example—create those tables on the delivery (destination) system.
3. On the source system, log in to the WebCenter Sites interface.
4. Create a **Publishing Destination** for the destination system that uses the Mirror to Server publishing method.
For help with this step, see [Section 16.3.5, "Create a Mirror Destination."](#)
5. Initialize the mirror destination.
For help with this step, see [Section 16.3.6, "Initialize the Destination."](#)
6. Configure the mirror destination.
For help with this step, see [Section 16.3.7, "Configure the Mirror Destination."](#)
Do **not** select start menu items, saved searches, or any of the workflow options.
7. If you created a flex family, log back in to the source system. Approve and then publish all of the data structure flex assets to the management (destination) system.
You can either approve each asset individually or use the **Approve Multiple Assets** feature on the **Admin** tab. For information about this feature, see the [Section 18.2, "Approving Multiple Assets."](#)
8. Approve and then publish all of the rest of the appropriate assets to the destination (delivery) system: template, page, collection, query, CSElement, and SiteEntry assets, and so on.
9. If there are any elements or page entries in the SiteCatalog table that are not CSElement or SiteEntry assets, use CatalogMover to mirror those items to the management system. For help with CatalogMover, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
10. If there are any images in the online site that are not assets, be sure to copy them to the delivery system.

18.2 Approving Multiple Assets

Each publishing destination has an option to Approve Multiple Assets. It is especially useful for upgrades and for the first publishing session to a destination.

Note that the Approve Multiple Assets feature is not the BulkApprover utility. The BulkApprover utility approves only those assets that have been imported into the WebCenter Sites database with the BulkLoader utility. Both BulkLoader and BulkApprover are described in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

To approve a group of assets

1. In the Admin tab, expand **Publishing**, then expand **Destinations**, and then expand the destination that you want to approve assets for.
2. Under that destination, double-click the **Approve Multiple Assets** option.

The Approve Assets form appears:

Figure 18–1 Approve Assets for Publish to Destination Form

Approve Assets for Publish to Destination 1 (static)

The screenshot shows a web form titled "Approve Assets for Publish to Destination 1 (static)". It features several sections:

- *Asset Types:** A list box containing "Article", "Article Flex", "Attribute Editor", "CSElement", "Collection", "Content", "Content Attribute", and "Content Attribute".
- Sample queries:** A list box containing "-Select sample query, if needed-", "Assets of selected asset type(s).", "Assets of selected asset type(s) in site:FirstSite1", "Assets of selected asset type(s) updated before a date", and "Placed Page assets".
- *Query:** A large empty text box for entering a custom query.
- Approve in Batches of:** A text input field with the value "500".
- Approve Previously Approved Assets:** Two radio buttons, "Yes" and "No", with "No" selected.
- Buttons:** "Cancel" and "Approve for Publish" buttons at the bottom.

3. In the **Asset Types** section, from the list on the left, select the asset types that you want to approve.
4. From the **Sample Queries** list on the right, select a query. If the exact query that you need is not present, select the query that is most like the one that you want.
WebCenter Sites creates a SQL query based on the items that you selected and displays it in the **Query** box.
5. (Optional) If necessary, edit the SQL query.
6. Click in the **Approve in Batches of** field and enter a numeric value. The default is set to 500.

Note: When you click the Approve for Publish button, the approval system approves batches of assets. The number of assets in each batch is determined by value in the Approve in Batches of field. Because the approval process is not a background process, it is possible that the browser session could time out while the batch is being approved.

When using the Approve Multiple Assets feature, to ensure that the session does not time out, adjust the following settings:

- The `cs.timeout` property in the `futuretense.ini` file, which sets the browser session timeout value. Start by setting this value to 1800 (which means 30 minutes). (See the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.)
 - The value you specify in the **Approve in Batches of** field. Start by using the default of 500 and lower it if necessary.
-

7. Specify the **Approve Previously Approved Assets** option as follows:

- If you want WebCenter Sites to ignore all previous approvals and re-approve all selected assets, select **Yes**. This is the option to use if previous approvals may have become invalid for some change such as a change in default templates for that destination.
 - If you want WebCenter Sites to approve only the assets that have not yet been approved, select **No**. This is the correct choice when you are sure that previously approved assets are still valid.
8. Click **Approve for Publish**.

18.3 Creating Destinations

The procedures for creating destinations are described in the previous section. For information about creating new destinations, use one of the procedures in the following list, as appropriate:

- To create an export destination, see [Section 15.4.3, "Configure an Export Destination."](#)
- To create a mirror destination, see [Section 16.3.5, "Create a Mirror Destination."](#)
- To create an export assets destination, see [Section 17.2.3, "Configure an Export Assets to XML Destination."](#)

18.4 Editing Destinations

To edit a destination

1. In the Admin tab, expand **Publishing**, then expand **Destinations**.
2. Double-click on the destination that you want to edit.

The Publish Destination form appears for the selected destination.

Figure 18–2 Publish Destination Form

Publish destination: Destination 2 (dynamic)

 [Inspect](#)  [Edit](#)  [Delete](#)

[Approve Multiple Assets](#)

Name:  Destination 2 (dynamic)

Delivery Type: [Mirror to Server](#): Copy database rows to remote dynamic server

[Initialize Mirror Destination...](#)

Destination address: http://[targetserver]/cs/

Remote User: [user]

Remote Password: *****

Send Email on Failure: false

Verbose Output: false

More Arguments:

Sites: Burlington Financial
GE Lighting

Roles (Approve for Publish): Any

Roles (Publish): Any

Publish Event: No existing publish event

[Set Publish Event...](#)

ID: 1329314485736

[Force Approve Published Assets](#)

[» List all Publish Destinations](#)

3. Click the **Edit** icon, and then make the appropriate changes.

For information about the values that can be entered in the fields in this form, see the delivery-type specific procedures in the previous sections:

- For an Export to Disk destination, see [Section 15.4.3, "Configure an Export Destination."](#)

- For a Mirror to Server destination, see [Section 16.3.5, "Create a Mirror Destination."](#)
- For an Export Assets to XML destination, see [Section 17.2.3, "Configure an Export Assets to XML Destination"](#).

18.5 Deleting Destinations

To delete a destination

1. In the Admin tab, expand **Publishing**, then expand **Destinations**.
2. Double-click on the destination that you want to delete.
3. Click the **Delete** icon.
WebCenter Sites displays a verification message.
4. Click **Delete Destination**.

18.6 Creating Export Starting Points

Export starting points are defined in [Section 15.3.5, "Export Starting Points"](#). For information about creating them, see [Section 15.4.5, "Create the Export Starting Points"](#).

18.7 Scheduling Publish Events

You can set up publishing events for all of your publishing destinations. Because publish events run as background processes and there is no publish queue, you can set up publish events for different destinations that run at the same time, if necessary.

Note the following:

- There can be only **one** publish event schedule for **each** destination.
- If any of your systems serve as both source and destination, be sure that incoming and outgoing publishing sessions do not overlap. For example, you publish from the development system to the management system and you publish from the management system to the delivery system. Never schedule a publish event from the development system to the management system for a time when the management system is publishing to the delivery system.

To schedule a publish event

1. In the Admin tab, expand **Publishing**, then expand **Destinations**.
2. Double-click on the destination that you want to schedule a publish event for.
3. In the Publish Destination form, click the **Set Publish Event** button.

WebCenter Sites displays the Edit Publish Event form:

Figure 18–3 Edit Publish Event for Destination Form**Edit Publish Event for Destination: Destination 2 (dynamic)**

Enabled Dates and Times

Hours: All	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled
Minutes: Every 5 Minutes of each selected hour	
Dates: All	
Days: All	
Months: Monthly	

Recurrence Pattern

Select: **All** | None

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
--------	--------	---------	-----------	----------	--------	----------

Information:
Event will run only on the selected days of the selected months.
Note: Selecting none and selecting all both have the same effect on the schedule.

Times of Recurrence

Hours												Minutes			
12	1	2	3	4	5	6	7	8	9	10	11	AM	0	5	10
12	1	2	3	4	5	6	7	8	9	10	11	PM	15	20	25
													30	35	40
													45	50	55

Information:
Event will run only on the selected minutes during the selected hours.

Now you select days, months, hours, and minutes to set a schedule.

The form for setting a publish event is similar to the form for setting a timed action event in workflow. See [Section 10.3.4, "Setting Up the Timed Action Event"](#) for more information.

Example Schedule

For example, if you want to set a schedule so that WebCenter Sites publishes the assets approved for this destination at 7 a.m., 4 p.m., and 8 p.m. every day of the week except Sunday, you would complete these steps:

1. In the **Recurrence Pattern**, **Days of the week** tab, click on **Monday** and drag to **Saturday** to select all days of the week except Sunday. The days highlight in gray as you select them.

Figure 18–4 Days of the Week Tab

Days of the week	Days of the month	Months				
Select: All None						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

2. Skip the **Days of the month** tab, since you have already selected the days you want.
3. In the **Months** tab, click **All** to select all months. All months become highlighted in gray.

Figure 18–5 Months Tab

Days of the week	Days of the month	Months			
Select: All None					
Jan	Feb	Mar	Apr	May	Jun
Jul	Aug	Sep	Oct	Nov	Dec

4. In the **Times of Recurrence** section, **Hours** selector, click on **7am**, **4pm**, and **8pm**.
5. In the **Minutes** selector, click on 0.

Figure 18–6 Hours and Minutes Section

Hours												Minutes			
12	1	2	3	4	5	6	7	8	9	10	11	AM	0	5	10
12	1	2	3	4	5	6	7	8	9	10	11	PM	15	20	25
													30	35	40
													45	50	55

6. Double-check the dates and times selected in the **Enabled Dates and Times** section of the form.

It should look like the sample shown below:

Figure 18–7 Schedule Detail

Hours: 7AM, 4PM, 8PM
Minutes: at :00 of each selected hour
Dates: Any
Days: Monday-Saturday
Months: Monthly

7. In the Enabled Dates and Times section, click **Enabled**.
8. Click the **Save** button.

A summary of the schedule now appears in the **Publish Event** section of the Publish Destination form.

18.8 Reading the Schedule Abbreviations

Once scheduled events are created, these events are displayed in string format in the WebCenter Sites interface.

In the Edit Publish Events form the schedule is displayed in easy-to-read format on the Enabled Dates and Times section at the top of the page. In the Publish Destination forms, the schedule is displayed in the Publish Event field as a string.

Here is the key to the schedule string:

hours:minutes:seconds weekdays/days of month/months of year

This information is displayed as follows:

- Hours are displayed as numbers from 0 (midnight) through 23 (11 p.m.) and are separated from the minutes with a colon.

When multiple hours are selected, the selected hours display with commas separating them.

In the previous example (see ["Example Schedule"](#) on page 18-8), 7 a.m., 4 p.m., and 8 p.m. were set as the times to publish. This displays as 7,16,20:

- Because you can set minutes in increments of 5, minutes are displayed as numbers 0 – 55 in five minute increments. Minutes are separated from the seconds with a colon.

If the schedule describes more than one minutes increment, the scheduled increments are listed with commas separating them.

When minutes are displayed as a 0, it means the event runs on the hour.

The hours and minutes for the example are displayed like this: 7,16,20:0:

- Seconds are always set to zero. Therefore, the complete expression of the time in the display of the example is: 7,16,20:0:0
- The time and the days are separated with a space.
 - Days of the week are expressed as numbers 0 (Sunday) through 6 (Saturday) and end with the slash (/) character.

- If either no days or all days were selected, the publish event runs every day of the week. An asterisk (*) character appears in the field in that case.
- If more than one day is scheduled, the selected days are displayed with commas separating them.

The example schedule includes all the days of the week except Sunday. Therefore, the days of the week are appended to the display as follows: 7,16,20:0:0 1,2,3,4,5,6/

- Days of the month are also displayed as numbers (1–31) and end with the slash (/) character.
 - The list of days is separated with a comma.
 - If no days of the month were selected, the publishing event runs every day of the month. An asterisk (*) character appears in the field in that case.
 - If you specify both days of the week and days of the month, the event runs when either setting matches the current day.

The example schedule does not specify days of the month. Therefore, the value for this item is appended to the display as an asterisk: 7,16,20:0:0 1,2,3,4,5,6/* /

- At the end of the string are the months, displayed as numbers from 1 (January) through 12 (December).
 - The list of months is separated with a comma.
 - If all the months are selected, the publishing event will run every month. An asterisk (*) character appears in the field in that case.

The example schedule specified all the months. Therefore, the final result is:

7,16,20:0:0 1,2,3,4,5,6/* /*

18.9 Editing Publish Events

To edit a publish event

1. In the Admin tab, expand **Publishing**, then expand **Destinations**.
2. Double-click on the destination that you want to schedule a publish event for.
3. In the Publish Destination form, click **Set Publish Event**.

WebCenter Sites displays the Publish Event form. The text at the top of the form describes how the event is currently configured.

4. Edit the event as needed and click **Schedule**. To clear your selections, click **Cancel**.

18.10 Overriding the Schedule

If you want to start a publishing session immediately, complete the following steps:

1. In the WebCenter Sites interface, click the **Publishing** button in the toolbar.
2. In the **Publishing Console**, from the drop-down list, select the destination that you want to publish to.
3. Click the **Select Destination** button.

WebCenter Sites determines whether there are any assets that can be published to the destination and displays a summary form like this one:

Figure 18–8 Publish Destination Form**Publish destination: FSII Destination (RealTime)****Destination:**  FSII Destination (RealTime) using RealTime**Destination address:** http://localhost:8380/cs/

Arguments: REMOTEUSER=fwadmin
 VERBOSE=false
 PUBLISHOPTION=Complete
 EMAILALERT=false
 REMOTEPASS=*****
 371 assets are ready for publish.

Configure Destination...**Edit Publish Schedule...****Publish****Cancel**

If there are assets that can be published to the destination, the **Publish** button is displayed. If there are no assets to publish, there is no **Publish** button.

4. Click Publish.

WebCenter Sites starts the publishing session.

Note that if a publishing session for this destination is already underway, the publish event that you just tried to run fails and WebCenter Sites displays a status message describing that another session for the same destination is in progress. If you still want to run this event, wait until the current session completes and then repeat this procedure.

18.11 Assigning Approval or Preview Templates

When assets are approved for a Export to Disk publishing destination, the approval system examines the template assigned to the asset to determine its dependencies.

However, when the system publishes the asset, it does not necessarily use the template that is assigned to the asset. This is because the code in another element could determine that a different template is used for that asset in certain cases.

Consider the sample site. An article asset from this sample site can be rendered by several different templates, depending on the context.

So when someone approves an article asset for the site, which template should the approval process use to determine the dependencies for the article? The one that contains the most representative set of dependencies for all of the templates.

What if the template that contains the most representative set of dependencies is not the template that you want to assign to the asset? Set it as the Default Approval Template for assets of that type.

Note that when you assign default templates to assets that are published to mirror destinations, those templates are not used to calculate dependencies, but they are used when someone previews the asset from the Status form.

To set a default template

1. In the **Admin** tab, expand **Publishing**, then expand **Destinations**.
2. Expand the destination that you want to configure default templates for and then select the **Set Default Templates** item.
3. WebCenter Sites displays a Default Templates form.
4. Click **Edit Configuration**.

The system displays a form similar to this one:

Figure 18–9 Default Templates Form

Default templates for: Destination 1 (static)

Asset Type	Subtype	Template	Wrapper page
Article	Columnist	– None specified; use asset's template field –	– None –
	Standard	– None specified; use asset's template field –	– None –
Article Flex	DrillHierarchyCT	– None specified; use asset's template field –	– None –
	Story	– None specified; use asset's template field –	– None –
CSElement	(no subtype)	– None specified; use asset's template field –	– None –
Collection	Article	– None specified; use asset's template field –	– None –
	AArticles	– None specified; use asset's template field –	– None –
	Collection	– None specified; use asset's template field –	– None –
	Content_C	– None specified; use asset's template field –	– None –

5. For each asset type that you need to configure, select the template and wrapper page that you want WebCenter Sites to use as the approval or preview template for assets of that type. If there are subtypes configured for an asset type, you can specify a default template for each subtype of that asset type.
6. Click the **Save** button.

You are returned to the Default Templates form for that destination.

18.12 Monitoring Sessions in the Publishing Console

You use the Publish Console to monitor publishing sessions. To open the Publish Console, click the Publishing button at the top of the WebCenter Sites interface:

Figure 18–10 Publish Console Form

Publish Console

Publish destination FSII Destination (RealTime) (using RealTime) ▼

Select Destination

Active (0) Scheduled (1) History (2)

No Running Publish Sessions

The console displays a summary of all the publishing activity, including any sessions that are currently running, scheduled, or completed.

18.13 Verifying Publishing Readiness

To determine whether a publishing session is ready or whether there are any problems with unapproved assets

1. In the Publish Console, select the destination from the drop-down list.
2. Click the **Select Destination** button.

A summary form appears. It lists any assets that need to be published but are not ready (they are held) and assets that can be published. For example:

Figure 18–11 Publish Destination Form**Publish destination: AviSports Destination (RT)**

Destination: AviSports Destination (RT) using RealTime

Destination address: `http://localhost:8280/cs/`

Arguments: REMOTEUSER=fwadmin
VERBOSE=false
EMAILALERT=false
REMOTEPASS=*****
PUBLISHOPTION=Complete
[19 assets are held for publish.](#)
[80 assets are ready for publish.](#)

3. To see the details, click on the link.

WebCenter Sites displays lists of the assets that are either approved or approved and still held for dependency reasons. On one tab is the list of Approved Assets, and the other tab are the assets that are in the On-Demand Queue.

For example:

Figure 18–12 Publish Destination Form

Publish destination: Avisports Destination (RT)

Destination: Avisports Destination (RT) using RealTime

Arguments:
 REMOTEUSER=fwadmin
 VERBOSE=false
 EMAILALERT=false
 REMOTEPASS=*****
 PUBLISHOPTION=Complete

423 assets available for Publishing
 Show 10 20 30 50 items per page

<input type="checkbox"/>	Type	Name	Locale	Start Date	End Date	Days Expired
<input type="checkbox"/>	●	Running	🇯🇵	-	-	-
<input type="checkbox"/>	●	Skiing	🇯🇵	-	-	-
<input type="checkbox"/>	●	Running Home	🇯🇵	-	-	-
<input type="checkbox"/>	●	FW_AttributeEditors/SegRa...	🇯🇵	-	-	-
<input type="checkbox"/>	●	SegRating	🇯🇵	-	-	-
<input type="checkbox"/>	●	FW_AttributeEditors/SegRa...	🇯🇵	-	-	-
<input type="checkbox"/>	●	TypeAhead	🇯🇵	-	-	-
<input type="checkbox"/>	●	FW_AttributeEditors/TypeA...	🇯🇵	-	-	-
<input type="checkbox"/>	●	FW_AttributeEditors/Calen...	🇯🇵	-	-	-
<input type="checkbox"/>	●	Couple Surfing	🇯🇵	-	-	-

Page 1 of 43 Next ▶ Last

[← Back](#)

[🚀 Publish On-Demand Queue](#)

If you need to resolve an issue, click the link to the asset and make any changes that are necessary (and perhaps approve it):

- To place assets in the On-Demand Queue, select them and click **Add to On-Demand Queue**. To remove assets from the On-Demand Queue (from the On-Demand Queue tab) select the assets and click **Remove from On-Demand Queue**.
- To publish the assets in the On-Demand Queue, click **Publish On-Demand Queue** from either tab.
- For information about why assets can be held, see [Table 13–4, "Approval States Assigned by the Approval System"](#).
- For procedures that describe how to work with assets, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

18.14 Managing Publishing History Information

Publishing sessions that are completed are listed in the Publish History section of the Publish Console, which is displayed when you click the Publishing button. Each publish session in the list has a status; Running, Done, or Failed.

Each item in the list has two icons:

- The inspect icon (the circled letter "i"), which displays a summary of that session. Examining the summary is the first step in troubleshooting a failed session.
- The delete icon (the trash can), deletes the session. When you click this icon, WebCenter Sites deletes the row from the `PubSession` table, the publishing log file for the session, and any messages for the session from the `PubMessage` table.

Note that the greater the number of publishing sessions listed in this section, the longer it takes WebCenter Sites to open the Publish Console.

If a session shows its status to be Failed, click the inspect icon, note any error messages, and then consult [Section 18.16, "Troubleshooting."](#)

About Session History

The publishing history log files are displayed in the History tab of the Publishing Console as summaries of publishing sessions.

The text in a history file is a list of all the references that were published during that session. The term reference means something different for each publishing method:

- For Mirror to Server publishing, a reference is an asset. Each asset that is published is listed by its asset type and its ID.
- For Export to Disk publishing and Export Assets to XML, a reference is a generated file. Each file that is created is listed by its file name.

To see the Publishing History log, select the particular session from the History tab to view the Publishing Status form for that session.

Figure 18–13 Publishing Status Form**Publishing Status**

Session ID: 1330098633208

Status: done

Destination: FSII Destination (RealTime) using RealTime (Complete publish)

Start Time: Friday, February 24, 2012 11:15:01 AM PST

▶ ◀ ◻

Stage	Status	% Completed
Gathering data to publish	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Serializing data	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Sending data to target	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Deserializing and saving	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Updating page caches	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%

[▶ View Log](#)
[▶ Download Log](#)
[▶ Publish History](#)
[▶ Configure Destination](#)

Click **View Log** to see the history log. You can also click **Download Log** to download the log and view it separately.

If there was a problem with the session, an error message is written to the `PubMessage` table and it is displayed as the history instead. If you see an error message in the history for a publishing session, see the section called [Section 18.16, "Troubleshooting"](#) and attempt to resolve the problem.

18.15 Publishing All Approved Assets

The publishing system conducts incremental publishing sessions. That is, during any session, only those assets that have been approved since the last session for this destination are published.

However, you may find a situation in which you need to publish all the approved assets in your entire site. (For example, after data repair on the delivery system.) When this is the case, use the **Force Publish All Items Next Time** button in the Edit Destination form.

When you click this button, it is the equivalent of changing the status of all approved assets to that of "never been published." That means that the next publishing session to the destination will publish all approved assets, regardless of whether they have already been published and have not changed since they were published.

18.16 Troubleshooting

This section describes error messages and other system indicators that reveal configuration, system, or data errors and suggests corrective actions for each. It contains the following topics:

- [Section 18.16.1, "About Publishing System Error Messages"](#)
- [Section 18.16.2, "Numeric Messages"](#)
- [Section 18.16.3, "Other Indicators of System or Configuration Issues"](#)

18.16.1 About Publishing System Error Messages

The publishing system reports information about publishing sessions in the following ways:

- Displays a status for each session in the **Publish History** list in the **Publish Console**. If you see a status of Failed, Not Found, or false, there was a problem with the publishing session.
- Writes information about the assets that were published to the log file for the publishing session. When you click the inspect icon for a publishing session in the Publish Console, the history log file for that session is displayed.

The `request.folder` property in the `batch.ini` file defines where the publishing history log files are located. Each log file is named as follows:

```
PubSession ID + Output.html
```

For example, if the PubSession ID for the session you are interested in is 9876544, then the log file is named `9876544Output.html`, and it is located in the directory specified by the `request.folder` property.

- Writes error messages about the publishing session to the `PubMessage` table. These messages are displayed in the **Messages** text area when you examine the publishing history in the Publishing Console for a specific publishing session.

You can use the PubSession ID to look up messages in the `PubMessage` table about a specific publishing session. Note that when you select the `Verbose` parameter on the destination configuration, status messages are also written to the `PubMessage` table.

- Writes other information about the session to the `PubSession` table. For example, to determine what time a publishing session started, look up the session by its PubSession ID in the `PubSession` table.
- Writes error messages to the `futuretense.txt` file on both the source and the destination (target) systems.

You can also find error messages about failed publishing sessions in the following locations:

- The application server log files on both the source and the destination systems.
- The `stdout` and `stderr` logs on both the source and the destination system.

18.16.2 Numeric Messages

This section contains descriptions and suggested corrective actions for the numeric error messages that the publishing system might report in any of the log files mentioned in the list above.

Note: The following list is not a complete list of all the error messages that your WebCenter Sites system can report. For a complete list of all error messages, see the error conditions chapter in the *Oracle Fusion Middleware WebCenter Sites: Tag Reference*.

This section contains the following topics:

- [Section 18.16.2.1, "-2"](#)
- [Section 18.16.2.2, "-3"](#)
- [Section 18.16.2.3, "-103"](#)
- [Section 18.16.2.4, "-611"](#)
- [Section 18.16.2.5, "-612"](#)
- [Section 18.16.2.6, "-12011"](#)
- [Section 18.16.2.7, "-12044"](#)
- [Section 18.16.2.8, "-12045"](#)
- [Section 18.16.2.9, "-12046"](#)
- [Section 18.16.2.10, "-12047"](#)
- [Section 18.16.2.11, "-13054"](#)
- [Section 18.16.2.12, "-13055"](#)

18.16.2.1 -2

This error indicates that either the mirror user name or password is not identified correctly. The publishing system on the source began the publishing process, but the destination system couldn't authenticate the user that is identified as the mirror user.

Corrective Action

Correct the mirror user information. See the procedure [Section 16.3.3, "Identify the Mirror User to the Source System"](#) for help.

18.16.2.2 -3

This error indicates that the mirror user does not have the correct permissions to save the assets to the WebCenter Sites database on the destination system. In other words, the destination system user identified to the source as the mirror user does not have all the ACLs it needs.

Corrective Action

Edit the mirror user's account and assign it the appropriate ACLs.

See the procedure [Section 16.3.2, "Set Up the Destination System"](#) for a list of the ACLs that the mirror user needs.

If you still cannot determine which ACL is missing from the mirror user's account, examine the `futuretense.txt` file on the destination system. There should be an entry that describes which table the mirror process failed to update and the name of the table may help you determine which ACL is missing.

For example, if the mirror session is failing on a visitor table, it is likely that your mirror user does not have the Visitor or VisitorAdmin ACL assigned to it.

18.16.2.3 -103

This error indicates that one of the tables whose data is being mirror published does not exist on the destination system. This error typically occurs when there is an asset type on the source that does not yet exist on the destination. It is also possible that someone created a custom table to support a function that was custom designed for your site but that table has not yet been created on the destination system.

Corrective Action

Examine the sites.log file on the destination system and look for the lines that list the -103 error. The text in the message should mention which table is missing.

Then use the appropriate tool to create the table on the destination system. That is, if it is a basic asset type, use AssetMaker. If it is a flex asset type, use Flex Family Maker. If it is a custom table, use Sites Explorer.

18.16.2.4 -611

This error message is one of the two generic mirror publishing error messages (the other is -12011). Typically -611 means that there was a problem when the publishing system tried to access the destination system and there should be other error messages reported, too.

Corrective Action

Examine the sites.log file and application server logs on the destination system to look for additional messages.

18.16.2.5 -612

This error message indicates that the definition of a mirror destination is incorrect in some way. Perhaps the syntax of the destination address is incorrect or there is a typographical error of some kind.

Corrective Action

Log into the WebCenter Sites interface on the source, examine the definition of the destination, and determine that it is correct. For help, see [Section 16.3.5, "Create a Mirror Destination."](#)

18.16.2.6 -12011

This error message is one of the two generic messages (along with -611) which mean that the mirror process failed. It is unlikely that this message will appear without other error messages being reported. Typically the other errors will give more information about what went wrong.

Corrective Action

Examine the messages in the PubMessage table and the sites.log file on the source to look for additional error messages.

18.16.2.7 -12044

This error indicates that the publishing system could not begin the publishing session because it could not contact the destination to start the session. That is, there was no response from the destination; some part of the destination system is offline.

Corrective Action

Start by verifying that the web server and application server are running on the destination system. If they are not, start them. Verify that the two systems can connect to each other through the network (perhaps the network went off-line, for example).

18.16.2.8 -12045

This error indicates that the publishing system was able to start the session, but it then failed in some way.

Corrective Action

Examine the `sites.log` file on the destination system and look for error and exception messages in the log.

18.16.2.9 -12046

This error indicates that there was a problem at the end of a publishing session when the publishing system began to clean up the temporary tables that it creates for each session. When the cleanup process began, the connection with the destination failed. That is, the source system could get no response from the destination when the publishing process tried to clean up the temporary tables.

Corrective Action

Determine whether the web server and application server on the destination system are running. If they are not, start them. Check for network connection problems, as well.

18.16.2.10 -12047

This error indicates that the publishing system was able to begin the cleanup process but it then failed in some way during the operation.

Corrective Action

Examine the `sites.log` file on the destination and look for error exception messages and exceptions that describe the problem.

18.16.2.11 -13054

This error occurs for complex and flex assets only. In this case, the publishing system began publishing a complex or flex asset but the data did not reach the destination. It can mean that the data itself was corrupt or that the HTTP request connection failed.

Corrective Action

Check the `sites.log` file on the source system. If the problem was the data itself, there will be additional messages in this log file that can help you determine which asset caused the problem. However, if the HTTP request was dropped before the destination system could respond, there will not be an additional message in the log.

18.16.2.12 -13055

This error occurs for complex and flex assets only. In this case, the data for the flex or complex asset was delivered, but the destination system could not save the asset to its WebCenter Sites database.

This message indicates that there is a problem with your data. For example, if you are attempting to publish from more than one source to the same destination, there can be ID collisions and other problems with data integrity that will cause this error.

Corrective Action

Examine the sites.log and application server log files on the destination system. There is often a stack trace message in the stderr log that describes what went wrong.

18.16.3 Other Indicators of System or Configuration Issues

This section describes symptoms your Sites system may exhibit when the publishing system discovers configuration errors or system problems.

This section contains the following topics:

- [Section 18.16.3.1, "Publishing Session Does Not End and Displays an Odd Status"](#)
- [Section 18.16.3.2, "The System Stops Altogether"](#)
- [Section 18.16.3.3, "Pages Are Not Refreshed After the Publishing Session Ends"](#)
- [Section 18.16.3.4, "DB2 Systems, Troubles When Publishing Assets with Associations"](#)

18.16.3.1 Publishing Session Does Not End and Displays an Odd Status

When a publishing session cannot begin or end, typically the batch user has not been configured correctly. Examine the Status listed for the publishing session. If the Status is Not Found or false, it is likely that your batch host is not configured correctly.

Note that the Initialize Mirror Destination feature does not use the batch user. If you can initialize a destination correctly, but the publishing session has errors, it is likely that the batch user settings are incorrect.

Corrective Action

See [Section 15.4.1, "Create the Batch User Account \(If One Does Not Exist\)"](#) and verify that you have configured the batch user correctly:

- The batch user identified by the `xcelerate.batchuser` property must have the appropriate read/write privileges. These are listed in that procedure.
- The name of the batch user identified by the `xcelerate.batchuser` property must be spelled correctly and the password identified by the `xcelerate.batchpass` property must be correct.
- The server identified by the `xcelerate.batchhost` property must identify the correct server. This property should identify the **application server** that hosts the source system, not the destination, and not the load balancer if you are using a load balancer. Note that if the port number is something other than 80, you must also specify the port number. For example: `myserver:7001`

If the WebCenter Sites environment is clustered, only one batch host is supported. The `xcelerate.batchhost` property must be set on each cluster member to point to the dedicated host.

If you see the message "Cannot retrieve output for publish session" in the publishing history, it is likely that the server name is incorrect.

- The port number for the batch host server must also be identified correctly. If you see the message "Cannot retrieve output for publish session" in the publishing

history, it is likely that either the port number is incorrect, or if a port number is not specified, that the default port (80) is incorrect.

18.16.3.2 The System Stops Altogether

If your system simply stops, this behavior indicates that there may not be enough disk space available for the publishing system to function.

Corrective Action

Check the `stdout` and `stderr` files. If there are any Java write errors, you are out of disk space.

- For Export to Disk publishing and Export Assets to XML, you need enough disk space in your file system to store all the files that are being generated.
- For Mirror to Server publishing, you need space equal to four times the size of the data that is being mirrored available on both the source and the destination system or the mirror operation will fail.

Note that the publishing process will also fail if the Java temp directory is not large enough.

18.16.3.3 Pages Are Not Refreshed After the Publishing Session Ends

If you are using Satellite Server—either the co-resident Satellite on a management or development system or a remote Satellite Server for a delivery system—and you notice that cached pages are not being regenerated after a publishing session, it is likely that you have configured Satellite Server incorrectly.

Corrective Action

Examine the `sites.log` file on the destination system and look for messages like these:

- "Number of satellite servers must match number of usernames and passwords."
This message indicates that there is something wrong with the values specified for the `cs.satellitehosts`, `cs.satelliteuser`, and `cs.satellitepassword` properties in the `futuretense.ini` file.
- "-100.FormPoster failed flushing URL."
This message indicates that either a Satellite servlet wasn't running or that the destination system couldn't reach a Satellite servlet that it tried to reach:

Open the `futuretense.ini` file, examine the properties on the **Satellite** tab, and note the values set for the user names, passwords, and host names. Open the `satellite.ini` files for the co-resident Satellite Server and your remote Satellite Server applications. Compare the values specified for the `cs.satellitehosts`, `cs.satelliteuser`, and `cs.satellitepassword` properties in the `futuretense.ini` file to the values set for the `username` and `password` properties in the `satellite.ini` files. They must match.

Remember that the order in which you specify host names in the `futuretense.ini` file must match the order in which you identify user names and passwords.

18.16.3.4 DB2 Systems, Troubles When Publishing Assets with Associations

When you notice that there are problems with publishing assets that have associations and your system uses a DB2 database, it is likely that the `LOCKLIST` parameter is not set correctly.

Corrective Action

The WebCenter Sites installation guide that describes DB2 installations recommends that this property be set to at least 1000. If you find that you are having difficulties publishing assets with associations, increase this value.

RealTime Publishing Overview

RealTime publishing is a dynamic publishing process, and the preferred method for publishing. Using the RealTime user interface, an administrator can monitor the progress of a publishing session, publish selected assets on demand, bulk unapprove assets approved for publishing, cancel publishing, and redo publishing sessions.

This chapter contains the following sections:

- [Section 19.1, "Overview"](#)
- [Section 19.2, "How RealTime Publishing Works"](#)

19.1 Overview

RealTime publishing is a dynamic publishing method named for its user interface. In the user interface, publishing status messages update in real time: Progress bars and mouse-over tooltips illustrate the progression of a publishing session as it unfolds, relaying status information from both source and destination systems.

Figure 19–1 Publishing Status Form

Publishing Status

Session ID: 1330406570281

Status: ...

Destination: RealTime Destination Overnights using RealTime (Delayed publish)

Start Time: Thursday, March 1, 2012 9:05:04 AM PST



Stage	Status	% Completed
Gathering data to publish	Success	<div style="width: 100%;"><div style="width: 100%;"></div></div> 100%
Serializing data	Success	<div style="width: 100%;"><div style="width: 100%;"></div></div> 100%
Sending data to target	Success	<div style="width: 100%;"><div style="width: 100%;"></div></div> 100%
<i>Deserializing and saving</i>	<i>Not Started</i>	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%
<i>Updating page caches</i>	<i>Not Started</i>	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%

▶ [View Log](#)

▶ [Download Log](#)

▶ [Publish History](#)

▶ [Configure Destination](#)

The RealTime user interface is also interactive. For example, you can interact with the list of approved assets. From the approved assets list you can select as many items as you like for immediate publication by placing the items into the On Demand queue. You can also remove one or more assets from the approved assets list using Unapprove. In both On Demand publishing and asset unapproval, asset dependencies are taken into account to ensure against broken links or missing assets on the destination system web site.

Figure 19–2 Approved Assets Tab

1655 assets available for Publishing

Show 10 20 30 50 Items per page

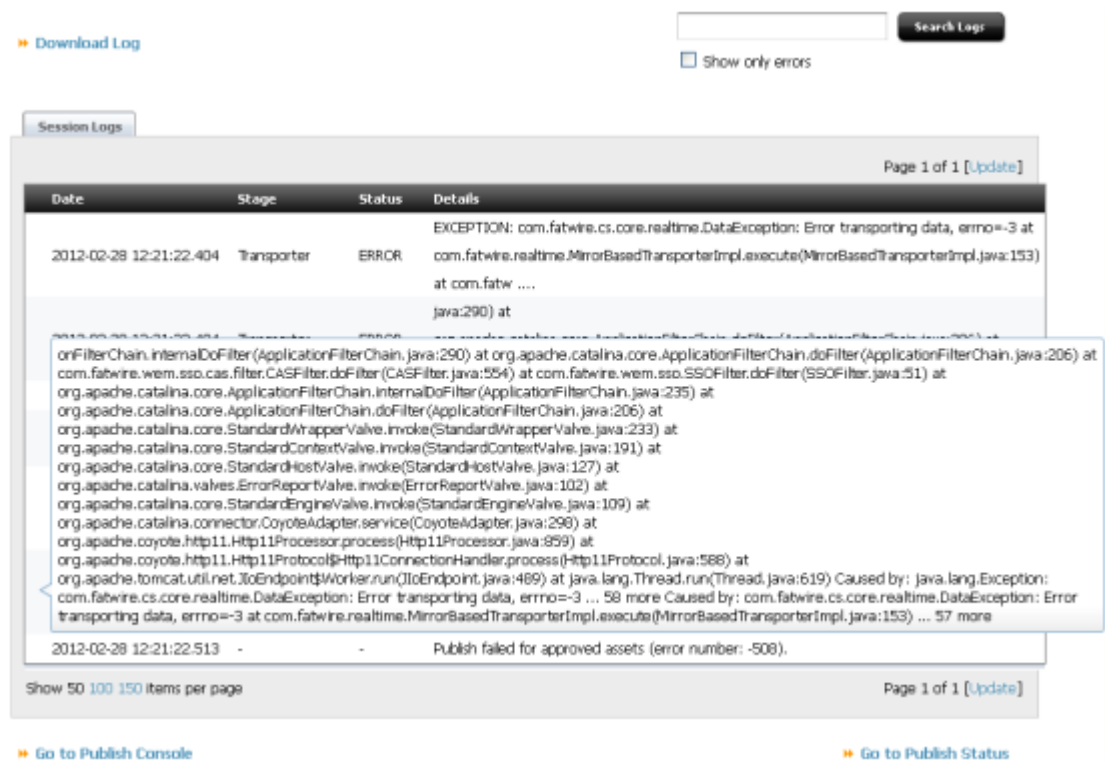
[Add to On-Demand Queue](#) [Unapprove](#)

<input type="checkbox"/>	Type	Name	Locale	Start Date	End Date	Days Expired
<input type="checkbox"/>	🔥	story1	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	story2	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	story3	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	Frequent Visitor	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	Burlington Promotion	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	FSIITopStories	en_US	-	-	-
<input type="checkbox"/>	🔥	FSIIItemOfTheWeek	en_US	-	-	-
<input type="checkbox"/>	🔥	FSIIAdvertisement	en_US	-	-	-
<input type="checkbox"/>	🔥	FSIIHotItems	en_US	-	9/30/09 11:59 PM	880
<input type="checkbox"/>	🔥	FSIILatestNews	en_US	-	9/30/09 11:59 PM	880

Page 1 of 166 [Next](#) [Last](#)

In addition to status messages and asset approval functions, the RealTime user interface features interactive logging forms. On the logging form, you can search, page through, and filter log entries. Mousing over a summary log message's **show all** link expands the entry to a detailed report.

Figure 19–3 Session Log



Underlying the RealTime user interface is a multi-stage publishing process. WebCenter Sites replicates, transfers, and stores web site data in five stages. The Publishing Status form below shows the five stages of publishing. These stages are described in detail in the section [Section 19.2, "How RealTime Publishing Works."](#)

Figure 19–4 Publishing Status Form


Publishing Status

Session ID: 1330406570281

Status: ...

Destination: RealTime Destination Overnights using RealTime (Delayed publish)

Start Time: Thursday, March 1, 2012 9:05:04 AM PST



Stage	Status	% Completed
Gathering data to publish	Success	100%
Serializing data	Success	100%
Sending data to target	Success	100%
<i>Deserializing and saving</i>	<i>Not Started</i>	0%
<i>Updating page caches</i>	<i>Not Started</i>	0%

[▶ View Log](#)[▶ Download Log](#)[▶ Publish History](#)[▶ Configure Destination](#)

One advantage of breaking RealTime publishing into discrete stages is that it introduces the option of stopping the publishing process after a stage completes and picking up the publishing session from the next stage.

RealTime publishing includes two modes of publishing: complete and delayed. In complete publishing mode, WebCenter Sites runs all stages of publishing in one continuous sequence. In delayed publishing mode, there is a pause midway through the process so the administrator can control when the system begins writing assets to the WebCenter Sites destination database.

Delayed publishing can be used to minimize the impact of publishing on web server performance. Using delayed publishing, the administrator can time a destination system update to occur during off-peak traffic hours, an option that is particularly useful when publishing a large number of assets.

Another advantage of the multi-stage publishing process is the ability to cancel a running session at will and then, if you wish, restart the cancelled session from just after the last successfully completed stage of publishing. You can also redo a publishing session that does not complete due to errors or power interruptions.

Figure 19–5 Publish Console**Summary**

When you use the RealTime publishing method, you can:

- Monitor an active publishing session. Progress bars on the Publishing Status form show the current progress of the session on both source and destination WebCenter Sites systems.
- Selectively publish approved assets using On-Demand publishing.
- Selectively unapprove assets.
- Review interactive, sortable session logs.
- Configure your destination for Complete Publishing mode to run the entire publishing process in an uninterrupted session.
- Configure your destination for Delayed Publishing mode to pause the publishing session midway so you can control when the system begins writing assets to the destination database.
- Cancel an active publishing session.
- Restart a cancelled publishing session from just after the last completed stage.
- Redo a failed publishing session.

19.2 How RealTime Publishing Works

Dynamic publishing is the process of copying assets and their underlying schema from one WebCenter Sites system to another. In RealTime publishing, this process is broken up into five stages.

WebCenter Sites always performs the five stages in order. When the destination is configured for complete publishing mode, the process proceeds without interruption. When configured for delayed publishing mode, the process pauses before the fourth step. User interaction is needed for the session to complete.

As the publishing session proceeds, you can monitor the progress of each stage on the Publishing Status form. Instructions for monitoring a publishing session can be found in [Chapter 21, "Working with RealTime Publishing Modes."](#)

Figure 19–6 Publishing Status Form


Publishing Status

Session ID: 1330406570281

Status: ...

Destination: RealTime Destination Overnights using RealTime (Delayed publish)

Start Time: Thursday, March 1, 2012 9:05:04 AM PST



Stage	Status	% Completed
Gathering data to publish	Success	100%
Serializing data	Success	100%
Sending data to target	Success	100%
<i>Deserializing and saving</i>	<i>Not Started</i>	0%
<i>Updating page caches</i>	<i>Not Started</i>	0%

[▶ View Log](#)[▶ Download Log](#)[▶ Publish History](#)[▶ Configure Destination](#)

The stages of publishing are:

1. Gathering data to publish:

- a. The WebCenter Sites source system locks the assets that are approved for this publishing session. Locking approved assets prevents them from being edited during publishing.
- b. On the source system, RealTime publishing uses data from the approval system to create a manifest, a list of assets to publish. The manifest identifies resource groups, groups of assets that must exist together on the destination to prevent broken links and incomplete pages.

2. Serializing data:

- a. The WebCenter Sites source system mirrors the following items:
 - Asset types
 - Auxiliary tables (identified in the destination configuration)

- Approved assets
- b. The source system serializes (translates to binary format) the mirrored data.

Note: This step can require considerable space to complete. It is recommended that the temporary space assigned to the JVM (via `java.io.tmp`) is at least three times the size of the data to be published. The source and destination databases must also have enough space for the table `FW_PUBDATASTORE` to grow. Since this can be equal to the size of the entire source site (both objects stored on disk and the database) during an initial publish, this means that the space required is likely to exceed what WebCenter Sites is currently using (particularly when large binaries are included).

3. Sending data to the target:

- a. The destination WebCenter Sites database locks assets that are approved for this publishing session. When assets on the target are locked, they cannot be edited.
- b. The WebCenter Sites source system sends the manifest and all serialized data to the destination. The data has not yet been committed to the destination database.

A delayed publishing session pauses at this point. It continues to stage 4 only after the administrator selects the **Resume** button on the Publish Console or Publishing Status form.

4. Deserializing and saving data:

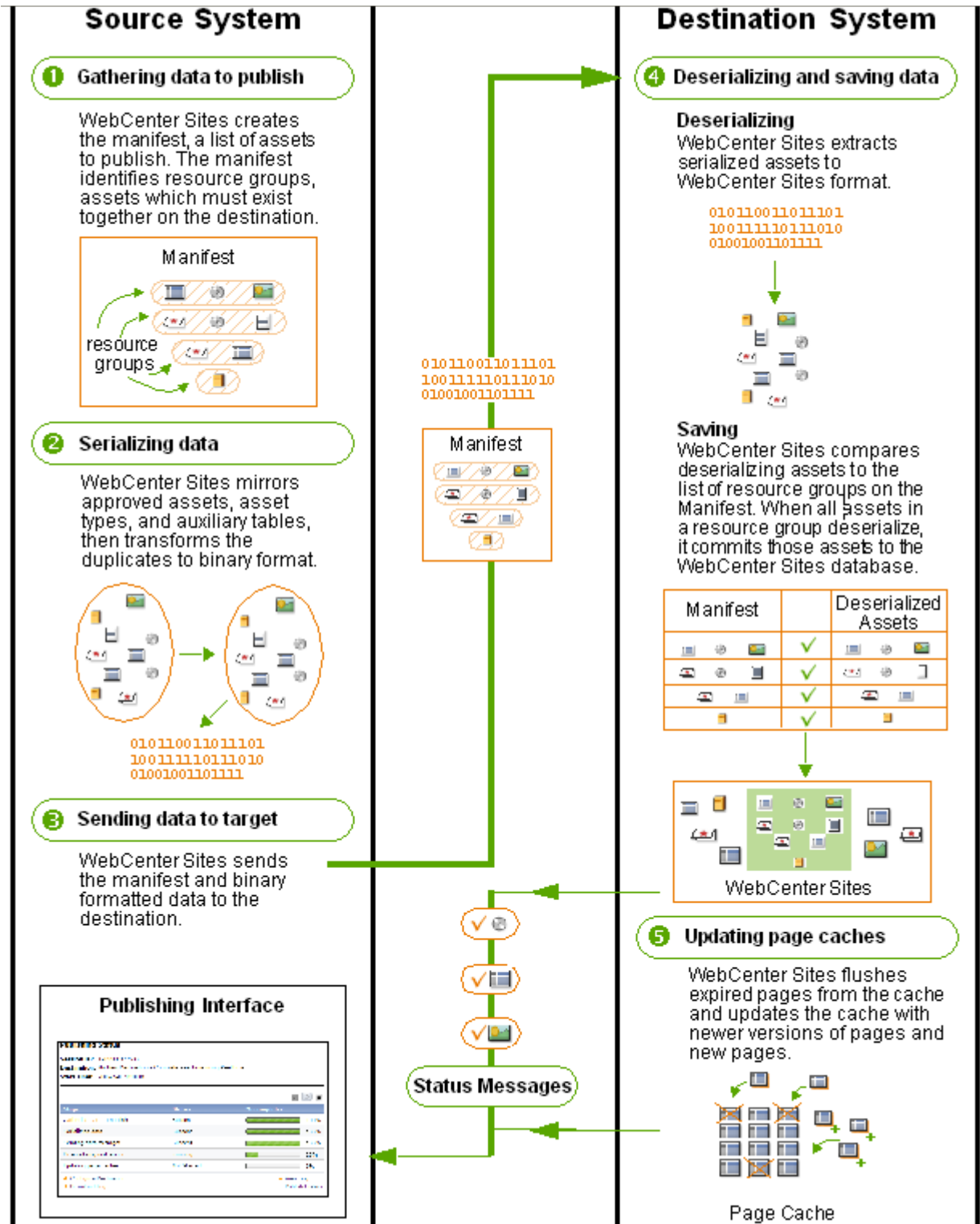
- a. The WebCenter Sites destination system deserializes (inflates back to WebCenter Sites format) the mirrored asset types, auxiliary tables, and approved assets.
- b. The destination database updates asset types and auxiliary tables with the deserialized ones, or adds them as new data.
- c. The destination system uses the manifest to determine when to commit assets to its database. Assets in a resource group are committed only when every asset in the group has deserialized.
- d. When the destination system commits all assets in a resource group:
 - a. It unlocks the assets in its database for editing. As the publishing session runs, the number of published, editable assets in the destination database increases.
 - b. The destination system sends a message to the source system identifying assets that have been successfully published. Assets in that resource group are unlocked on the source system for editing. As the publishing session runs, the number of editable assets on the source system increases.

5. Updating page caches:

The destination system completes the publication process:

- a. Flushes the page cache, which removes expired pages.
- b. Adds new pages to the cache.

Figure 19–7 The Five Stages of Publishing



Configuring the RealTime Publishing Process

When you configure the RealTime publishing process, you define the publishing destination and mirror site configuration data to that destination. In addition, you can set the publishing schedule. This chapter also includes instructions for using the RealTime process to synchronize site data.

This chapter contains the following sections:

- [Section 20.1, "Overview"](#)
- [Section 20.2, "Configuring Your System for RealTime Publishing"](#)
- [Section 20.3, "Adding a New RealTime Destination Definition"](#)
- [Section 20.4, "Synchronizing Site Configuration Data"](#)
- [Section 20.5, "Performance Enhancement"](#)

20.1 Overview

Dynamic publishing is the process of copying assets and their underlying schema from one WebCenter Sites system to another. In order for the RealTime publishing process to work, both the source and target systems must be set up for RealTime publishing.

Source System Configuration

The source system requires the following items to be configured:

- **A batch user account.** This user allows publishing to run as a background process and designates the location of publishing logs.
- If there is a proxy, the identity of the local **proxy server**. This is identified to WebCenter Sites in the `futuretense.ini` file.
- For large publishes (such as the initial publication of a site), the session timeout may need to be increased beyond the default 900 seconds (for example, 9000 seconds). To do so, open `futuretense.ini` in property editor and change `cs.timeout` to 9000, then restart WebCenter Sites. When the publish is completed this value can be reset.
- The **publishing destination**, which identifies:
 - The destination URL and port
 - The publishing method
 - Publishing log and email notification options

- User roles with approval and publishing privileges
- The RealTime user, who has publishing privileges on the destination

Additionally, though not required for publishing to work, you can configure a publishing schedule which will publish approved assets on a regular basis.

Destination System Configuration

The destination system requires:

- A **RealTime user account**, which has publishing privileges on the destination.
- **Mirrored site configuration data** from the source system for the sites to be published:
 - WebCenter Sites
 - Publication and SitePlanTree table data
 - Auxiliary tables (such as Source and MimeType)
 - Asset types
 - Roles
 - Start menu items*
 - Workflow groups and processes*
 - Tree tabs*
 - Saved searches*

** Items marked with an asterisk are typically not necessary on a delivery system.*

Additionally, if you will not publish from the destination system, we recommend that you turn off asset invalidation.

Follow the steps in this chapter to configure your system for RealTime publishing.

20.2 Configuring Your System for RealTime Publishing

Note: Configuring a RealTime publishing environment requires you to provide information that is specific to your WebCenter Sites installation and your business practices. Before starting procedures in this section, it is best to read them in order to gather and confirm the information they prompt you to provide.

This section contains the following topics:

[Section 20.2.1, "Create the Batch User Account \(If One Does Not Exist\)"](#)

[Section 20.2.2, "Set Up the Destination System"](#)

[Section 20.2.3, "If a Proxy Server Exists, Identify it to the Source System"](#)

[Section 20.2.4, "Create a RealTime Destination Definition on the Source System"](#)

[Section 20.2.5, "Initialize the Destination Database"](#)

[Section 20.2.6, "Mirror the Site Configuration Data to the Destination Database"](#)

[Section 20.2.7, "Test the Publishing Process"](#)

[Section 20.2.8, "Test the Published Site"](#)

[Section 20.2.9, "Set Up the Schedule"](#)

[Section 20.2.10, "Turn Off Asset Invalidation on the Delivery System"](#)

20.2.1 Create the Batch User Account (If One Does Not Exist)

Note: This procedure needs to be performed only once on each source system, regardless of the number and types of publishing destinations you are configuring.

If a batch user account already exists on your source system, skip to [Section 20.2.2, "Set Up the Destination System."](#)

A batch user account is required before any type of publishing can take place.

The purpose of a batch user account is two-fold:

- Enable the publishing process to run as a background process on the source system.
- Specify the location where publish logs will be stored.

If you have not already done so, create the batch user account. For instructions, see [Section 15.4.1, "Create the Batch User Account \(If One Does Not Exist\)."](#)

Note: Set `xcelerate.batchhost` property to the host name of the application server that is hosting the source system. The source system is the batch host. If the port number of the web server is anything other than 80, you must also include the port number. For example, `myserver:7001`. In a clustered WebCenter Sites environment, only one batch host is supported. The `xcelerate.batchhost` property must be set on each cluster member to point to the dedicated host.

20.2.2 Set Up the Destination System

Data cannot be dynamically published unless the same database schema exists on the source and destination systems. In this step, you will ensure that the structure of auxiliary tables on the source database is reproduced in the destination database. The data, itself, you will mirror in a later step ([Section 20.2.5, "Initialize the Destination Database"](#)).

The destination also requires a mirror user to complete database transactions on the destination system during publishing. You will ensure that a mirror user exists on the destination. (The mirror user is different from the batch user, which exists on the source system.)

To set up the destination system

1. Determine which auxiliary tables on the source system support the types of assets that users will publish, then manually re-create the table structure on the destination system.

Note: Auxiliary tables are custom-defined tables that support asset types. The tables are not created via WebCenter Sites' AssetMaker or Flex Family Maker. Examples of auxiliary tables are Source, MimeType, and lookup tables (for different attributes in asset types).

Database properties on the source and destination systems, if not an exact match, must be compatible, particularly database schema options (set in `futuretense.ini`). Be sure to restart the application server if you make changes to `futuretense.ini`.

2. Create the RealTime user on the destination system (using the administrator's interface) and note the following:

Note: The RealTime user has the same privileges as the mirror user configured for Mirror to Server publishing. If a mirror user is configured for Mirror to Server publishing, the same user may serve as the RealTime user.

- **Basic privileges:** The RealTime user must be assigned the following ACLs:
 - Browser
 - ElementEditor
 - PageEditor
 - TableEditor
 - Visitor
 - VisitorAdmin
 - xceladmin
 - xceleditor
- **Additional privileges:** Because the RealTime user account is used by the CacheManager to regenerate the page cache after a publishing session, the RealTime user must have sufficient privileges to regenerate all the pages in the cache. Therefore, the RealTime user account must be assigned all the ACLs that are assigned to page entries in the `SiteCatalog` table or database tables that hold data to be rendered.

If you need help with creating the mirror user, see [Section 5.4.1, "Creating a New User."](#)

20.2.3 If a Proxy Server Exists, Identify it to the Source System

Note: This procedure needs to be performed only once on each source system, regardless of the number and types of publishing destinations you are configuring.

Skip to [Section 20.2.4, "Create a RealTime Destination Definition on the Source System"](#) if the local proxy has already been identified to your source system, or a proxy server is not used.

To identify the local proxy server to the source system for all destinations

1. Open `futuretense.ini` in the Property Editor on the source system, select the **Export/Mirror** tab, and specify values for the following properties:
 - `cs.mirrorproxyserver`
Set this property to either the name or the IP address of the local proxy server.
 - `cs.mirrorproxyserverport`
Set this property to the port number of the local proxy server.
2. Save the property file and close the Property Editor.
3. Restart the application server.

To identify the local proxy server to the source system per destination

1. In the Admin interface, select the Admin tab. Expand Publishing, then expand Destinations, and open the destination.
The destination opens in the main form.
2. Click Edit.
The Edit Destination form for the destination displays.
3. In the More Arguments field, enter
`PROXYSERVER=wcs.example.com&PROXYPORT=8080`
Where `wcs.example.com` is the local proxy server, and 8080 is the port the proxy is running on.
4. Click Save.

20.2.4 Create a RealTime Destination Definition on the Source System

In this step, you will define the publishing destination to the source system.

Note: While you can define multiple destinations, bear in mind that only sequential publishing is supported.

To create a RealTime destination definition

1. On the source system, log in to the administrator's interface.
2. In the **Admin** tab, expand **Publishing**, then **Destinations**.
3. Under **Destinations**, double-click **Add New**.
The Add New Destination form appears.

Figure 20-1 Add New Destination Form

Add New Destination

***Name:**

Delivery Type:

Select the desired publishing option

Complete publish

Delayed publish

***Destination address:**

***Remote User:**

***Remote Password:**

Send Email on Failure:

Verbose Output:

More Arguments:

***Sites:**
 AdminSite
 Burlington Financial
FirstSite II
 GE Lighting
 Hello Asset World

***Roles (Approve for Publish):**
 AdvancedUser
 Analyst
 Analytics
 Approver
 ArtworkAuthor

***Roles (Publish):**
 AdvancedUser
 Analyst
 Analytics
 Approver
 ArtworkAuthor

4. Fill in the Add New Destination form. For guidelines, use the information below.
 - **Name:** Enter a unique name for the destination.

- **Delivery Type:** Select **RealTime: Copy assets to remote dynamic server**, then choose one of the following options:
 - **Complete publish** to copy assets to the destination address without interruption to the publishing process.
 - **Delayed publish** to pause the publishing process before assets are committed to the destination database. To resume the session, a user must initiate the fourth stage of publishing (deserializing [inflating] data and saving it to the destination database). More information on delayed publishing can be found in [Section 21.3.2, "Resuming a Delayed Publishing Session."](#)
 - **Destination Address:** Enter the URL in the format shown. For `[targetserver:port]`, delete the brackets, enter the hostname or IP address of the destination system, and specify the port to be used by the destination. (**A slash is required at the end of the URL** because this URL is appended dynamically.)
 - **Remote User:** Enter the name of the RealTime user that you created in [Section 20.2.2, "Set Up the Destination System."](#) This user will be invoked by the publishing system to unpack the RealTime queue on the destination system.
 - **Remote Password:** Enter the password of the RealTime user.
 - **Send Email on Failure:** If publishing fails and email notices to that effect are required, select the checkbox and fill in the field that is displayed:
 - **EmailIDs:** Enter the recipient's email address. (This field is available only when **Send Email on Failure** is selected.)
 - **Verbose Output:** Select this option to activate detailed error logging during the publishing process. When this option is selected, messages in addition to error messages are written to the `PubMessage` table. Because additional information lengthens the publishing process, select this parameter only for troubleshooting.
 - **More Arguments:** This parameter is reserved; no additional arguments may be specified at this time.
 - **Sites:** Select the sites whose assets can be approved for and published to this destination.
 - **Roles (Approve for Publish):** Select the roles to which you are assigning asset approval privileges. All users assigned these roles will be able to approve assets.
 - **Roles (Publish):** Select the roles to which you are assigning publish privileges. All users assigned these roles will be able to publish and view the Publish Console.
5. Click **Add New Destination**.

WebCenter Sites writes your information to the `Pubdestination` table and displays the destination definition in the Inspect form:

Figure 20–2 Publish Destination Form

Publish destination: FSII Destination (RealTime)

Inspect
Edit
Delete
Approve Multiple Assets

Name: ● FSII Destination (RealTime) ←

Delivery Type: RealTime Copy assets to remote dynamic server
⊕ Complete publish

Initialize Realtime Destination

Destination address: http://localhost:8080/cs/

Remote User: fwadmin

Remote Password: *****

Send Email on Failure: false

Verbose Output: false

More Arguments:

Sites: FirstSite II

Roles (Approve for Publish): Any

Roles (Publish): Any

Publish Event: Publish event scheduled at:
*:0,5,10,15,20,25,30,35,40,45,50,55:0 */**

Set Publish Event...
Delete

ID: 1330406369412

Force Approve Published Assets

[▶ List all Publish Destinations](#)

Note this field.
If a red button is displayed, it means that the target server cannot be located. Refer to the note below for possible causes.

Note: Before proceeding to the next step, check the **Name** field. A red button next to the destination name means that the target server cannot be located. Common causes are the following: The target URL is incorrect or the target server is not running.

6. If you need to create more destination definitions, repeat steps 3 through 5 for each new definition.
7. When you are ready to initialize the destination system, proceed to the next section, [Section 20.2.5, "Initialize the Destination Database."](#) Begin at step 3 of that procedure.

20.2.5 Initialize the Destination Database

You must initialize the destination database before you can publish to it. To initialize, you will mirror two constructs:

- WebCenter Sites
- Data in auxiliary tables (which support the types of assets users will be publishing)

The Publication and PublicationTree tables will be updated with the site names and table data you have chosen to mirror.

To initialize the destination

1. In the **Admin** tab on the source system, expand **Publishing**, then **Destinations**.
2. Under **Destinations**, double-click the RealTime destination you want to initialize.
3. In the Publish Destination form, click **Initialize RealTime Destination**.

The Initialize RealTime Destination form appears:

Figure 20–3 Initialize Mirror Destination Form

Initialize Mirror Destination: FSII Destination (RealTime)

Sites: Select all sites that will be supported on this destination:

AdminSite
Burlington Financial
FirstSite II
GE Lighting
Hello Asset World

Auxiliary Tables: Add any other non-Asset tables whose data is referenced or displayed. These tables must exist on the destination before data can be mirrored over.

Source
MimeType
Category

4. In the **Sites** field, select the sites whose assets you want to publish to this destination.
5. In the **Auxiliary Tables** field, enter the names of the following tables:
 - *Source*, if you are using the source feature for any of your asset types.
 - *MimeType*, if you are using Sites DocLink, flex filter assets, the ImageFile asset type, or if you are using the MimeType table for any of your custom asset types.
 - *Filters*, if you are using flex filters on your site. (The *Filters* table lists the classes that are used by the flex filters. You will need to copy jar files and classes manually.)
 - Any other auxiliary tables (such as lookup tables) for your asset types.

Note: Only five fields are provided for table names. If you have to enter more than five table names, complete this procedure through step 6 for the first five tables, then repeat steps 3–6 for the remaining tables.

6. Click **Mirror**.
 If initialization is successful, WebCenter Sites displays a confirmation message (otherwise, an error message is displayed).
 - Based on the sites that you selected in step 4, the corresponding rows from the Publication, SitePlanTree, and AssetType tables are copied to the destination.

- If you specified auxiliary tables in step 5, all rows in those tables are copied to the destination.

20.2.6 Mirror the Site Configuration Data to the Destination Database

In this step, you will mirror the source's site configuration data (such as asset types and start menu items) to the destination database.

To mirror the site configuration data

1. In the **Admin** tab on the source system, expand **Sites** and double-click the site whose data you want to publish to the destination.
2. In Publish Destinations (near the bottom of the form), click **Mirror site configuration for *destination***.

WebCenter Sites displays the Mirror Site Configuration form:

Figure 20–4 Mirror Site Configuration Form

Mirror site configuration for: [FirstSiteII](#)

Publish destination: TennisPub

Asset Types: enabled in FirstSiteII

- Attribute Editor
- CSElement
- Content
- Content Attribute
- Content Definition

Asset Subtypes for selected asset types

Asset Associations for selected asset types

Categories for selected asset types

Sites Desktop configuration for selected asset types

Sites DocLink configuration for selected asset types

Start Menu Items: available in FirstSiteII

- CS-Desktop Content_C
- CS-DocLink Document
- CS-DocLink Media
- Find Attribute Editor, FirstSiteII
- Find Content Attribute, FirstSiteII

Workflow Processes: available in FirstSiteII

- FSII: Approval for Artwork
- FSII: Approval for Content
- FSII: Approval for Documents
- FSII: Approval for Products
- FSII: Approval for Promotions

Workflow States used by selected workflow processes

All Workflow Actions

All Workflow Conditions

Workflow Groups: available in FirstSiteII

- FSII Home Page Translations

Workflow Group Actions

Tree Tabs: available in FirstSiteII

- Site Plan
- Admin
- Site Admin
- Workflow
- Bookmarks

Saved Searches: available in FirstSiteII

-

Roles: available in all sites

- AdvancedUser
- Analyst
- Analytics
- Approver
- ArtworkAuthor

3. Select the data you wish to mirror to the destination:

Note: Which data you choose to mirror typically varies with the purpose of the destination system:

- If the destination system is strictly for delivery, it makes sense to mirror only asset types.
 - When content management functions are to be supported, mirror the remaining data as well.
-
-

- **Asset Types:** Select the asset types that you want to make available on the destination.

If your asset types have subtypes and categories, choose the appropriate options from the list in the **Asset Types** section and select the corresponding check boxes. For example, the `AssocNamed` table contains information about associations for asset types. The table will be mirrored only if you select **Asset associations for selected asset types** and then the relevant asset types.

Select the Sites Desktop / Sites DocLink configuration check boxes to configure the selected asset types for Sites Desktop and Sites DocLink.

- **Start Menu Items:** Select any start menu items that you or the developers designed for your content providers to use.
- **Workflow Processes:** If there are workflow processes for your asset types, select these processes and the appropriate workflow items.
- **Workflow Groups:** Select any workflow groups that you or the developers designed for your content providers to use.
- **Tree Tabs:** Select the tree tabs that you want to make available on the destination.
- **Saved Searches:** Select any saved searches that you or the developers designed for your content providers to use.
- **Roles:** Select the roles that must exist on the destination.

4. Click **Mirror**.

Based on which asset type configuration options you selected in step 3, appropriate rows from the `AssocNamed`, `AssocNamed_Subtypes`, and `Category` tables are copied to the destination:

- If you selected **Start Menu Items** or **Saved Searches**, appropriate rows from the tables that implement those features are copied to the destination.
- Based on which workflow configuration options you selected in step 3, appropriate rows from the workflow tables are copied to the destination.

5. Repeat steps 1–4 for each site whose assets you want to publish to the destination.

20.2.7 Test the Publishing Process

You can test your publishing process by using all publishable assets or a smaller collection.

This section contains the following topics:

- [Section 20.2.7.1, "Approve Assets"](#)
- [Section 20.2.7.2, "Publish the Assets"](#)

20.2.7.1 Approve Assets

Do one of the following:

- To fully test your publishing process, you must approve all the assets that you intend to publish. See [Section 18.2, "Approving Multiple Assets"](#) for help with approving multiple assets at once.
- If you wish to perform a simple test of your configuration, temporarily create a home page asset with just a few dependents.

20.2.7.2 Publish the Assets

To run a test publishing session on the approved assets

1. On the source system, click **Publishing** in the button bar.
2. In the **Publish Console**, select your RealTime destination from the drop-down list and click **Select Destination**.

WebCenter Sites displays information about the assets that are ready to be published to this destination.
3. If for some reason the destination has not yet been initialized, it must be initialized now or the publishing session will fail. To initialize the destination, click **Create Site on Target** in the **Publish Console** (or follow instructions in [Section 20.2.5, "Initialize the Destination Database"](#)). This creates the basic site information on the destination system.
4. Click **Publish**.

The publishing system mirrors all the approved assets for this destination to the WebCenter Sites database on the destination system.

20.2.8 Test the Published Site

To test your published site, you will point your browser to the home page asset on the destination system and examine the site.

If you have not already done so, you must first determine the home page asset's URL. A WebCenter Sites URL is constructed by concatenating the following values, as shown:

```
http://<targetserver:port>/<cgipath>/ContentServer?pagename=<your_home_page>/<other_info>
```

where

- `<targetserver:port>` is the hostname or IP address (and sometimes the port number) of the destination system.
- `<cgipath>` is the value of the `ft.cgipath` property in the `futuretense.ini` file. For example, `<cgipath>` is `/servlet/` for WebLogic and other application servers with servlet architectures.
- `ContentServer?pagename=` sets the name of the page to be displayed at this URL.
- `<your_home_page>` is the home page's name from the SiteCatalog entry.
- `<other_info>` is additional information that is passed in with the WebCenter Sites page criteria variables, `c`, `cid`, `tid`, and `p` (see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for information about these variables).

For example, the URL for the Burlington Financial home page looks like this when it is rendered by the WebCenter Sites JumpStart Kit:

```
http://localhost:7001/servlet/
  ContentServer?pagename=BurlingtonFinancial/Page/Home
```

To determine the URL of your home page and test the site

1. Start the Property Editor on the destination system (if you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*).
 - a. Open the `futuretense.ini` file.
 - b. In the **App Server** tab, locate the `ft.cgipath` property and write down its value.
 - c. In the **Compatibility** tab, locate the `ft.approot` property and write down its value.
 - d. Save the property file and close the Property Editor.

2. Open a text editor and assemble the URL:

- a. Enter `http://` or `https://` followed by: `<targetserver:port>/<cgipath>/`

For example:

```
http://bigfatsun.example.com:8080/servlet/
```

(for WebLogic and WebSphere)

- b. At the end of the string, add the following text:

```
ContentServer?pagename=<your_home_page>
```

Now the URL should look similar to the following example (for WebLogic and WebSphere):

```
http://bigfatsun.fatwire.com:8080/servlet/
  ContentServer?pagename=ExampleSite/Page/Home
```

3. Point your browser to the URL you assembled.
4. Scan the page for errors and test all links to ensure they work.

Note: If you detect an error related to content that is not stored as assets (the content is stored outside WebCenter Sites), use the appropriate transfer protocol to publish that content. For more information, see [Section 20.2.9, "Set Up the Schedule."](#)

It is recommended that you conduct a complete test of the system at strategic times: under peak load conditions after you have mirrored the entire site for the first time, and at regular points thereafter.

20.2.9 Set Up the Schedule

Once you have ensured that your destination is correctly set up, finish configuring your source's publishing system.

To finish configuring the publishing system

- On the source system, schedule times running publishing sessions to the destination. For help with this step, see [Section 18.7, "Scheduling Publish Events."](#)

- If your site uses image files that are stored not as assets but as images on the web server, plan how you will move the image files from the CM system web server to the delivery system web server. For example, you can set up a regular FTP transfer.
- If you are using elements and SiteCatalog page entries that are not CSElement and SiteEntry assets, you must use the CatalogMover tool to mirror them to the destination system. For help with CatalogMover, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

20.2.10 Turn Off Asset Invalidation on the Delivery System

By default, the destination invalidates chain publishing. That is, the publishing system (on the destination) marks each published asset as changed, due solely to the fact that the asset was published. Hence, an asset must be approved from the current destination before it can be published to yet another destination.

For assets on a final destination—such as a delivery system—asset invalidation should be turned off. It is an unnecessary (and time consuming) operation, as assets will not be published from that destination.

To turn off asset invalidation

1. Start the Property Editor on the final destination system (delivery, in this example) and open the `futuretense_xcel.ini` file. (If you need help with this step, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.)
2. In the **Publishing** tab, locate the `xcelerate.publishinvalidate` property and set its value to `false`.

20.3 Adding a New RealTime Destination Definition

Once you have configured your source system to publish to one RealTime destination, you can define other RealTime destinations, if necessary. Follow only the steps below:

[Section 20.2.2, "Set Up the Destination System"](#)

[Section 20.2.4, "Create a RealTime Destination Definition on the Source System"](#)

[Section 20.2.5, "Initialize the Destination Database"](#)

[Section 20.2.6, "Mirror the Site Configuration Data to the Destination Database"](#)

[Section 20.2.7, "Test the Publishing Process"](#)

[Section 20.2.8, "Test the Published Site"](#)

[Section 20.2.9, "Set Up the Schedule"](#)

[Section 20.2.10, "Turn Off Asset Invalidation on the Delivery System"](#)

20.4 Synchronizing Site Configuration Data

In addition to publishing assets, you can synchronize site configuration between the source and destination. For example, you can synchronize site configuration data from a development system to a management system or to a delivery system.

Use the Mirror Type Configuration page:

- To move configuration items that support asset types (start menu items, associations, asset types, and so on).
- If you or your developers add categories or subtypes for existing asset types.

- To move workflow configuration data from a development system to a management system.

See [Section 20.2.6, "Mirror the Site Configuration Data to the Destination Database"](#) for instructions on using the Site Configuration form.

Use the Initialize Mirror Destination page:

- If you or your developers add sites, or auxiliary tables to your system that you wish to publish.
- When you need to troubleshoot your configuration. If you can successfully initialize a RealTime destination, the source and the destination systems are communicating.

See [Section 20.2.5, "Initialize the Destination Database"](#) for instructions on using the Initialize Mirror Destination form.

20.5 Performance Enhancement

To improve the efficiency of high-volume dynamic publishing, an approval query can be implemented. For more information, see [Section 16.4, "Approval Query for Performance Enhancement."](#)

Working with RealTime Publishing Modes

One of the advantages of RealTime publishing is the feedback that the system generates and displays to the publishing administrator throughout the publishing process. The publishing administrator can monitor sessions in progress, review publishing schedules, inspect history records, and explore logs.

This chapter contains the following sections:

- [Section 21.1, "Complete and Delayed Publishing"](#)
- [Section 21.2, "Publish Console"](#)
- [Section 21.3, "Working with Active Publishing Sessions"](#)
- [Section 21.4, "Working with Scheduled Publishing Sessions"](#)
- [Section 21.5, "Working with Prior Publishing Sessions"](#)
- [Section 21.6, "Working with Publishing Logs"](#)

21.1 Complete and Delayed Publishing

A publishing session can run in one of two modes: complete or delayed. The mode must be explicitly specified, as shown in [Section 20.2, "Configuring Your System for RealTime Publishing."](#)

- In a complete publishing session, the five stages of RealTime publishing run in one uninterrupted sequence. You have the option to stop and resume a session, or cancel it.
- In delayed publishing, the session pauses after the second stage (data serialization). For the session to run to completion, you must resume the session. The final stages follow automatically.

Delayed publishing enables you to control when the system begins writing to the destination database. For example, if you are publishing a large number of assets to a heavily trafficked web site, delayed publishing offers you the option to wait for a low-traffic period, when saving data to the destination database is a more efficient process.

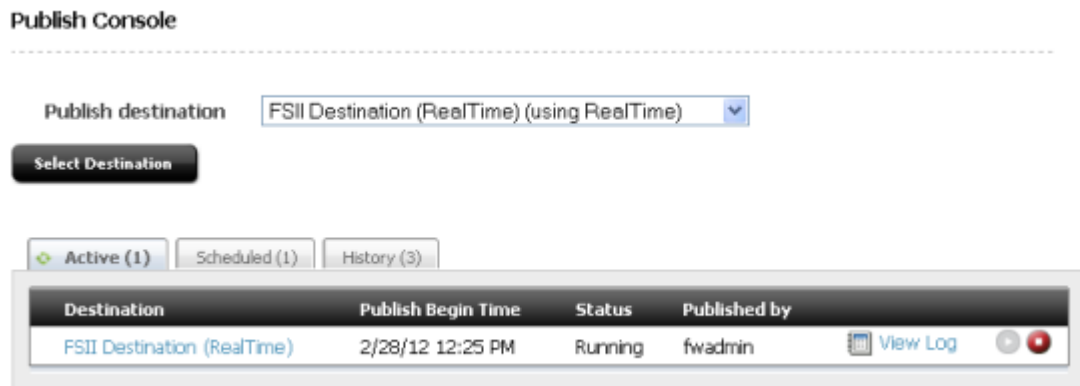
This chapter contains instructions for monitoring complete and delayed publishing sessions, adjusting publishing schedules, and retrieving the histories of past sessions.

21.2 Publish Console

Throughout RealTime publishing, you will use the Publish Console. In the Publish Console, the system communicates information about current, scheduled, and past publishing sessions for all configured publishing destinations.

To open the Publish Console, log in to the administrator's interface and click **Publishing** on the button bar.

Figure 21–1 Publish Console



- The **Active** tab shows a summary of ongoing publishing sessions. While a publishing session is underway, you can open the Publishing Status form to learn detailed information about the publishing process in real-time. You can also explore interactive on form logs. Upon completion of the publishing session, you can download text formatted logs. More information about using the functions on the Active tab can be found in [Section 21.3, "Working with Active Publishing Sessions"](#). Information on working with logs can be found in [Section 21.6.1, "Viewing and Searching Logs."](#)
- The **Schedule** tab displays the number of approved assets for each destination and the schedule of publishing sessions. More information about using the functions on the **Schedule** tab can be found in [Section 21.4, "Working with Scheduled Publishing Sessions."](#)
- The **History** tab shows summary information on prior publishing sessions. More information on working with the **History** tab can be found in [Section 21.5, "Working with Prior Publishing Sessions"](#). From this tab, you can also access logs of prior publishing sessions. For information about logs, see [Section 21.6, "Working with Publishing Logs."](#)

21.3 Working with Active Publishing Sessions

The **Active** tab of the Publish Console shows information about all publishing sessions currently running. From the Active tab, you can click a destination name to access the Publishing Status form for the selected session. On the Publishing Status form you can monitor the progress of the session as it runs.

Instructions in this section apply to both modes of publishing: complete and delayed (described in [Section 21.1, "Complete and Delayed Publishing"](#)).

This section contains the following topics:

- [Section 21.3.1, "Monitoring a Publishing Session"](#)

- Section 21.3.2, "Resuming a Delayed Publishing Session"
- Section 21.3.3, "Canceling a Publishing Session"

Note: When a delayed publishing session pauses, it remains listed in the **Active** tab. **Canceled** sessions are moved to the History tab.

21.3.1 Monitoring a Publishing Session

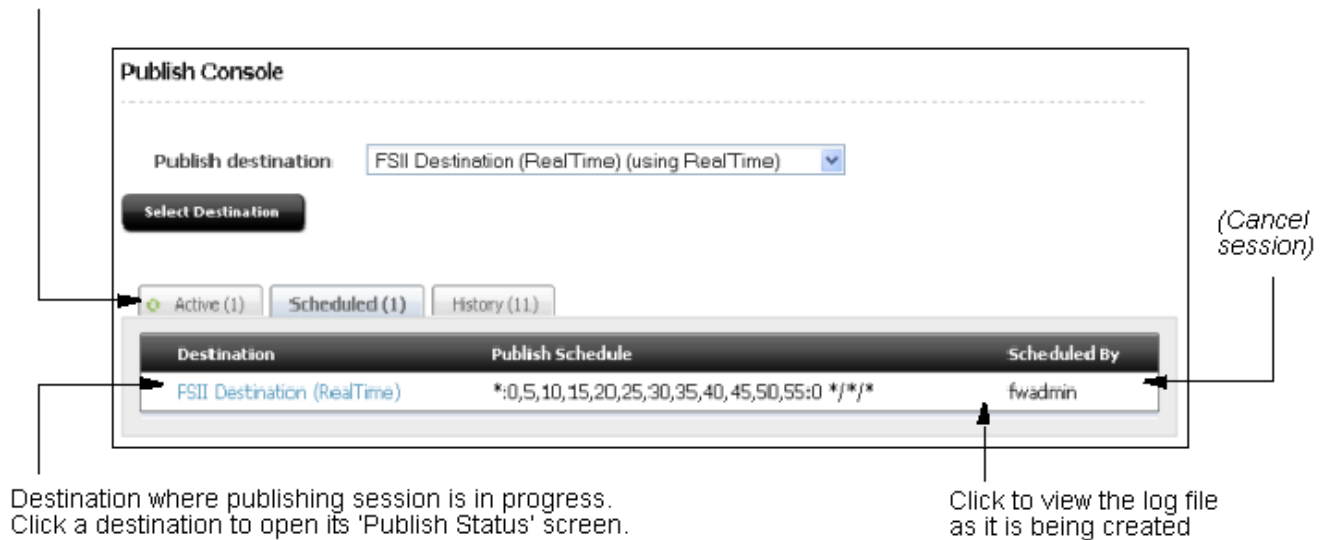
To monitor a publishing session

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

The Publish Console opens with the **Active** tab at the front.

Figure 21–2 Publish Console

The circling green arrows indicate that at least one session is in progress.
The number indicates the number of sessions that are in progress.



2. To monitor a session, stay in the **Active** tab and select the destination where the session is running.

The Publishing Status form opens.

Figure 21–3 Publishing Status

Pause over a status bar to view the stage's percent completion and last asset type processed

Session ID: 1330406433612
 Status: ...
 Destination: FSII Destination (RealTime) using RealTime (Complete publish)
 Start Time: Wednesday, February 29, 2012 3:08:17 PM PST

Stage	Status	% Completed
Gathering data to publish	Success	100%
Serializing data	Running	19%
Sending data to target	Not Started	0%
Deserializing and saving	Not Started	0%
Updating page caches	Not Started	0%

Serializing ASSET:Products:993403846823

View Log Download Log Configure Destination

Click to view the log file as it is created

Click to download a text formatted log to your computer

(Cancel session)

Notice the five status bars, one for each stage of publishing. The status bars are interactive.

- To view the percent completion of a stage, hold your cursor over the stage's status bar.
- As you hover, another tooltip opens showing the current asset type being processed. If you continue to hover, you will see the tooltip updating in real-time, always displaying the publishing system's last action.

(While a session is running, you can also view its log.)

21.3.2 Resuming a Delayed Publishing Session

Delayed publishing allows you to control when the system begins writing to the destination WebCenter Sites database. This mode of publishing can be used if you are publishing a large number of assets and wish to wait for a low traffic period on your web site before you begin saving data to the destination database.

When a session runs in delayed publishing mode, it pauses after the second stage (data serialization). Although the session pauses, no other publishing session can run

to the destination until the current session completes. For the session to complete, you must resume the session. The final stages follow automatically.

To resume a delayed

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).
2. Stay in the **Active** tab and navigate to the destination where the publishing session is running. (If the destination name is not displayed, no session is running on that destination.)

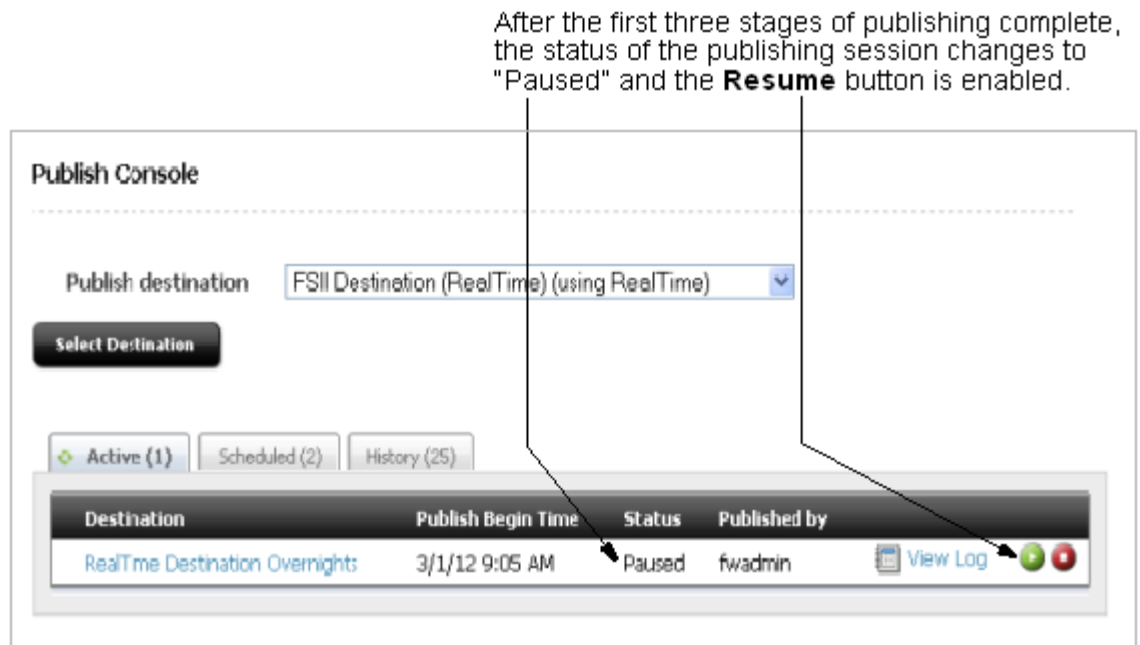
Figure 21–4 Publish Console Form, Active Tab

The screenshot shows the 'Publish Console' interface. At the top, there is a 'Publish destination' dropdown menu set to 'FSII Destination (RealTime) (using RealTime)' and a 'Select Destination' button. Below this are three tabs: 'Active (2)', 'Scheduled (1)', and 'History (5)'. The 'Active (2)' tab is selected, displaying a table with the following data:

Destination	Publish Begin Time	Status	Published by	View Log	Cancel session
FSII Destination (dynamic)	2/29/12 3:35 PM	Running	fwadmin	View Log	
FSII Destination (RealTime)	2/29/12 3:35 PM	Running	fwadmin	View Log	

Annotations in the image include an arrow pointing to the 'Active (2)' tab with the text 'When a publishing session is running, it appears in the **Active** tab of the "Publish Console"', and another arrow pointing to the 'Cancel session' button with the text '(Cancel session)'.

3. Read the destination's Status column and do one of the following:
 - If the status is paused, it means that the first three stages of publishing are complete. To continue the session, click the **Resume** button (which is enabled only during a pause).

Figure 21–5 Publish Console Form paused.

- If the status is running, you can determine which stage is in progress. Open the Publishing Status form:
 - a. Select the destination where the session is running.
 - b. In the Publishing Status form, note the progress bars **and wait for the second stage to complete**. If the second stage is not allowed to complete, the session cannot be resumed.

Note that the status bars are interactive:

To view the percent completion of a stage, hold your cursor over the stage's status bar.

As you hover, another tooltip opens showing the current asset type being processed. If you continue to hover, you will see the tooltip updating in real-time, always displaying the publishing system's last action.

(While a session is running, you can also view its log.)

- c. When the second stage is complete, click the **Resume** button.

Figure 21–6 Resume Session Icon Location on Publishing Status

Resume session


Publishing Status

Session ID: 1330406570281

Status: ...

Destination: RealTime Destination Overnights using RealTime (Delayed publish)

Start Time: Thursday, March 1, 2012 9:05:04 AM PST



Stage	Status	% Completed
Gathering data to publish	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Serializing data	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
Sending data to target	Success	<div style="width: 100%; height: 10px; background-color: green;"></div> 100%
<i>Deserializing and saving</i>	<i>Not Started</i>	<div style="width: 0%; height: 10px; background-color: grey;"></div> 0%
<i>Updating page caches</i>	<i>Not Started</i>	<div style="width: 0%; height: 10px; background-color: grey;"></div> 0%

» [View Log](#)

» [Publish History](#)

» [Download Log](#)

» [Configure Destination](#)

- d. When the publishing session is complete, all five stages display "Success" in the Status column.
4. Once a session is complete, you can access its summary information and complete log on the **History** tab of the Publish Console. To access the **History** tab:
 - a. Return to the Publish Console (click **Publishing**, on the button bar).
 - b. Click the **History** tab. More information about options on the **History** tab can be found in [Section 21.4, "Working with Scheduled Publishing Sessions."](#)

21.3.3 Canceling a Publishing Session

Ongoing publishing sessions can be canceled (and restarted) by any user with publishing rights. When a session is canceled, session information is moved from the **Active** tab to the **History** tab. Here, the session can be rerun, but only if at least the second publishing stage was completed successfully before the cancellation **and** the session is the most recently canceled.

Note: You can cancel a publishing session from the Publish Console or from the Publishing Status form. Be careful about which option you choose. If you prematurely cancel a session that you intend to redo, that session cannot be redone. When in doubt, cancel sessions from the Publishing Status form, where you can verify that at least the second stage has been completed.

To cancel a publishing session from the Publish Console

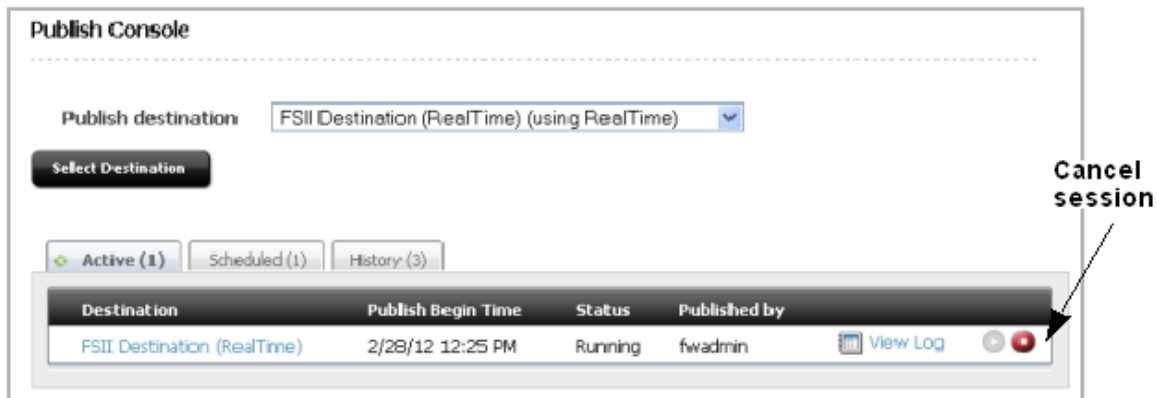
1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

Note: Use the following procedure only if you are sure that the canceled session will not be rerun. Otherwise, if you plan to rerun the session, follow the steps on page 21-9 to cancel the session from the Publishing Status form.

The Publish Console appears, displaying the active publishing sessions in the **Active** tab. (If the **Active** tab is empty, no publishing sessions are in progress.)

2. Stay in the **Active** tab, navigate to the destination where the session is running and click its **Cancel** button.

Figure 21–7 Cancel Session Icon on Publish Console Form



A message that the cancellation request has been sent appears above the list of active publishing sessions. WebCenter Sites looks for a stable stopping point; it may take a few moments for the session to cancel. Session information is then moved from the **Active** tab to the **History** tab.

To cancel a publishing session from the Publishing Status form

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

The Publish Console appears, displaying the active publishing sessions in the **Active** tab. If the **Active** tab is empty, no publishing sessions are currently in process.

Note: You must use the following procedure if you plan to rerun the session you wish to cancel. The Publish Status form enables you to monitor the publishing process to ensure that at least the second stage (data gathering and serialization) runs to completion. Canceling prior to that stage invalidates rerunning the session.

Note also that the option to redo a session is available if all of the following conditions hold:

- The first two stages of publishing were successfully completed.
 - The session is the most recently canceled publishing session.
-

2. Stay in the **Active** tab and select the destination where the session is running.
3. In the Publishing Status form, note the progress of the session and wait for at least the second stage to complete.
4. When the second (or later) stage has completed, click the **Cancel** button.

Figure 21–8 Publishing Status Form**Publishing Status**

Session ID: 1330406480153

Status: ...

Destination: FSII Destination (RealTime) using RealTime (Complete publish)

Start Time: Wednesday, February 29, 2012 7:33:02 PM PST

Stage	Status	% Completed
Gathering data to publish	Success	100%
Serializing data	Running	51%
<i>Sending data to target</i>	<i>Not Started</i>	0%
<i>Deserializing and saving</i>	<i>Not Started</i>	0%
<i>Updating page caches</i>	<i>Not Started</i>	0%

▶ [View Log](#) ▶ [Download Log](#)

▶ [Publish History](#) ▶ [Configure Destination](#)

A message appears above the status column indicating that the cancellation request has been sent. WebCenter Sites looks for a stable stopping point; it may take a few moments before the session is canceled. Session information is then moved from the **Active** tab to the **History** tab.

- When you are ready to rerun the session, select the **History** tab, navigate to the session, and click the **Redo** button. For more information, see [Section 21.5.2, "Redoing a Publishing Session."](#)

21.4 Working with Scheduled Publishing Sessions

If a destination is configured for scheduled publishing, its schedule is listed in the Schedule tab of the Publish Console. For instructions on configuring publishing schedules, see [Section 18.7, "Scheduling Publish Events."](#)

Use the Schedule tab to:

- View a destination's publishing schedule

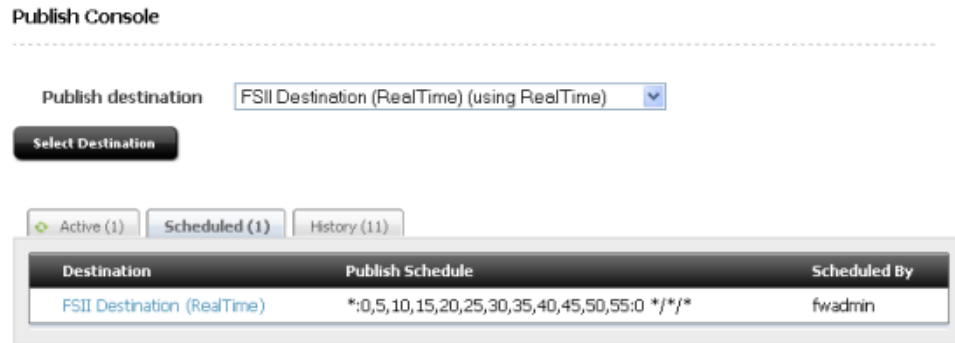
- Edit publishing schedules

To view or edit publishing schedules for all configured destinations

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).
2. Click the **Scheduled** tab.

The **Scheduled** tab displays only destinations with a publishing schedule.

Figure 21–9 Scheduled Tab on Publish Console Form



Note: A publishing schedule is expressed in the string format that is used by the SystemEvents table:

hours:minutes:seconds weekdays/days of month/months of year

For information about syntax, see [Section 18.8, "Reading the Schedule Abbreviations"](#).

3. To view a more user-friendly version of the publishing schedule, hold your cursor over the schedule.
4. To edit a schedule, click the destination name and specify the publishing conditions. (You can also disabled the schedule.)

21.5 Working with Prior Publishing Sessions

The Publishing Console's History tab provides summary information and detailed logs on every publishing session that has run in the past. Past sessions include completed, canceled, and failed sessions, but not delayed publishing sessions that have paused.

Note: When delayed publishing sessions pause, they remain listed on the **Active** tab. They are considered to be active sessions.

The History tab is interactive. You can hold your cursor over each destination to view the destination's summary information in a pop-up window, or you can select a destination to obtain its session log. To locate information quickly, you can sort the list of destinations by the following criteria: Destination, Publish End Time, Status, and Published by

On the History tab, you can:

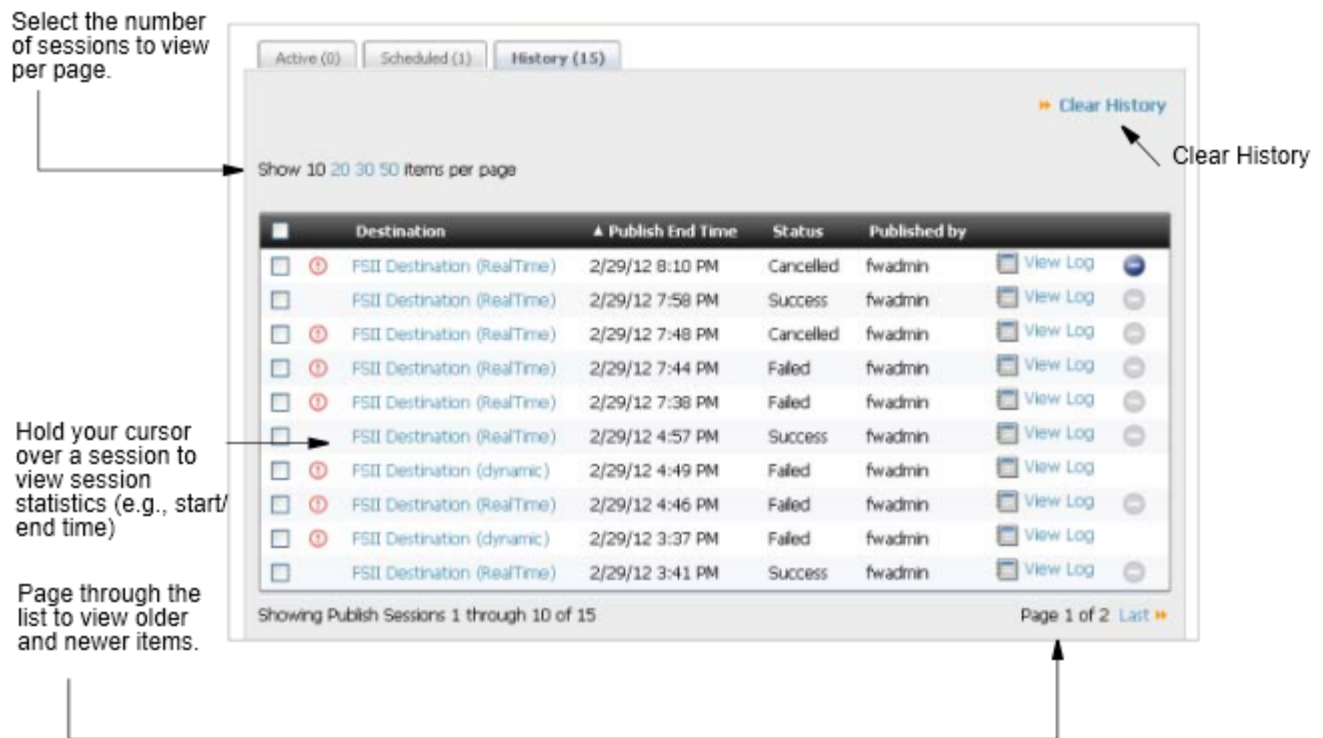
- View summary information about past publishing sessions. See [Section 21.5.1, "Viewing Past Publishing Sessions"](#)
- Access the final Publishing Status form for completed sessions. See [Section 21.5.1, "Viewing Past Publishing Sessions"](#)
- Redo a canceled session. See [Section 21.5.2, "Redoing a Publishing Session"](#)
- Access session logs
- Clear the entire history list or delete selected sessions

21.5.1 Viewing Past Publishing Sessions

To view past publishing sessions

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).
2. Click the **History** tab.

Figure 21–10 History Tab on the Publish Console Form



3. To view a session's summary information, hold your cursor over the destination where the session ran. (You can sort the **Destination** column by clicking its title.)

A pop-up window displays summary information, as shown next.

The pop-up window shows when the session started and ended, how long it took to publish the assets, and the number of assets published.

Figure 21–11 History Tab on Publish Console Form, Showing Context Window

The screenshot shows the 'History' tab in the Publish Console. At the top, there are three tabs: 'Active (1)', 'Scheduled (1)', and 'History (13)'. A 'Clear History' link is visible in the top right. Below the tabs, there is a 'Show 10 20 30 50 items per page' dropdown. The main area contains a table of publishing sessions. A context window is open over the fourth row, displaying summary statistics for that session.

Destination	Start Time	End Time	Total Time	Asset Count
FSII Destination (RealTime)	Wednesday, February 29, 2012 4:50:02 PM PST	Wednesday, February 29, 2012 4:57:20 PM PST	7 minutes 18 seconds	1655

The table below shows the list of sessions:

Destination	Time	Status	User	Action
FSII Destination (RealTime)	2/29/12 4:49 PM	Failed	fwadmin	View Log
FSII Destination (RealTime)	2/29/12 4:46 PM	Failed	fwadmin	View Log
FSII Destination (dynamic)	2/29/12 3:37 PM	Failed	fwadmin	View Log
FSII Destination (RealTime)	2/29/12 3:41 PM	Success	fwadmin	View Log
FSII Destination (RealTime)	2/29/12 3:15 PM	Success	fwadmin	View Log
FSII Destination (RealTime)	2/28/12 12:30 PM	Success	fwadmin	View Log

At the bottom, it says 'Showing Publish Sessions 1 through 10 of 13' and 'Page 1 of 2 Last'.

4. To view and download a session log:
 - a. Select the destination where the session ran.
 - b. In the Publishing Status form, click **View Log**.
If a log entry exceeds 200 characters, click the **show all** link to display the entire entry. You can also search the log for specific entries.
 - c. To download the log, click the **Download Log** link.

21.5.2 Redoing a Publishing Session

When a session stops (because of system error or user cancellation), you can rerun the session, meaning that you start the session again. When the session starts, it continues the publishing process from the last successfully completed stage. The option to redo a session is available if all of the following conditions hold:

- The data to be published was successfully gathered and serialized (that is, the first two stages of publishing were successfully completed).
- The session is the most recently failed or canceled publishing session.

The Redo option is useful when a disruption, such as a power failure, interferes with publishing.

To redo a publishing session

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).
2. Click the **History** tab to view past publishing sessions.

The list of past publishing sessions appears. The **Redo** button is active (in the right-hand column) only for the sessions that can be redone.

If the **Redo** button is disabled, then either this publishing session did not progress far enough to be redone, or it is not the most recently canceled (or failed) session.

Figure 21–12 Redo Button on History Tab

Click the **Redo** button to begin a session at stage 3 of publishing (sending data to target)

The screenshot shows a web interface with three tabs: Active (0), Scheduled (1), and History (15). The History tab is selected. Below the tabs, there is a 'Clear History' link and a 'Show 10 20 30 50 items per page' dropdown. A table lists publishing sessions with columns for Destination, Publish End Time, Status, and Published by. The first row is highlighted, and a red arrow points to the Redo button (a blue circle with a white arrow) in the right column of that row.

	Destination	Publish End Time	Status	Published by	
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 8:10 PM	Cancelled	fwadmin	View Log Redo
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 7:58 PM	Success	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 7:48 PM	Cancelled	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 7:44 PM	Failed	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 7:38 PM	Failed	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 4:57 PM	Success	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (dynamic)	2/29/12 4:49 PM	Failed	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 4:46 PM	Failed	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (dynamic)	2/29/12 3:37 PM	Failed	fwadmin	View Log
<input type="checkbox"/>	FSII Destination (RealTime)	2/29/12 3:41 PM	Success	fwadmin	View Log

Showing Publish Sessions 1 through 10 of 15 Page 1 of 2 Last

3. Do one of the following:

- Click the **Redo** button to restart the publishing session.
- Redo the session from the Publishing Status form:
 - a. Click the destination link for the session you wish to redo.

The final Publishing Status form for the session opens.
 - b. Click the **Redo** button to restart the publishing session.

The message `Restart request has been sent` appears and the session moves from the **History** tab to the **Active** tab. The circling green arrows rotate in the corner of the **Active** tab, indicating a publishing session is active.

4. To monitor the session as it completes, click the **Active** tab followed by the destination name. (For more information, see [Section 21.3, "Working with Active Publishing Sessions."](#))

21.6 Working with Publishing Logs

- [Section 21.6.1, "Viewing and Searching Logs"](#)
- [Section 21.6.2, "Downloading Logs"](#)
- [Section 21.6.3, "Deleting Publish History"](#)

21.6.1 Viewing and Searching Logs

RealTime publishing logs store a detailed account of the transactions that take place during a publishing session. The logs are viewable for active and past publishing sessions. They are also available for download, at any time, while sessions run and when they end.

Note: WebCenter Sites can be configured to generate verbose log files, containing more detailed information than regular log files. Because verbosity lengthens the publishing process, this option should be used only when needed (typically for troubleshooting).

WebCenter Sites can also be configured to automatically email text-formatted log files at the completion of a publishing session.

To configure log files and specify the email option, see [Section 20.2.4, "Create a RealTime Destination Definition on the Source System."](#)

To view publishing logs

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

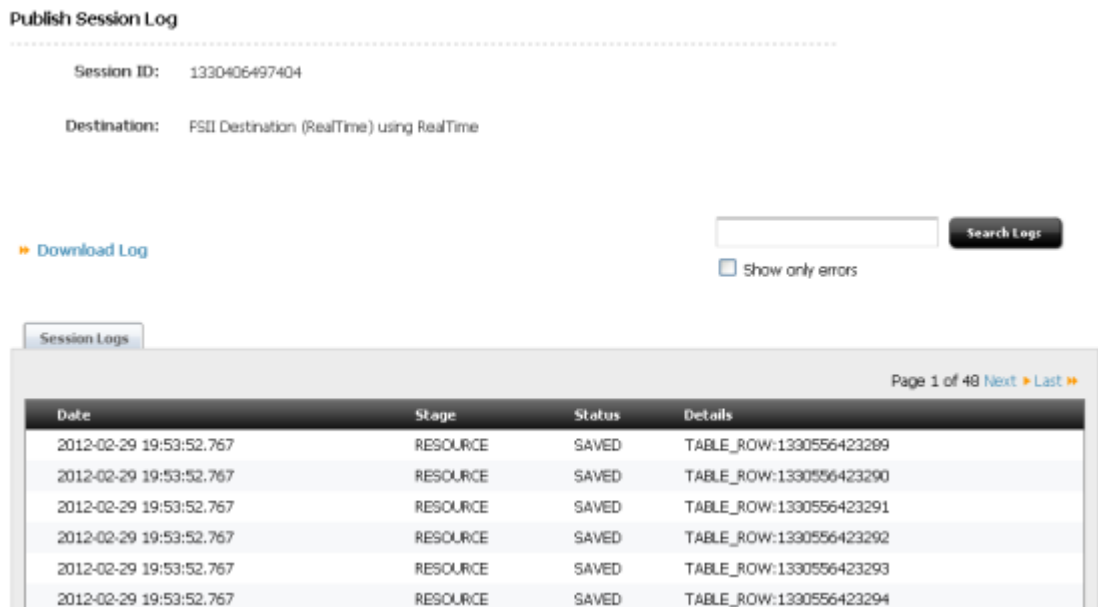
The Publish Console appears with the **Active** tab open, displaying all currently running RealTime sessions.

Figure 21–13 Active Tab on Publish Console



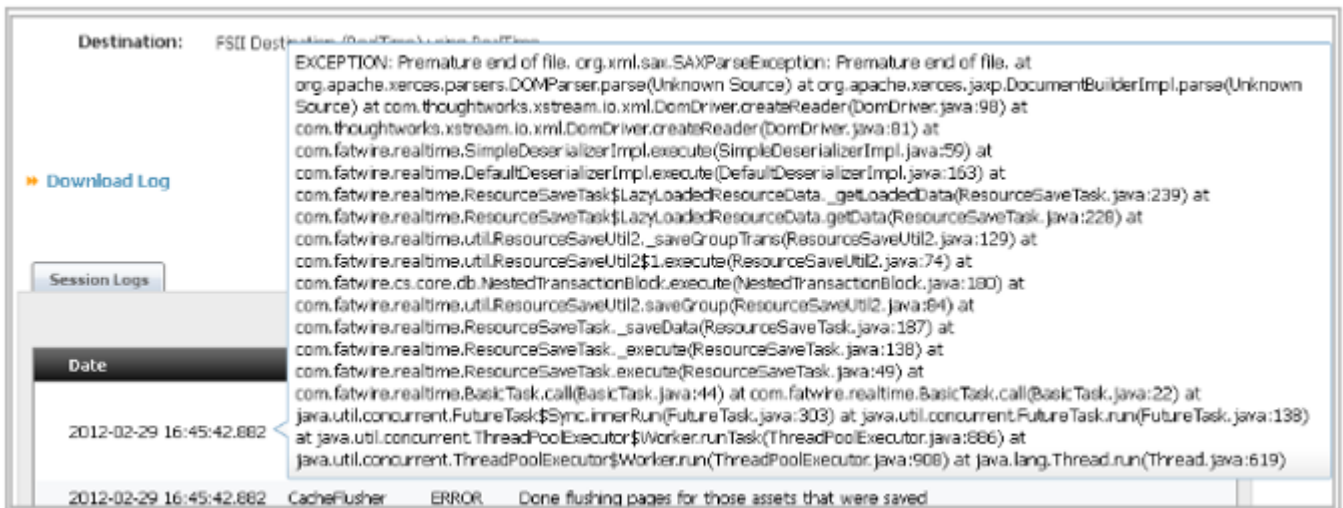
2. To view the log of a current session, stay in the **Active** tab, navigate to the session, and click **View Log**. To view the log of a past session, click the **History** tab, navigate to the session, and click **View Log**.
 - The Publish Session Log form opens. Additional pages are added as the session continues.

Figure 21–14 Publish Session Log Form



- If a log entry exceeds 200 characters, an ellipsis (three periods) will appear at the end. To see the entire entry, hover the mouse over the entry. The entire entry opens in a window that floats over the log file. Note that the cause of the log event is identified.

Figure 21–15 Pop-Over Window Showing Information for the Log Event



3. You can search the log file in several ways:
 - To search for specific entries, enter a search term in the **Search** field (at the top right of the form).

Note: The search utility recognizes terms that are used in the following columns: Stage, Status, and Details. Wildcards and boolean operators are not recognized.

- To view only error messages, select the **Show only errors** checkbox and leave the **Search** field empty.
 - To filter error messages by specific terms, select the **Show only errors** checkbox and enter a search term in the **Search** field.
- a. Click **Search**.

The **Search Results** tab opens (next to the **Session Logs** tab).

Figure 21–16 Search Results Tab on the Publish Session Log Form

Publish Session Log

Session ID: 1330406459756

Destination: FSII Destination (RealTime) using RealTime

[Download Log](#)

 Show only errors

[Session Logs](#)
[Search Results](#)

Date	Stage	Status	Details
2012-02-29 16:45:42.882	CacheFlusher	ERROR	Done flushing pages for those assets that were saved
2012-02-29 16:46:10.614	__JobMonitor__	TIMINGS	Messenger timings: global 10240ms, log writes 0ms, progress reports 204ms, asset status 0ms, send/receive 10176ms
2012-02-29 16:46:10.629	-	-	Publish failed for approved assets (error number: -508).

[Go to Publish Console](#)
[Go to Publish Status](#)

- b. To exit the **Search Results** tab, click the X on the tab.

You are returned to the **Session Logs** tab.

21.6.2 Downloading Logs

A publishing log is available for download at any time during the session and when the session ends. You can download a log file from several places in the RealTime publishing interface, as shown in the steps below.

To download a session log

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

The Publish Console opens with the **Active** tab at the front. The **Active** tab lists publishing sessions currently in progress.

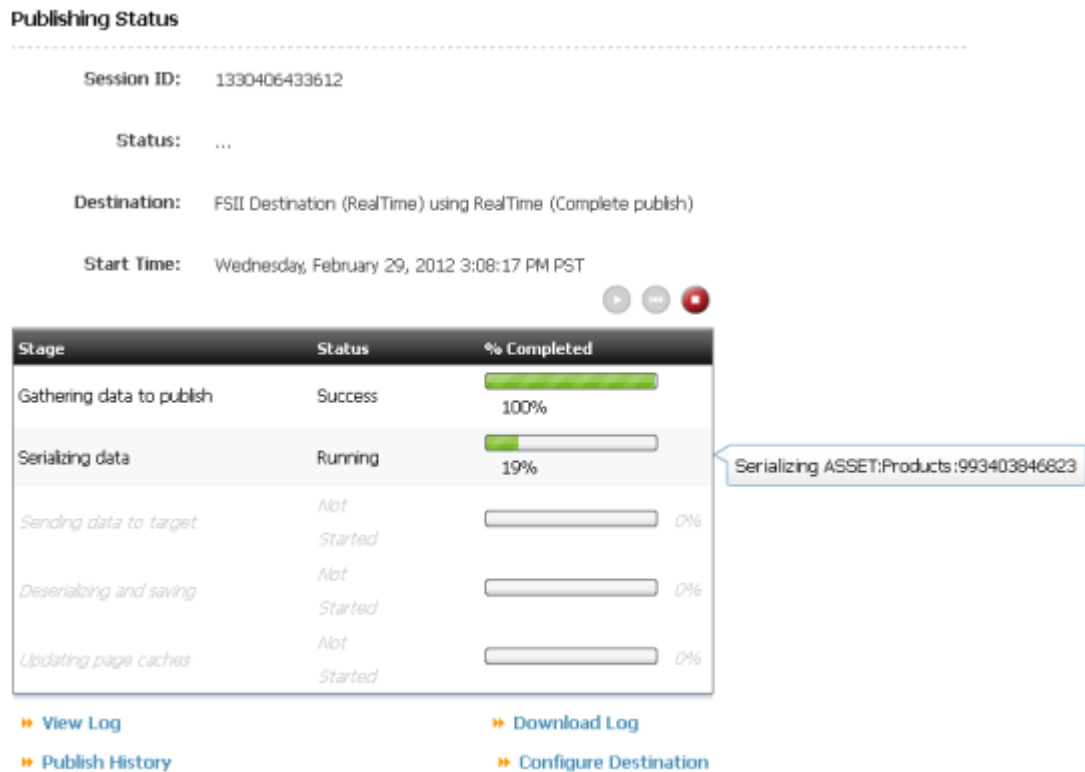
Figure 21–17 Publish Console Form



2. Download the log file for an active session or a past session:
 - **Active** tab: Select a destination name, and click **Download Log**.
 - **History** tab: Select a destination name, click **View log** and select **Download Log**.

The Publishing Status form opens.

Figure 21–18 Publishing Status Form



3. When the session is complete, click **Download Log** and follow the prompts. The log file is in text format and can be opened in any text editor.

21.6.3 Deleting Publish History

It is recommended that you delete the publish history regularly ensure proper operation and performance, as all data related to publishing is stored in the database, and the size of these tables can become a performance problem over time. While regular publishes will not greatly increase the size of this table, large publishes (such as the initial publish of a site, especially when it has failed) can leave large amounts of data behind.

Deleting a publish session will do the following:

- Delete all messages associated with the session from the database tables.
- Clear out any data associated with it from FW_PUBDATASTORE on this system (data is often left in this table when a publish fails).
- Attempt to access the destination and clear associated data from FW_PUBDATASTORE (data is often left in this table when a publish fails).

To delete a session log

1. Open the Publish Console (in the administrator's interface, click **Publishing** on the button bar).

The Publish Console opens with the **Active** tab at the front.

2. Select the History tab.

All available logs are displayed.

Figure 21–19 History Tab on Publish Console Form

The screenshot shows the 'History (1)' tab selected in the Publish Console. At the top right, there is a 'Clear History' link. Below it, a dropdown menu shows 'Show 10 20 30 50 items per page'. The main content is a table with the following columns: Destination, Publish End Time, Status, and Published by. A single row is visible with a checked checkbox in the first column, indicating it is selected for deletion. The row data is: Destination: DEST2, Publish End Time: 4/5/12 5:01 PM, Status: Success, Published by: fwadmin. To the right of the row is a 'View Log' link. Below the table, it says 'Showing Publish Sessions 1 through 1 of 1' and 'Page 1 of 1'. At the bottom left, there is a 'Delete' button.

<input type="checkbox"/>	Destination	▲ Publish End Time	Status	Published by	
<input checked="" type="checkbox"/>	DEST2	4/5/12 5:01 PM	Success	fwadmin	View Log

3. Check the box next to each log or logs you want to delete.

4. Click **Delete**.

The selected logs are deleted.

Note: As data in FW_PUBDATASTORE is stored using Large Binary Objects it may be necessary have the DBA manually clean up the Large Binary Data after performing this operation.

On Demand Publishing and Asset Unapproval

If you are using the RealTime publishing method, you can selectively publish approved assets. You can also unapprove multiple assets in bulk.

This chapter contains the following sections:

- [Section 22.1, "Using On-Demand Publishing"](#)
- [Section 22.2, "Unapproving Selected Assets from a Publishing Session"](#)

22.1 Using On-Demand Publishing

In RealTime publishing, you can publish assets on-demand. That is, you can immediately and selectively publish any of the assets on the approved assets list.

As with scheduled RealTime publishing, an asset will be published with all its dependent assets. Therefore, the dependent assets are also placed in the On-Demand Queue.

The list of approved assets is searchable, making it easier to find the assets you need to publish. After the selected assets are published, they are removed from the approved assets list for the next publishing session.

Note: On Demand publishing can be either complete or delayed. For more information about publishing modes, see [Chapter 21, "Working with RealTime Publishing Modes."](#)

To select approved assets for immediate publication

1. Open the Publish Console by clicking **Publishing** on the button bar.
2. From the Publish Destination drop-down list:
 - a. Select the RealTime publishing destination to which you will publish the assets.
 - b. Click **Select Destination**.
3. On the Publish Destination form, click the **x assets are ready for publish** link.

Figure 22–1 Publish Destination Form

Publish destination: FSII Destination (RealTime)

Destination: FSII Destination (RealTime) using RealTime

Destination address: http://localhost:8080/cs/

Arguments: REMOTEUSER=fwadmin
 VERBOSE=false
 PUBLISHOPTION=Complete
 EMAILALERT=false
 REMOTEPASS=*****

1655 assets are ready for publish.

The **On Demand Queue** tab opens. The **Approved Assets** tab lists all assets approved for the upcoming publishing session.

Figure 22–2 On-Demand Queue Tab

Approved Assets | On-Demand Queue

1655 assets available for Publishing

Show 10 20 30 50 items per page

<input type="checkbox"/>	Type	Name	Locale	Start Date	End Date	Days Expired
<input type="checkbox"/>	🔴	story1	🇺🇸	-	-	-
<input type="checkbox"/>	🔴	story2	🇺🇸	-	-	-
<input type="checkbox"/>	🔴	story3	🇺🇸	-	-	-
<input type="checkbox"/>	🔴	Frequent Visitor	🇺🇸	-	-	-
<input type="checkbox"/>	🔴	Burlington Promotion	🇺🇸	-	-	-
<input type="checkbox"/>	🔴	FSII Top Stories	🇺🇸 en_US	-	-	-
<input type="checkbox"/>	🔴	FSII Item Of The Week	🇺🇸 en_US	-	-	-
<input type="checkbox"/>	🔴	FSII Advertisement	🇺🇸 en_US	-	-	-
<input type="checkbox"/>	🔴 📄	FSII Hot Items	🇺🇸 en_US	-	9/30/09 11:59 PM	860
<input type="checkbox"/>	🔴 📄	FSII Latest News	🇺🇸 en_US	-	9/30/09 11:59 PM	860

Page 1 of 166 [Next](#) [Last](#)

- (Optional) Search through the approved assets.

- a. Enter a search term in the Search text box.
 - Search runs across asset names and descriptions.
 - You cannot use wildcard and boolean operators.
- b. Click **Search**.
The **Search Results** tab opens, displaying the results of your search.

Figure 22–3 Search Results Tab

Type	Name	Locale	Start Date	End Date	Days Expired
	BurlingtonFinancial/Site/...		-	-	-
	FSIISummary		-	-	-
	BurlingtonFinancial/Site/...		-	-	-
	OpenMarket/CSElementTemplate		-	-	-
	BurlingtonFinancial/Commo...		-	-	-
	OpenMarket/SiteEntryTemplate		-	-	-
	BurlingtonFinancial/Site/...		-	-	-
	FSIISideNav		-	-	-
	storeSchemaOct		10/1/09 12:00 AM	10/31/09 11:59 PM	850

5. Add assets to the On-Demand Queue.

Note: On-Demand supports immediate publication. Therefore, selections that you place into the On-Demand queue remain there only throughout the current WebCenter Sites session. Once you log out (or the system logs you out), the On-Demand queue is cleared.

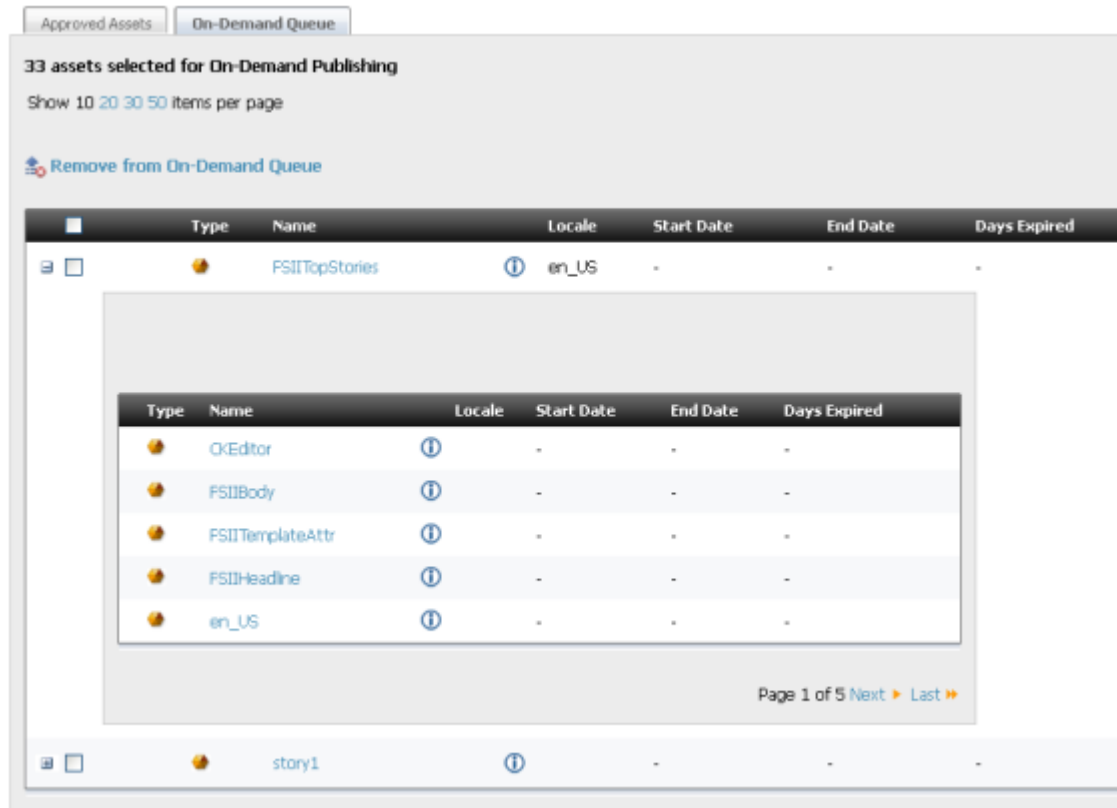
On either the **Approved Assets** tab or the **Search Results** tab, select the assets you wish to add to the On-Demand Queue, then click **Add to On-Demand Queue**.

A message is displayed while the dependencies are calculated. Once this process completes, the selected assets are removed from the **Approved Assets** tab (and **Search Results** tab, if open). The On-Demand Queue is updated with the selected assets and their dependent assets.

6. Publish the On-Demand Queue. Do one of the following:
 - If you are sure that the assets you selected can be published, click **Publish On-Demand Queue** (on the **Approved Assets** tab).
 - If you wish to review the assets before you publish them, switch to the On-Demand Queue by clicking the **On Demand Queue** tab.

- To view an asset's dependencies, click the plus mark at the far left of the asset's row.
- If you decide not to publish assets after you place them in the On-Demand Queue, select them and click **Remove from On-Demand Queue**. The assets will be removed from the On-Demand Queue and placed back in the **Approved Assets** list. (A dialog box prompts you to confirm your action.)

Figure 22-4 On-Demand Tab



7. Click **OK**.
The Publish Console opens showing the publishing session in process.
8. For instructions on observing the publishing session, see [Chapter 21, "Working with RealTime Publishing Modes."](#)

22.2 Unapproving Selected Assets from a Publishing Session

If you change your mind about publishing approved assets, you can selectively remove them from the approved assets list.

To unapprove assets

1. Open the Publish Console by clicking **Publishing** on the button bar.
2. From the Publish Destination drop-down list:
 - a. Select the RealTime publishing destination you wish to publish assets to.
 - b. Click **Select Destination**.

3. On the Publish Destination form, click the link **x assets ready for publish**.

Figure 22–5 Publish Destination Form

Publish destination: FSII Destination (RealTime)

Destination:  FSII Destination (RealTime) using RealTime

Destination address: http://localhost:8080/cs/

Arguments: REMOTEUSER=fwadmin
VERBOSE=false
PUBLISHOPTION=Complete
EMAILALERT=false
REMOTEPASS=*****

1655 assets are ready for publish.

Configure Destination...

Edit Publish Schedule...

Publish

Cancel

The On Demand Publishing form opens. This form displays the **Approved Assets** tab and the On-Demand Queue. The **Approved Assets** tab lists all assets approved for the upcoming publishing session.

Figure 22–6 Approved Assets Tab

Approved Assets On-Demand Queue

1655 assets available for Publishing

Show 10 20 30 50 items per page

[Add to On-Demand Queue](#) [Unapprove](#)

	Type	Name	Locale	Start Date	End Date	Days Expired
<input type="checkbox"/>	●	story1	ⓘ	-	-	-
<input type="checkbox"/>	●	story2	ⓘ	-	-	-
<input type="checkbox"/>	●	story3	ⓘ	-	-	-
<input type="checkbox"/>	●	Frequent Visitor	ⓘ	-	-	-
<input type="checkbox"/>	●	Burlington Promotion	ⓘ	-	-	-
<input type="checkbox"/>	●	FSII Top Stories	ⓘ en_US	-	-	-
<input type="checkbox"/>	●	FSII Item of the Week	ⓘ en_US	-	-	-
<input type="checkbox"/>	●	FSII Advertisement	ⓘ en_US	-	-	-
<input type="checkbox"/>	●	FSII Hot Items	ⓘ en_US	-	9/30/09 11:59 PM	880
<input type="checkbox"/>	●	FSII Latest News	ⓘ en_US	-	9/30/09 11:59 PM	880

Page 1 of 166 [Next](#) [Last](#)

4. (Optional) Search through the approved assets.
 - a. Enter a search term in the Search text box.
 - Search runs across asset names only.
 - You cannot use wildcard and boolean operators.
 - b. Click **Search**.
The **Search Results** tab opens, displaying the results of your search.

Figure 22–7 Search Results Tab

<input type="checkbox"/>	Type	Name	Locale	Start Date	End Date	Days Expired
<input type="checkbox"/>	🔥	BurlingtonFinancial/Site/...	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	FSIISummary	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	BurlingtonFinancial/Site/...	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	OpenMarket/CSElementTemplate	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	BurlingtonFinancial/Commo...	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	OpenMarket/SiteEntryTemplate	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	BurlingtonFinancial/Site/...	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	FSIISideNav	🇺🇸	-	-	-
<input type="checkbox"/>	🔥	storeSchemaOct	🇺🇸	10/1/09 12:00 AM	10/31/09 11:59 PM	850

5. On either the **Approved Assets** tab or the **Search Results** tab, select assets to unapprove. Click **Unapprove**.

The assets you selected are removed from the Approved Assets list.

If the asset you unapprove has a dependent asset that is approved for this publishing session, it is placed into a hold queue.

6. (Optional) Approve assets held from publishing:
 - a. Click **Back** to return to the Publish Destination form. Note that there are now two text links. One lists the assets held for publishing, the other the assets ready for publishing. Click the **x assets are held for publish** link.

Figure 22–8 Publish Destination

Publish destination: FSII Destination (RealTime)

Destination: FSII Destination (RealTime) using RealTime

Destination address: http://localhost:8080/cs/

Arguments: REMOTEUSER=fwadmin
 VERBOSE=false
 PUBLISHOPTION=Complete
 EMAILALERT=false
 REMOTEPASS=*****

10 assets are held for publish.1609 assets are ready for publish.

[Configure Destination...](#) [Edit Publish Schedule...](#)
[Cancel](#)

The held queue opens.

Figure 22–9 Assets in Held Queue

10 assets are being held from publishing

Approve	Type	Name	Locale	Start Date	End Date	Days Expired	
Held		Home		-	-	-	
Held		Portfolio		-	-	-	
Held		Stocks		-	-	-	
Held		PortfolioTop		-	-	-	
Held		StocksTop		-	-	-	
Held		FSIIHome	en_US	-	-	-	
Held		FSIILayout		-	-	-	
Held		Home (fr)	fr_FR	-	-	-	
Held		Home (de)	de_DE	-	-	-	
Held		Home (es)	es_ES	-	-	-	


- b.** Click the **Held** link for the asset you wish to approve for the publishing session.

The **Assets Preventing Asset From Being Published to Destination** form opens.



















- If the asset has no dependencies, a message appears stating that the asset is approved.
- If the asset has dependencies, a list of those dependent assets is shown.

Figure 22–10 Assets Preventing Site From Being Published Form

Assets preventing FSIIHome from being published to destination: FSII Destination (RealTime)

 This asset cannot be published until dependent assets have been approved.

You must approve the following assets for destination FSII Destination (RealTime) before FSIIHome can be published:

<input type="checkbox"/>	Asset Type	Name	Locale	Start Date	End Date	Days Expired	
<input type="checkbox"/>		FSIILatestNewsFromJan	 en_US	1/1/10 12:00 AM	-	-	
<input type="checkbox"/>		FSIIHotItemsFromJan	 en_US	1/1/10 12:00 AM	-	-	
<input type="checkbox"/>	 	FSIIHotItemsNov	 en_US	11/1/09 12:00 AM	11/30/09 11:59 PM	820	
<input type="checkbox"/>	 	Hot Items December	 en_US	12/1/09 12:00 AM	12/31/09 11:59 PM	789	
<input type="checkbox"/>	 	FSIILatestNewsNov	 en_US	11/1/09 12:00 AM	11/30/09 11:59 PM	820	

After reviewing the list of dependent assets, select all of the assets on the list by checking the check box in the header row of the table. Then click **Approve**.

The following occurs: A message indicates that the assets have been approved and are ready for publishing, the asset and its dependent assets are placed in the approved assets queue, and you are returned to the held queue.

- c. (Optional) Repeat step b to approve another held asset.

Configuring the User Interfaces

There are several WebCenter Sites user interface features that must be configured before content providers can use them.

Some of these features are alternate interfaces to the WebCenter Sites interfaces: Sites Desktop, Sites DocLink, and the Web Mode of the Contributor interface. Others change the WebCenter Sites interfaces (locale settings) and the behavior of the preview function.

This chapter contains the following sections:

- [Section 23.1, "Setting the Locale for the WebCenter Sites Interfaces"](#)
- [Section 23.2, "Sites Desktop"](#)
- [Section 23.3, "Sites DocLink"](#)
- [Section 23.4, "Web Mode"](#)
- [Section 23.5, "Maintaining Separate Browser Sessions for Preview"](#)

Note: The following features are deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Sites Desktop, Sites DocLink.

23.1 Setting the Locale for the WebCenter Sites Interfaces

Once WebCenter Sites is installed, English is set as the default locale for the system. As the administrator, you can configure a default locale for WebCenter Sites.

There are several options and properties that you use to set the locale:

- The **Locale** item in the **Admin** tab sets the default locale for the WebCenter Sites interfaces. Individual users can override the default and use a different locale (if one is present) by setting a preference.
- The `cs.emailcharset` and `cs.emailcontenttype` properties in the `futuretense.ini` file determine the character set that is used in the email messages that the workflow system sends. These properties—which must be set to a character set that can appropriately represent characters for all of the locales that your content providers can select—are usually set during installation.
- If your system is using the UTF-8 character set and you are using the Mirror to Server publishing delivery type, you must set the value for the `cs.mirrorrowsperpost` property in the `futuretense.ini` file to 4 or lower.

This section contains the following topics:

- [Section 23.1.1, "System Default Locale for the WebCenter Sites Users"](#)

- [Section 23.1.2, "System Locale Properties"](#)
- [Section 23.1.3, "Single-Language Restrictions"](#)
- [Section 23.1.4, "Locale and Asset Types"](#)
- [Section 23.1.5, "Locale and the Article Asset Type"](#)

23.1.1 System Default Locale for the WebCenter Sites Users

When the language packs are installed, the installer writes the language and country code for the locale of the language pack to the `LocaleMap` database table. For example, Portuguese Brazilian is `pt_BR`. When there is more than one locale listed in the `LocaleMap` table the administrator should specify which should be the default locale. This can be done in the following ways:

- **Set a new global default language.** You can use the WebCenter Sites Local Manager to replace the current default language for your WebCenter Sites system with a different language pack. This language will become the new default language globally on your WebCenter Sites system. You also have an option to set a different default language for individual users, but not from the Locale Manager.
- **Set a new default language on a per-user basis.** This option allows you to select a default language for individual users that is different from the global default language that you select in the Locale Manager form. See [Section 5.5, "Working with User Profiles and User Attributes"](#) for information about how to set a user's locale preference.

WebCenter Sites determines which language to use for an individual user through its universal session variable `locale`. This variable contains the language and country code associated with the user who is logged in.

When a user logs in to WebCenter Sites, WebCenter Sites obtains the value for the `locale` variable as follows:

- If the user profile has a value for the `locale` user attribute, WebCenter Sites obtains and uses it.
- If the user profile does not specify a preference, WebCenter Sites attempts to match the locale setting of that user's browser to a locale from the `LocaleMap` table.
- If the user profile does not set a preference and WebCenter Sites cannot match that user's browser locale, WebCenter Sites uses the default locale that is set for the system.

Note: When you select a site and set a new system default language from that site, the language is automatically set as the global default language for all sites on your WebCenter Sites system.

To set the system default locale

1. In the **Admin** tab, double-click **Locale**.

WebCenter Sites displays the Locale Manager form.

Figure 23–1 *Locale Manager Form***Locale Manager**

Language	Locale	Default
  English (United States)	en_US	<input checked="" type="radio"/>
  French (Canada)	fr_CA	<input type="radio"/>
  Spanish (United States)	es_US	<input type="radio"/>

Add New

2. Click the **Edit** icon next to the desired locale.
A radio button will appear on the form in the Default column.
3. Select the **Default** radio button next to the desired locale.
4. Click **Save**.

The next time users log in to WebCenter Sites, the system will be displayed in the new default language.

Note: An individual user's language preference (which is set from the User Profiles option) overrides the language preference set for the WebCenter Sites system (which is set from the Locale Manager). See [Section 5.5, "Working with User Profiles and User Attributes"](#) for information about how to set a user's locale preference.

23.1.2 System Locale Properties

The properties in the following list should have been configured during the installation of the language pack on your Sites system. They are listed here for troubleshooting purposes. For example, if the workflow emails are displaying odd characters, perhaps the email properties were not set correctly.

For instructions on how to start and use the Property Editor to verify or modify property values, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

Table 23–1 System Locale Properties

Property	Property File	Description
cs.contenttype	futuretense.ini	Determines the HTTP header character set for all WebCenter Sites pages, as well as other HTTP interaction (Sites Explorer, CatalogMover, and so on). This value is set to <code>text/html; charset=UTF-8</code> by default. This value must be set to an appropriate value for the online site that your Sites system is delivering.
cs.emailcharset	futuretense.ini	Determines the character set used in the subject lines of email messages sent by WebCenter Sites. (Typically these are workflow emails.)
cs.emailcontenttype	futuretense.ini	Determines the character set used in the body of email messages sent by WebCenter Sites.
cs.mirrorrowsperpost	futuretense.ini	Specifies the number of table rows that can be mirrored during each HTTP POST while a mirror publish session is underway. If you are using UTF-8 and your database serves non-ASCII text, this value must be set to 4 or lower.
xcelerate.charset	futuretense_xcel.ini	Determines the HTTP header character set used for all WebCenter Sites pages. This value must be set to UTF-8.

23.1.3 Single-Language Restrictions

Although you can configure a multi-lingual management system, certain parts of the user interface can be displayed in one language only.

For example, the names of tables and columns in the WebCenter Sites database as well as individual items such as categories and source codes can have one name only. This means that although much of the text on an individual WebCenter Sites form can be displayed in any of the languages that you have installed on your Sites system, items such as field names and asset type names can be displayed in one language only.

Following is a list of items in WebCenter Sites that can have one name only, which means that they can be displayed in one language only:

- Asset type names
- Field names
- Asset names
- Categories
- Source codes
- Tree tab names
- Site names

- Names of workflow building blocks (actions, email objects, conditions, states, steps, processes)
- Role names
- Start menu items—both Search and New

On a system that provides two or more languages, you must determine which language is going to be used by the majority of content providers and then use that language to name your sites, tabs, asset types, and so on.

23.1.4 Locale and Asset Types

When your site developers create asset types (either basic or flex), they can specify the field names for those asset types in whatever language they have available on the Sites system. However, there are several core columns/fields that WebCenter Sites creates for all asset types by default:

- id
- name
- description
- status
- createdby
- createddate
- updatedby
- updateddate
- startdate
- enddate
- subtype
- filename
- path
- template
- category

These column names are created in English, by default. When WebCenter Sites displays them as fields in its interfaces, it uses the language that is set as the system default language.

For descriptions of the default columns for asset types, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

23.1.5 Locale and the Article Asset Type

If your site developers decided to use the sample site asset type named Article, you must also modify the value for the `xcelerate.body.length` property in the `futuretense_xcel.ini` file. This property determines how many characters are stored in the `urlbody` column of the Article table.

Because `urlbody` is a URL column, the data entered in it is actually stored as a file outside of the WebCenter Sites database. However, the first n number of characters, where n equals the value specified for the `xcelerate.body.length` property, is also

stored in the urlbody column so that you can search for text in the body of an article asset with the search feature in the administrator's interface.

When this column holds non-ASCII text, the value for this property should be set to 500.

23.2 Sites Desktop

Sites Desktop enables Microsoft Word users to create, import, and edit Word documents as WebCenter Sites assets using the familiar Word user interface. They save these assets to, and open them from, the WebCenter Sites database.

Note: Sites Desktop supports the use of standard Word documents and templates. In Microsoft Word 2000, web page templates that enable Word users to create HTML pages using Word were introduced.

The Word web page templates are neither supported nor needed by Sites Desktop.

This section contains the following topics:

- [Section 23.2.1, "Overview"](#)
- [Section 23.2.2, "Configuring the Word Asset Types for Sites Desktop"](#)
- [Section 23.2.3, "User Accounts and Sites Desktop"](#)
- [Section 23.2.4, "Installing the Sites Desktop Client Application"](#)
- [Section 23.2.5, "Specifying Locale for the Client Application"](#)
- [Section 23.2.6, "Testing the Sites Desktop Configuration"](#)
- [Section 23.2.7, "Configuring Word Templates for the Word Assets"](#)

23.2.1 Overview

When Sites Desktop is configured for a system, users can log in to WebCenter Sites from a special toolbar in the Microsoft Word user interface.

You, the administrator, configure which asset types will be available in the Word interface. You enable the asset types for Sites Desktop and then determine which fields the Word users can enter data for. This information determines the appearance and the functionality of the WebCenter Sites toolbar when users create assets of those types in Word. Your content providers use the WebCenter Sites toolbar to insert field markers that mark the data for each field.

When someone saves the Word document as an asset, Sites Desktop sends that asset to WebCenter Sites. A temporary copy of the document is also saved locally on the user's hard drive.

The file conversion subsystem converts the binary form of the document into XML, which, in turn, is parsed into a form suitable for storing in the WebCenter Sites database. The asset is then managed as follows:

- The Word document is also stored with the asset. That document is retrieved when a user opens the asset in Word.
- When a visitor to your online site (on the delivery system) requests the asset, WebCenter Sites serves the asset, not the Word document.

Note: If you use WebCenter Sites' interfaces to edit an asset that was created in Sites Desktop, the link between the asset and Sites Desktop will be irreversibly severed when you save the asset. You will no longer be able to edit the asset in Sites Desktop. WebCenter Sites warns you of this fact when you open the asset in the Edit form. A link to the Sites Desktop client installer is also provided at that time.

This section contains the following topics:

- [Section 23.2.1.1, "Sources of Data for a Word Asset"](#)
- [Section 23.2.1.2, "Preserving the Formatting of the Word Data"](#)

23.2.1.1 Sources of Data for a Word Asset

The Microsoft Word document is the complete source for a word asset. There are several ways to enter data for an asset's field in the Word interface:

- A user enters text and then marks it with the appropriate field marker.
- Certain required fields such as **Name** and **Description** are hidden in Word. For these fields, the user selects the field name from the WebCenter Sites toolbar and then enters the data in a dialog box.

This kind of data is considered to be Word metadata.

In Word, you can select **File**, then select **Properties**, and click the **Custom** tab to see these hidden fields and other metadata that results from storing the Word asset in the WebCenter Sites database. Do not modify this data.

- Obtain information from Word template .dot files. You can create Word templates for the Word assets that provide field/value pairs for specific fields. (You can also use Word templates to create a window for your content providers to use to enter information about their Word assets.)

23.2.1.2 Preserving the Formatting of the Word Data

The administrator determines how the data from an individual field is managed by using the Preserve formatting option when you configure asset types for Sites Desktop.

When you select Preserve formatting for a field, WebCenter Sites converts the Word format to the corresponding HTML format and stores the HTML in the column that represents the field in the WebCenter Sites database. Paragraph breaks are converted to <p> tags, bold text is marked with pairs, tables are stored as HTML tables, and so on.

Therefore, it is best if you use the Preserve formatting option for fields that contain text only. Do not use the Preserve formatting option for fields that contain files (image or other types of files).

It is also best to keep text and images separate—that you do not allow the Sites Desktop users to save both text and images to the same blob attribute or upload field. For example, if you have an article asset type that contains an image with a caption, use one attribute or field for the caption and another attribute or field for the image file.

By keeping text and images in separate fields and controlling whether the formatting is preserved or not, your template writers can know what kind of data will be extracted from a particular field and can code their templates to display that data

appropriately. If you decide to allow users to save both text and image to the same blob attribute, both you and your template developers should note the following:

Table 23–2 Format Preservation from Word Document Conversion

Format Preserved	Word Content Marked to Be Saved in the Blob Attribute	How Word Content Is Stored in the Database
Yes	text and image	The image is written to the directory identified by the <code>transformer.imgdir</code> property in <code>futuretense_xcel.ini</code> . The text is stored as HTML in the URL column; the HTML has an unmanaged link to the image in the <code>transformer.imgdir</code> directory. In such a case, you must implement a process that regularly mirrors the <code>transformer.imgdir</code> directory to the delivery system because the publishing system does not mirror this directory.
Yes	text only	The text is stored as HTML in the URL column.
Yes	image only	The image file is written to the URL column.
No	text and image	The text is stripped out and the image file is written to the URL column.
No	text only	The formatting of the text is stripped out and the text is written to the URL column.
No	image only	The image file is written to the URL column.

23.2.2 Configuring the Word Asset Types for Sites Desktop

The term "Word asset type" refers to any WebCenter Sites asset type that is configured to be created and edited in Microsoft Word. It is not a new asset type, but rather, an asset type already known to WebCenter Sites that you (or another administrator) have enabled for Word (Sites Desktop).

While there are no restrictions on which asset types might fall into this category, practical limits require that you restrict the qualifying asset types to those that have text content or are otherwise document-related. When configuring asset types for Sites Desktop ensure the following:

- Enable the asset types that you want to make available through Sites Desktop.
- For each subtype for the enabled basic asset types and for each definition for the enabled flex asset types, specify which fields will appear on the WebCenter Sites toolbar within the Microsoft Word interface.
- For each field that you enable, determine whether WebCenter Sites should retain the formatting of the data that was entered in the Microsoft Word application. For fields that hold blobs, use the formatting option to determine how text and files are managed when they are written to that field.
- Create Sites Desktop start menu items for each of the Word asset types.

This section contains the following topics:

- [Section 23.2.2.1, "Enabling Asset Types for Sites Desktop"](#)

- Section 23.2.2.2, "Specifying Which Fields Are on the WebCenter Sites Toolbar"
- Section 23.2.2.3, "Creating Sites Desktop Start Menu Items"

23.2.2.1 Enabling Asset Types for Sites Desktop

You enable an asset type for Sites Desktop by adding it to those that are selected for Sites Desktop.

To enable asset types for Sites Desktop

1. In the tree, do one of the following:
 - In the **Admin** tab, expand **Sites**, then the site for which you are configuring Sites Desktop.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Expand **Sites Desktop** and double-click **Enable Sites Desktop**.

WebCenter Sites displays the Enable Asset Types for Sites Desktop form, showing asset types that have not yet been enabled for Sites Desktop in this site:

Figure 23–2 Enable Asset Types for Sites Desktop Showing Disabled Asset Types

Edit Configuration of asset type Content, subtype FSII Article for Sites Desktop

Fieldname	Description	Enable?	Preserve formatting?
Attribute_FSIIHeadline	Headline	Required	<input type="checkbox"/>
enddate	End Date	<input type="checkbox"/>	
Attribute_FSIISubheadline	Subheadline	<input type="checkbox"/>	<input type="checkbox"/>
SegRating		<input type="checkbox"/>	
Attribute_FSIIByline	Byline	<input type="checkbox"/>	<input type="checkbox"/>
template	Template	<input type="checkbox"/>	<input type="checkbox"/>
Group_FSII ContentCategory	FSII ContentCategory	Required	
Dimension		<input type="checkbox"/>	
description	Description	<input type="checkbox"/>	<input type="checkbox"/>
path	Path	<input type="checkbox"/>	<input type="checkbox"/>
name	Name	Required	<input type="checkbox"/>
filename	Filename	<input type="checkbox"/>	<input type="checkbox"/>
Attribute_FSIIPostDate	PostDate	<input type="checkbox"/>	
startdate	Start Date	<input type="checkbox"/>	
Attribute_FSIIAbstract	Abstract	Required	<input type="checkbox"/>
Publist	Publication	Required	
renderid		<input type="checkbox"/>	
Attribute_FSIIBody	Body	Required	<input type="checkbox"/>

Cancel Save

3. Select the asset types you want to enable for Sites Desktop in this site.

4. Click **Enable Asset Types**.
5. The Enable Asset Types Start Menu selection form displays.

Figure 23–3 Enable Asset Types Form

Enable Asset Types: FirstSiteII

Start Menu selection

Asset Type	<input type="checkbox"/> Available Start Menus
Attribute Editor	<input type="checkbox"/> Create Sites Desktop start menu for Attribute Editor
CSElement	<input type="checkbox"/> Create Sites Desktop start menu for CSElement
Content	CS-Desktop Content_C already enabled for this site.
Content Attribute	<input type="checkbox"/> Create Sites Desktop start menu for Content Attribute
Content Definition	<input type="checkbox"/> Create Sites Desktop start menu for Content Definition
Content Filter	<input type="checkbox"/> Create Sites Desktop start menu for Content Filter
Content Parent	<input type="checkbox"/> Create Sites Desktop start menu for Content Parent
Content Parent Definition	<input type="checkbox"/> Create Sites Desktop start menu for Content Parent Definition

6. Select the asset types to include in the Start Menu.
7. Click **Enable Asset Types**.
8. Continue to step 4 in [Section 23.2.2.2, "Specifying Which Fields Are on the WebCenter Sites Toolbar."](#)

23.2.2.2 Specifying Which Fields Are on the WebCenter Sites Toolbar

Before you begin this procedure, be sure that you have enabled the asset type(s) that you want to configure for Sites Desktop. If you have not yet enabled the asset type(s), see [Section 23.2.2.1, "Enabling Asset Types for Sites Desktop."](#)

To specify which fields of an asset type will be displayed on the WebCenter Sites toolbar in the Microsoft Word interface

1. In the tree, do one of the following:
 - In the **Admin** tab, expand **Sites**, then the site for which you are configuring Sites Desktop.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Expand **Sites Desktop**, then expand asset type you want to configure.
3. Double-click the desired subtype (basic asset types) or definition (flex and flex parent asset types).

The Configuration page for the asset opens.

4. Click **Edit Configuration**.

WebCenter Sites displays a configuration form similar to this one:

Figure 23–4 Edit Configuration of Asset Type Form

Edit Configuration of asset type Content, subtype FSII Article for Sites Desktop

Fieldname	Description	Enable?	Preserve formatting?
Attribute_FSIIHeadline	Headline	Required	<input type="checkbox"/>
enddate	End Date	<input type="checkbox"/>	
Attribute_FSIISubheadline	Subheadline	<input type="checkbox"/>	<input type="checkbox"/>
SegRating		<input type="checkbox"/>	
Attribute_FSIIByline	Byline	<input type="checkbox"/>	<input type="checkbox"/>
template	Template	<input type="checkbox"/>	<input type="checkbox"/>
Group_FSII ContentCategory	FSII ContentCategory	Required	
Dimension		<input type="checkbox"/>	
description	Description	<input type="checkbox"/>	<input type="checkbox"/>
path	Path	<input type="checkbox"/>	<input type="checkbox"/>
name	Name	Required	<input type="checkbox"/>
filename	Filename	<input type="checkbox"/>	<input type="checkbox"/>
Attribute_FSIIPostDate	PostDate	<input type="checkbox"/>	
startdate	Start Date	<input type="checkbox"/>	
Attribute_FSIIAbstract	Abstract	Required	<input type="checkbox"/>
Publist	Publication	Required	
renderid		<input type="checkbox"/>	
Attribute_FSIIBody	Body	Required	<input type="checkbox"/>

Cancel Save

5. In the Edit Configuration form, select the **Enable** check box for each field you want to appear in the Sites Desktop toolbar in Word. As a general rule, select only text-entry fields. Note that required fields are pre-elected and cannot be deselected.
6. (Optional) If you want the format of the text that is entered in a field through Word to be preserved when that data is stored in the WebCenter Sites database, select the **Preserve formatting** option for that field.
As a general rule, enable the **Preserve formatting** option only for fields that store text. Do not enable the **Preserve formatting** option for fields that hold uploaded files.
7. Click **Save**.
8. (Optional) Repeat steps 3 – 7 for each additional subtype or definition you want to configure.
9. (Optional) Repeat steps 2 – 8 for each additional asset type you want to configure.

23.2.2.3 Creating Sites Desktop Start Menu Items

The subtype configuration determines which fields are available on the WebCenter Sites toolbar in the Microsoft Word interface, but your Word asset types do not become available in Word until you create start menu items for them.

To create Sites Desktop start menu items

1. In the **Admin** tab, expand **Asset Types**, then the desired asset type.
2. Under the selected asset type, expand **Start Menu** and double-click **Add New**.
3. In the **Name** field, enter a short, descriptive name for the item. This is the name that is displayed in the list of asset types that are available from the Sites Desktop toolbar in Word.
4. In the **Description** field, enter a short, informative description for the start menu item.
5. In the **Type** field, select **Sites Desktop**.

Note: You cannot set default field values for assets whose source is Word. Ignore the Default Values section when creating start menu items for Word asset types.

6. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl-click** to select more than one site.
7. In the **Roles** field, select all the roles that can have access to the start menu item. Use **Ctrl-click** to select more than one role.
8. (Optional) To configure workflow process details for the assets that are created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the Select Workflow form, select the appropriate workflow process from the drop-down list and click **Select Workflow**.
 - c. In the Set Participants form, select at least one user for each role that appears and then click **Set Participants**.
 - d. Click **Continue**.

WebCenter Sites saves your changes and redisplay the Start Menu form.
9. In the Start Menu form, click **Save**.
10. Repeat steps 1–9 for each additional asset type you want to configure.

23.2.3 User Accounts and Sites Desktop

Like the content providers who use the WebCenter Sites interfaces, the content providers who use the Sites Desktop application must have WebCenter Sites user accounts as well. User accounts for the individuals who use Sites Desktop must have the following ACLs:

- Browser
- ElementReader
- PageReader
- RemoteClient

- xceleditor
- UserReader
- Visitor

For information about creating user accounts, see [Section 5.4.3, "Editing a User,"](#) and [Section 31.4, "Required ACLs for Custom Users."](#)

Give the Sites Desktop users the following information:

- Their user name/password combination
- The appropriate URL for logging in to your Sites management system

They need both pieces of information to log in to WebCenter Sites through Sites Desktop.

The WebCenter Sites URL uses the following convention:

```
http://<server>:<port>/<context>
```

where:

- <server> is the host name or IP address of the machine running WebCenter Sites.
- <port> is the port on which WebCenter Sites is listening for connections.
- <context> refers to the application context root (URI) assigned to the WebCenter Sites application when it was installed.

23.2.4 Installing the Sites Desktop Client Application

In order to use the Sites Desktop feature, the Sites Desktop client must be installed on the client machine.

Note: Before installing Sites Desktop, make sure that the client machine:

- Is running supported versions of Microsoft Windows and Word, and
 - Supports the entry and display of text using the UTF-8 character set.
-
-

To install the Sites Desktop client

1. In the shipped zip file, the Sites Desktop installer is located on the following path:
ContentServer/Xcelerate/cs_client
where there are /32bit and /64bit folders to choose depending on your system.
2. Extract the downloaded archive to a temporary directory and open that directory.
3. Launch the Sites Desktop installer (.msi) and follow the instructions it displays.

Note: If one or more language packs are installed on your Sites system, the page lists localized versions of the Sites Desktop client corresponding to the installed language packs. In such case, download the version appropriate to the user's language preference. (Language packs are available with patchset 1)

23.2.5 Specifying Locale for the Client Application

The Download page is generated dynamically based on the language packs that are installed on your Sites system. If your Sites system has a language pack installed, the Download page presents a Sites Desktop option in that language and the `csdesktop.exe` file installs a version of the help file in that language.

The first time a Sites Desktop user logs in to WebCenter Sites, the Log In dialog box is in English. The user clicks the Select Language link to select the language that matches the version of Sites Desktop help file that he or she installed.

After a user selects a language, Sites Desktop copies the appropriate XML language file from WebCenter Sites to that user's hard drive. Sites Desktop uses the information in that language file for menu names and dialog box text until the user selects a different language.

Additionally, whenever the language file on the WebCenter Sites server is updated, be sure to remind your Sites Desktop users to update their language file on their hard disks by using the Select Language link again the next time they log in to WebCenter Sites.

23.2.6 Testing the Sites Desktop Configuration

To determine whether your Sites Desktop application is configured correctly, you must create and save an asset from the Microsoft Word interface and then examine it through WebCenter Sites.

To determine if Sites Desktop is configured correctly

1. Open Microsoft Word.
2. On the button bar, select **WebCenter Sites**, then select **Login**.
3. In the **Login** dialog box, enter the following information:
 - The name and password of a user who has the RemoteClient ACL.
 - The server URL, as follows:
`http://<server>:<port>/<context>`
where:
 - `<server>` is the host name or IP address of the machine running WebCenter Sites.
 - `<port>` is the number of the port on which WebCenter Sites is listening for connections.
 - `<context>` refers to the application context root (URI) assigned to the WebCenter Sites application when it was installed.
4. Select the site, asset type, and subtype, as necessary, to set the field markers list.
5. Enter text in the Word document.
6. Select (highlight) the entire range of text.
7. Click the **Field Markers** button on the toolbar and select the marker for the field that you are entering text for.
8. Repeat steps 5 – 7 for all required asset fields that require you to enter text in the Word document.

9. Set values for the field markers containing ellipses (...). These field markers represent hidden fields; that is, their values do not appear in the Word document, but are part of the asset definition, and are thus stored in the WebCenter Sites database. When you click a field marker with an ellipsis, a dialog box opens. Enter the appropriate data.
10. In the button bar, select **WebCenter Sites**, then **Save to WebCenter Sites**.
11. Log in to the administrator's interface as a user who has access to the type of asset that you just created.
12. Select the site that you created the asset for and then click **Search**.
13. Find the asset that you entered through Sites Desktop.
14. Click **Inspect**. (Do not select **Edit**.)
15. Examine the data listed for the fields and verify that the data that you marked in Sites Desktop matches the data displayed in the Inspect form.

23.2.7 Configuring Word Templates for the Word Assets

The field markers on the WebCenter Sites toolbar are Word bookmarks. Therefore, you can create .dot files for your Word assets, marking the appropriate areas of those templates with either the WebCenter Sites field markers or your own Word bookmarks.

Create a new Word document and mark placeholder text with the appropriate field markers, just as you would for an asset of that type (that is, be sure that you do not mark text for the metadata fields such as Name and Description and so on). Then save the document as a template (a .dot file) rather than to WebCenter Sites.

23.3 Sites DocLink

Oracle WebCenter Sites DocLink (Sites DocLink) provides a drag-and-drop interface for uploading and downloading documents, graphics, or other files that are managed as flex assets by WebCenter Sites. This application presents the hierarchical data structure of the flex parents and flex assets in your WebCenter Sites database as folders and files in the Windows Explorer application.

This section contains the following topics:

- [Section 23.3.1, "Overview"](#)
- [Section 23.3.2, "Configuring the Sites DocLink Asset Types"](#)
- [Section 23.3.3, "Users and Sites DocLink"](#)
- [Section 23.3.4, "Installing the Sites DocLink Client Application"](#)
- [Section 23.3.5, "Testing the Sites DocLink Configuration"](#)

23.3.1 Overview

Sites DocLink enables your content providers to use their third-party tools to create content for your online sites, and then save it to the WebCenter Sites database without having to log in to the WebCenter Sites interfaces.

Sites DocLink enables content providers to drag and drop Word files, Excel spreadsheets, graphic files, and so on—that is, single binary files—into the WebCenter Sites database by dropping each file onto the appropriate Microsoft Windows Explorer

folder that represents the asset type or parent asset type that the document should be saved as.

After a content provider drops the file onto a folder, Sites DocLink displays a dialog box that prompts the user to enter any metadata that is required, depending on how you, the administrator, have configured your system. Examples of such metadata include name, description, and so on.

Sites DocLink differs from Sites Desktop for Word as follows:

- With Sites Desktop, content providers produce the content and tag the data in that document that should be inserted into various fields from within the Word interface. The tagging process creates a structured asset when the document is stored in the database as an asset.
- With Sites DocLink, content providers produce their content in their third-party applications but they do not use those applications to provide structure to the data.

They drop a binary file into the Windows Explorer interface, and then provide information about the file through the Sites DocLink dialog boxes that appear in response to the drop. The information required depends on the kind of asset type the file is going to be stored as, based on the place in the graphical representation of the database that the user dropped the file.

23.3.2 Configuring the Sites DocLink Asset Types

The term Sites DocLink asset type refers to a flex or flex parent asset type that is configured to be accepted by Sites DocLink. Sites DocLink asset types must be flex asset types that store an uploaded, binary file. That is, one (and only one) of the flex attributes that define the asset must be of type blob. When configuring the Sites DocLink asset types, complete the following:

- Verify that the flex or flex parent asset type is enabled for the site. Be sure that the asset type has **one** flex attribute of type **blob**.
- Enable the flex or flex parent asset type for Sites DocLink.
- For each definition for those enabled asset types, specify which fields will be available in Sites DocLink and specify which field is the upload field.
- Create Sites DocLink start menu items for each of the Sites DocLink asset types.

This section contains the following topics:

- [Section 23.3.2.1, "Enabling Asset Types for Sites DocLink"](#)
- [Section 23.3.2.2, "Specifying Which Fields Are Available in Sites DocLink"](#)
- [Section 23.3.2.3, "Creating Sites DocLink Start Menu Items"](#)

23.3.2.1 Enabling Asset Types for Sites DocLink

You enable an asset type for Sites DocLink by adding it to those that are selected for Sites DocLink.

To enable asset types for Sites DocLink

1. In the tree, do one of the following:
 - In the **Admin** tab, expand **Sites**, then the site for which you want to enable Sites DocLink.

- If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Under the selected asset type, expand **Sites DocLink**, then double-click **Enable Sites DocLink**.

WebCenter Sites displays the Enable Asset Types form, listing asset types that have not yet been enabled for Sites DocLink in the selected site.

Figure 23–5 Enable Asset Types for Sites DocLink Form

Enable Asset Types for Sites DocLink: FirstSiteII

Enable for Sites DocLink:

	Name	Description
<input type="checkbox"/>	Content_C	Content
<input type="checkbox"/>	Product_C	Product
<input type="checkbox"/>	FSIIVisitor	Site Visitor

3. Select the check boxes next to the asset types that you want to enable for Sites DocLink in the selected site.
4. Click **Enable Asset Types**.
5. The Start Menu Selection form appears. Select the items to include in the Start Menu.

The enabled asset types (the items selected on the Enable Asset Types form in step 2) appear in the **Admin** tab under the Sites DocLink node for the selected site.

6. Continue to step 3 in [Section 23.2.2.2, "Specifying Which Fields Are on the WebCenter Sites Toolbar."](#)

23.3.2.2 Specifying Which Fields Are Available in Sites DocLink

Before you begin this procedure, be sure that the following conditions are true:

- You have enabled the asset type that you want to configure for Sites DocLink. If you have not yet enabled the asset type, see [Section 23.3.2.1, "Enabling Asset Types for Sites DocLink."](#)
- The asset type that you want to configure has one attribute of type blob.

To specify which of an asset type's fields will be available to content providers who are using Sites DocLink to create or edit assets of that type

1. In the tree, do one of the following:
 - In the **Admin** tab, expand **Sites**, then the site for which you are configuring Sites DocLink.

- If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Under the selected site, expand **Sites DocLink**, then expand the desired asset type and double-click the subtype.

The configuration form for the asset subtype is displayed.

3. Click **Edit Configuration**.

WebCenter Sites displays a form similar to this one:

Figure 23–6 Configuration of Asset Type for Sites DocLink Form

Configuration of asset type Media_C, subtype FSII_Image for Sites DocLink

Fieldname	Description	Enable?	Accepts Documents?
enddate	End Date	<input checked="" type="checkbox"/>	-
Attribute_FSII_ImageFile	Image File	<input checked="" type="checkbox"/>	- + + (No size restriction)
SegRating		<input type="checkbox"/>	-
template	Template	<input type="checkbox"/>	-
Attribute_FSII_AltText	Alt Text	<input checked="" type="checkbox"/>	-
Dimension		<input checked="" type="checkbox"/>	-
Group_FSII_ImageCategory	FSII_ImageCategory	<input type="checkbox"/>	-
description	Description	<input checked="" type="checkbox"/>	-
path	Path	<input type="checkbox"/>	-
name	Name	Required	-
filename	Filename	<input type="checkbox"/>	-
startdate	Start Date	<input checked="" type="checkbox"/>	-
Relationships		<input type="checkbox"/>	-
Publist	Publication	<input type="checkbox"/>	-
renderid		<input type="checkbox"/>	-

Select Document Types

Cancel Save

4. In the Configuration of Asset Type form, select the **Enable** option for each field that you want to be displayed in the Sites DocLink interface. An enabled field means that users can view and edit the data in that field.

As a general rule, enable the following kinds of fields:

- The blob attribute that will hold the file (just one)
- Text entry fields (**Description**, for example)

Note:

- Do **not** enable the **Publist** field—assets cannot be shared with other sites through the Sites DocLink interface.
- If users are required to enter a value into a field, they will be prompted for the value **when they create the asset**—whether or not you have selected the **Enable** option for that field. If you wish to have a required field write-protected once the asset is saved, do not enable the field. When Sites DocLink users right-click the asset and select **Properties**, the required field will not be displayed to them.

5. Click **Select Document Types** next to the upload field (flex attribute) you want to specify as the field in which documents are to be stored.

Note: If your developers have created a flex filter for assets of this type, the flex attribute that you specify as the upload field must match the field that is designated as the input attribute for the flex filter.

For information about flex filters, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* and talk to your developers.

WebCenter Sites displays a form similar to this one:

Figure 23–7 Configuration of Asset Type Form - Select Document Types

Configuration of asset type Media_C, subtype FSII_Image for Sites DocLink

Fieldname	Document Types	Max file size
Image File	<ul style="list-style-type: none"> Any BMP image file CSS StyleSheet HTM page HTML page Image in GIF format 	<input type="text"/> bytes

6. In the **Document Types** field, select the types of documents that this upload field can store. Use **Ctrl-click** to select more than one.

The values displayed in the **Document Types** field come from the `description` column of the entries in the `MimeType` table. If the document type that you want to select does not appear in the list, use Sites Explorer to add it to the `MimeType` table.

Note: When you select the Any option, it means that Sites DocLink accepts any type of file for assets of this type, whether or not there is an entry for it in the MimeType table.

However, when there is no entry for a file type in the MimeType table, a file of that type cannot be displayed correctly in the WebCenter Sites interfaces.

7. In the **Max file size** field, specify a size limit (if there is one) for the documents that can be stored in this upload field.
8. Click **Save**.
9. In the Configuration of Asset Type form, click **Save** again.
10. Repeat steps 2 – 8 for each additional subtype or definition you want to configure.
11. Repeat steps 2 – 10 for each additional asset type you want to configure.

23.3.2.3 Creating Sites DocLink Start Menu Items

The Sites DocLink configuration determines which fields are available in the Sites DocLink interface, but your content providers cannot create assets of this type in Sites DocLink until you create start menu items for those asset types.

To create Sites DocLink Start Menu items

1. In the **Admin** tab, expand **Asset Types**, then the desired asset type.
2. Under the selected asset type, expand **Start Menu** and double-click **Add New**.
3. In the **Name** field, enter a name for the item.
4. In the **Description** field, enter a short, informative description for the start menu item.
5. In the **Type** field, select **Sites DocLink**.

Note: You cannot set default field values for assets whose source is Sites DocLink in this version. Ignore the Default Values section when creating Sites DocLink start menu items.

6. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl+click** to select more than one site.
7. In the **Roles** field, select all the roles that can have access to the start menu item. Use **Ctrl+click** to select more than one role.
8. (Optional) To configure workflow process details for the assets that are created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the Select Workflow form, select the appropriate workflow process from the drop-down list and then click **Select Workflow**.
 - c. In the Set Participants form, select at least one user for each role that appears and then click **Set Participants**.
 - d. Click **Continue**.

WebCenter Sites saves the workflow information and redisplay the Start Menu form.

9. In the Start Menu form, click **Save**.
10. Repeat this procedure for each additional asset type you want to configure.

23.3.3 Users and Sites DocLink

Like the content providers who use the WebCenter Sites interfaces, the content providers who use the Sites DocLink application must have WebCenter Sites user accounts as well. Accounts for users who use Sites DocLink must have the following ACLs:

- Browser
- ElementReader
- PageReader
- RemoteClient
- UserReader
- xceleditor

For information about creating user accounts, see [Section 5.4.1, "Creating a New User."](#)

Give the Sites DocLink users the following information:

- Their user name/password combination
- The appropriate URL for logging in to WebCenter Sites

They need both pieces of information to log in to WebCenter Sites through Sites DocLink.

The WebCenter Sites URL uses the following convention:

```
http://<server>:<port>/<context>
```

where:

- <server> is the host name or IP address of the machine running WebCenter Sites.
- <port> is the port on which WebCenter Sites is listening for connections.
- <context> refers to the application context root (URI) assigned to the WebCenter Sites application when it was installed.

23.3.4 Installing the Sites DocLink Client Application

In order to use the Sites DocLink feature, the Sites DocLink client must be installed on the client machine.

To install the Sites DocLink client

1. In the shipped zip file, the Sites DocLink installer is located on the following path:
ContentServer/Xcelerate/cs_client
where there are /32bit and /64bit folders to choose depending on your system.
2. Extract the downloaded archive to a temporary directory and open that directory.
3. Launch the Sites DocLink installer (.msi) and follow the instructions it displays.

Note: If one or more language packs are installed on your WebCenter Sites installation, the page lists localized versions of the Sites DocLink client corresponding to the installed language packs. In such case, download the version appropriate to the user's language preference. (Language packs are available with patchset 1)

23.3.5 Testing the Sites DocLink Configuration

To determine whether your Sites DocLink application is configured correctly, you must create and save an asset from the Sites DocLink interface and then examine it through WebCenter Sites.

To test the Sites DocLink configuration

1. Open WebCenter Sites Explorer.
2. Double-click **Sites DocLink**.
3. In the **Login** dialog box, enter the following information:
 - The Sites DocLink URL. See [Section 23.3.3, "Users and Sites DocLink"](#) for the appropriate URL for your system.
 - The user name and password of a user who has the necessary ACLs (listed in [Section 23.3.3, "Users and Sites DocLink"](#)).
4. Drag a document from the file system to the file structure represented under the Sites DocLink node and drop it in the appropriate place.
5. If there are required fields that must be filled in, the **Properties** dialog box appears. Enter the required information and save the asset.
6. Open your browser and log in to the administrator's interface as a user who has access to the type of asset that you just created.
7. If prompted, select the site in which you created the asset.
8. Find the asset that you created through Sites DocLink and open it in the Inspect form.
9. Examine the data listed for the fields and verify that the data that you specified in Sites DocLink matches the data displayed in the Inspect form.

23.4 Web Mode

Web Mode is a WebCenter Sites feature that enables infrequent users or users who perform a limited role to find, edit, and submit content directly from the rendered (Preview) version of an asset.

Enabling this feature requires three general steps:

- You, the administrator, set the `xcelerate.enableinsite` property in the `futuretense_xcel.ini` file to `true` on the Sites management system.
- Your developers code templates that invoke the Web Mode feature for the fields that you want content providers to be able to edit in this way. Note that these same templates do not display Web Mode on the WebCenter Sites delivery system because the `xcelerate.enableinsite` is set to `false` on that system.
- You determine the appropriate URL and then make it available to the WebCenter Sites users who need to use Web Mode.

To enable Web Mode

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. On the **xcelerate** tab, select `xcelerate.enableinsite` and set its value to `true`.
3. Save the property file and close the Property Editor.

To configure your servlet and host name used in Web Mode and Preview

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. On the **xcelerate** tab, do one or both of the following:
 - a. Set the `xcelerate.previewhost` property as
`http://<server>:<port>/servlet/`
 - b. Change the value of the `xcelerate.previewervlet` property from `ContentServer` to `Satellite`

For Example:

- If content providers are not required to include the port number in the URL that they use to log in to WebCenter Sites, include it in the value for the `xcelerate.previewhost` property.
- If content providers are required to include the port number, specify a different port number for the property. Use either the web server port number or the port number of a redirect request to the web server.
- Create an alias and use it only for the `xcelerate.previewhost` property.
- Use the `xcelerate.previewervlet` property to specify the Satellite Server servlet rather than the WebCenter Sites servlet.

23.5 Maintaining Separate Browser Sessions for Preview

If the online site that you are serving from your WebCenter Sites delivery system requires visitors to log in, or it logs them in by default, your content providers need a separate browser session for preview if they must log in to the previewed site on the Sites management system with a user name that is different from the one they are using in WebCenter Sites.

Without a new, separate browser session for preview, the WebCenter Sites session ends when the preview session begins, which means that content providers can lose any unsaved work.

You can do the following to ensure that a preview session does not end the WebCenter Sites session:

- Have your site designers configure the online site so that your content providers are logged in to the site that they are previewing with their WebCenter Sites user accounts.
- Configure your Sites management system so that the web server opens a new browser session for the preview window without closing the current session in the WebCenter Sites window.

To configure your Sites management system so that the web server opens a new browser session for the previewed asset, you configure properties in the `futuretense_xcel.ini` file: `xcelerate.previewhost` and `xcelerate.previewervlet`.

You use these properties to point to the URL for the Sites management system, but the URL must be different in some way from the URL that your content providers use to log in to the management system. For example:

- If content providers are not required to include the port number in the URL that they use to log in to WebCenter Sites, include it in the value for the `xcelerate.previewhost` property.
- If content providers are required to include the port number, specify a different port number for the property—the web server port number or the port number of a redirect request to the web server.
- Create an alias and use it only for the `xcelerate.previewhost` property.
- Use the `xcelerate.previewervlet` property to specify the Satellite Server servlet rather than the WebCenter Sites servlet.

Start the Property Editor, open the `futuretense_xcel.ini` file, select the `xcelerate` tab, and do one or both of the following:

- Set the `xcelerate.previewhost` property as follows:
For most application servers, including Sun JES Application Server:
`http://<server>:<port>/servlet/`
- Change the value of the `xcelerate.previewervlet` property from `ContentServer` to `Satellite`

Configuring Vanity URLs

Starting with Oracle WebCenter Sites 11.1.1.8.0, WebCenter Sites allows for the creation of vanity URLs for all asset types. Vanity URLs can be short, readable, and easy manage URLs. They are usually made up of human readable information often including the name, or other descriptive information about the content of the web page the vanity URL points to.

Vanity URLs do not replace the existing URLs in the product. They exist alongside of the existing URLs, as either may be used to access the content. A vanity URL can be auto-generated as well as custom created by a content contributor in the WebCenter Sites Contributor interface.

An example "normal" WebCenter Sites URL could be

```
http://www.example.com/cs/Satellite?c=Page&cid=3451344545&pagename=avisports/Page/HomeLayout
```

where the vanity URL for the same asset could be

```
http://www.example.com/Running/Home
```

This shows how the Vanity URL is useful, being both simpler to type (particularly on a Mobile device), as well as easier for a user to read and recall later.

Vanity URLs consist of two parts. First, the WebRoot, which is the control of how the vanity URL is interpreted in different environments. The other part of a vanity URL is the URL Pattern or Free Form entry. The URL Pattern is a rule defined for an entire asset type, while the Free Form entry is created individually by the content contributor for each asset.

This chapter contains the following sections:

- [Section 24.1, "WebRoots"](#)
- [Section 24.2, "Generating Vanity URLs"](#)
- [Section 24.3, "Using Vanity URL with System Tools"](#)
- [Section 24.4, "Resolving Vanity URLs Using a Rewriter Filter"](#)
- [Section 24.5, "Vanity URL for Mobile Sites"](#)
- [Section 24.6, "Using the Web Server with Vanity URLs"](#)
- [Section 24.7, "Using mod_rewrite with Vanity URLs"](#)

24.1 WebRoots

The WebRoot is used to control how the vanity URL is interpreted. WebRoots are like assets. Once configured, a WebRoot must be published to the destinations for it to work in delivery system.

WebRoots come in two forms: absolute and relative. An absolute WebRoot must contain the entire URL prefix (including host and port information), it may also optionally contain a PATH prefix, it is unique to every server. A relative WebRoot contains only information related to the PATH and does not contain any information on the host or port. Both URLs are handled identically by WebCenter Sites; however, with relative WebRoot only a single WebRoot is needed across multiple environments, for instance development, staging, and production environments. With absolute WebRoot each of these environments will have unique WebRoot.

To eliminate this limitation, the concept of VirtualRoot is supported. VirtualRoot requires setting environment identifier in the `futuretense.ini` in order to identify that they are valid for the given environment, if the parameter is missing then the WebRoot is used. It is important to determine and understand the types of WebRoots as you may have reason to have both absolute and relative roots defined at the same time. [Table 24-1](#) lists the advantages and disadvantages of these types of WebRoots.

Table 24-1 Advantages and Disadvantages of WebRoot Types

WebRoot Type	Advantages	Disadvantages
Absolute WebRoot	<ul style="list-style-type: none"> Full URL is displayed to the Content Contributor 	<ul style="list-style-type: none"> A unique URL is required for every environment Virtual roots require setting a WebCenter Sites property Difficult to test vanity URLs prior to publishing
Relative WebRoot	<ul style="list-style-type: none"> Single WebRoot works across all Systems Vanity URLs can be easily tested before publishing 	<ul style="list-style-type: none"> Content Contributors will not see the full URL, only the Path is displayed May require additional steps in rewriting the URL
Combination	<ul style="list-style-type: none"> Full URL is displayed to the Content Contributor Single WebRoot works across all Systems. Vanity URLs can be easily tested before publishing 	<ul style="list-style-type: none"> More URLs are present and need to be stored/published Content contributors will see both Relative as well as absolute WebRoot URLs. Since there are more URLs it may be confusing to the Content Contributor.

Examples of the different types of WebRoots required for a URL

The following examples use the same vanity URL used in the introduction — `http://www.example.com/Running/Home` — and will illustrate using and accessing it from management and production servers.

The following vanity URLs are used for each environment:

- Production: `http://www.example.com/Running/Home`
- Management: `http://management.example.com/Running/Home`

For more information on the Path portion of `Running/Home` path of the URL, see [Section 24.2, "Generating Vanity URLs."](#)

The server name used for the production server is `http://www.example.com/` and the server name used for the management server is `http://management.example.com/`.

For an absolute WebRoot, a unique WebRoot for each environment is required, with the root URL as `http://www.example.com`. In `futuretense.ini` set the `sites.environment` property to `management`. The site will be AVISports.

It is important to note that the value specified in WebRoot for the virtual root URI and the value specified in the `sites.environment` property must be the same.

For a relative WebRoot, a single Web Root is used all environments. The root URL is `/`. The site will be AVISports.

To create a new WebRoot

1. From the **Admin** tab, expand the **WebRoots** node. Double-click **Add New**.

The WebRoot edit form will display, as in [Figure 24–1](#), but with all fields blank.

Figure 24–1 WebRoot Edit Form

WebRoot: Avi

Host Name:

Root URL:

Virtual Root URL:

Environment	Root URL
<input type="text"/>	<input type="text"/>

Sites:

2. Enter a name in the **Host Name** field. This is the unique identifier for the WebRoot instance. This can be used in render tags to get a specific vanity URL of an asset.

This value cannot be changed once the WebRoot is saved.

3. Enter a URL in the **Root URL** field. The value for the Root URL can be either absolute or relative. For an absolute WebRoot, typically, this is the URL that is used to render the asset in delivery environment, but for the relative WebRoot this would be `/<Optional PATH Prefix>`.

For the Content Contributor this is the URL that is used as a prefix when displaying the defined vanity URLs. For example:

- Absolute Root URL using only the hostname
 - Production URL: `http://www.example.com`
 - Complete vanity URL: `http://www.example.com/Running/Home`
- Absolute Root URL including both the hostname and prefix for the path
 - Production URL: `http://www.example.com/Prefix`

- Complete vanity URL: `http://www.example.com/Prefix/Running/Home`
 - Absolute Root URL including the hostname for a mobile site
 - Production URL: `http://www.example.mobi`
 - Complete vanity URL: `http://www.example.mobi/Running/Home`
 - Relative Root URL
 - Production URL: `/cs/example`
 - Complete vanity URL: `http://<Current Host Name>/cs/example/Running/Home`
 - Relative Root URL
 - Production URL: `/`
 - Complete vanity URL: `http://<Current Host Name>/Running/Home`
4. Enter one or more virtual URLs. The virtual URL has two parts:
- **Environment:** The identifier for the other environments. Environments must be identified by setting `property sites.environment=<name>` in `futuretense.ini`.
 - **Root URL:** This is the URL that is used to render the asset in other environments such as editorial, staging, and so forth.

When first displayed, there is only one row for adding an environment and root URL. Additional lines of **Environment** and **Root URL** fields are added by clicking the **Add New** button.

The virtual URL entered must have an absolute Root URL. For example:

- For a management URL of `http://management.example.com/Running/Home` enter:
 - Environment: `management`
 - Root URL: `management.example.com`
 - `sites.environment=management` in `futuretense.ini`
- For a staging URL of `http://staging.example.com/Running/Home` enter:
 - Environment: `staging`
 - Root URL: `staging.example.com`
 - `sites.environment=staging` in `futuretense.ini`
- For a development URL (including port) of `http://dev.example.com:8080/Running/Home` enter:
 - Environment: `dev`
 - Root URL: `dev.example.com:8080`
 - `sites.environment=dev` in `futuretense.ini`

Note: Virtual Root URL is not used in display in the contributor UI. It is used only for rendering the web pages in different environments, and only when the Environment matches the value provided in the `sites.environment` property in `futuretense.ini`.

5. Select one or more sites. The WebRoot can be enabled for specific sites using the **Sites** field.

In the proceeding examples we would set the Site to `AVISports`, since these URLs are unique to this site.

6. Once all information is added, click the Save icon. The WebRoot is created.

Once the WebRoot is created for the first time, it will be available as an asset type in the **Admin** tab under the **Asset Type** node.

To view a WebRoot

1. From the **Admin** tab, expand the **WebRoots** node. Double-click one of the listed WebRoots.

The WebRoot Content form will display, as in [Figure 24-2](#), showing the saved information for that WebRoot.

Figure 24-2 WebRoot Content Form

WebRoot: Avi

Host Name: Avi

Root URL: http://www.avisports.com

Virtual Root URL:	Environment	Root URL
	editorial	http://editorial.avisports.com
	staging	http://staging.avisports.com

Sites: avisports

To edit an existing WebRoot

1. From the **Admin** tab, expand the **WebRoots** node. Double-click one of the listed WebRoots.

The WebRoot edit form will display, as in [Figure 24-2](#).

2. Click the Edit icon to edit the WebRoot.

The WebRoot Edit form will display, as in [Figure 24-3](#).

Figure 24–3 WebRoot Edit Form

WebRoot: Avi

*Host Name: Avi

*Root URL:

Virtual Root URL:

Environment	Root URL
<input type="text"/>	<input type="text"/>

Sites:

- Any
- AdminSite
- FirstSite1
- avisports

- The stored information will display in the **Host Name**, **Root URL**, **Virtual Root URL**, and **Sites** fields.

The **Host Name** field is the unique identifier for the WebRoot instance. This can be used in render tags to get a specific vanity URL of an asset.

The **Root URL** field is typically the URL used to render the asset in delivery environment. This is the URL that is used for display for the Content Contributor.

The **Virtual Root URL** has two parts:

- **Environment:** The identifier for the other environments. Environments must be identified by setting `property sites.environment=<name>` in `futuretense.ini`.
- **Root URL:** This is the URL that is used to render the asset in other environments such as editorial, staging, and so forth.

To add additional virtual URLs, additional lines of **Environment** and **Root URL** fields are added by clicking the **Add New** button.

Select one or more sites. The WebRoot can be enabled for specific sites using the **Sites** field.

- Once all information is added, click the Save icon. The WebRoot is updated.

To Approve and Publish WebRoot

WebRoot can be approved for publishing in the same way other assets are. This is done by clicking the green checkmark icon when previewing the WebRoot.

WebRoots are not set as blocking assets. The administrator must approve and publish the WebRoot separately.

24.2 Generating Vanity URLs

Vanity URLs can be constructed in two ways. One way a vanity URL can be constructed manually, in a one-off manner by Content Contributors (see the *Oracle Fusion Middleware WebCenter Sites User's Guide* for more information). The other method is described in this section, and automatically generates a vanity URL from a pattern.

This section contains the following topics:

- [Section 24.2.1, "Configuring Auto-Generated URLs"](#)
- [Section 24.2.2, "Working with URL Patterns"](#)
- [Section 24.2.3, "Publishing URL Patterns"](#)

24.2.1 Configuring Auto-Generated URLs

Vanity URLs based upon patterns are automatically generated for individual assets. Once a pattern is registered, the vanity URL is generated when the asset is saved for the first time. Patterns can be created for every asset type; however, these patterns are only valid for a single site. The same pattern must be created multiple times if multiple sites share the asset type.

While creating vanity URLs for assets, users edit the default vanity URLs of the assets and Oracle WebCenter Sites allows the creation of duplicate vanity URLs. Therefore, the auto generated URLs are changed to user generated URLs and users can decide if they want to delete the URLs, redirect them to another URL, and so on.

Note: Changes to patterns are only processed when an asset is created or saved. If you modify a pattern or create a new one it only applies to assets created after the pattern is saved.

To configure an auto-generated vanity URL

1. On the **Admin** tab, expand the **Asset Types** node, then expand the node for the specific asset type you are creating an auto-generated URL for. Under the specific asset type node, expand **URL Pattern** and double-click **Add New**.

The URL Pattern Form displays.

Figure 24-4 URL Pattern Form - Asset

URL Pattern Form

Asset **Blob**

Name:

Site:

Host:

Subtype:

Device Group:

Template:

Wrapper:

Pattern:

(e.g. Media_C/{name}/{formatDate(updateddate,"yy/MM/dd")}/{description.toLowerCase()})

Name	Type
createdby	string
createddate	date
description	string
enddate	date
fwtags	array
id	long
locale	string
name	string
startdate	date
status	string
subtype	string
template	string
updatedby	string
updateddate	date

Attributes

Functions

Figure 24–5 URL Pattern Form - Blob (Detail)

URL Pattern Form

Asset
 Blob

Name:

Site:

Host:

Subtype:

Field: **Is Downloadable**

Blob Headers:

(e.g. Last-Modified=S{updateddate})

Pattern:

(e.g. Media_C/\$f{getFileName(BLOB_ATTRIBUTE)})

2. Enter a name for the pattern in the **Name** field. This name is only used to reference this pattern and is unique; it may be changed at any time.
3. In the **Pattern** field, enter the rule that defines the vanity URL. Lists of valid attributes and functions which may be used are displayed on the right side of the screen. For more information on creating patterns, see [Section 24.2.2, "Working with URL Patterns."](#)
4. In the **Blob Headers** field (available only if a blob is selected), enter the rule used to define the blob headers in the vanity URL.
5. The **Site** is the site the pattern will be applied for, and will only list sites for which the asset type is enabled. The selection of the site will also narrow down the selections for the **Host** (WebRoot), **Subtype**, **Template** (asset only), and **Wrapper** (asset only).

Note: Patterns defined will remain even if the asset type is later disabled for that site; however, the use of these URLs might result in a broken page.

6. The **Host** is the WebRoot which the vanity URL is applied to. Only WebRoots that are defined for this site or "any" are displayed. The pattern will not work without a WebRoot.
7. The **Subtype** is the subtype of the asset type which the vanity URL is applied to. The selection of **Subtype** further restricts the available items for **Template** and **Wrapper**.

Note: The **Subtype** field for blobs will not show *Any* as an option for flex assets. The **Subtype** field for blobs will not appear for basic assets.

8. The **Field** is the blob to be rendered by this URL. This field appears only when a blob is selected. There may be multiple items available if the asset contains more than one blob entry.

Selecting the **Is Downloadable** checkbox will have the browser display a save dialogue, when the vanity URL is accessed, instead of attempting to view the blob object within the page.
9. **Device Group** provides a list of possible device groups currently defined. This field appears only when an asset is selected. All device groups are formatted into lists depending upon the extension and regardless of their current state within the site. For more information on device groups, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
10. The **Template** is the template for the URL and all templates that are valid for the asset type, subtype and site are displayed regardless of the existence of a mobile variation. All templates that are valid will be shown, however not all templates may have a variation for the specified **Device Group**. It is up to the administrator to ensure the proper variation exists when using a device group other than the default.
11. The **Wrapper** will provide a list of all wrapper elements that are valid for the selected **Template**. If none exist then this field will be disabled.
12. At this point you may click **Save** to save the pattern and continue working, or evaluate the URL pattern.

The Evaluate URL Pattern section of the form allows you to see an example of what a pattern generated vanity URL will look like. This section may be used either before or after saving the pattern; however, it is recommended to evaluate the URL pattern before saving to validate your pattern.

Note: If no asset of the given type exists then you will be unable to use this feature.

13. The **Asset** drop-down allows you to select an asset currently defined for the pattern type. You may also use the "look ahead" feature to type in an asset's name. Once an asset is selected, WebCenter Sites will evaluate the created pattern and show how this asset's URL will appear.

Remember that this is an example of the URLs final appearance. The URL is not yet valid, until the pattern has been saved and the asset in question has also been edited and saved.

24.2.2 Working with URL Patterns

Automatic patterns are the heart of vanity URLs. There is a another component to vanity URLs which allows contributors to create custom URLs for each asset (see *Oracle Fusion Middleware WebCenter Sites User's Guide* for more information), however, as an administrator pattern matched URL will likely make up the bulk of the vanity URLs present on a site. Patterns take their form from a combination of paths and variables, where the variables are either fields from the asset or predefined functions.

Note: All the parts of URL are URL encoded before the save.

The actual variables which are present will differ based on the asset type selected, but all asset types will share some common variables. [Table 24–2](#) lists the most common variables.

Table 24–2 Common Asset Type Variables

Variable Name	Variable Type
createdby	string
createddate	date
description	string
enddate	date
fwtags	array
id	long
locale	string
name	string
startdate	date
status	string
subtype	string
template	string
updatedby	string
updateddate	date

Clicking a variable will cause it to appear at the end of the pattern field. It does not make a difference where the cursor is, the new pattern is always the last item.

Variables are specified by `${<Name>}`. For example, to call the ID of an asset, use `${id}`.

Variables of types string, int, and long can be used directly, without modification.

Variables of type array need to be specified such that only a single value is extracted. For example, to get the first tag defined for an asset, use `${fwtags[0]}`, or use the function `listAsPath`. Array numbering begins with 0 and using a value outside of bounds of the array will result in "null."

Variables of type date will need to be formatted; a function `formatDate` is provided for this.

Other types, such as url, may not be usable or may require the use of a function to convert to a form that is acceptable for generating a URL. Use the Evaluate URL Pattern option to assist you with these variables.

Variables are defined using Java standard types so that normal Java functions can also be added to a variable. For example, on the string variable "name," `name.toLowerCase()` will convert the name to lower case.

This section contains the following topics:

- [Section 24.2.2.1, "Functions"](#)
- [Section 24.2.2.2, "Examples of Using Patterns"](#)
- [Section 24.2.2.3, "Deleting Patterns"](#)

24.2.2.1 Functions

Functions are useful in converting and formatting data to include in the vanity URL. There are predefined functions in WebCenter Sites will will be useful in creating vanity URLs.

Clicking a function will cause it to appear at the end of the pattern field. It does not make a difference where the cursor is, the new pattern is always the last item.

Predefined functions

- `spaceToUnderscore(java.lang.String)`
Converts spaces to `_`. For example: `Page 001` would become `Page_001`.
- `formatDate(java.util.Date, java.lang.String)`
Formats a date to a number and slash format. For example, if `updatedate` was `"Thu Feb 28 12:57:12 UTC 2013"` using the `formatDate` function we can display it as `"2/28/2013"`. The string used to format the date should match that used by Java's `SimpleDateFormat` function.
- `spaceToDash(java.lang.String)`
Converts a space to `-`. For example, `Page 001` would become `Page-001`.
- `listAsPath(java.util.List, int)`
Converts an array to a list, for the first `x` items. For instance, if an array looked like `[1, 2, 3, 4, 5, 6]` then using `listAsPath` with `x = 4` the resulting list would appear as `1\2\3\4`.
Note that `x`, defined by `int`, is required.
- `getFileName(com.fatwire.assetapi.data.BlobObject)`
Gets the file name for a blob object as text. For instance, if the blob name was `xyz.png` this would return the text `xyz.png`. Normally this is the very last parameter used, so that the correct file extension is present.
This function cannot be used for binary fields of the Asset Maker asset.
- `property(java.lang.String, java.lang.String, java.lang.String)`
Reads a property from ini file or a property file. This uses `ics.GetProperty`. Parameters for this method are 1) property name, 2) default value if the property is not present and 3) comma separated list of file names.

Functions are specified using the format `${f:<name>(<variables>)}`, so for the `spaceToUnderscore` example given above the function would be written as

```
${f:spaceToUnderscore("Page 001")}
```

If you make a mistake or pass in an invalid parameter, then in the Evaluate URL Pattern section a null is returned for that part of the path. While only the outermost variable must contain `{}`, all functions must begin with `f:`.

24.2.2.2 Examples of Using Patterns

The examples are based off AVISports article asset type. The asset name is "Baker Likely to Stay With Stars."

Figure 24–6 URL Pattern Form - Asset

URL Pattern Form

Asset **Blob**

Name:

Site:

Host:

Subtype:

Device Group:

Template:

Wrapper:

Pattern:

(e.g. Media_C/\${name}/\${f:formatDate(updateddate,\"yy/MM/dd\")}/\${description.toLowerCase()})

Name	Type
createdby	string
createddate	date
description	string
enddate	date
fwtags	array
id	long
locale	string
name	string
startdate	date
status	string
subtype	string
template	string
updatedby	string
updateddate	date

Attributes

Functions

Note: The first part of the URL is always the WebRoot. This will be the local system name and context root in the case of AVI Sports. The next part, `Avi`, is the filter prefix (for information on Filters see `Mod_rewrite` section). Only the actual unique path is provided in these examples.

Given the asset "Baker Likely to Stay With Stars," there are a number of options available to generate a URL. Each example shows the function and the resulting URL.

1. Start with a simple name pattern
 - `/Article/${name}`
 - `/Article/Baker+Likely+to+Stay+With+Stars`
2. Convert the name to lower case

- /Article/\${name.toLowerCase() }
 - /Article/baker+likely+to+stay+with+stars
3. Remove the spaces from this URL
 - /Article/\${f:spaceToUnderscore(name) }
 - /Article/Baker_Likely_to_Stay_With_Stars
 4. Use multiple variables
 - /Article/\${id}/\${createdby}
 - /Article/1328196049037/fwadmin
 5. Use a function within a function
 - /Article/\${(f:formatDate(createddate, "yy/MM/dd")) }
 - /Article/12-04-27

24.2.2.3 Deleting Patterns









Patterns can be deleted via the navigation tree.

To delete a pattern

1. From the **Admin** tab, expand **Asset Types**. Double-click the asset type containing the pattern.
The selected asset type will expand.
2. Double-click **URL Pattern**.

Figure 24–7 Saved URL Patterns by Asset Type

URL Patterns:

Action	Type	Name	Host	Site	Subtype
  		ArticleImagesPattern	http://localhost:9080/cs/Avi	avisports	ArticleImage
View More					
  		ImagesPattern	http://localhost:9080/cs/Avi	avisports	Image
View More					

[Add New](#)

The currently saved patterns are displayed, and can be inspected, edited, or deleted as any other asset.

24.2.3 Publishing URL Patterns

Patterns are automatically published with the asset type. There is no special approval or publishing needed for URL patterns.

24.3 Using Vanity URL with System Tools

Using the System Tools node, an Admin can view all presently defined vanity URLs.

This section contains the following topics:

- [Section 24.3.1, "Viewing All Vanity URLs"](#)
- [Section 24.3.2, "Resolving Vanity URL Conflicts Using System Tools"](#)

24.3.1 Viewing All Vanity URLs

To view created vanity URLs

1. From the **Admin** tab, expand the **System Tools** node.
2. Double-click **URL**.

The URL Utility Form is displayed.

Figure 24–8 URL Utility Form

URL Utility

Use this utility to view and manage URLs

Results per page Search Criteria

Host	URL	Template	Wrapper	HTTP Status
Avi	SkiBrampton	avisports/Page/SectionLayout		200
Avi	SkiBC	avisports/Page/SectionLayoutGreen		200
Avi	m	avisports/Page/SectionLayout		200

Page 1 of 1

Since every URL defined (for all sites) are present on this list, it will be necessary to filter it using the search box provided to locate URL(s) of interest. From this menu you can manually delete a vanity URL, or check if it is a problem or confirm where a conflict is originating. However, if the URL is based upon a pattern then it will be automatically regenerated when the asset is next saved.

24.3.2 Resolving Vanity URL Conflicts Using System Tools

Vanity URLs are unique in that there cannot be more than one with the same path (even across WebCenter Sites).

For instance, if there are two sites and each site has a Rule pattern — `/Page/${name}` — and if the sites use the same name for a page (for instance, "Home"), then a conflict will occur.

This means that whichever page is saved first will succeed and create a vanity URL, but, the second page will fail to create a vanity URL. When this occurs it might be necessary to use the System Tool to locate what is blocking the creation of the Vanity URL and if needed remove the existing URL to allow the new one to be created. This issue can be mitigated by ensuring that the WebRoot is unique to a single site.

24.4 Resolving Vanity URLs Using a Rewriter Filter

There are two ways to resolve a vanity URL. The first is by using a web server and rewriting the URL. The second way is designed for instances where a web server is not present such as often in development, and relies upon a filter.

WebCenter Sites contains a URL rewriter filter that is used, by default, only for sample sites. This rewriter filter can be configured to be used along with a WebRoot for any site. The filter will intercept any call through it (based upon a string match it begins with) and pass the call through to WebCenter Sites as a vanity URL filter.

In simple terms, it allows vanity URLs to operate without a rewriting rule on the web server. The major limitation of using this filter is that the context root, and then a unique prefix, must be given to ensure that WebCenter Sites knows which URLs to treat as vanity URLs.

Note: "FSII" and "Avi" are the parameters pre-configured for URL rewrite filter in the `web.xml` when sample sites are installed.

To enable the rewriter filter

1. Copy the following code into `web.xml` after the final existing `</filter>`. Then replace `Avi` with a unique string (for instance the site name), then restart WebCenter Sites.

Note: Multiple SitePrefix can be added by specifying comma separated list in the `param-value`. For each SitePrefix ensure a WebRoot is created inside Sites that matches the prefix as shown in step 2.

```
<filter>
  <filter-name>URLRewriteFilter</filter-name>
  <filter-class>COM.FutureTense.Servlet.URLRewriteFilter</filter-class>
  <init-param>
    <param-name>SitePrefix</param-name>
    <param-value>Avi</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>URLRewriteFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

2. Create a WebRoot inside WebCenter Sites that matches the prefix. For example, if using "Avi" then you need a WebRoot beginning with `<contextroot>/Avi`. If using a relative WebRoot then this all that is necessary. If using an absolute WebRoot prefix, the Hostname and Port are required as well. Make sure to select the site for which this WebRoot is defined before saving.
3. Once saved, create a single URL using the new WebRoot.
4. Verify that the vanity URL works in a browser.

For example:

Assume a relative Web Root of `/Avi` and a path of `/Running/Home`, the resulting URL using the filter would look like:

```
http://<Sites Host>:<Sites Port>/<Context Root>/Avi/Running/Home
```

Since a filter is used, the use of a prefix (specifically `Avi` in this example) is required both in the filter and in the WebRoot.

Note: Vanity URLs support query parameters. While required parameters have been removed from vanity URLs it is understood that sometimes additional data must be passed as part of the URL. In such cases the existing parameters will need to be concatenated onto the existing URL. For instance, if passing in `time=true` then the URL above sent to sites would look like:

```
http://<Sites Host>:<Sites Port>/<Context
root>/Avi/Running/Home?time=true
```

24.5 Vanity URL for Mobile Sites

In most cases, those using a mobile-specific site will want to use a different URL for mobile users. Not only will mobile users benefit from a shorter URL, but there is often a separate domain and different URLs. To set up a different domain and different URL for the mobile pages of a website the following steps should be followed

1. WebRoot must be set for the Mobile domain names. The Root URL of the WebRoot should have mobile domain names.
2. User should choose the right host names when creating vanity URLs for mobile device groups. This will ensure that the links are correctly generated for the mobile domain names.
3. All other steps including Web Server URL Rewrite are same as mentioned in [Section 24.1, "WebRoots,"](#) [Section 24.2, "Generating Vanity URLs,"](#) and [Section 24.3, "Using Vanity URL with System Tools."](#)

24.6 Using the Web Server with Vanity URLs

Using a web server with URL rewriting will allow you to access vanity URLs in a similar way to the filter. and you will also have all the control and power that URL rewriting entails.

When using web server, all rewritten vanity URLs must be split into two sections before sending them to WebCenter Sites. One section is the WebRoot, the other is the path.

Creating an example: use the WebRoot `http://www.example.com` and a pattern that generates `/Running/Home` for the URL `http://www.example.com/Running/Home`. In order to pass this URL in from a web server we must break up the URL into the WebRoot and path. These are then passed, separately, as query parameters to a special filter in WebCenter Sites The vanity URL filter used to handle rewritten URLs is named "Sites", and is accessed by `<content root>/Sites`.

The two parts of the URL are broken up into `lookuphost`, and `lookuppage`. The `lookuphost` refers to the WebRoot (full server name, port and prefix if absolute WebRoot, or only the prefix if relative WebRoot). The `lookuppage` refers to the remainder of the PATH after the prefix. Using the above URL as an example, the URL passed to WebCenter Sites would need to be rewritten to look like:

```
http://<Sites Host>:<Sites Port>/<contextpath>/Sites?lookuppage=/Running/Home
```

```
&lookuphost=http://www.example.com
```

Note: Vanity URLs support query parameters. While required parameters have been removed from vanity URLs, it is understood that sometimes extra data must be passed in as part of the URL. In such cases the existing parameters will need to be concatenated onto the existing URL.

For instance, if passing in `time=true` then the URL above sent to sites would look like:

```
http://<Sites Host>:<Sites  
Port>/<contextpath/Sites?lookuppage=/Running/Home  
&lookuphost=http://www.example.com&time=true
```

24.7 Using mod_rewrite with Vanity URLs

Every web server and site will be unique, so the following should be viewed as guide lines and examples as to how to use `mod_rewrite` (which is used in OHS, Apache and IBM HTTP Server to rewrite URLs). Your system will likely require modifications to the rules to operate correctly.

There are two methods to use `mod_rewrite`. It can be used to either directly transform the URL for the Sites filter, or it could be used in conjunction with the URL rewriter filter to perform the same action but in a way that will allow for much simpler `mod_rewrite` rules (likely closer to what existing sites have used previously to remove the context root). Examples of both methods will be provided. To determine which is right for you - first decide if the extra step (and hence cost) of having the Sites filter break apart the vanity URLs is worth the simplified rewrite rules.

Note: The guide will assume that you are familiar with `mod_rewrite`. If not, start by reviewing the Oracle documentation located at:

```
http://docs.oracle.com/cd/B12037_  
01/server.101/b12255/confmods.htm#1009986
```

All subsequent steps are based off Apache 2.4.x manually compiled. If using a version shipped with an OS the actual changes needed to enable and configure `mod_rewrite` may differ, but the rules and concepts will not.

This section contains the following topics:

- [Section 24.7.1, "Setting Up mod_rewrite"](#)
- [Section 24.7.2, "Converting with mod_rewrite"](#)
- [Section 24.7.3, "Using the Sites Filter with mod_rewrite"](#)
- [Section 24.7.4, "Using the Rewriter Filter with mod_rewrite"](#)
- [Section 24.7.5, "Using Static Content with mod_rewrite"](#)

24.7.1 Setting Up mod_rewrite

First enable `mod_rewrite` by editing `httpd.conf` and uncommenting the following line:


```
LoadModule rewrite_module modules/mod_rewrite.so
```

Second, enable the Rewrite engine, by adding this line to httpd.conf:

```
RewriteEngine On
```

Third, for debugging, only enable logging on mod_rewrite (this is very high cost operation so only use it when debugging):

```
LogLevel alert rewrite:trace6
```

Save the changes to httpd.conf; mod_rewrite is now enabled (without any rules). Lastly ensure that the web server starts.

Note: There are web pages that allow for much easier construction and testing. It is suggested that you use one of these sites when initially writing up the mod_rewrite rules.

Setup for Provided Examples

In order to provide some examples of mod_rewrite rules, we first need to define what the environment looks like. Below is a sample based upon the vanity URLs used in this guide; however, you should be able to use the same elements and just change out the values for your environment as a starting point.

The vanity URLs a customer is expected to see.

- Production:

```
http://www.example.com/Page/Surfing
http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars
http://www.example.com/Home
http://www.example.com/Home?time=true
```

- Development:

```
http://dev.example.com:8080/Running/Home
```

Generic Sites information:

- Production context root: /cs
- Production rewrite filter: /AVISPORTS
- Development context root: /csdev
- Development rewrite filter: /Avi

Web Roots Defined for each Server:

- Production - absolute: http://www.example.com/
- Development - relative: /Avi

URL structure:

- Production: http://www.example.com/Page/Surfing
 - WebRoot: http://www.example.com
 - Path: /Page/Surfing
- Production: http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars

- WebRoot: `http://www.example.com`
- Path: `/Article/Baker_Likely_to_Stay_With_Stars`
- Production: `http://www.example.com/Home`
 - WebRoot: `http://www.example.com`
 - Path: `/Home`
- Production: `http://www.example.com/Home?time=true`
 - WebRoot: `http://www.example.com`
 - Path: `/Home`
 - Query Parameters: `time=true`
- Development: `http://dev.example.com:8080/Running/Home`
 - WebRoot: `/Avi`
 - Path: `/Running/Home`

There are two filters. The first is the Sites vanity URL filter, the second is the Sites rewrite filter. When using the rewrite filter it passes the data to the vanity URL filter transparently.

When using the Sites vanity URL filter, the URLs are expected in this manner:

- Production:
 - `http://www.example.com/Page/Surfing`
`lookuphost=http://www.example.com &lookuppage=/Page/Surfing`
 - `http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars`
`lookuphost=http://www.example.com &lookuppage=/Article/Baker_Likely_to_Stay_With_Stars`
 - `http://www.example.com/Home`
`lookuphost=http://www.example.com &lookuppage=/Home`
 - `http://www.example.com/Home?time=true`
`lookuphost=http://www.example.com &lookuppage=/Home&time=true`
- Development:
 - `http://dev.example.com:8080/Running/Home`
`lookuphost=/cs/Avi&lookuppage=/Running/Home`

When using the WebCenter Sites rewriter filter, the URLs are expected in this manner:

- Production:
 - `http://www.example.com/Page/Surfing`
`/cs/AVISPORTS/Page/Surfing`
 - `http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars`
`/cs/AVISPORTS/Article/Baker_Likely_to_Stay_With_Stars`
 - `http://www.example.com/Home`
`/cs/AVISPORTS/Home`

- `http://www.example.com/Home?time=true`
`/cs/AVISPORTS/Home?time=true`
- **Development:**
 - `http://dev.example.com:8080/Running/Home`
`/csdev/Avi/Running/Home`

24.7.2 Converting with mod_rewrite

After choosing some URLs and breaking them down to component parts, it is now necessary to convert these URLs using `mod_rewrite`. The rules for `mod_rewrite` consist of two parts: the first is a regular expression which matches the URL provided to the web server, and the second is how it will appear when passed on. If the first part fails to match, then the URL is passed on to the next rewrite rule; if no defined rule matches, then the URL is passed on unchanged. Thus the order of the rules become important if more than a single rule is present.

Note: `mod_rewrite` is an optional component that you have previously enabled. If you create a `mod_rewrite` rule without actually enabling `mod_rewrite` then the rule is skipped. There are no errors and nothing is rewritten. For this reason it is suggested to ensure debugging is active during development and monitor the `error.log` for details related to `mod_rewrite`.

What is discussed in this section only covers a very small portion of what `mod_rewrite` can handle and for most sites a more complex rule is likely to be required. For more information, see the vendor's documentation.

The component parts of a mod_rewrite rule

This example will use the first URL for production:

`http://www.example.com/Page/Surfing`

Create this rule for it:

```
RewriteRule ^(.*)$ /cs/Sites?lookuphost=http://www.example.com&lookuppage=$1 [L]
```

This rule has four component parts:

1. `RewriteRule` — a statement that the following entry is a `mod_rewrite` rule
2. `^(.*)$` — a regular expression, which breaks down as follows:
 - `^` — start of line indicator
 - `(.*)` — `s` select match all text indicator; placing it in parentheses also saves that text to a variable
 - `$` — end of line indicator

This expression then indicates that between the start and end of string save all text to a variable.

3. `/cs/Sites?lookuphost=http://www.example.com&lookuppage=$1` — the WebCenter Sites context root, filter, and path for vanity URL, which breaks down further as

- /cs/Sites — the Sites Context Root (cs) followed by the vanity URL filter (Sites)
 - ? — indicates that parameters coming after this are query parameters
 - lookuphost=http://www.example.com
 - lookuphost — provides the vanity URL filter with the name of the WebRoot
 - http://www.example.com — the WebRoot for the production server (it is an absolute WebRoot).
 - lookuppge=\$1
 - lookuppge — the path portion of the vanity URL (in this case /Page/Surfing)
 - \$1 — a replacement variable, for the first piece of text extracted from the input URL. In this case /Page/Surfing which was matched by the (.*)
4. [L] — informs mod_rewrite that this is the end of the rules. All mod_rewrite rules must end with this.

24.7.3 Using the Sites Filter with mod_rewrite

The following rule will handle /Page/Surfing, but it will also handle any other page sent in, extracting everything after the host to the end of the line.

```
RewriteRule ^(.*)$ /cs/Sites?lookuphost=http://www.example.com& lookuppge=$1 [L]
```

So the rule will also handle:

```
http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars
http://www.example.com/Home
```

However, this rule will not handle

```
http://www.example.com/Home?time=true
```

as this URL has a query parameter being passed in along with the host and path. To handle the query parameter a mod_rewrite built in variable "%{QUERY_STRING}" must be used, thus the previous rule:

```
/cs/Sites? lookuphost=http://www.example.com& lookuppge=$1
```

Must be updated to:

```
/cs/Sites? lookuphost=http://www.example.com& lookuppge=$1&{%{QUERY_STRING}}
```

If no Query parameters are passed in the URL then %{QUERY_STRING} will be empty. Thus it will not only work for the three previous URLs, but also for this URL.

Next we look at the Development URL:

```
http://dev.example.com:8080/Running/Home .
```

This must be passed in to the vanity URL filter as

```
lookuphost=/cs/Avi&/cs/Sites?lookuppge=/Running/Home .
```

Since mod_rewrite does not deal with host and port by default (there are special parameters that contains this information), the rewrite rule is very similar to this rule:

```
RewriteRule ^(.*)$ /cs/Sites?lookuppge=$1&lookuphost=/cs/Avi [L]
```

In this case since development utilizes a relative WebRoot, we need to adjust the name, since the rest of the URL is unchanged the same rule still applies.

24.7.4 Using the Rewriter Filter with mod_rewrite

When using the rewrite filter, URLs do not need to be broken down the way they are for the vanity URL filter. Instead, the Sites Context Root and rewriter Filter Prefix are added to the URL (This is very similar to using WebCenter Sites prior to the introduction of vanity URLs, so it is likely that modify existing rules can be easily modified to support vanity URLs.

Using the example URL

```
http://www.example.com/Page/Surfing
```

Sites expects

```
/cs/AVISPORTS/Page/Surfing
```

To handle this, create a new mod_rewrite rule:

```
RewriteRule ^(.*)$ /cs/AVISPORTS$1 [L]
```

This rule simply takes what is given and prefixes /cs/AVISPORTS to it. The same rule works for:

```
http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars
```

```
http://www.example.com/Home
```

To handle

```
http://www.example.com/Home?time=true
```

add the %{QUERY_STRING} parameter to the rule so that it will become:

```
RewriteRule ^(.*)$ /cs/AVISPORTS$1%{QUERY_STRING} [L]
```

24.7.5 Using Static Content with mod_rewrite

One of the biggest issues with using mod_rewrite and vanity URLs is accessing existing CSS and static images. Often these are stored either on the application server or on the web server in a directory named after the site. This means that often the rewrite rules will result in making them inaccessible.

Using the example of

```
http://www.example.com/Page/Surfing
```

and assume that the CSS files are kept in a directory named AVISPORTS_STATIC.

To handle this a second mod_rewrite rule is required. This rule will precede the former rule which ensures that calls to this directory are not rewritten.

This can be done as:

```
RewriteRule ^.*AVISPORTS_STATIC/(.*)$ /cs/CSPerformance/$1 [L,PT].
```

With these two rules, the first ensures that all static pages and CSS are not converted to a vanity URL, and the second rule (which was constructed in [Section 24.7.2, "Converting with mod_rewrite"](#)) sends the remaining URLs to the vanity URL filter.

```
RewriteRule ^.*/AVISPORTS_STATIC/(.*)$ /cs/CSPerformance/$1 [L,PT]  
RewriteRule ^(.*)$ /cs/Sites?lookuphost=http://www.example.com&lookuppage=$1 [L]
```

Configuring the Lucene Search Engine

Lucene is a third-party search engine that is integrated with WebCenter Sites. Lucene powers the search feature in WebCenter Sites' Contributor interface and enables the Public Site Search API to support search capabilities on websites. Before you can search for assets in the Contributor interface, you must set up and configure the Lucene search engine. This chapter shows you how to set up and maintain Lucene on your system.

This chapter contains the following sections:

- [Section 25.1, "Overview"](#)
- [Section 25.2, "Setting Up Search Indices"](#)
- [Section 25.3, "Disabling the Lucene Search Engine"](#)
- [Section 25.4, "Maintaining Search Indexes"](#)
- [Section 25.5, "Writing Code that Queries the Search Index"](#)

25.1 Overview

When you install WebCenter Sites, the search feature in the WebCenter Sites database and Lucene are available. Lucene is enabled and configured out of the box. The Lucene engine is set up as Sites is installed, allowing content contributors, website visitors, and third party applications will be able to search for assets.

This chapter includes information detailing how to use the Lucene search engine, how to make additional assets searchable and how to pause or disable the search engine.

This section contains the following topics:

- [Section 25.1.1, "Indexing for Search Functions"](#)
- [Section 25.1.2, "WebCenter Sites Search Functions"](#)

25.1.1 Indexing for Search Functions

Contributor searches are run against a search index that is powered by Lucene, not WebCenter Sites' database. A search index is built by an automated process called indexing, which collects, parses, and stores asset data in a format that can be quickly retrieved during a search query.

Search results are returned based solely on the data that is available in the index at the time the search is performed. The more assets you include in the index, the longer it takes to build and to search.

You select the types of assets to index on the search configuration forms. Selected assets will be indexed and therefore, searchable. Asset types you omitted from indexing will not be indexed and, therefore, will not be searchable.

Once the index is built, the Lucene search engine runs an event every thirty seconds that checks for changes made to assets of the types selected for indexing. If changes were made (such as creating a new index item, editing an existing entry, or deleting an entry) Lucene updates the index automatically. By default, index data is stored in the <cs_shared_dir>/lucene directory (where <cs_shared_dir> is the Sites shared file system directory).

25.1.2 WebCenter Sites Search Functions

WebCenter Sites includes the following search functions: Global search, Asset Type search, and the most specific of the searches, Configure Attributes for Asset Type Index. This search option is a subset of the Asset Type Index option, as it allows you to specify attributes that will be searchable for the indexing-enabled asset types.

You can enable all of the searches on your system. The searches differ in how they store the indexed user-defined attributes.

Global Search indexes system-defined attributes individually, allowing users to search by specific attributes. All of an asset's user-defined attribute values are stored together in one table cell; attribute names are omitted. This means, when Global search is configured, users are restricted to searching across all user-defined attributes per asset type.

For example, suppose you have an article asset. You could search for the string "Jane Doe." However, you could not limit your search to just one specific user-defined attribute, because all of the user-defined attribute data are stored together in a single cell that does not differentiate one attribute from another.

Asset Type Search indexes each attribute value, for both system-defined and user-defined attributes, in its own individual cell, by attribute name. Asset type searches are used for the Public Site Search API, which enables search capabilities on the website. For more information on public site search, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Using the example above of an article asset, with Asset Type search enabled you could, as with Global search, look for the string "Jane Doe" in the system-defined attribute Name. You could also search for the string in the user-defined attribute Byline, or in any other attribute for that asset type. Since user-defined attributes are now stored under their attribute names, you can search by those specific attributes.

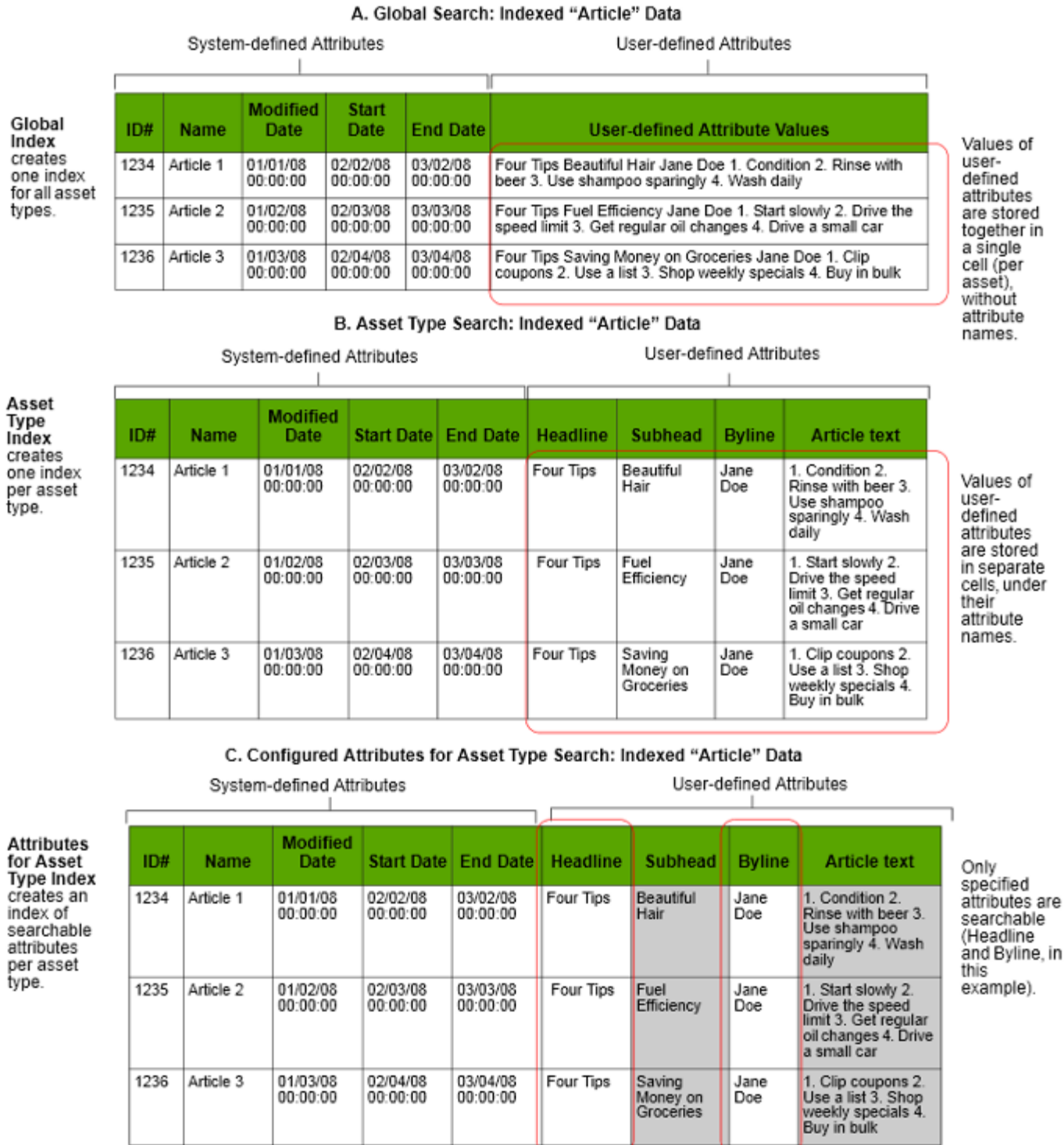
Asset type search, therefore, has a level of specificity to its search results that Global search does not. In addition to the obvious benefits of this targeted search capability, an asset type search can return search results more quickly than a Global search. By limiting searches to only the relevant attributes, Asset Type search can eliminate the need for a search to run across unnecessary index data.

Configure Attributes for Asset Type Search enables you to limit a search to only the attributes that you specify for the asset types that are enabled by Asset Type Search. Using the example above of an article asset, if you have configured indexing for specific attributes, such as Headline and Byline, users can search for the string "Jane Doe" in only those attributes.

The tables illustrate the differences in Lucene-based searches and their levels of granularity. Each table represents an index for the same article type asset, but for a different type of search function. Across the tables, only the system-defined attribute data is stored in the same way.

Note: The following tables illustrate the granularity of global searches, asset type searches, and attribute-specific searches. They are not meant to indicate how the search engine actually stores indexed data.

Figure 25–1 Search Function Tables and Differences



Besides the way they index data, the searches enable different functions, as summarized below:

Global Index	Asset Type Index
Enables searches across all selected asset types.	Enables searches per asset type (multiple asset types can be enabled).
Creates one index for all selected asset types.	Creates one index per asset type.
Supports searching by system-defined attributes.	Supports searching by system-defined and custom attributes.
System-defined attributes can be filtered (by using Configuring Attributes for Asset Type Index).	System-defined and custom defined attributes can be filtered (by using Configuring Attributes for Asset Type Index).
Supports public searches on live site.	Supports public searches on live site.

For more information about the Public Site Search API, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

25.2 Setting Up Search Indices

The steps for setting up Lucene are as follows:

1. [Enabling the Lucene Search Engine](#). Before you can configure Lucene, you must enable it on your system.
2. [Adding Asset Types to the Search Index](#). This indicates to Lucene which assets it should index. You can add asset types to the global index and the asset type index.

Once you have selected which assets to index and enabled binary file indexing (if desired), you can start the indexing process.

During indexing, Lucene examines the contents of assets of the selected asset types (and the binary files the assets reference, if applicable) and creates entries for those assets in the index. A global index creates one index for all the selected asset types, while an asset type index creates one index per asset type. Once the global index has been created and users conduct a search, assets will be returned by the search feature in WebCenter Sites' Contributor interface or on the live site. Once the asset type index is created, and users conduct a search, items will be returned by the search feature on the live site.

3. [Configuring Attributes for Asset Type Index](#). Once you have created an index, you can select specific attributes to search on. Once an index for these specified attributes is created and populated, and you conduct a search, the specified attributes will be returned by the search feature on the live site.
4. [Enabling Indexing of Binary Files](#). If one or more asset types which you added to the index are set up to reference binary files, you can configure Lucene to convert the contents of those files to text when indexing assets that reference them.

This section contains the following topics:

- [Section 25.2.1, "Enabling the Lucene Search Engine"](#)
- [Section 25.2.2, "Adding Asset Types to the Search Index"](#)
- [Section 25.2.3, "Configuring Attributes for Asset Type Index"](#)
- [Section 25.2.4, "Binary File Indexing"](#)

25.2.1 Enabling the Lucene Search Engine

This section shows you how to enable the Lucene engine.

To start the Lucene Engine

1. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
2. Click **Start Search Engine**.

Figure 25–2 Enable Indexing Form

Search - Enable Indexing

Use this screen to enable or disable search engine indexing.

A dark, rounded rectangular button with the text "Start Search Engine" in white.

Lucene is now enabled to start indexing selected data. The time it takes to index data varies with the number of assets being indexed and the speed of your system.

Once the Lucene Search Engine is started, it will continue to run until it is disabled. While indexing is running, changes to selected asset types are detected and the index is updated. The status of the asset type is listed as Enabled, while the index is running. If you wish to remove search capability, in addition to stopping indexing, you will also need to delete index data. See [Section 25.4.3, "Deleting Index Data."](#)

25.2.2 Adding Asset Types to the Search Index

This section shows you how to add asset types to the Global Search Index and the Asset Type index. Once each initial index has been created, Lucene checks for changes every 30 seconds. By default, index data is stored in the `<cs_shared_dir>/lucene` directory (where `<cs_shared_dir>` is the Sites shared file system directory). Once the data is added, it will be maintained until indexing is stopped entirely or paused for a selected asset type. Assets of the selected types will not be returned by the search feature in WebCenter Sites' Contributor interface or on the live site until Lucene has indexed them.

To add new asset types to the search index

1. Enable the Lucene engine:
 - a. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
 - b. Click **Start Search Engine**.
2. Add asset types to the global search index:
 - a. Double-click **Configure Global Search**.

- b. In the For index: drop-down list, select **Add**. WebCenter Sites displays a list of asset types that are not currently being indexed.

Figure 25–3 Configure Global Search Form

Configure Global Search

This screen is used to configure the global search index. Actions will not take effect unless global index is running.
 Step 1: Select an action > Step 2: Select Asset Types from table rows > Step 3: Click OK

For index:

- Add:** add asset types to index
- Pause:** pause indexing on asset types
- Delete:** delete asset types from index
- Re-index:** re-index asset types

Start or Stop binary file indexing. Starting this index will allow content of binary files to be searched.

<input type="checkbox"/>	Asset Types	Status
<input type="checkbox"/>	AArticles	Index Enabled
<input type="checkbox"/>	AImages	Index Enabled
<input type="checkbox"/>	AdvCols	Index Enabled
<input type="checkbox"/>	Article	Index Enabled
<input type="checkbox"/>	AttrTypes	Index Enabled
<input type="checkbox"/>	CAttributes	Index Enabled
<input type="checkbox"/>	CGroupTmpls	Index Enabled
<input type="checkbox"/>	Collection	Index Enabled
<input type="checkbox"/>	ContentGroups	Index Enabled

- c. In the Asset Types list, select the asset types you want to index.
 - d. Click **OK**.
 - e. In the confirmation pop-up dialog, click **OK**. The asset type status changes to Enabled and indexing is enabled for the selected asset type. The index is created for that asset type as soon as the first asset of that type is created.
3. Add asset types to the asset type search index.
 - a. Double-click **Configure Asset Type Search**.
 - b. In the **For index:** drop-down list, select **Add**. WebCenter Sites displays a list of asset types that are not currently being indexed.

Figure 25–4 Configure Asset Type Search Form**Configure Asset Type Search**

This screen is used to configure the attribute based asset search index. Actions may not take effect unless attribute based asset index is running.

Step 1: Select an action > Step 2: Select Asset Types from table rows > Step 3: Click OK

For index:

Add: add asset types to index

Pause: pause indexing on asset types

Delete: delete asset types from index

Re-index: re-index asset types

Start Binary Indexing

Start or Stop binary file indexing. Starting this index will allow content of binary files to be searched.

<input type="checkbox"/>	Asset Types	Status
<input type="checkbox"/>	AArticles	Index Disabled
<input type="checkbox"/>	AIImages	Index Disabled
<input type="checkbox"/>	AdvCols	Index Disabled
<input type="checkbox"/>	Article	Index Disabled
<input type="checkbox"/>	AttrTypes	Index Disabled
<input type="checkbox"/>	CAttributes	Index Disabled
<input type="checkbox"/>	CGroupTmpIs	Index Disabled

- c. In the list, select the asset types you want to index.
 - d. Click **OK**. In the confirmation pop-up dialog that appears, click **OK**.
The asset type status changes to Enabled and indexing is enabled for the selected asset type. The index is created for that asset type as soon as the first asset of that type is created.
4. Enable binary file indexing, if desired (click **Start Binary Indexing**). For more information on binary file indexing, see [Section 25.2.4, "Binary File Indexing."](#)

Note: Global search indexing creates one index for all asset types. Asset type index creates one index for each individual asset type.

25.2.3 Configuring Attributes for Asset Type Index

You can configure indexing on specific attributes for specific asset types. The selected asset type needs to be enabled for indexing first before you can select any specific attributes for the asset type. Once you have enabled Lucene and conducted a search on the live site, assets with indexed attribute data matching the search terms will be returned.

To configure attributes for a selected asset type

1. If you have not already done so, add the asset type you want to configure to the asset type index. For instructions, see [Section 25.2.2, "Adding Asset Types to the](#)

[Search Index.](#)"

2. In the **Admin** tab, expand **Search** and double-click **Configure Attributes for Asset Type Index**.
3. In the **Asset Type:** drop-down list, select **the asset type you want to configure**. WebCenter Sites displays a list of attributes for the selected asset type.

Figure 25–5 Configure Assets for Asset Type Index Form

Configure Attributes for Asset Type Index

This screen is used to configure the index source information of the attribute for the selected asset type. Asset types needs to be enabled for search first before they can appear below in the drop down. Actions may not take effect unless attribute based asset index is running.

Asset Type: ▾

Enabled	Attribute	Type	Tokenized	Stored
<input checked="" type="checkbox"/>	createdby	TEXT	false	true
<input checked="" type="checkbox"/>	description	LARGE_TEXT	false	true
<input checked="" type="checkbox"/>	enddate	DATETIME	false	true
<input checked="" type="checkbox"/>	name	TEXT	false	true
<input checked="" type="checkbox"/>	startdate	DATETIME	false	true
<input checked="" type="checkbox"/>	updateddate	DATETIME	false	true

4. The following information is also displayed:
 - **Enabled:** Indicates if the specified attribute is enabled for indexing for that specific asset type. To disable an attribute, deselect its checkbox.
 - **Type:** Indicates the type of data the attribute stores. For example, "Numeric" can indicate that price information is stored. You can also have "Text" and "DateTime" data.
 - **Tokenized:** Select True if you want this data to be converted to text before it is indexed. Data that is not tokenized is indexed as a single word and may not be interpreted by the reader.
 - **Stored:** Indicates if the original entire text of the specified attribute has been added to the index. Select True to store the entire text to the index.

Note: At least one attribute must be enabled for each listed asset type.

5. Once you have made your changes to the selected asset types, click **Save**.

25.2.4 Binary File Indexing

Binary files are files of type other than text, such as Word and PDF documents. You may choose to not enable this option if your assets do not reference binary files, or if the files they reference contain content that is not indexable, such as images and videos.

This section contains the following topics:

- [Section 25.2.4.1, "Enabling Indexing of Binary Files"](#)
- [Section 25.2.4.2, "Disabling Indexing of Binary Files"](#)

25.2.4.1 Enabling Indexing of Binary Files

If one or more asset types which you added to the indexing queue are set up to reference binary files stored in the WebCenter Sites file system, you can configure Lucene to convert the contents of those files to text when indexing the assets that reference them. (By default, Lucene is set up to ignore all binary files referenced by assets being indexed.)

To enable binary file indexing

1. If you have not already done so, enable the Lucene engine:
 - a. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
 - b. Click **Start Search Engine**.

The button name changes to **Stop Search Engine**.

2. In the **Admin** tab, expand **Search**.
3. Do one of the following:
 - To enable binary file indexing for Global search, double-click **Configure Global Search**.
 - To enable binary file indexing for Asset Type search, double-click **Configure Asset Type Search**.
4. Click **Start Binary Indexing**.

Lucene will now convert to text all binary files that are referenced by the assets it indexes.

25.2.4.2 Disabling Indexing of Binary Files

If you decide that you no longer want Lucene to convert the contents of binary files referenced by assets it indexes, you can disable this feature to improve performance.

To disable binary file indexing

1. If you have not already done so, enable the Lucene engine:
 - a. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
 - b. Click **Start Search Engine**.

The button name changes to **Stop Search Engine**.

2. In the **Admin** tab, expand **Search**.
3. Do one of the following:

- To disable binary file indexing for Global search, double-click **Configure Global Search**.
 - To disable binary file indexing for Asset Type search, double-click **Configure Asset Type Search**.
4. Click **End Binary Indexing**.
Lucene will now ignore all binary files referenced by the assets it indexes.

25.3 Disabling the Lucene Search Engine

You can stop the Lucene engine if you want to improve performance. Once the engine is stopped, you will no longer be able to add or delete assets, or pause indexing. You will also no longer be able to re-index assets.

To stop indexing

1. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
2. Click **Stop Search Engine**.

The button name changes to **Start Search Engine**.

Indexing with the Lucene engine is now disabled. The index data is preserved; a search on the Contributor interface or public site will return assets included during the last time the index was built. If you wish to delete the index data (and therefore remove search capability), see [Section 25.4.3, "Deleting Index Data."](#)

Note: If you stop Global search indexing, the search index used for the Contributor interface search function and the public site will not be updated. Therefore, search results will not be accurate.

- To remove search capability from Contributor, you will also need to delete all assets from the Global index. See [Section 25.4.3, "Deleting Index Data"](#) for instructions.
 - If you add or remove assets while indexing is stopped, you need to rebuild the index to create an accurate search index when you restart indexing. See [Section 25.4.2, "Re-indexing"](#) for information on rebuilding indexes.
-
-

25.4 Maintaining Search Indexes

Once you have set up Lucene, you may need to perform tasks such as temporarily suspending indexing in order to perform bulk operations on assets, re-indexing, deleting index data, or writing code to specifically query the search engine.

This section contains the following topics:

- [Section 25.4.1, "Pausing and Resuming Indexing"](#)
- [Section 25.4.2, "Re-indexing"](#)
- [Section 25.4.3, "Deleting Index Data"](#)

25.4.1 Pausing and Resuming Indexing

Pausing and stopping indexing are similar functions. When you pause indexing, you have the ability to pause indexing for selected asset types, whereas when you stop the search index you stop indexing on all assets.

When you add and delete large numbers of assets, you can speed up the process by temporarily pausing indexing on the assets of the type you are adding or deleting. To reflect these changes in your search index, you will then need to index all assets of the type that you added or deleted, using the re-indexing function.

- [Section 25.4.1.1, "Pausing Global and Asset Type Indexing"](#)
- [Section 25.4.1.2, "Resuming Global and Asset Type Indexing"](#)

25.4.1.1 Pausing Global and Asset Type Indexing

When indexing is enabled, every asset that is added or updated to the WebCenter Sites database is indexed after it is saved. Saving a large number of assets will proceed faster if you pause the indexing of assets of that type. For example, you can pause indexing when performing a bulk import of assets into the WebCenter Sites database. You can then resume indexing and re-index all assets of that type after the assets are added to the database, indexing all the new (and existing) assets at one time.

When you pause indexing for an asset type, Lucene does the following:

- Stops indexing assets of the selected type.
- Preserves the index data for assets of the selected type.

When indexing is paused, searches continue to return results against the existing index. However, changes to the database made after indexing is paused are not indexed. Therefore, search results will not reflect changes made to the database after indexing was paused.

To pause indexing

1. If you have not already done so, enable the Lucene engine:
 - a. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
 - b. Click **Start Search Engine**.

The button name changes to **Stop Search Engine**.

2. In the **Admin** tab, expand **Search**.
3. Do one of the following:
 - To pause indexing for Global search, double-click **Configure Global Search**.
 - To pause indexing for Asset Type search, double-click **Configure Asset Type Search**.
4. In the **For index:** drop-down list, select **Pause**. WebCenter Sites displays the list of asset types for which you can pause indexing.

Figure 25–6 Configure Global Search Form

Configure Global Search

This screen is used to configure the global search index. Actions will not take effect unless global index is running.

Step 1: Select an action > Step 2: Select Asset Types from table rows > Step 3: Click OK

For index:

Start Binary Indexing

Start or Stop binary file indexing. Starting this index will allow content of binary files to be searched.

- Add:** add asset types to index
- Pause:** pause indexing on asset types
- Delete:** delete asset types from index
- Re-index:** re-index asset types

<input type="checkbox"/>	Asset Types	Status
<input type="checkbox"/>	AArticles	Index Enabled
<input type="checkbox"/>	AImages	Index Enabled
<input type="checkbox"/>	AdvCols	Index Enabled
<input type="checkbox"/>	Article	Index Enabled
<input type="checkbox"/>	AttrTypes	Index Enabled
<input type="checkbox"/>	CAttributes	Index Enabled
<input type="checkbox"/>	CGroupTmps	Index Enabled
<input type="checkbox"/>	Collection	Index Enabled

5. In the **Asset Types** list, select the asset types for which you want to pause indexing.

Note: If no asset types are displayed when you select Pause from the drop-down list, stop here. Either indexing is already paused for all asset types, or no asset types have yet been selected for indexing.

6. Click the **OK** button next to the drop-down list of operation selections.
7. In the confirmation pop-up dialog that appears, click **OK**.

The Lucene Search Engine pauses indexing on assets of the selected types and preserves their index data. The status of the asset type changes to Paused.

Changes to the database after indexing was paused are not indexed; therefore, search results will not reflect changes made to the database after indexing pauses.

25.4.1.2 Resuming Global and Asset Type Indexing

After pausing or disabling indexing, you will need to re-index to ensure that all your asset data is in the index. Follow the steps below to restart indexing.

To resume indexing

1. Restart indexing on the paused asset types. For instructions, see [Section 25.2.2, "Adding Asset Types to the Search Index."](#)
2. If you added assets to the database while indexing was paused, you must re-index to ensure that new data is included in the index. Proceed to [Section 25.4.2, "Re-indexing."](#)
3. If you deleted assets while indexing was paused, the regular indexing process will detect which assets were deleted and remove that data from the index. However, if a large number of assets were deleted, it may be faster to delete the entire index for assets of the type you deleted and then re-index.
 - For more information on deleting index data, see [Section 25.4.3, "Deleting Index Data."](#)
 - Information on re-indexing can be found in the section [Section 25.4.2, "Re-indexing."](#)

25.4.2 Re-indexing

While indexing is paused or stopped, WebCenter Sites does not track the additional assets added to the database. Therefore, to search those assets, all assets of the type for which indexing was paused must be re-indexed.

The time it takes to re-index assets varies with the number of assets being indexed and your system configuration. Updated search results for assets of the selected types will be returned only after the Lucene search engine has indexed them.

To re-index assets

1. If you have not already done so, enable the Lucene engine:
 - a. In the **Admin** tab, expand **Search** and double-click **Start/Stop Search Engine Indices**.
 - b. Click **Start Search Engine**.
2. In the **Admin** tab, expand **Search**.
3. Do one of the following:
 - To re-index assets for Global search, double-click **Configure Global Search**.
 - To re-index assets for Asset Type search, double-click **Configure Asset Type Search**.
4. In the **For index:** drop-down list, select **Re-index**.

WebCenter Sites displays the asset types currently selected for indexing.

Note: If no asset types are displayed when you select Re-index from the drop-down list, stop here. No asset types are in the indexing queue or indexing has been paused for all asset types in the queue.

Figure 25–7 Configure Global Search Form

Configure Global Search

This screen is used to configure the global search index. Actions will not take effect unless global index is running.
 Step 1: Select an action > Step 2: Select Asset Types from table rows > Step 3: Click OK

For index:

Add: add asset types to index
Pause: pause indexing on asset types
Delete: delete asset types from index
Re-index: re-index asset types

Start or Stop binary file indexing. Starting this index will allow content of binary files to be searched.

<input type="checkbox"/>	Asset Types	Status
<input type="checkbox"/>	AArticles	Index Enabled
<input type="checkbox"/>	AImages	Index Enabled
<input type="checkbox"/>	AdvCols	Index Enabled
<input type="checkbox"/>	Article	Index Enabled
<input type="checkbox"/>	AttrTypes	Index Enabled
<input type="checkbox"/>	CAttributes	Index Enabled
<input type="checkbox"/>	CGroupTmps	Index Enabled
<input type="checkbox"/>	Collection	Index Enabled

5. In the list, select the asset types whose index data you want to build (or rebuild).
6. Click **OK**.
7. In the confirmation pop-up dialog that appears, click **OK**.

Indexing begins.

The status of the selected asset types changes to Enabled.

Updated search results for assets of the selected types will be returned only after the Lucene search engine has indexed them.

25.4.3 Deleting Index Data

If you no longer need to perform searches on assets of a particular type, search results will be returned more quickly if the unnecessary data is removed from the index.

You may also wish to delete indexes if you stopped indexing and then deleted a large number of assets. In this case, it may be faster to delete the relevant index data and then re-index the remaining assets than to allow the regular indexing process to run through its normal process to detect which assets have been deleted. More information on pausing and restarting indexing can be found in [Section 25.4.1, "Pausing and Resuming Indexing."](#)

When you delete index data, WebCenter Sites does the following:

- Pauses indexing on assets of the selected asset types.
- Deletes the index data for assets of the selected asset types.

After you perform the steps below, index data is no longer available for assets of the selected types. Search results will no longer return data from assets of the selected types.

To delete data from the index

1. In the **Admin** tab, expand **Search**.
2. Do one of the following:
 - To delete assets from Global search, double-click **Configure Global Search**.
 - To delete assets from Asset Type search, double-click **Configure Asset Type Search**.
3. In the **For index:** drop-down list, select **Delete**.
4. WebCenter Sites displays the asset types currently being indexed.

Figure 25–8 Configure Global Search Form

Configure Global Search

This screen is used to configure the global search index. Actions will not take effect unless global index is running.
 Step 1: Select an action > Step 2: Select Asset Types from table rows > Step 3: Click OK

For index:

Add: add asset types to index
Pause: pause indexing on asset types
Delete: delete asset types from index
Re-index: re-index asset types

Start or Stop binary file indexing. Starting this index will allow content of binary files to be searched.

<input type="checkbox"/>	Asset Types	Status
<input type="checkbox"/>	AArticles	Index Enabled
<input type="checkbox"/>	AImages	Index Enabled
<input type="checkbox"/>	AdvCols	Index Enabled
<input type="checkbox"/>	Article	Index Enabled
<input type="checkbox"/>	AttrTypes	Index Enabled
<input type="checkbox"/>	CAttributes	Index Enabled
<input type="checkbox"/>	CGroupTmpIs	Index Enabled
<input type="checkbox"/>	Collection	Index Enabled

Note: If no asset types are displayed when you select **Delete** from the drop-down list, stop here. No assets are being indexed.

5. In the list, select the asset types whose index data you want to delete.
6. Click **OK**.
7. In the confirmation dialog box that appears, click **OK**.

WebCenter Sites pauses indexing on assets of the selected types and deletes their index data.

In Configure Global Search, the status of the asset types changes to Paused. This status indicates that no new assets will be added to the existing index.

In Configure Asset Type Search, the status of the asset types changes to Disabled. This status indicates that this asset is no longer eligible for indexing.

The assets will no longer be returned by the search feature in WebCenter Sites' Contributor interface or the public site.

To make the assets searchable again, you must add the asset types back to indexing. For instructions, see [Section 25.2.2, "Adding Asset Types to the Search Index."](#)

25.5 Writing Code that Queries the Search Index

The following sample code illustrates how to query the Lucene search engine index. This code is based on the assumption that the user wants to search against a particular site and a particular asset type, where the site is passed in as variable `currentSite` and type is passed in as `assetType`. The user would use the following code to write a query against the Global index. The Lucene search engine would return all the assets or the `maxResults` (if total is greater than `maxResults`) specified of the type that belongs to the specified site.

```
ICS ics = Factory.newCS();
IndexSourceConfig srcConfig = new IndexSourceConfigImpl(ics);
SearchEngineConfig engConfig = new SearchEngineConfigImpl(ics);
IndexSourceMetadata sourceMd =
    srcConfig.getConfiguration("Global");
String engineName = sourceMd.getSearchEngineName();
SearchEngine eng = engConfig.getEngine(engineName);
String currentSite = (String)
    props.get(SearchIndexFields.Global.SITEID);
QueryExpression siteExpr = new
    QueryExpressionImpl(SearchIndexFields.Global.SITEID,
        Operation.CONTAINS, currentSite);
siteExpr = siteExpr.or(SearchIndexFields.Global.SITEID, Operation.EQUALS, "0");
QueryExpression typeQ = new
    QueryExpressionImpl(SearchIndexFields.Global.ASSET_TYPE, Operation.EQUALS,
        assetType);
QueryExpression qe = typeQ.and(siteExpr);
qe.setMaxResults(maxResults);
SearchResult<ResultRow> res =
    eng.search(Collections.singletonList("Global"), qe);
```

Revision Tracking

WebCenter Sites provides revision tracking functionality that prevents a row in a table from being edited by more than one user at a time. When you enable revision tracking for a table, WebCenter Sites maintains multiple versions of a row in that table.

WebCenter Sites applies this revision tracking functionality to your asset types. You decide which asset types should be tracked and determine how many revisions to store. The content providers then check their assets in and out and can compare versions, if necessary.

This chapter describes how to enable revision tracking and how to manage versions of assets.

This chapter contains the following sections:

- [Section 26.1, "Overview"](#)
- [Section 26.2, "Enabling Revision Tracking"](#)
- [Section 26.3, "Disabling Revision Tracking"](#)
- [Section 26.4, "Unlocking Revisions"](#)

26.1 Overview

WebCenter Sites provides revision tracking functionality through its revision tracking API. This API is used to provide additional revision tracking functionality for asset type tables.

When you enable revision tracking for an asset type, WebCenter Sites creates a new table, called a tracker table, for assets of that type. You specify how many versions you want to keep and you also specify a storage directory that the tracker table uses to store supporting files for the assets that it is tracking.

WebCenter Sites' implementation of revision tracking provides the following features:

- **Check out and check in.**

Check out locks an asset so that only one user can edit it at a time.

Check in releases the lock on the asset, increments the version number, and determines whether the number of versions falls within the configured limit. If the new version exceeds the limit, the oldest version is deleted to make room for the next version.

- Storage of multiple versions of an asset.

When you enable revision tracking for an asset type, WebCenter Sites stores versions in a tracker table; upload data is stored in a storage directory that you

specify, as shown in [Table 26.1.2, "The RTInfo Table"](#). Because past versions are stored (that is, a history exists), a user can roll back an asset to a previous version or examine the differences between two versions of the asset.

- Administrative or maintenance features.

An administrator can delete past versions of an asset or clear the checkout for an asset by overriding the check out on it and checking it back in.

When an asset type is being tracked, WebCenter Sites provides checkin, checkout, and other revision tracking features on the New and Edit forms for assets of those types.

This section contains the following topics:

- [Section 26.1.1, "Tracker Tables and Storage Directories"](#)
- [Section 26.1.2, "The RTInfo Table"](#)
- [Section 26.1.3, "Revision Tracking and the Two Asset Models"](#)
- [Section 26.1.4, "Implicit vs. Explicit Checkin and Explicit Checkout"](#)
- [Section 26.1.5, "Revision Tracking and Non-Asset Tables"](#)
- [Section 26.1.6, "How Many Versions?"](#)

26.1.1 Tracker Tables and Storage Directories

When you enable revision tracking for an asset type, WebCenter Sites creates a tracker table that stores revision information for the records in the source table. A tracker table has the same name as the main storage table for the asset type with `_t` appended to it. For example, the tracker table for the article asset type would be named `Article_t`. The tracker table for the attribute type asset type would be named `AttrTypes_t`.

For each record in a tracked asset type table, there are several rows in the corresponding tracker table that stores its version information. Tracker tables have two kinds of columns:

- Columns that store the system information that the revision tracking system needs to keep track of all the versions
- Columns that hold the IDs of text files that are stored in a storage directory

When a new version of an asset is checked in, WebCenter Sites creates a separate text file to hold the data in each of the asset type's upload (URL) fields and in any text fields that are configured to hold more than 64 characters. These files are stored in `shared/rtStorage/AssetType`. There is a set of these text files stored in the storage directory for each version of the asset.

Note: Because tracker tables hold system information only, they are hidden in Sites Explorer. Do not attempt to modify the information in any of the tracker tables with a database tool.

26.1.2 The RTInfo Table

While the tracker tables are kept hidden, the RTInfo table is visible through Sites Explorer. This table holds information about which tables are being revision tracked. It has the following columns:

Table 26–1 RTInfo Table

Column	Description
tblname	The name of a table that is being revision tracked. For asset types, this is the name of the main asset type table.
versions	The number of versions to store for each asset of this type
storage	The path to the storage directory that holds the text files for each version of assets of this type. The path is in the shared folder, located in <i>shared/rtStorage/AssetType</i> .
recordupdate	A timestamp of the last time a version was stored in the tracker table.
trackingupdate	The time at which revision tracking was enabled for the source table.

26.1.3 Revision Tracking and the Two Asset Models

Because the data model for basic assets is different from the data model for flex assets, the revision tracking system works differently for the two asset models:

- For basic assets, only the row in the main asset storage table is tracked.
- For flex assets and the other multi-table asset types (template and CSElement), the information from the appropriate rows from all of their tables are serialized into an object and stored in the tracker table for that asset type.

For example, if you enable revision tracking for template assets, the appropriate rows from the `Template`, `SiteCatalog`, and `ElementCatalog` tables are serialized into an object and stored in the `Template_t` table.

26.1.4 Implicit vs. Explicit Checkin and Explicit Checkout

When revision tracking is on for an asset type, WebCenter Sites provides both implicit (or automatic) and explicit (or manual) checkout, and explicit checkin functionality. When users create or edit assets of a type that is being revision tracked, they do not have to manually check out the asset: it is automatically assigned to them. However, implicit checkin is not allowed; when they click Save, the asset is not checked back in. Users must manually check assets in.

This may or may not be the behavior that you want. For example, if an author is making extensive revisions to an asset that was checked out implicitly, the changes will not be available until the user also explicitly checks in the asset. This will help prevent the user from overwriting an older version that he or she really wanted to keep. An additional example is when an author implicitly checks out an asset to make edits, then navigates to a different area, then comes back to the asset. Once an implicitly checked out asset is navigated away from, it must be explicitly checked in when editing is complete.

Implicit checkouts are used for quick edits where the editing and saving would be done in one operation. These types of updates that are expected to be done with minimal clicks. Explicit checkouts are for changes to an asset over time.

When a content provider implicitly or explicitly checks out an asset, a version is not stored, no matter how many times he or she saves it, until it is manually checked back in.

26.1.5 Revision Tracking and Non-Asset Tables

In addition to using revision tracking for your asset types, you can implement revision tracking on your non-asset tables.

To do so, you use the User Access Management feature in the Admin tab to enable tracking for the table. WebCenter Sites then creates a corresponding tracker table to support the tracked table. It is named the same way as a tracker table for an asset type: `nameOfTable_t`. For example, if you enabled revision tracking for the Source table, the tracker table would be called `Source_t`.

When revision tracking is enabled for a non-asset table, you can use either the revision tracking features accessible from the menus in Sites Explorer or the **User Access Management** node on the **Admin** tab to lock (check out) a row and then unlock the row (check it back in) when you are finished with it.

If you need to provide additional support outside of the Sites Explorer tool for revision tracking of a non-asset table, your developers can code additional forms, using the WebCenter Sites revision tracking API and revision tracking XML or JSP tags. For information, see the *Oracle Fusion Middleware WebCenter Sites: Tag Reference* and the *Oracle Fusion Middleware WebCenter Sites: Java API Reference*.

Note: If you need to delete a non-asset table from the WebCenter Sites database and that table is being revision tracked, be sure to untrack the table before deleting it.

26.1.6 How Many Versions?

Each revision of an asset or a database row occupies disk space. Therefore, your decision about how many revisions to keep must be based on the following factors:

- The amount of disk space that you have available
- The typical data size of the asset (or row)
- The likelihood that there could be a need for a rollback of several versions

For example, an asset that consists of a small amount of ASCII data occupies so little space that a large number of revisions would take little disk space. However, each version of an asset that holds a large amount of binary data could occupy a significant amount of disk space. In the second case, you must strike the appropriate balance, storing the fewest number of versions necessary for rollback purposes.

26.2 Enabling Revision Tracking

This section shows you how to enable revision tracking for assets and for non-asset tables.

Note: If Revision Tracking is turned on at the publishing target destination for either ElementCatalog or SiteCatalog, then the publishing of template will fail and the template may become corrupted.

It is not recommended to enable Revision Tracking on the target destination.

This section contains the following topics:

- Section 26.2.1, "Enabling Revision Tracking for Assets"
- Section 26.2.2, "Enabling Revision Tracking for Non-Asset Tables"
- Section 26.2.3, "Editing Revision Tracking Settings"

26.2.1 Enabling Revision Tracking for Assets

To enable revision tracking for assets

1. In the **Admin** tab, expand **Asset Types**, then expand the asset type for which you want to enable revision tracking.
2. Under the selected asset type, expand **Revision Tracking** and double-click **Track**.
The Track Asset Type form appears:

Figure 26–1 Revision Tracking for Asset Type Form

Revision Tracking for Asset Type: ▶Collection

*Revisions to Keep:

Cancel

Enable Revision Tracking

3. In the **Revisions to Keep** field, enter the number of revisions you want WebCenter Sites to store for each asset of the selected asset type. Once this many revisions are stored, the oldest revision is overwritten by the next revision.

If you are using revision tracking solely for its record-locking feature and you do not need the ability to roll back to previous versions, you can set this field to 1.

4. Click **Enable Revision Tracking**.

26.2.2 Enabling Revision Tracking for Non-Asset Tables

To enable revision tracking for database tables that do not hold assets, use the **User Access Management** node.

To enable revision tracking for non-asset tables

1. In the **Admin** tab, expand **User Access Management** and double-click **Revision Tracking**.
2. In the form that appears, select **Track Tables** and click **OK**.
3. In the **Enter root storage directory** field, enter the full path to the directory in which revisions of table rows will be stored. Do not add a slash (or backslash) character after the directory name.
4. In the **Enter number of revisions to keep** field, enter the number of revisions WebCenter Sites should store for rollback purposes.
5. Select the **Track?** check box next to each table for which you are enabling revision tracking.
6. Click **Track Tables**.

26.2.3 Editing Revision Tracking Settings

You must have the SiteGod, Xceladmin, Visitor Admin or Remote Client ACL to either inspect or edit revision tracking settings. If you attempt to examine the current revision tracking settings for a tracked asset type or non-asset table and you do not have one of those ACLs, the system does not display the name of the root storage directory or the number of revisions.

Because revision tracking settings directory affect the database, you must be careful when changing these settings.

This section contains the following topics:

- [Section 26.2.3.1, "Changing the Root Storage Directory"](#)
- [Section 26.2.3.2, "Changing the Number of Revisions"](#)

26.2.3.1 Changing the Root Storage Directory

Caution: If you change the root storage directory for an asset type or non-asset table, you will lose all the versions currently stored in it.

If you must change the root storage directory, follow these steps:

1. Disable revision tracking for the asset type or table. When you do this, all the version data for the asset type is orphaned.
2. Enable revision tracking for the asset type or table, entering the new root storage directory.

26.2.3.2 Changing the Number of Revisions

The following sections provide information about increasing and decreasing the number of revisions to be stored for an asset type or table.

Increasing

If you want to increase the number of revisions to be stored for an asset type, simply increase the value in the Revisions to Keep field.

To increase the value, in the **Admin** tab, expand **Asset Types**, expand the asset type you want to modify, **Revision Tracking**, and double-click **Set Revisions**. Use the form that appears to increase the value.

Decreasing

Although you might need to decrease the number of versions while you are testing configuration settings on a WebCenter Sites development system or while you are fine-tuning the Sites management system, it is best if you do not decrease the number of versions being stored by Revision Tracking on a fully functioning management system.

If you decrease the value in the Revisions to Keep field to a number that is less than the number of revisions currently being stored for that asset type or table, the following occurs:

- The text files in the storage directory for the extra versions are orphaned.
- The rows in the tracker table for the extra versions are orphaned.

You can avoid creating orphan rows in the tracker table by deleting the extra versions (the oldest ones) before you decrease the number of revisions. However, you cannot avoid creating orphan text files in the storage directory and you should not attempt to delete them because it is very difficult to determine which ones to delete.

If you must decrease the number of revisions, use the form associated with the asset type. In the **Admin** tab, expand **Asset Types**, the asset type you want to modify, **Revision Tracking**, and double-click **Set Revisions**. Use the form that appears to decrease the value.

26.3 Disabling Revision Tracking

When you disable revision tracking for an asset type or a non-asset table, the following occurs:

- The tracker table is inactivated, but not deleted. This means that the versions stored in it are orphaned, but not deleted. And, after revision tracking is disabled for a table, you can no longer delete versions by using the WebCenter Sites revision tracking forms.
- All links to the text files in the storage directory are broken, but the files themselves are not deleted. They, too, are orphaned.

If you later decide to enable revision tracking for that asset type, the old versions and text files are ignored. They remain orphaned.

Because these old versions could take up unwanted space, it may be necessary to delete them manually. Manually delete all the text files from the revision tracking directory located inside the storage directory for that asset type.

For example, if the storage directory for the asset type is `/Storage/AssetType`, then delete all the files from the `/Storage/AssetType/AssetType` revision tracking directory.

Note: Do not delete any files from the asset type storage directory.

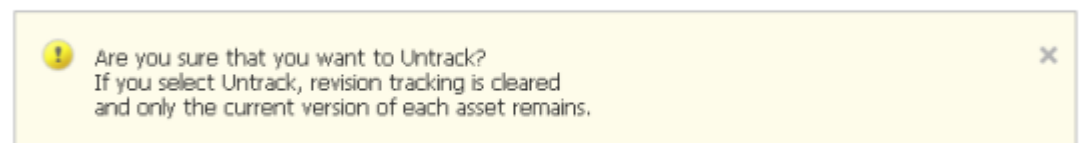
To disable revision tracking for asset types

1. In the **Admin** tab, expand **Asset Types**, then the asset type for which you want to disable revision tracking.
2. Under the selected asset type, expand **Revision Tracking** and double-click **Untrack**.

WebCenter Sites displays a warning message:

Figure 26–2 Revision Tracking for Asset Type Warning Message

Revision Tracking for Asset Type: [▶ Article Flex](#)



Cancel Untrack

3. Click **Untrack**.

26.4 Unlocking Revisions

Occasionally, user operations may leave an asset in an inappropriate locked state. Resolving these states is an administrative responsibility.

For asset types, use the **Clear Checkouts** function in the **Admin** tab.

Clearing Checkouts for Assets

Before you begin, find out the name of the user who has the asset locked. This information is listed on the Inspect form for the asset.

To clear checkouts for assets

1. In the **Admin** tab, double-click **Clear Checkouts**.

WebCenter Sites displays the Search for Checkouts form:

Figure 26–3 Search for Checkouts Form

Search for Checkouts

*User Name:

Show Checkouts

2. Enter the name of the user who has left the asset locked and click **Show Checkouts**.
3. In the list of assets checked out to the selected user, select the **Clear** check box next to each asset for which you want to undo the checkout.
4. Click **Clear Checkouts**.

inCache Framework

Oracle WebCenter Sites and Satellite Server ship with a caching system called the *inCache framework*, or simply *inCache*. This system is built on top of Ehcache, an open source, standards-based product from Terracotta. Compared to the legacy method of caching to the database, inCache offers improved website performance. This chapter introduces the inCache framework.

This chapter contains the following sections:

- Section 27.1, "What is inCache?"
- Section 27.2, "How inCache Works"
- Section 27.3, "Restarting a Node"
- Section 27.4, "Page Regeneration During RealTime Publishing"
- Section 27.5, "Double-Buffered Caching"
- Section 27.6, "inCache Features for Remote Satellite Server"
- Section 27.7, "Summary"
- Section 27.8, "Next Steps"

27.1 What is inCache?

inCache is a high performance, memory-based page and asset caching system that eliminates the need to cache Oracle WebCenter Sites' data in a central, shared repository and shared file system. inCache is based on Ehcache, an open source Java caching framework from Terracotta, and can be implemented on top of any caching system. (For more information about Ehcache, see <http://ehcache.org/>.)

The inCache framework replaces the legacy page caching framework. inCache also supports asset caching, by default. The inCache framework offers the following benefits:

- High performance. Fast response times allow for a greater frequency of publishing sessions. Efficient page and asset invalidation ensures that updated content is quickly served.
- Decentralized architecture. Nodes maintain their own local caches. The inCache framework eliminates database caching and shared disks.
- Broadcasting system. Each individual node no longer has to have a complete view of the entire cache. Nodes listening for changes to content respond to broadcasts as necessary, depending on their individual content.
- Improved linear scalability, enhanced for data on disk and in memory.

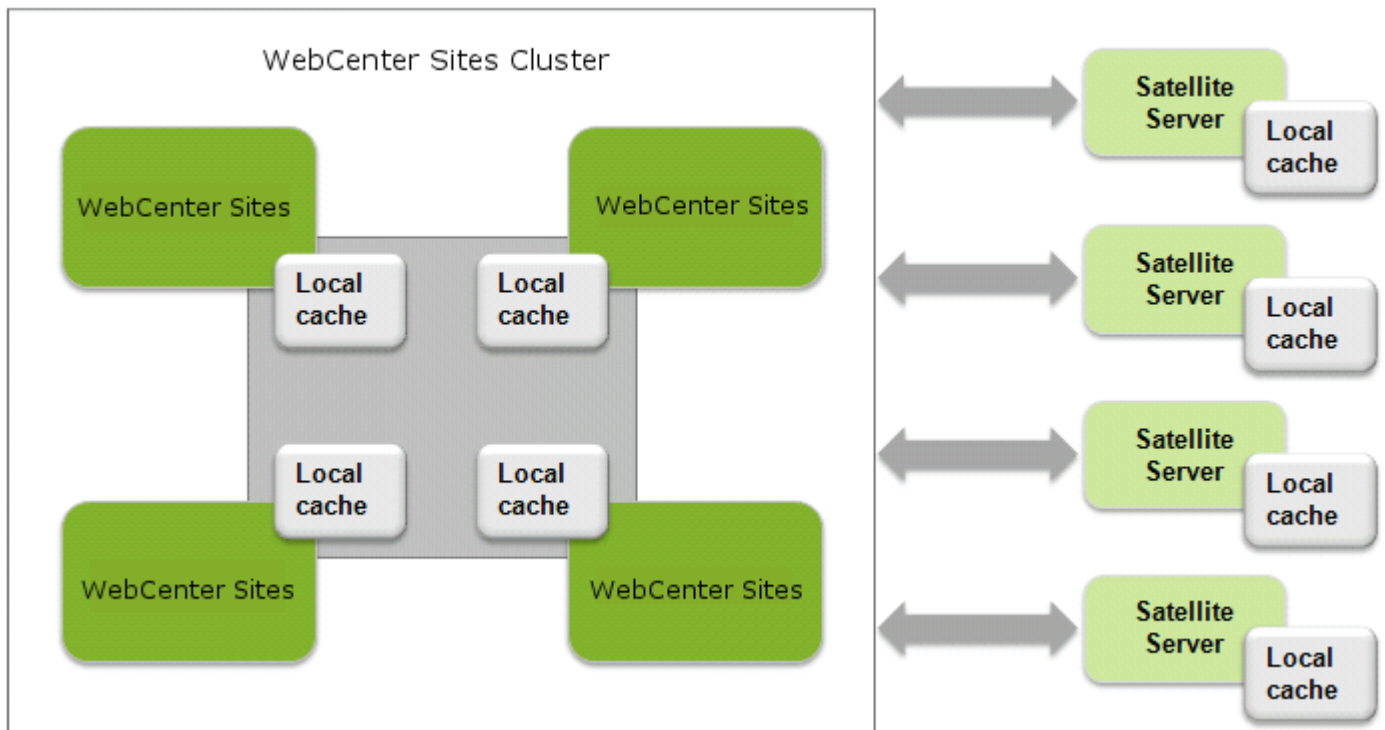
- Failover and persistence. Nodes that are shut down retain data in their local caches and update themselves upon restart against a centrally managed record of invalidations.
- On-demand page evaluation and invalidation. Nodes validate and update their currently cached content only when the content is requested.

This chapter describes mainly page caching. Throughout, the term "page" is used to mean rendered page rather than page asset. To follow the discussion of asset caching, we recommend first reading the rest of this chapter, especially [Section 27.2, "How inCache Works,"](#) for a basic understanding of cache containers and how they function in relation to each other.

27.2 How inCache Works

In a clustered environment, inCache is implemented in a decentralized architecture outside the cluster's database. Each node has its own local cache, as shown in [Figure 27-1, "Local Caches"](#). Each Oracle WebCenter Sites node keeps its local cache current by listening for broadcasts from other Oracle WebCenter Sites nodes and communicates updates to remote Satellite Server via HTTP.

Figure 27-1 Local Caches



A node's local cache is partitioned:

- The `pageByQry` cache stores the node's web pages.
- The `dependencyRepository` cache stores identifiers of the assets that make up the web pages.

Note: The `pageByQry` cache is also called page cache.

The `dependencyRepository` cache is also called dependency cache.

- The `notifier` cache broadcasts identifiers of assets that have been modified by editorial or publishing processes. Broadcasts are initiated by the nodes on which assets are modified. How a listening node responds to a broadcast depends on the node. For example:
 - a. A WebCenter Sites user at node A edits asset A.
 - b. The `notifier` cache on node A broadcasts a notice of change to all other nodes.
 - c. Every WebCenter Sites node that contains asset A responds by invalidating the asset in its own local cache. Each node refers to its own `dependencyRepository` cache, marks the asset's identifier as invalid, and increments the dependency generation counter for the entire cache. Because the invalidated asset is no longer available to pages that reference the asset, the pages themselves are invalidated. However, the node does not evaluate the pages until they are requested. Herein lies the performance benefit.

When a node responds to a request for a page that has a dependency on the invalidated asset, the node refers to its `pageByQry` cache, evaluates the page, determines the page to be invalid, flushes the page, generates the new page, serves the page to the visitor's browser, and records the new page along with its **dependencies** in the local cache (dependencies are recorded as identifiers of the assets that make up the page). From that point on, the same page is served from the node's local cache until the page expires or its assets are again invalidated.

WebCenter Sites and Satellite Server use the same caching framework, with one main difference: Satellite Server nodes still communicate with WebCenter Sites nodes via HTTP.

Note: While the `pageByQuery` cache is used only for page caching, it illustrates how a cache container works. For asset caching, the `inCache` framework introduces a counterpart container called `AssetCache`. It works in a similar manner to `pageByQuery` and interacts with the `dependencyRepository` and `notifier` caches, described above. Additional information about asset caching is available in [Chapter 29, "inCache for Asset Caching."](#)

27.3 Restarting a Node

When a WebCenter Sites or Satellite Server node is shut down, data in its cache persists, but can become quickly outdated as the active nodes continue to invalidate assets. As a result, restarting a node requires updating its cache. Update on restart is ensured by a common invalidation memory, which is stored as a table in the database of the WebCenter Sites cluster and kept available to all nodes in the cluster for use upon restart. The table is named `FW_InvalidationMemory`.

The invalidation memory stores records of asset invalidations, specifically, identifiers of assets that have been modified or deleted during a content management or publishing process. Table growth is checked by a timer-based cleanup mechanism that runs at 15-minute intervals to purge invalidation records for the oldest period of time.

When a node is restarted, it attempts to recover information that it missed during its inactive period and therefore refers to the invalidation memory:

- If invalidation records exist for the node's inactive period, the node replays them on itself.
- If no assets were invalidated during the inactive period, the node continues to operate as if it were never shut down.
- If invalidation records were purged by the cleanup mechanism while the node was inactive, the node's cache self-destructs and must be rebuilt.

When a remote Satellite Server is restarted, it obtains the information it missed by sending a request to WebCenter Sites for an update.

27.4 Page Regeneration During RealTime Publishing

Enabling `inCache` deactivates the `donotregenerate` flag for the RealTime publishing process. Because the flag is no longer recognized, crawling is used to regenerate pages during RealTime publishing sessions. Crawling is a computationally expensive option. If it is not implemented, pages will be regenerated only when they are requested.

The crawling option requires specifying a set of URLs to be analyzed by the WebCenter Sites page regenerator. Typically, they are the URLs of the home page and other high-traffic pages. In addition, you can specify the depth to which those pages will be crawled. For example, a depth of 1 means that the specified pages and pages they link to will be crawled, while a depth of 0 means that only the specified pages will be crawled. Crawled pages are regenerated only if their component assets have been invalidated during the publishing session, or the pages are not cached. All pagelets on the specified pages are regenerated in the process.

The list of URLs to crawl and the crawl depth must be specified in the `FW_RegenCriteria` table, which is created on the delivery system during the first publishing session after `inCache` is configured. The `ft_ss` parameter can be included in the URL to specify that requests are handled either directly by WebCenter Sites or by remote Satellite Server. Instructions for configuring `inCache` and enabling page regeneration can be found in [Chapter 28, "inCache for Page Caching."](#)

27.5 Double-Buffered Caching

WebCenter Sites' double-buffered page caching method uses the WebCenter Sites and Satellite Server caches in tandem on live web sites. Double buffering ensures that pages are always kept in cache, either on WebCenter Sites or Satellite Server, to protect WebCenter Sites from an overload of page requests and prevent the live web site from displaying blank pages and broken links.

To maintain the traditional system of double-buffered caching for the `inCache` framework, we ensured that remote Satellite Server continues to communicate with WebCenter Sites via http requests. Satellite Server still reads page data via http requests and caches in the usual way. However, page data now includes dependency information in the form of a comma-separated list of asset identifiers which is also streamed to remote Satellite Server.

27.6 inCache Features for Remote Satellite Server

Remote Satellite Server is used only in page caching. It can be configured to support advanced functionality such as page propagation and page regeneration in background.

This section contains the following topics:

- [Section 27.6.1, "Page Propagation"](#)
- [Section 27.6.2, "Page Regeneration in Background"](#)

27.6.1 Page Propagation

The page propagation option enables all WebCenter Sites nodes and Satellite Server nodes to host the same pages without each node having to regenerate the pages. Instead of referring to the database to regenerate pages, nodes receive newly generated and regenerated pages into their local caches from the nodes on which the pages were (re)generated and cached. Caching the pages triggers their propagation. Instructions for configuring page propagation can be found in [Chapter 28, "inCache for Page Caching."](#)

27.6.2 Page Regeneration in Background

Remote Satellite Server can be configured to serve invalidated pagelets while they are being regenerated by a background process. See [Section 28.3.4, "Configuring for Pagelet Regeneration in Background"](#) for more information.

27.7 Summary

The inCache framework significantly increases performance. Nodes can retain cache on disk and recover from failure. Decentralized architecture prevents bottlenecks (although the lack of a central repository for cached items can make it difficult to determine the overall state of all caches). Page propagation eliminates the need to regenerate pages, while page regeneration in background enables remote Satellite Servers to continue serving pages while their replacements are being generated.

This new framework deactivates the `donotregenerate` flag for RealTime publishing. Pages that must be regenerated during a publishing session must be specified in the `FW_RegenCriteria` table. Otherwise, pages are regenerated when they are requested.

27.8 Next Steps

For more information about page caching and configuration methods, see [Chapter 28, "inCache for Page Caching."](#) The same chapter contains information about options such as striping the disk, enabling page regeneration to occur during RealTime publishing, setting up page propagation, configuring page regeneration in background, and returning to the traditional system of page caching.

For information about asset caching, see [Chapter 29, "inCache for Asset Caching."](#)

inCache for Page Caching

When WebCenter Sites is installed or upgraded, inCache for page caching is enabled by default. inCache for page caching overrides the legacy method of page caching. Two configuration files, `cs-cache.xml` and `ss-cache.xml`, are provided with WebCenter Sites for configuring local caches and peer-to-peer communication.

- [Section 28.1, "Overview"](#)
- [Section 28.2, "Configuring Your System for inCache Page Caching"](#)
- [Section 28.3, "Tuning Options"](#)

28.1 Overview

inCache supports disk striping and affects RealTime publishing by deactivating the `donoteregenerate` flag. The option to enable page regeneration during RealTime publishing requires populating the `FW_RegenCriteria` table, on the delivery system, with the URLs of pages to be crawled and regenerated. Page propagation is also an option, used to ensure that all nodes host the same pages without each node having to regenerate the pages. In addition, remote Satellite Server can be configured to continue serving stale pages for a short duration while their replacements are being regenerated.

Note: You can return to the legacy page caching method, as necessary, by:

1. Setting the VM argument on all WebCenter Sites and Satellite Server nodes as follows: `-Dcs.useEhcache=false`
2. Reconfiguring all WebCenter Sites and Satellite Server nodes to use the database and shared file system.
3. You will also have to purge the legacy page cache before resuming operations, given that it is likely to have become outdated since the time it was disabled.

In general, switching between caching methods requires you to reconfigure the WebCenter Sites and Satellite Server nodes to use the same cache, and to purge the cache before its use, as recommended by best practices.

28.2 Configuring Your System for inCache Page Caching

The configuration process consists of a set of required steps and optional optimization steps, shown below. For example, you will enable multiple WebCenter Sites nodes to

communicate with each other, and set storage properties for each local cache on the WebCenter Sites and remote Satellite Servers.

To configure your system for inCache page caching

1. Ensure that inCache for page caching is enabled. Look for the following VM argument: `-Dcs.useEhcache` and verify that it is either set to `true` or not set (in which case its value is assumed to be `true`).
2. Configure each WebCenter Sites and Satellite Server node to retain cache content on system restart. Pass the following VM argument:

```
-Dnet.sf.ehcache.enableShutdownHook=true
```

3. Optional optimization. Set the `diskStore` path for each WebCenter Sites and Satellite Server node. If multiple nodes are on the same machine (vertical cluster) ensure that each node has a unique `diskStore` path. In the `cs-cache.xml` and `ss-cache.xml` files, set the property `diskStore path="<path to disk store>". Both xml files are stored on each WebCenter Sites node: cs-cache.xml for WebCenter Sites and ss-cache.xml for co-resident Satellite Server. Only the ss-cache.xml file is stored on remote Satellite Server. The files are located under the WEB-INF/classes folder.`

For more information about the `diskStore` property, refer to the documentation on the Ehcache web site (at the time of this writing, the URL is <http://ehcache.org/documentation/configuration.html>).

Note: inCache exists on WebCenter Sites, remote Satellite Servers, and co-resident Satellite Server. Architecturally and functionally inCache is identical on co-resident and remote Satellite Servers. However, the following recommendations are applied to co-resident Satellite Server:

1. Disk persistence is turned off.
 2. The size of each cache is kept smaller than the size of WebCenter Sites' cache. This is to prevent overloading memory. We recommend keeping the sizes small except under special circumstances, such as the requirement for full double-buffered caching on co-resident Satellite Server.
-

4. Required for multiple nodes. Configure automatic node detection for WebCenter Sites clusters (Satellite Server nodes cannot be clustered). As an example, this step uses three WebCenter Sites nodes named CS1, CS2, and CS3.
 - a. On each WebCenter Sites node ensure that `cs-cache.xml` and `ss-cache.xml` (in `WEB-INF/classes`) specify a `cacheManagerPeerListenerFactory`, which will be used to create a `CacheManagerPeerProvider`. The provider detects other nodes in the cluster. Configure automatic detection as shown in the following example:

```
<cacheManagerPeerProviderFactory
  class="net.sf.ehcache.distribution.RMCacheManagerPeer
  ProviderFactory"
  properties="peerDiscovery = automatic,
  multicastGroupAddress = 230.0.0.1,
  multicastGroupPort = 4444, timeToLive = 32"/>
```

Note: The `multicastGroupPort` property must be set identically across cluster members. For example, if CS1 specifies `multicastGroupPort=4444` in its `cs-cache.xml` file, then CS2 and CS3 must specify the same setting in their files.

The `timeToLive` property specifies the number of hops allowed to a packet, which determines how far the packet propagates. Valid values range between 0 and 255 with the following restrictions:

- 0 - same host
 - 1 - same subnet
 - 32 - same site
 - 64 - same region
 - 128 - same continent
 - 255 - unrestricted
-

- b. On all Satellite Servers, co-resident and remote, do one of the following:
- Set the `multicastGroupPort` property in each `ss-cache.xml` file to a unique value.
 - To support cache replication, set the `multicastGroupPort` property in each `ss-cache.xml` file to the same value.

Note: The value(s) that you set in the `ss-cache.xml` files must be different from the value for WebCenter Sites' `multicastGroupPort` property in the `cs-cache.xml` files.

When inCache is configured, the system will start using the configurations specified in `cs-cache.xml` and `ss-cache.xml`. Caches are initialized upon the first call to any page in WebCenter Sites, cached or not.

5. Optional optimization. Configure the local caches as necessary, using [Table 28–1, "Cache Configuration Properties"](#) as a property reference. The local cache for each WebCenter Sites and remote Satellite Server node is partitioned. Each part is defined in its own `<cache>` tag in the `cs-cache.xml` and `ss-cache.xml` files. The parts are named as follows:
- `pageByQry`: This is the cache for the page data itself, keyed by the query url.
 - `dependencyRepository`: This is the cache of dependencies on which the pages are built. When pages are added to the `pageByQry` cache, entries are automatically created in the `dependencyRepository` cache. Each entry in this cache is an asset id or `unknowndep` or `unknowndep-<type>`. Therefore, this cache contains at most the number of assets in the system and a handful of items for different variations of `unknowndeps`.

When a page with no dependencies is cached, that page logs a dependency on `_NODEP_` (a single item in the `dependencyRepository` cache used to identify pages with no dependencies). The `_NODEP_` item remains associated with the page until that page either expires or is manually flushed. On remote Satellite Servers, you can disable caching of pages without dependencies by setting the JVM option:

- Dignore_nodep_pages to true (per Satellite Server).
 - notifier: This cache is used by active WebCenter Sites cluster members to notify other WebCenter Sites cluster members of changes to content.
6. Restart all configured WebCenter Sites and Satellite Server nodes.
 7. Verify that all active cluster members are in the caching network and recognize each other. Use the Cluster Info diagnostic tool, which lists all WebCenter Sites members that have a notifier cache. To invoke Cluster Info:
 - a. Bootstrap inCache by rendering a cacheable page.
 - b. Log in to the WebCenter Sites Admin interface as a general administrator (fwadmin/xceladmin, by default).
 - c. Verify that WebCenter Sites cluster members and co-resident Satellite Servers are communicating with each other. Open the **Admin** tab and go to **System Tools** then expand **Cache Management**, then expand **Sites Cache**, and double-click **Cluster Info**.

Figure 28–1 Cluster Info Form

Cluster Info

Number of remote cluster peer = 1

Name	Url	Port	Guid
notifier	192.0.2.201	54014	oracle-desk/192.0.2.201–3a43be2d:136c6d640a2

Various types of page caching statistics are available in the Cache Configuration tool. For more information, see [Chapter 30, "System Tools."](#)

8. If you wish to stripe disks, enable page regeneration during RealTime publishing, or enable page propagation, see [Section 28.3, "Tuning Options."](#)

Table 28–1 Cache Configuration Properties

Property	Required?	Description
name	Y	Specifies the name of the cache. Legal Values: <ul style="list-style-type: none"> ▪ pageByQry ▪ dependencyRepository ▪ notifier For descriptions, see step 5.
diskPersistent	N	Specifies whether to persist data on disk between restarts of the JVM. Default / recommended value: true Note: The diskPersistent setting must be consistent across the PageByQry, dependencyRepository, and notifier caches to prevent potential conflicts in generation count.

Table 28–1 (Cont.) Cache Configuration Properties

Property	Required?	Description
maxElementsInMemory	Y	Specifies the maximum number of objects to store in memory. Default value: 200000
maxElementsOnDisk	Y	Specifies the maximum number of objects to store on disk. Disks can be striped. For information, see Section 28.3.1, "Striping the Disk Cache." Default value: 1000000
eternal	Y	Specifies whether cache will be cleared by WebCenter Sites (it is never cleared by inCache). Default value: true Do not change the value of this property.
overflowToDisk	Y	Specifies whether the memory cache is allowed to overflow to disk-based cache. Default value: true Do not change the value of this property.
diskSpoolBufferSizeMB	N	Specifies the size of the disk buffer, in megabytes. If you expect many disk I.Os (that is, disk cache is much larger than memory cache) set the buffer to 20MB. Default value: 5
memoryStoreEvictionPolicy	N	Specifies how to remove items from cache. The recommended and default value is LFU (Least Frequently Used). LRU (Least Recently Used) is also a legal value. Default / recommended value: LFU
clearOnFlush	N	Specifies whether to clear memory as it is flushed to disk. Default value: false Do not change the value of this property.

28.3 Tuning Options

- [Section 28.3.1, "Striping the Disk Cache"](#)
- [Section 28.3.2, "Configuring for Page Regeneration During RealTime Publishing"](#)
- [Section 28.3.3, "Setting Up Page Propagation"](#)
- [Section 28.3.4, "Configuring for Pagelet Regeneration in Background"](#)

28.3.1 Striping the Disk Cache

The inCache framework supports striping of the pageByQry cache to reduce the contention that occurs when a large portion of the cache must be written to disk.

Complete the following steps on WebCenter Sites nodes and remote Satellite Servers in the inCache framework:

To stripe the pageByQry cache

1. To enable striping, add the following VM argument:

```
-DnumOfDiskStores=X
```

where *X* is the number of stripes. Set the number of stripes to the number of unique spindles that are available to stripe over (for instance if you have 5 drives, then set *X* to 5).

Note: Drives used for striping the disk-based cache should not be used for any other purpose.

The size of each DiskStore is the size specified by the property `maxElementsOnDisk` in the xml configuration file (see [Table 28-1](#)). For example, if you are using 5 stripes and `maxElementsOnDisk` is set to 100000 items, a total of 500000 items can be stored.

2. Create a symbolic link or else mount the drive physically in the correct location so that the stripes are properly distributed.

For each defined cache, the system creates a directory under the `diskStore` path (configured in step 3 in [Section 28.3.3, "Setting Up Page Propagation"](#)). Under each directory it also creates a group of numbered directories, starting at 0. Each directory points to a different drive.

For example, items would be stored as follows in a disk-based cache on WebCenter Sites for `-DnumOfDiskStores = 5`:

```
<custom_path>/cs-cache:
  Directory: 0
  Directory: 1
  Directory: 2
  Directory: 3
  Directory: 4
  File: dependencyRepository.data
  File: dependencyRepository.index
  File: notifier.data
  File: notifier.index
<custom_path>/cs-cache/0:
  File: pageByQry.data
  File: pageByQry.index
<custom_path>/cs-cache/1:
  File: pageByQry.data
  File: pageByQry.index
<custom_path>/cs-cache/2:
  File: pageByQry.data
  File: pageByQry.index
<custom_path>/cs-cache/3:
  File: pageByQry.data
  File: pageByQry.index
<custom_path>/cs-cache/4:
  File: pageByQry.data
  File: pageByQry.index
```

In this example, the root cache contains dependency and notifier caches. Spread among the directories 0–4 are the stripes of the pageByQry cache. The directories 0–4 in this example were placed on five separate drives by the use of symbolic links.

28.3.2 Configuring for Page Regeneration During RealTime Publishing

Pages are regenerated only when they are requested.

To configure page regeneration during RealTime publishing

1. By default, the PageCacheUpdater is set by the WebCenter Sites installer to use ParallelCacheRegenerator:

Open the AdvPub.xml file in the WEB-INF/classes folder on the delivery WebCenter Sites system and verify that the PageCacheUpdater section has the lines shown below:

Note: Do not change any of the values in the PageCacheUpdater section except for:

- numThreadsPerServer, to specify the number of simultaneous threads for crawling
 - regenServers, to point to the WebCenter Sites nodes by <address> and <port of server where pages will be regenerated>
-

```
<bean id="PageCacheUpdater"
class="com.fatwire.realtime.regen.ParallelRegeneratorEh" singleton="false">
  <property name="id" value="CacheFlusher" />
  <property name="numThreadsPerServer" value="3" />
  <property name="regenServers">
    <list>
      <value>http://address:port of server where pages will be
regenerated/servlet/ContentServer</value>
    </list>
  </property>
</bean>
```

2. Restart the delivery system.
3. RealTime publish to the delivery WebCenter Sites to create the FW_RegenCriteria table on that system.
4. Open the FW_RegenCriteria table, using Sites Explorer. Specify the pages to be regenerated and the depth of links to be followed. The ft_ss parameter can be included in the URLs to specify whether page requests will be handled by WebCenter Sites directly or by remote Satellite Server.

Note: If a URL assembler is used, specify the internal URLs of the pages to crawl. The regenerator does not recognize URL-assembler URLs.

Example:

```
pagename=SiteName/HomePage&ft_ss=true
Level=1
```

WebCenter Sites will regenerate HomePage and the pages that are linked from HomePage. Given that ft_ss=true, the requests are treated as if they are generated from Satellite Server.

5. RealTime publish to the delivery WebCenter Sites.

During the publishing session, the delivery system crawls all of the pages specified in the `FW_RegenCriteria` table, but regenerates only pages for which component assets were invalidated (it also generates the uncached pages).

For example:

- a. Updated assets are published to the delivery system.
- b. The delivery system invalidates the existing dependency information for the re-published assets by marking their asset identifiers as invalid in the `dependencyRepository` cache and incrementing the dependency generation counter. Because the invalidated assets are no longer available to pages that reference the assets, the pages are invalidated.
- c. The page regenerator crawls pages with the URLs that are specified in the `FW_RegenCriteria` table and regenerates the invalidated pages. It also generates the uncached pages.

28.3.3 Setting Up Page Propagation

Page propagation enables all nodes in a WebCenter Sites cluster (including the Satellite Server nodes) to host the same pages without each node having to regenerate the pages.

When configured to use `inCache`, each WebCenter Sites has a separate JVM and thus maintains a separate, local cache. If one WebCenter Sites generates and caches a new page, none of the other WebCenter Sites have the newly cached page, nor are they informed of that page. Upon receiving a request for the same page, each node must generate the page by referring to the database and shared file system, thus putting extra load on both components. Page propagation prevents different nodes from regenerating the same page by propagating the page across the cluster.

Page propagation is triggered when pages are loaded into a node's local cache. It works as illustrated in the scenario below, starting with basic `inCache` functionality.

1. `inCache` Page Caching for Newly Generated Pages:

- a. Node A, a WebCenter Sites node, receives a request for a new page.
- b. Node A generates the requested page.
- c. Node A caches the new page with complete dependency information (identifiers of component assets) into its local page and dependency caches.
- d. When page propagation is enabled, step 3 follows.

2. `inCache` Page Caching for Regenerated Pages:

- a. Node B, also a WebCenter Sites node, receives a request for a page that has been invalidated (a component asset was modified. The asset is marked as invalid in the node's local dependency cache and in the dependency caches of all other nodes containing the asset.)
- b. Node B regenerates the requested page.
- c. Node B caches the regenerated page and updates its dependency (in step 2a) by incrementing the generation counter.
- d. When page propagation is enabled, step 3 follows.

3. `inCache` Page Caching with Page Propagation:

When the node caches the (re)generated page, it also propagates the page to all other nodes. Propagated information consists of:

- The page's complete dependency information. For example, if the page has x dependencies (asset identifiers), all x dependencies are propagated from the local dependency cache to the dependency caches of other WebCenter Sites nodes.
- The page itself. The page is propagated from the local page cache to the page caches of other WebCenter Sites nodes.

If the page already exists on a receiving node, the node ignores the propagation.

The following list summarizes page propagation events and conditions:

- When a page is cached on a WebCenter Sites node, its complete page information (described in step 3, above) is propagated to all other WebCenter Sites nodes.
- When a page is cached on a remote Satellite Server node, its complete page information (described in step 3, above) is propagated to other remote Satellite Servers over Java Remote Method Invocation (RMI).
- When a page is propagated:
 - The page's last updated time stamp is preserved on all WebCenter Sites and Satellite Server nodes.
 - The generation count on a given node remains independent of generation count on all other nodes. For example, Node A has a generation count of 10, but other nodes use a generation count of 10 for a different type of dependency. Even though the same object is propagated across WebCenter Sites or Satellite Servers, it may be assigned different generation counts by each WebCenter Sites or Satellite Server node. The object and its last updated/modified time remain the same across nodes.
 - The propagation is ignored by nodes on which the page already exists.
- If any of a cached page's dependencies fail to propagate to a node where the page does not exist, the page is not cached on that node until it is requested and generated on that node.
- Because blobs are still stored in the Satellite Server caches, they are replicated across Satellite Servers.

This section contains the following topics:

- [Section 28.3.3.1, "Enabling Page Propagation"](#)
- [Section 28.3.3.2, "Setting Up Page Propagation on Restart"](#)

28.3.3.1 Enabling Page Propagation

Start with a system that is configured to use inCache.

To enable page propagation

1. On all WebCenter Sites nodes:
 - a. Set `propagatecache=true` in the `futuretense.ini` property file.
 - b. In `cs-cache.xml`, verify that `multicastGroupPort` is the same for all WebCenter Sites nodes in the cluster to ensure they can communicate with each other. (The file is located in `WEB-INF\classes`.)
2. On all remote Satellite Servers:
 - a. Set `propagatecache=true` in the `satellite.properties` file.

- b. In `ss-cache.xml`, ensure `multicastGroupPort` is identical for all Satellite Servers intended for cache propagation, but different from `multicastGroupPort` for the WebCenter Sites nodes.
3. On all nodes, initialize page propagation by rendering any page and verify that the system responds as described below:
 - On WebCenter Sites nodes, caching of the page triggers its propagation from the local cache to the caches of other WebCenter Sites nodes (as described in step 3 in [Section 28.3.3, "Setting Up Page Propagation"](#)) while preserving the page's last update/modified time stamp across all WebCenter Sites. Propagation is ignored by WebCenter Sites nodes on which the page already exists.
 - The response on Satellite Servers proceeds as on WebCenter Sites nodes, but over Java RMI.

28.3.3.2 Setting Up Page Propagation on Restart

If a node enabled for page propagation is restarted, its local caches must be re-initialized in order for the node to recognize page propagations. To re-initialize a local cache, render a cacheable page on the node; caching triggers the node to propagate the page to other nodes and recognize pages that are propagated by other nodes. When a node is restarted, the propagations it missed during its period of inactivity are not reproduced on the node, even though its local caches are re-initialized.

28.3.4 Configuring for Pagelet Regeneration in Background

Remote Satellite Server can be configured to serve invalidated pagelets while they are being regenerated by a background process. To enable serving of invalidated pagelets, add `serveStale=true` to remote Satellite Server's `satellite.properties` file. If a pagelet is then invalidated, it will be regenerated in one of the following ways:

- If a browser requests the pagelet *within* the next 30 minutes, remote Satellite Server starts a background process that sends a request to WebCenter Sites to regenerate the page. While the background process is running, the browser is served the invalidated pagelet from remote Satellite Server's cache. All subsequent requests will be served the invalidated pagelet until it is regenerated by the background process.
- If a browser requests the pagelet *after* 30 minutes, remote Satellite Server uses its normal process for regenerating pagelets; that is, it sends a request to WebCenter Sites to regenerate the pagelet. The requesting browser must wait until the page is regenerated.

To obtain information about the background process, set the `com.fatwire.logging.cs.cache.ehcache logger` to `DEBUG` in remote Satellite Server's `commons-logging.properties` file. `DEBUG` produces the following message at the start of the background process:

```
"Data for <cache key> is found to have been invalidated. A new request has started processing new data in the background."
```

At the end of the process, the message reads:

```
"Background process for request <cache key> has completed successfully, data in cache is updated."
```

inCache for Asset Caching

The WebCenter Sites inCache framework supports asset caching by default. Asset caching can be customized.

This chapter contains the following sections:

- [Section 29.1, "Introduction to Asset Caching"](#)
- [Section 29.2, "AssetCache Container"](#)
- [Section 29.3, "Asset Caching Operations"](#)
- [Section 29.4, "Types of Asset Caching"](#)
- [Section 29.5, "Customizing Asset Caching"](#)
- [Section 29.6, "Disabling Asset Caching"](#)
- [Section 29.7, "Cache Management Tool"](#)

29.1 Introduction to Asset Caching

Asset caching is automatically enabled on WebCenter Sites systems during installation and upgrade. Asset caching, a memory-based system, is built on the inCache framework to protect WebCenter Sites' performance by taking up load that would otherwise affect the database. Asset caching uses its own container – the `AssetCache` container – which functions by interacting with other inCache components, in particular the `dependencyRepository` and `notifier` caches.

To follow the discussion about asset caching, we recommend first reading [Chapter 27, "inCache Framework"](#) for a basic understanding of the inCache framework, especially how its cache containers – `pageByQuery`, `dependencyRepository`, and `notifier` – function in relation to each other in page caching. Note that while the `pageByQuery` cache is not used in asset caching, it serves to illustrate how a cache container works. Its counterpart `AssetCache` container works in a similar manner. Note, also, that asset caching and inCache page caching work independently of each other. Each type of caching can be enabled or disabled independently of the other.

The rest of this chapter describes the `AssetCache` container, discusses its functionality in terms of asset loading and invalidations, compares asset caching with page caching, and provides instructions for customizing caching of assets by asset type.

29.2 AssetCache Container

In WebCenter Sites, programmatic usage of assets consists of loading and reading their attributes. Given that assets are loaded by templates, which are stored in WebCenter Sites, `AssetCache` is used only on WebCenter Sites nodes.

- `AssetCache` is a component of the `inCache` framework. Similar to the `pageByQry` cache, `AssetCache` is independently available in the `inCache` framework as its own container on each WebCenter Sites node.
- The `AssetCache` container is enabled by default in each WebCenter Sites `cs-cache.xml` file, where it is named in the `<cache>` element. The `AssetCache` container functions by caching assets of all types and interacting with the `inCache` components listed below. Additional `AssetCache` containers for assets of specific types can be configured. For more information, see [Section 29.4, "Types of Asset Caching"](#) and [Section 29.5, "Customizing Asset Caching."](#)

- Interaction with the `dependencyRepository` cache

When an asset is loaded into `AssetCache`, `inCache` logs a dependency on that asset by creating entries in the `dependencyRepository` cache, as shown in [Table 29–1](#). Modifying or saving the asset removes its dependency information from the `dependencyRepository` cache and therefore invalidates the asset's entry in the `AssetCache`. In general, any save or delete operation on a cached asset invalidates that asset.

Note: When asset caching is enabled, assets must not be saved or deleted directly in database tables via Sites Explorer or the CatalogManager API. Save and delete operations made in this manner do not invalidate the asset cache.

- Interaction with the `notifier` cache

Invalidations that happen on any one node are reflected on other nodes by means of the `notifier` cache. This cache broadcasts information as to which asset was changed. If a receiving node contains the asset in its `AssetCache`, the node updates its own `dependencyRepository` cache accordingly.

Asset caching works independently of page caching, but shares various cache containers, as shown in [Table 29–1](#).

Table 29–1 Cache Containers Used by Asset and Page Caching

Asset Caching Function	Page Caching Function	AssetCache Container	pageByQry Container	dependency Repository Container	notifier Container
Off	Off	Not used	Not used	Not used	Not used
Off	On	Not used	Used in Page Caching	Used in Page Caching	Used in Page Caching
On	Off	Used in Asset Caching	Not used	Used in Asset Caching	Used in Asset Caching
On	On	Using in Asset Caching	Used in Page Caching	Used in Asset Caching and Page Caching	Used in Asset Caching and Page Caching

29.3 Asset Caching Operations

This section contains the following topics:

- [Section 29.3.1, "Asset Loading"](#)
- [Section 29.3.2, "Comparing Asset Caching with inCache Page Caching"](#)
- [Section 29.3.3, "Flushing AssetCache"](#)

29.3.1 Asset Loading

An asset's data can be read in several ways, by using APIs or `asset:load` tags.

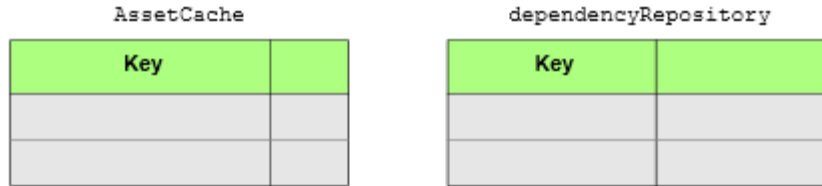
- The following APIs load assets into `AssetCache` in `readonly-complete` mode:
 - `AssetDataManager.read(AssetId)`
 - REST API GET for assets
- The following tag loads assets into `AssetCache` in different modes:
 - `asset:load(objectid="cid1" option="editable")`
 - `asset:load(objectid="cid1" option="readonly")`
 - `asset:load(objectid="cid1" option="readonly-complete")`

The option parameter determines whether an asset is cached, and how a cached asset's key is named in the `AssetCache` and in the `dependencyRepository`. Caching scenarios are summarized in [Figure 29-1](#). Note that in scenario D, you will find multiple entries in the `AssetCache` container for the same asset. This is due to the fact that an asset can be loaded in two modes: `readonly` and `read-only complete`.

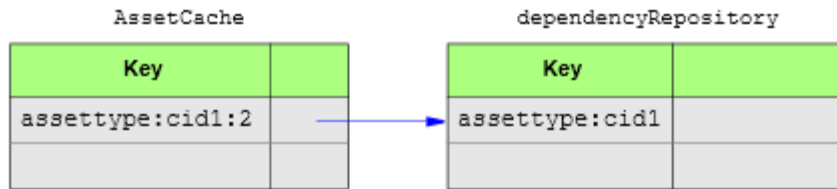
The following modes determine whether an asset is cached and how its key is named in the `AssetCache` and `dependencyRepository`. The mode is defined in the APIs and tags listed in [Section 29.3.1, "Asset Loading."](#)

Figure 29–1 Asset Caching Scenarios

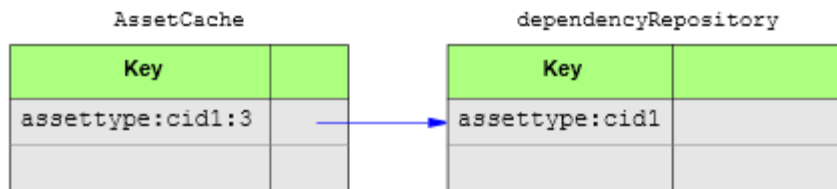
A. editable mode. When an asset is loaded in editable mode it is not cached.



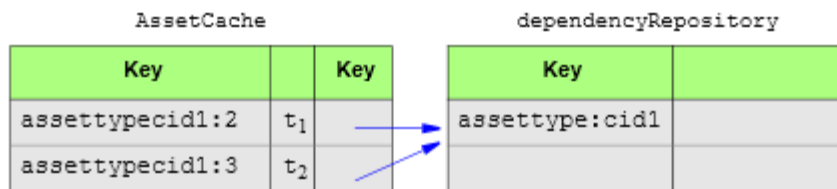
B. readonly mode. The asset is cached pointing to its dependency on itself. The 2 in assettype:cid1:2 specifies the readonly mode.



C. readonly_complete mode. The asset is cached pointing to its dependency on itself. The 3 in assettype:cid1:3 specifies the readonly_complete mode.



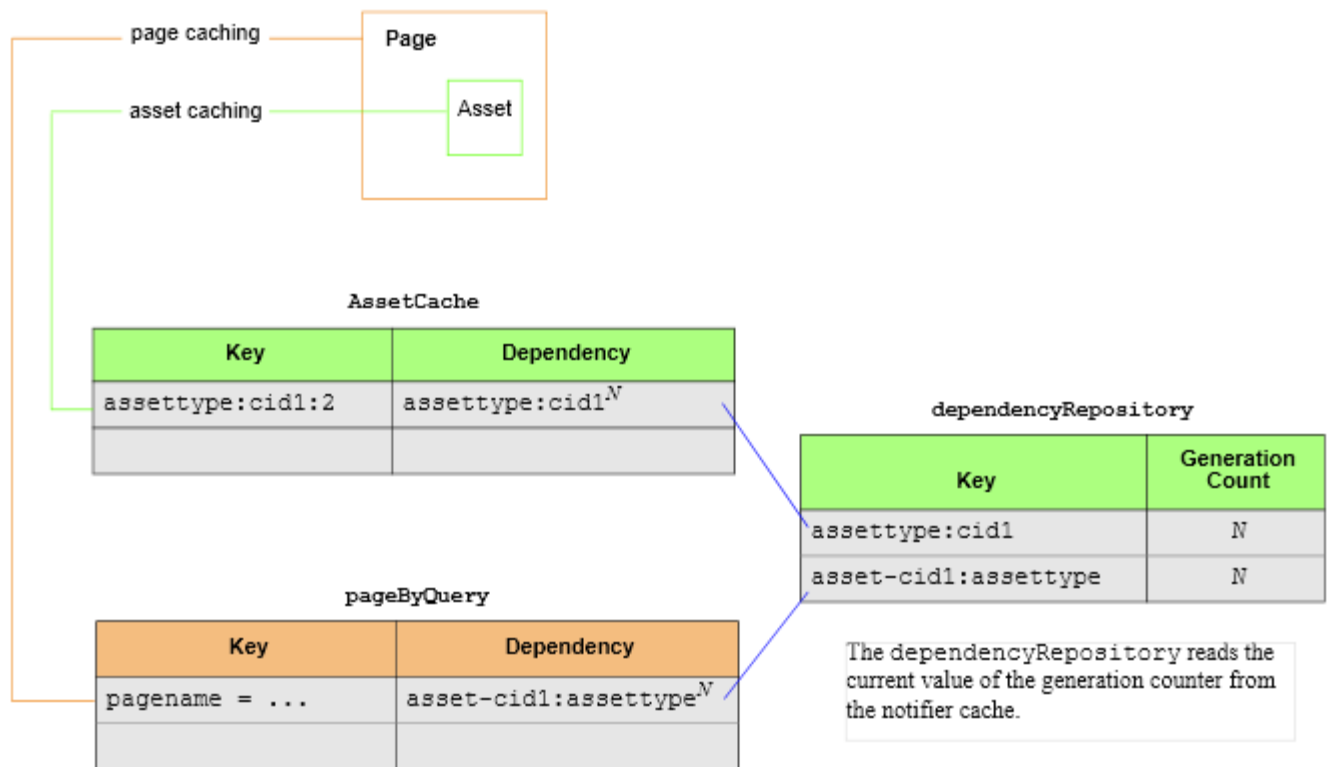
D. readonly mode at time t₁ and readonly_complete mode at time t₂. The asset is cached twice, pointing to its dependency on itself. The 2 in assettype:cid1:2 specifies readonly mode; the 3 in assettype:cid1:3 specifies the readonly_complete mode.



29.3.2 Comparing Asset Caching with inCache Page Caching

Keys are named differently for cached assets and pages, and for their dependencies, as shown in [Figure 29–2](#). Understanding the differences helps to interpret caching information displayed in the Cache Management tool (available under the System Tools node, in the Admin tab).

Figure 29–2 Dependence Naming Conventions in Asset Caching and Page Caching



For example, Figure 29–2 shows that in the AssetCache container, the key of a cached asset is named `assettype:cid1:2`, where the last integer represents the mode in which the asset was read (2 for readonly mode). If the same asset is also loaded in readonly-complete mode, its key would be `assettype:cid1:3`. When an asset is called in multiple modes, you will find multiple entries in the AssetCache container for that asset (see Figure 29–1).

The key of a cached page begins with `pagename`. Mode does not apply to pages; if a page is cached, it is cached only once.

For more information about asset loading modes, see Section 29.3.1, "Asset Loading." Information about the Cache Management tool is available in Chapter 30, "System Tools."

29.3.3 Flushing AssetCache

Assets can be flushed from the AssetCache if they are saved via the following APIs and tags:

- API
 - REST API PUT
 - AssetDataManagerWrite
- JSP Tags
 - `asset:save`
 - `asset:saveall`
 - `asset:delete`

- asset:deleteassettype
- asset:void
- asset:rollback
- asset:deleterevision
- asset:undocheckout
- insite:edit
- XML Tags
 - INSITE.EDIT
 - ASSET.VOID
 - ASSET.SAVE
 - ASSET.SAVEALL
 - ASSET.DELETE

29.4 Types of Asset Caching

When asset caching is enabled on a WebCenter Sites node, the `cs-cache.xml` file on that node contains at least one of the following `<cache>` elements:

- The default `<cache>` element, which specifies that assets of all types are cached in a generic cache container named `AssetCache`:

```
<cache name="AssetCache"
  diskPersistent="true" maxElementsInMemory="100"
  maxElementsOnDisk="100" eternal="true" overflowToDisk="true"
  diskSpoolBufferSizeMB="20" memoryStoreEvictionPolicy="LFU"
  clearOnFlush="false" />
```

The generic `AssetCache` container is automatically configured on all WebCenter Sites nodes during the WebCenter Sites installation process.

- A custom `<cache>` element, which specifies that assets of a given type are cached in their own container named `AssetCacheAsset_type_name`:

```
<cache name="AssetCacheAsset_type_name"
  diskPersistent="true" maxElementsInMemory="100"
  maxElementsOnDisk="100" eternal="true" overflowToDisk="true"
  diskSpoolBufferSizeMB="20" memoryStoreEvictionPolicy="LFU"
  clearOnFlush="false" />
```

All custom, type-specific `AssetCache` containers must be manually configured in the `cs-cache.xml` files. How many type-specific containers to create depends on which types of assets must be cached in their own containers and which conditions must be applied to caching in each container.

Note that the generic `AssetCache` container can either co-exist or be replaced with one or many type-specific asset caches. The contents of all asset caches are mutually exclusive. That is, the generic `AssetCache` container excludes assets of the type that are cached in a type-specific container, and a type-specific container excludes assets of all other types.

Also note that when asset caching is enabled, the Cache Management tool (on the **System Tools** node of the **Admin** tab) displays dialogs showing the contents of

AssetCache containers, corresponding entries in the dependencyRepository, and other types of information. Searches are also supported.

For more information about configuring asset caching, see [Section 29.5, "Customizing Asset Caching."](#) For more information about the Cache Management tool, see [Chapter 30, "System Tools."](#)

29.5 Customizing Asset Caching

The steps below show you how to configure type-specific asset caches.

To configure asset caching

1. When default asset caching is enabled on a WebCenter Sites node, you will see the following entry in the `cs-cache.xml` file on WebCenter Sites nodes, which specifies that assets of all types will be cached in a generic container named AssetCache (parameters, other than name, are defined in [Table 28-1](#)):

```
<cache name="AssetCache"
  diskPersistent="true" maxElementsInMemory="100"
  maxElementsOnDisk="100" eternal="true"
  overflowToDisk="true" diskSpoolBufferSizeMB="20"
  memoryStoreEvictionPolicy="LFU" clearOnFlush="false" />
```

Note: Given that the generic AssetCache container can either co-exist or be replaced with one or many type-specific asset caches, you have the option to do any of the following:

- You can keep the generic AssetCache container and add as many type-specific containers as necessary. Assets of all types will be cached. Assets having their own type-specific container will be cached *exclusively* in their own container, which can be individually tuned by use of the parameters shown in [Table 28-1](#).
- You can replace the generic AssetCache container with one or more type-specific containers. Only assets of the named types will be cached.

How many AssetCache containers to create depends on which types of assets must be cached in their own containers and which tunings must be applied. Configuration information is available in the steps below.

2. To cache assets of a given type in their own container, create a `<cache>` element for that asset type and set `name="AssetCacheAsset_type_name"` (only one asset type is supported for each custom `<cache>` element). Configure the cache by using the parameters in [Table 28-1](#).

For example:

```
<cache name="AssetCacheContent_C"
  diskPersistent="true" maxElementsInMemory="100"
  maxElementsOnDisk="100" eternal="true"
  overflowToDisk="true" diskSpoolBufferSizeMB="20"
  memoryStoreEvictionPolicy="LFU" clearOnFlush="false" />
```

In this example, `name="AssetCacheContent_C"` creates a type-specific `AssetCache` container that holds only assets of type `Content_C`. If the generic `AssetCache` container is also configured, it will not cache assets of type `Content_C`.

29.6 Disabling Asset Caching

If your `inCache` framework is set up to support asset caching and you wish to disable asset caching, do the following:

On each WebCenter Sites node in the `inCache` framework, open the `cs-cache.xml` file (in the `WEB-INF/classes/` directory) and remove the `<cache>` elements whose name begins with `AssetCache`. You can remove selected `<cache>` elements or all `<cache>` elements. Each element represents an `AssetCache` container. For information about the types of `AssetCache` containers, see [Section 29.4, "Types of Asset Caching."](#)

29.7 Cache Management Tool

When asset caching is configured, the Cache Management tool displays dialogs showing the contents of `AssetCache` containers and corresponding entries in the `dependencyRepository`, among other types of information. Searches are also supported. For more information about the Cache Management tool, see [Chapter 30, "System Tools."](#)

The System Tools node on the Admin tab provides a range of diagnostic utilities that general administrators can use to troubleshoot directly from the WebCenter Sites administrator's interface. Features include configuring log4j loggers, accessing various types of system information, managing caches, searching the contents of the WebCenter Sites log, and testing the performance of the shared file system.

This chapter contains the following sections, each covering a sub-node of System Tools:

- [Section 30.1, "Configure log4j"](#)
- [Section 30.2, "System Information"](#)
- [Section 30.3, "Cache Management"](#)
- [Section 30.4, "Cache Management - CAS Cache"](#)
- [Section 30.5, "Cache Management - Resultset Cache"](#)
- [Section 30.6, "Cache Management - Sites Cache"](#)
- [Section 30.7, "Cache Management - Co-Res Cache"](#)
- [Section 30.8, "Log Viewer"](#)
- [Section 30.9, "File System Test"](#)
- [Section 30.10, "Slots"](#)
- [Section 30.11, "URL"](#)
- [Section 30.12, "Purge Voided Assets"](#)

30.1 Configure log4j

If WebCenter Sites is set up to use the Apache log4j logging system, the Configure log4j tool enables you to view current logger levels, change logger levels, and add new loggers directly from the administrator's interface. In addition, since changes to logger levels and added loggers remain in effect only until WebCenter Sites is restarted, you can retain your changes by copying a test version of the logger properties to the log4j.properties file.

This section contains the following topics:

- [Section 30.1.1, "Setting Up Apache log4j"](#)
- [Section 30.1.2, "Viewing Loggers"](#)
- [Section 30.1.3, "Changing Logger Levels"](#)

- [Section 30.1.4, "Adding New Loggers"](#)

30.1.1 Setting Up Apache log4j

If WebCenter Sites was not set up to use log4j during the upgrade process, you can manually switch to log4j by configuring the `commons-logging.properties` and `log4j.properties` files in the `WEB-INF/classes` directory.

To set up log4j

1. In the `commons-logging.properties` file, update the following property to set log4j as WebCenter Sites' logging system:

```
org.apache.commons.logging.Log= org.apache.commons.logging.impl.Log4JLogger
```

2. Open the `log4j.properties` file and do the following:

- a. Add the `FWDefaultAppender` file appender properties:

```
log4j.rootLogger=INFO, FWDefaultAppender
log4j.appender.FWDefaultAppender.File=<path to
futuretense.txt>/futuretense.txt
```

Note: The `futuretense.txt` file is typically located in the WebCenter Sites installation directory.

```
log4j.appender.FWDefaultAppender.Append=true
log4j.appender.FWDefaultAppender.BufferSize=256
log4j.appender.FWDefaultAppender.MaxFileSize=10MB
log4j.appender.FWDefaultAppender.encoding=UTF-8
log4j.appender.FWDefaultAppender.layout.ConversionPattern=[%d] [%c{4}] %m%n
log4j.appender.FWDefaultAppender=org.apache.log4j.RollingFileAppender
log4j.appender.FWDefaultAppender.MaxBackupIndex=15
log4j.appender.FWDefaultAppender.layout=org.apache.log4j.PatternLayout
log4j.appender.FWDefaultAppender.bufferedIO=false
```

- b. Copy the loggers in the `commons-logging.properties` file to the `log4j.properties` file. Prepend `log4j.logger.` to each logger name. For example: `com.fatwire.logging.cs.auth` becomes `log4j.logger.com.fatwire.logging.cs.auth`.

For information about the file appender properties and other log4j-specific properties, refer to the Apache log4j documentation. For information about WebCenter Sites loggers, see the `commons.logging-properties` documentation in the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

30.1.2 Viewing Loggers

You can view a list of WebCenter Sites loggers and their verbosity levels. The list includes the following:

- Loggers that are registered in the `log4j.properties` file, in the `WEB-INF/classes` directory. The file is read only on system startup. If, during runtime, you make changes to the `log4j.properties` file and save them, your changes will not be displayed in the log4j Configuration form. They will be applied and displayed when WebCenter Sites is restarted.
- Loggers that are detected during runtime. The loggers are detected when:

- System features or add-on components are accessed for the first time.
- The loggers are added to the system manually by developers using the **Add Logger** feature, as shown in [Section 30.1.4, "Adding New Loggers."](#)

The collapsed view of the list shows the root logger and parent loggers, all of which are explicitly assigned verbosity levels. If you require a more granular view, you can expand the list to also show loggers that inherit levels from the root logger or parent loggers.

To view loggers

1. On the **Admin** tab, expand **System Tools** and double-click **Configure log4j**.

The list of loggers with explicitly assigned levels is displayed.

Figure 30–1 log4j Configuration Form

log4j Configuration

Configured Loggers

Num	Level	Logger	Set New Level
1	ERROR	root	TRACE DEBUG INFO WARN ERROR FATAL OFF
2	INFO	com.fatwire.cs.core.db.DBTransaction	TRACE DEBUG INFO WARN ERROR FATAL OFF
3	INFO	com.fatwire.cs.core.db.DBTransaction.callstack	TRACE DEBUG INFO WARN ERROR FATAL OFF
4	INFO	com.fatwire.logging.cs	TRACE DEBUG INFO WARN ERROR FATAL OFF
5	INFO	com.fatwire.logging.cs.auth	TRACE DEBUG INFO WARN ERROR FATAL OFF
6	INFO	com.fatwire.logging.cs.blobserver	TRACE DEBUG INFO WARN ERROR FATAL OFF
7	INFO	com.fatwire.logging.cs.cache.page	TRACE DEBUG INFO WARN ERROR FATAL OFF
8	INFO	com.fatwire.logging.cs.cache.resultset	TRACE DEBUG INFO WARN ERROR FATAL OFF
9	INFO	com.fatwire.logging.cs.core.http.HttpAccess	TRACE DEBUG INFO WARN ERROR FATAL OFF
10	INFO	com.fatwire.logging.cs.core.uri.assembler	TRACE DEBUG INFO WARN ERROR FATAL OFF
11	INFO	com.fatwire.logging.cs.core.uri.definition	TRACE DEBUG INFO WARN ERROR FATAL OFF
12	INFO	com.fatwire.logging.cs.db	TRACE DEBUG INFO WARN ERROR FATAL OFF

2. To expand the list to include loggers that inherit levels, click **Show All Known Loggers** at the bottom of the form.

Loggers that inherit levels are listed below their parent loggers.

Figure 30–2 log4j Configuration Form

log4j Configuration

All Known Loggers

Num	Level	Logger	Set New Level
1	ERROR	root	TRACE DEBUG INFO WARN ERROR FATAL OFF
2	(Inherited: ERROR)	com.fatwire.assetapi	TRACE DEBUG INFO WARN ERROR FATAL OFF
3	(Inherited: ERROR)	com.fatwire.buffering	TRACE DEBUG INFO WARN ERROR FATAL OFF
4	(Inherited: ERROR)	com.fatwire.cas.web.NoSuchFlowExecutionExceptionResolverModified	TRACE DEBUG INFO WARN ERROR FATAL OFF
5	(Inherited: ERROR)	com.fatwire.cas.web.flow.SendPOPHeaderAction	TRACE DEBUG INFO WARN ERROR FATAL OFF
6	INFO	com.fatwire.cs.core.db.DBTransaction	TRACE DEBUG INFO WARN ERROR FATAL OFF
7	INFO	com.fatwire.cs.core.db.DBTransaction.callstack	TRACE DEBUG INFO WARN ERROR FATAL OFF
8	(Inherited: ERROR)	com.fatwire.cs.core.logging.context.filter	TRACE DEBUG INFO WARN ERROR FATAL OFF
9	(Inherited: ERROR)	com.fatwire.cs.systemtools.util	TRACE DEBUG INFO WARN ERROR FATAL OFF
10	(Inherited: ERROR)	com.fatwire.csdt	TRACE DEBUG INFO WARN ERROR FATAL OFF
11	INFO	com.fatwire.logging.cs	TRACE DEBUG INFO WARN ERROR FATAL OFF
12	(Inherited: INFO)	com.fatwire.logging.cs.assetframework	TRACE DEBUG INFO WARN ERROR FATAL OFF
13	INFO	com.fatwire.logging.cs.auth	TRACE DEBUG INFO WARN ERROR FATAL OFF
14	INFO	com.fatwire.logging.cs.blobserver	TRACE DEBUG INFO WARN ERROR FATAL OFF
15	(Inherited: INFO)	com.fatwire.logging.cs.cache.ehcache	TRACE DEBUG INFO WARN ERROR FATAL OFF
16	INFO	com.fatwire.logging.cs.cache.page	TRACE DEBUG INFO WARN ERROR FATAL OFF

- To view only loggers with explicitly assigned levels once again, click **Show Configured Loggers** at the bottom of the form.

30.1.3 Changing Logger Levels

You can temporarily change logger levels to define the type and number of statements that get written to the WebCenter Sites log specified in the `log4j.properties` file. Your changes are applied right away; as soon as the loggers are executed, you can track the impact of their new levels in the log. The new levels remain in effect only until WebCenter Sites is restarted, but if you wish to retain the changes, you can copy a text version of the logger properties to the `log4j.properties` file in the `WEB-INF/classes` directory on your application server.

This option is particularly useful when you want to temporarily override inherited logger levels. For example, if a parent logger is set to a high-verbosity level such as TRACE and you want to prevent one of its child loggers from inheriting that level, you can assign the child logger a different level.

The following logger levels are available, listed in order of decreasing verbosity:

TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF

OFF indicates that a logger does not write any statements to the log. For more information about logger levels, refer to the Apache log4j documentation.

To change a logger level

- On the **Admin** tab, expand **System Tools** and double-click **Configure log4j**.

The list of loggers with explicitly assigned levels is displayed. If you require a more granular view, you can click **Show All Known Loggers** at the bottom of the form to expand the list to include loggers that inherit levels from parent loggers.

2. Find the logger that you want to change and select a new logger level in the **Set New Level** column.

The new level is displayed in the Level column. As soon as the logger is executed, you can track the impact of the new level in the log (see [Section 30.8, "Log Viewer"](#)).

Figure 30–3 log4j Configuration Form

log4j Configuration

Configured Loggers

Num	Level	Logger	Set New Level
1	ERROR	root	TRACE DEBUG INFO WARN ERROR FATAL OFF
2	INFO	com.fatwire.cs.core.db.DBTransaction	TRACE DEBUG INFO WARN ERROR FATAL OFF
3	DEBUG	com.fatwire.cs.core.db.DBTransaction.callstack	TRACE DEBUG INFO WARN ERROR FATAL OFF
4	INFO	com.fatwire.logging.cs	TRACE DEBUG INFO WARN ERROR FATAL OFF
5	INFO	com.fatwire.logging.cs.auth	TRACE DEBUG INFO WARN ERROR FATAL OFF
6	INFO	com.fatwire.logging.cs.blobserver	TRACE DEBUG INFO WARN ERROR FATAL OFF
7	INFO	com.fatwire.logging.cs.cache.page	TRACE DEBUG INFO WARN ERROR FATAL OFF
8	INFO	com.fatwire.logging.cs.cache.resultset	TRACE DEBUG INFO WARN ERROR FATAL OFF
9	INFO	com.fatwire.logging.cs.core.http.HttpAccess	TRACE DEBUG INFO WARN ERROR FATAL OFF
10	INFO	com.fatwire.logging.cs.core.uri.assembler	TRACE DEBUG INFO WARN ERROR FATAL OFF
11	INFO	com.fatwire.logging.cs.core.uri.definition	TRACE DEBUG INFO WARN ERROR FATAL OFF
12	INFO	com.fatwire.logging.cs.db	TRACE DEBUG INFO WARN ERROR FATAL OFF

3. If you want the new logger level to remain in effect after WebCenter Sites is restarted, do the following:
 - a. Click **Show Current Logger Levels As .properties** at the bottom of the form.
The loggers with explicitly assigned levels are displayed as text.

Figure 30–4 Form Showing Loggers with Explicitly Assigned Levels

```

log4j.logger.com.fatwire.cs.core.db.DBTransaction=INFO
log4j.logger.com.fatwire.cs.core.db.DBTransaction.callstack=DEBUG
log4j.logger.com.fatwire.logging.cs=INFO
log4j.logger.com.fatwire.logging.cs.auth=INFO
log4j.logger.com.fatwire.logging.cs.blobserver=INFO
log4j.logger.com.fatwire.logging.cs.cache.page=INFO
log4j.logger.com.fatwire.logging.cs.cache.resultset=INFO
log4j.logger.com.fatwire.logging.cs.core.http.HttpAccess=INFO
log4j.logger.com.fatwire.logging.cs.core.uri.assembler=INFO
log4j.logger.com.fatwire.logging.cs.core.uri.definition=INFO
log4j.logger.com.fatwire.logging.cs.db=INFO
log4j.logger.com.fatwire.logging.cs.errno=INFO
log4j.logger.com.fatwire.logging.cs.event=INFO
log4j.logger.com.fatwire.logging.cs.export=INFO
log4j.logger.com.fatwire.logging.cs.file=INFO
log4j.logger.com.fatwire.logging.cs.filelock=INFO
log4j.logger.com.fatwire.logging.cs.firstsite.filter=INFO
log4j.logger.com.fatwire.logging.cs.framework=INFO
log4j.logger.com.fatwire.logging.cs.install=INFO
log4j.logger.com.fatwire.logging.cs.jsp=INFO
log4j.logger.com.fatwire.logging.cs.realtime=INFO
log4j.logger.com.fatwire.logging.cs.request=INFO
log4j.logger.com.fatwire.logging.cs.satellite=INFO
log4j.logger.com.fatwire.logging.cs.satellite.cache=INFO

```

- b. Copy the changed logger to the `log4j.properties` file in the `WEB-INF/classes` directory on your application server.
- c. Save the `log4j.properties` file.

30.1.4 Adding New Loggers

If you wish to add new loggers to the log4j framework in your WebCenter Sites system and you do not want them to inherit verbosity levels from the root logger or parent logger, you can add the loggers via the administrator's interface and assign each of them a verbosity level. (If you add your own custom loggers, be sure to deploy the supporting `class` and `jar` files.)

Assigned verbosity levels remain in effect only until WebCenter Sites is restarted, but if you wish to apply the levels permanently, you can copy a text version of the logger properties to the `log4j.properties` file in the `WEB-INF/classes` directory on your application server.

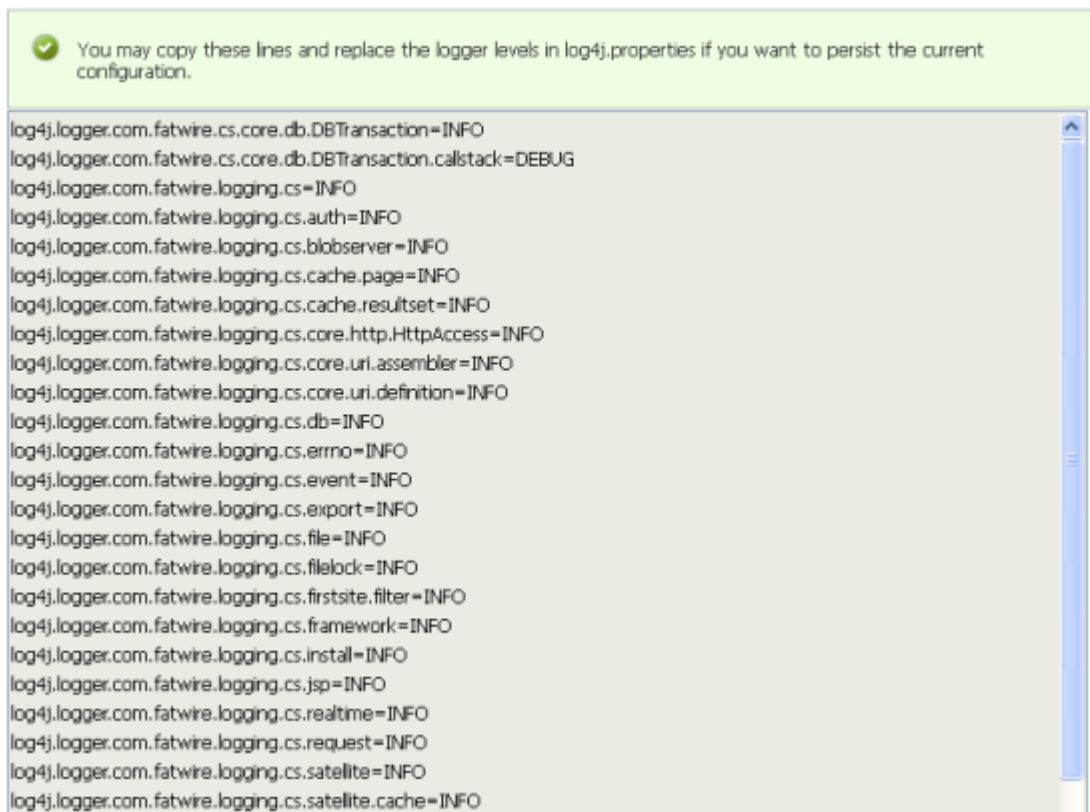
To add a new logger

1. On the **Admin** tab, expand **System Tools** and double-click **Configure log4j**.
The list of loggers with explicitly assigned levels is displayed.
2. In the field below the list of loggers, enter the name of the logger you want to add and select its verbosity level.
3. Click **Add Logger**.

The new logger is added to the list. As soon as the logger is executed, you can track the impact of the assigned level in the log (see [Section 30.8, "Log Viewer"](#)).

4. If you want the added logger to remain in effect after WebCenter Sites is restarted, do the following:
 - a. Click **Show Current Logger Levels As .properties** at the bottom of the form. The loggers with explicitly assigned levels are displayed as text.

Figure 30–5 Form Showing Loggers with Explicitly Assigned Levels



```

log4j.logger.com.fatwire.cs.core.db.DBTransaction=INFO
log4j.logger.com.fatwire.cs.core.db.DBTransaction.callstack=DEBUG
log4j.logger.com.fatwire.logging.cs=INFO
log4j.logger.com.fatwire.logging.cs.auth=INFO
log4j.logger.com.fatwire.logging.cs.blobserver=INFO
log4j.logger.com.fatwire.logging.cs.cache.page=INFO
log4j.logger.com.fatwire.logging.cs.cache.resultset=INFO
log4j.logger.com.fatwire.logging.cs.core.http.HttpAccess=INFO
log4j.logger.com.fatwire.logging.cs.core.uri.assembler=INFO
log4j.logger.com.fatwire.logging.cs.core.uri.definition=INFO
log4j.logger.com.fatwire.logging.cs.db=INFO
log4j.logger.com.fatwire.logging.cs.errno=INFO
log4j.logger.com.fatwire.logging.cs.event=INFO
log4j.logger.com.fatwire.logging.cs.export=INFO
log4j.logger.com.fatwire.logging.cs.file=INFO
log4j.logger.com.fatwire.logging.cs.filelock=INFO
log4j.logger.com.fatwire.logging.cs.firstsite.filter=INFO
log4j.logger.com.fatwire.logging.cs.framework=INFO
log4j.logger.com.fatwire.logging.cs.install=INFO
log4j.logger.com.fatwire.logging.cs.jsp=INFO
log4j.logger.com.fatwire.logging.cs.realtime=INFO
log4j.logger.com.fatwire.logging.cs.request=INFO
log4j.logger.com.fatwire.logging.cs.satellite=INFO
log4j.logger.com.fatwire.logging.cs.satellite.cache=INFO
  
```

- b. Copy the added logger to the `log4j.properties` file in the `WEB-INF/classes` directory on your application server.
- c. Save the `log4j.properties` file.

30.2 System Information

The System Information tool provides immediate access to various types of WebCenter Sites, database, and thread information for troubleshooting and checking the health of your application. You can download information to a file or view information directly in the administrator's interface.

This section contains the following topics:

- [Section 30.2.1, "WebCenter Sites Information"](#)
- [Section 30.2.2, "Database Information"](#)
- [Section 30.2.3, "Thread Information"](#)

Note: You can download all available system information and a list of all WebCenter Sites directories and files on the application server by right-clicking the System Information tool and selecting Download. If Sigar is configured, you can also download Sigar information to send to Technical Support by right-clicking the tool and selecting System Details (Sigar).

30.2.1 WebCenter Sites Information

You can download or view the following types of WebCenter Sites information:

- **Memory Information** – The amount of free memory in the application server Java Virtual Machine, the total amount of memory free and in use, and the maximum amount of memory that the Java Virtual Machine will attempt to use.
- **Sites Properties** – The `ini`, `xml`, and `properties` files in the WebCenter Sites installation directory.
- **Web Apps Properties** – The `xml` and `properties` files in the `WEB-INF` and `WEB-INF/classes` directories on the application server.
- **Other Sites Info** – The `jar` files in the `WEB-INF/lib` directory on the application server, the session variables for the user currently logged in to WebCenter Sites, and the Java run-time properties.

To download or view WebCenter Sites information

1. On the **Admin** tab, expand **System Tools**, then expand **System Information**, and double-click **Sites Info**.

The WebCenter Sites Information form shows memory information as well as additional information that you can select to download or view.

Figure 30–6 Sites Information Form

Sites Information

Memory Information In Bytes	
Free Memory	275164800
Total Memory	399769600
Max Memory	799145984

Sites Properties:

Available

- assetframework.ini
- batch.ini
- catalog.ini
- commercedata.ini
- databaseloader.ini
- dir.ini
- futuretense.ini
- futuretense_xcel.ini
- gator.ini
- logging.ini
- ms.ini
- propeditor-descriptions.prope

Add
Remove

Selected

Web Apps Properties:

Available

- commons-logging.properties
- fckeditor.properties
- log4j.properties
- satellite.properties
- ServletRequest.properties
- AdvPub.xml
- applicationContext.xml
- BufferingConfig.xml
- cas-cache.xml
- cs-cache.xml
- linked-cache.xml
- RestResources.xml

Add
Remove

Selected

Other Sites Info:

Jar Versions

Session Variables

System Variables

Show Results Download

2. In the Sites Properties and Web Apps Properties sections, select the files that you are interested in and click Add.
3. In the Other Sites Info section, do the following:
 - To obtain information about the jar files in the WEB-INF/lib directory on the application server, select the **Jar Versions** check box.
 - To obtain the session variables for the user currently logged in to WebCenter Sites, select the **Sites Session Variables** check box.
 - To obtain Java run-time properties, select the **System Variables** check box.
4. To download the selected WebCenter Sites information as a text file, click **Download** and then open or save SysInfo.zip. The text file is inside SysInfo.zip.

5. To view the selected WebCenter Sites information in the administrator's interface, click **Show Results**.
 - a. The selected files and information are displayed directly in the administrator's interface.
 - b. To view the details of a file or type of information, click its "plus" icon.
For ini and properties files, as well as WebCenter Sites session variables and system variables, the properties and values are displayed.

Figure 30–7 Properties and Values for batch.ini

batch.ini

Property	Value
debug	false
hidden.encrypted	true
request.folder	./dispatcher/
security.class	com.openmarket.Batch.DefaultSecurity
thread.count	32
thread.growcache	false
thread.idle	10
thread.wait	15

For xml files, the code is displayed.

Figure 30–8 XML File Code Display

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
    "http://www.springframework.org/dtd/spring-beans.dtd">
<beans/>
```

For Jar Versions, the jar names, versions, and implementation vendors are displayed. NA : NA indicates that a jar file does not provide its version and vendor.

Figure 30–9 Jar Versions and Vendors

Jar Versions

Jar Name	Version : Vendor
jersey-json-1.1.4.1.jar	1.1.4.1 : Sun Microsystems, Inc
analyticscs.jar	Build Date: Oct 13 2011 at 17:24:19 Analytics-bld- 10 Revision:140059 : FatWire Corporation
commons-io-1.4.jar	1.4 : The Apache Software Foundation
core-renderer-R8pre2.jar	NA : NA
httpcore-4.1.2.jar	4.1.2 : The Apache Software Foundation

30.2.2 Database Information

You can download or view the following types of database information:

- **Basic Information** – The database engine and version, default transaction isolation level, JDBC driver, URL, and version, and JNDI data source.
- **System Tables** – Tables that store core WebCenter Sites data. The ElementCatalog table, for example, lists all of the XML or JSP elements used in your system.
- **Tree Tables** – Tables that store parent-child relationships between objects. The AssetRelationTree, for example, stores information about the associations between assets.
- **Sites Object Tables** – Tables that store uniquely identified objects, such as asset types, asset attributes, and asset definitions.
- **Other Sites Tables** – Tables that store flat data that cannot be organized into parent-child relationships and the foreign tables that WebCenter Sites does not manage.

Note: You have permission to access a database table if at least one of the Access Control Lists (ACLs) assigned to the table is assigned to your user. For more information about database permissions, see [Section 4.2, "ACLs."](#) For more information about the different types of WebCenter Sites database tables, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

To download or view database information

1. On the **Admin** tab, expand **System Tools**, then select **System Information** and double-click **DB Info**.

The Database Information form shows basic information about your database as well as the tables that you can select to download or view.

Figure 30–10 Database Information Form**Database Information**

Basic Information	
Database Server Information	HSQL Database Engine 1.8.0
Default Transaction Isolation	TRANSACTION_READ_UNCOMMITTED
JDBC Driver Info	HSQL Database Engine Driver 1.8.0
JDBC Driver URI	jdbc:hsqldb:C:/FatWire/JSK/JSK_022312-8080/App_Server/apache-tomcat-6.0.32/default/data/hypersonic/csDB
JDBC Version	3.0
JNDI Data Source	java:comp/env/csDataSource

System Tables:

Available		Selected
<ul style="list-style-type: none"> ElementCatalog FW_InvalidationMemory RTInfo SiteCatalog SystemACL SystemEvents SystemIdGenerator SystemInfo SystemItemCache SystemLocalAlias SystemLocaleString SystemPageCache 	<p>Add</p> <p>Remove</p>	

Tree Tables:

Available	Selected
<ul style="list-style-type: none"> AssetRelationTree LocaleTree 	

2. In the System Tables, Tree Tables, Sites Object Tables, and Other Sites Tables sections, select the tables that you want to view and click Add.
3. To download the selected database information as a spreadsheet file, click **Download** and then open or save *SysInfo.zip*. The spreadsheet file is inside *SysInfo.zip*.
4. To view the selected database information in the administrator's interface, click **Show Results**.
 - a. For each selected table that you have permission to access, the following information is displayed:
 - Table Name
 - Primary key
 - Number of rows
 - CSz (number of resultsets to cache in memory)
 - Timeout (number of minutes to keep a resultset cached in memory)
 - Abs (how to calculate the expiration time for resultsets)
 - Time to run SQL
 - Special table key
 - b. To view a table's column and indices, click its "plus" icon. If an index is listed more than once, it extends across multiple columns.

Figure 30–11 Tree Tables Information

Tree Tables

Table Name	Primary Key	Number of rows	CSz	Timeout	Abx	Time to Run SQL	Special Table Key
AssetRelationTree	nid	492				31 ms	

Name	Type	Size
nid	(integer)	0
nparentid	(integer)	0
nrank	(integer)	0
otype	(text)	28
oid	(integer)	0
oversion	(integer)	0
ncode	(text)	16

Column Info:

Name	Type	Column	Ordinal Position	Non Unique
SYS_IDX_254	Non clustered	NID	1	false
ASSETRELATIONTREE1	Non clustered	OTYPE	1	true
ASSETRELATIONTREE1	Non clustered	OID	2	true
ASSETRELATIONTREE1	Non clustered	OVERSON	3	true
ASSETRELATIONTREE5	Non clustered	OTYPE	1	true
ASSETRELATIONTREE5	Non clustered	OID	2	true
ASSETRELATIONTREE7	Non clustered	NPARENTID	1	true

Index Info:

30.2.3 Thread Information

You can download or view thread dumps to analyze the runtime state of the application server. This can be especially useful when trying to detect problems that might result in resource starvation or thread hangs.

To download or view thread information

1. On the Admin tab, expand **System Tools**, then expand **System Information** and double-click **Thread Info**.

The Thread Information form is displayed.

Figure 30–12 Thread Information Form

Thread Information

Thread Name: Regular Expression:

States:

2. The **Regular Expression** check box is selected by default. This option enables you to use Java regular expressions to search for thread names.

Examples:

- **Thread.*** to find Thread-0, Thread-1, and Thread-3
- **.*thread.*** to find pool-6-thread-1, pool-8-thread-1, and pool-14-thread-2

If you deselect the check box, you must search for exact thread names.

3. In the **Thread Name** field, enter the thread name or text string that you want to search for.
4. If you want to limit your search to one or more thread states, select the state(s) in the **States** field. If you do not select a state, all states will be searched.
5. To download the selected thread information as a text file for use with a thread dump analyzer, click **Download** and then open or save *SysInfo.zip*. The text file is inside *SysInfo.zip*.

Note: Downloading thread information is possible only in the following cases:

- WebCenter Sites' JDK includes a compatible `jstack` utility.
- The first JDK listed in your operating system's library path includes a `jstack` utility that is compatible with WebCenter Sites' JDK.

If a compatible `jstack` is not found, the following message displays:
"Thread dump feature is not supported for this platform."

6. To view the selected thread information in the administrator's interface, click **Show Results**. For each thread matching the search, the following information is displayed:
 - Name
 - ID
 - State
 - Total time the CPU executed the thread
 - Total time the thread was in user mode
 - Number of times it was blocked and waited
 - How long it was blocked and waited
 - Object for which it is waiting
 - Stack trace

Figure 30–13 Thread Dump Results**Thread Dump**

```

Full Thread Dump at Tue Feb 08 10:02:38 EST 2011
"Timer-0" id=39 in TIMED_WAITING on lock=java.util.TaskQueue@fd9d27
total cpu time=0.00 ms, user time=0.00 ms
blocked count=1, blocked time=0.00 ms, wait count=23, wait time=0.91 ms
  at java.lang.Object.wait(Native Method)
  at java.util.TimerThread.mainLoop(Timer.java:509)
  at java.util.TimerThread.run(Timer.java:462)

"ScannerThread" id=42 in TIMED_WAITING
total cpu time=0.00 ms, user time=0.00 ms
blocked count=49, blocked time=0.00 ms, wait count=252, wait time=0.96 ms
  at java.lang.Thread.sleep(Native Method)
  at org.jboss.deployment.scanner.AbstractDeploymentScanner$ScannerThread.loop
(AbstractDeploymentScanner.java:280)
  at org.jboss.deployment.scanner.AbstractDeploymentScanner$ScannerThread.run
(AbstractDeploymentScanner.java:225)

```

30.3 Cache Management

If the inCache framework is set up for page caching, asset caching, or both, the Cache Management tool enables you to monitor the caching processes on WebCenter Sites cluster members and their co-resident Satellite Servers. Features include cache searching, force writing to disk, and flushing individual pages, assets, dependencies, and entire caches. For information about setting up the inCache framework, see [Chapter 28, "inCache for Page Caching"](#) and [Chapter 29, "inCache for Asset Caching."](#)

Note: The **Cache Management** tool provides caching information only for WebCenter Sites cluster members and their co-resident Satellite Servers. To access cache information for remote Satellite Servers, use the following URL:

```

http://<server>:<port>/<application_context_
path>/Inventory?username=<SatelliteServer>&password=<password>&detail=debug

```

30.4 Cache Management - CAS Cache

This section includes the following topics:

- [Section 30.4.1, "Summary"](#)
- [Section 30.4.2, "Cluster Info"](#)

30.4.1 Summary

This form lists the following basic data for each cache (values are estimates):

- Total Count (total number of cached elements)
- Memory Store Count (number of cached elements in memory)
- Disk Store Count (number of cached elements on disk)
- Cache Hits (number of cached elements retrieved from both memory and disk)
- In Memory Hits (number of cached elements retrieved from memory)

- On Disk Hits (number of cached elements retrieved from disk)
- Cache Misses (number of times elements were not found in the cache)
- Eviction Count (number of elements evicted from the cache)
- Average Get Time (average number of milliseconds for retrieving an element)
- Max Elements In Memory (maximum number of elements allowed in memory)
- Max Elements On Disk (maximum number of elements allowed on disk)
- Overflow to Disk (whether elements are written to disk when memory is full)

This form also provides the following options. They should be used with caution, especially on a delivery system, as they can impact the performance of your site.

- Writing a cache's entire memory store to disk – This option can be particularly useful if WebCenter Sites is not set up to automatically write a cache's memory store to disk before shutting down.
- Flushing a cache's entire memory and disk stores.

Note: If legacy page caching is enabled, cached page data is stored in the `SystemPageCache` table and information about the stored data is not displayed in the Summary window.

To view and work with summary cache information

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **CAS Cache** and double-click **Summary**.

The Summary page displays.

Figure 30-14 CAS Cache Summary

Summary

Node:http://localhost:8380 on Monday, February 25, 2013 10:22:58 AM PST

Name	Ticket Cache	Session Cache
Total Count	0	2
Memory Store Count	0	2
Disk Store Count	0	2
Cache Hits	0	206
In Memory Hits	0	206
On Disk Hits	0	0
Cache Misses	0	2
Eviction Count	0	0
Average Get Time	0.0	0.0
Max Elements In Memory	1000000	1000000
Max Elements On Disk	10000000	1000000
Overflow To Disk	true	true
Write To Disk	<input type="checkbox"/>	<input type="checkbox"/>
Flush Cache	<input type="checkbox"/>	<input type="checkbox"/>

2. Select the checkbox for the **Ticket Cache** or the **Session Cache** (or both) and click **Write To Disk** to write the selected cache's entire memory store to disk.
3. Select the checkbox for the **Ticket Cache** or the **Session Cache** (or both) and click **Flush Cache** to flush the memory and disk stores for the selected caches.

30.4.2 Cluster Info

If Sigar is configured, this form lists the name, URL, port, and GUID of each additional WebCenter Sites cluster member.

To view the cluster information, expand **System Tools** on the **Admin** tab, then select **Cache Management**, then expand **CAS Cache** and double-click **Cluster Info**.

Figure 30–15 Cluster Info Form**Cluster Info**Number of remote cluster peer = **1**

Name	Url	Port	Guid
notifier	192.0.2.201	54014	oracle-desk/192.0.2.201-3a43be2d:136c6d640a2

30.5 Cache Management - Resultset Cache

Oracle WebCenter Sites caches resultsets to help optimize system performance. Caching prevents overloading the database with repeated queries for the same information, and serving from cache is always faster than performing another database lookup.

By default, Oracle WebCenter Sites stores resultsets within the inCache framework, instead of hash tables. Both caching frameworks use Java memory. You can switch between frameworks by setting the `rsCacheOverInCache` property in `futuretense.ini` to either `true` or `false`.

When resultset caching over inCache is enabled, the **System Tools** node displays the **Resultset Cache** tool, which provides statistical information about the caches and their contents as described in this section. Additional information about resultset caching is available in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

This section contains the following topics:

- [Section 30.5.1, "Enabling and Configuring Resultset Caching over inCache"](#)
- [Section 30.5.2, "Summary"](#)
- [Section 30.5.3, "Cluster Info"](#)

30.5.1 Enabling and Configuring Resultset Caching over inCache

If you do not enable resultset caching over inCache, caching in hash tables is enabled by default.

To enable and configure resultset caching

1. Enable resultset caching:
 - a. Make sure the `linked-cache.xml` configuration file is present in the application server's classpath (`WEB-INF/classes` directory).
 - b. In the `futuretense.ini` file, set the `rsCacheOverInCache` property to `true`.

2. Configure resultset caching:

In the `futuretense.ini` file, set values for the following resultset caching properties:

- `cc.cacheResults` – Specify the default number of resultsets to cache.
- `cc.cacheResultsTimeout` – Specify the number of minutes to keep a resultset cached.

- `cc.cacheResultsAbs` – Specify how expiration time will be calculated for cached resultsets.

For detailed information about configuring resultset caching properties, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

30.5.2 Summary

The summary view of resultset cache displays statistical information about all caches. You can select a cache from the summary table to view the database queries it contains. You can further view the resultset elements of the database queries contained in a cache.

To view cache information

1. On the **Admin** tab, expand **System Tools** then select **Cache Management**, then expand **Resultset Cache** and double-click **Summary**.

The summary table displays.

Figure 30–16 Resultset Cache over InCache Summary

Resultset Cache Over inCache

Statistics in the table below are not presented in real time. They are provided to help you estimate the status of the caches.

Refresh

Filter Clear Invalidate

Cache	Count	Max Count	Fill Ratio	Hits	Misses	Hit Ratio	Misses Exp	Get time	Puts	Updates	Expired	Removed	Evicted	Eternal?	TTL	TTI
<input type="checkbox"/> FW_InvalidationMemory	0	500	0%	0	0	0%	0	0.0	0	0	0	0	0	false	300	300
<input type="checkbox"/> SystemACL	1	25	4%	3	1	75%	0	0.0	1	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> SystemInfo	3	500	0%	26	3	89%	0	0.0	3	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> SystemLocaleString->locale	1	500	0%	336	2	99%	0	0.0	2	0	1	0	0	false	300	300
<input type="checkbox"/> SystemUserAttr	13	500	2%	115	26	81%	0	0.0	26	0	13	0	0	false	300	300
<input type="checkbox"/> SystemUsers	13	100	13%	113	13	89%	0	0.0	13	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> XMLElements	73	1000	7%	2026	128	94%	2	0.0070	143	370	70	0	0	false	3600	3600
<input type="checkbox"/> ac->ac-hash	12	25	48%	9059	11	99%	0	0.0	12	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> cached-catalogs	37	1000000	0%	25800	1956	92%	33	0.0060	115	0	78	0	0	false	1800	1800
<input type="checkbox"/> com.fatwire.assetapi.def.AssetTypeDefManager	2	1000	0%	18	2	90%	0	0.0	2	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> com.openmarket.assetmaker.asset.AMAsset	2	1000	0%	9	2	81%	0	0.0	2	0	0	0	0	false	315360000	315360000
<input type="checkbox"/> csDataSourceApprovalQueue	0	500	0%	0	48	0%	0	0.0	24	0	0	0	0	false	300	300
<input type="checkbox"/> csDataSourceAssetPublication	0	500	0%	0	4	0%	0	0.0	4	0	4	0	0	false	300	300
<input type="checkbox"/> csDataSourceAssetQueues	0	500	0%	0	1536	0%	0	0.0	0	0	0	0	0	false	300	300

2. Click one of the following options, located above the summary table:
 - **Refresh** refreshes the results of the summary table.
 - **Filter** (caches) narrows down the results shown in the summary table to only selected caches.
 - **Clear** clears selected caches.
 - **Invalidate** invalidates selected caches. All invalidated resultset elements are purged by the cleaning thread, which runs periodically.

Each resultset element contains a list of tables on which it depends. When an invalidated resultset element is requested, it will be read from the database and reloaded into cache.

Table 30–1 Descriptions of Columns in the Summary Table

Column Name	Description
Cache	Name of the cache.
Count	Number of resultset elements in the cache. A resultset element is a key-value pair, where the key is the query and the value is the result returned by the query.
MaxCount	Capacity of the cache (maximum number of resultset elements allowed in the cache).
Fill Ratio	Percentage of cache filled: $(\text{Count} / \text{MaxCount}) \times 100$
Hits	Total number of times all resultset elements were read from the cache.
Misses	Number of times the resultset element was not found in cache.
Hit ratio	Percentage of hits: $[\text{Hits} / (\text{Hits} + \text{Misses})] \times 100$
Misses Exp	Number of misses due to resultset elements that are expired but not removed.
Get time	Average time (in ms) for get operations.
Puts	Number of times resultset elements were placed in the cache following a database query.
Updates	Number of times individual resultset elements in the cache have been updated.
Expired	Number of expired resultset elements. They are removed by a cleaning thread that runs periodically in the background.
Removed	Number of resultset elements removed for any reason (such as a manual clear operation or cleaning thread).
Evicted	Number of evicted resultset elements. Eviction occurs when the capacity of the cache is exceeded.
Eternal	Indicates whether all resultset elements in the cache have infinite expiration time.
TTL	Time to live. Length of time (in seconds) that resultset elements will remain in cache.
TTI	Time to idle. If a resultset element is not hit (read from cache) within this time period, it is marked as idle.



- To view detailed information about the contents of a cache, click the cache name in the summary table.

The detail view opens to show the keys (queries), their related statistics, and a search field for locating the cache's content by key, expiration time, and dependency.

To refresh your form, click **Search**.

Figure 30-17 Detail View for Cache Contents

Cache : XMLElements

Size	Capacity	Fill Ratio	Hits	Misses	Hit Ratio	Misses Exp	Get time	Puts	Updates	Expired	Removed	Evicted	Eternal?	TTL	TTI
21	1000	 2%	1542	86	 94%	1	0.01	98	447	77	0	0	false	3600	3600

Items Per Page Search By

Table Legend

- Key=csDataSource-Content_C-*-id=1114083739888 Invalidated Page
- esDataSourceContent_C Updated
- Dependency

Number of items in cache = 21 1 2 3 of 3 pages last

Key	Dependencies	Created Time	Last accessed time	Expired time
fatwire/systemtools/logs/logViewer	XMLElements	Tue Mar 06 19:55:14 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 20:55:14 PST 2012
OpenMarket/Xcelerate/Search/Event	XMLElements	Tue Mar 06 20:01:00 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 21:01:00 PST 2012
OpenMarket/Gator/UIFramework/LoadAdminTree	XMLElements	Tue Mar 06 20:18:43 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 21:18:43 PST 2012
OpenMarket/Xcelerate/Util/SetLocale	XMLElements	Tue Mar 06 20:19:12 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 21:19:12 PST 2012
OpenMarket/Gator/UIFramework/Admin/LoadLinkedCacheDetails	XMLElements	Tue Mar 06 20:18:43 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 21:18:43 PST 2012
fatwire/systemtools/FSTest/TestFS	XMLElements	Tue Mar 06 19:55:46 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 20:55:46 PST 2012
fatwire/systemtools/logs/logTailViewer	XMLElements	Tue Mar 06 19:43:44 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 20:43:44 PST 2012
OpenMarket/Gator/UIFramework/BuildTreeNodeID	XMLElements	Tue Mar 06 20:18:44 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 21:18:44 PST 2012
OpenMarket/Xcelerate/Scripts/DateTimeWidget	XMLElements	Tue Mar 06 19:33:53 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 20:33:53 PST 2012
OpenMarket/Gator/UIFramework/UITimeOutPopup	XMLElements	Tue Mar 06 19:43:45 PST 2012	Tue Mar 06 20:22:10 PST 2012	Tue Mar 06 20:43:45 PST 2012

Table 30-2 Descriptions of Columns in the Cache Detail Table

Column Name	Description
Key	Key (query) for the resultset element.
Dependencies	List of tables against which the query was run.
Created time	When this resultset element was placed in the cache.
Last accessed time	When this cached resultset element was last accessed.
Expired time	When this resultset element will expire.

- To view the resultset element for a key, click the desired key (query).
Columns in the table that opens are dynamic and depend on your site's data model and how the query is set up.

Figure 30–18 Result Set Key Query Form**Result Set**

```
csDataSource-SELECT TBLNAME, ACL FROM SYSTEMINFO WHERE TBLNAME IN
('ELEMENTCATALOG', 'SITECATALOG')[Number of positions=0] data [
```

acl	tblname
SiteGod,ElementEditor,ElementReader	ElementCatalog
SiteGod,PageEditor,PageReader	SiteCatalog

30.5.3 Cluster Info

If Sigar is configured, this form lists the name, URL, port, and GUID of each additional WebCenter Sites cluster member.

To view the cluster information, expand **System Tools** on the **Admin** tab, then select **Cache Management**, then expand **Resultset Cache** and double-click **Cluster Info**.

Figure 30–19 Cluster Info Form**Cluster Info**

Number of remote cluster peer = **1**

Name	Url	Port	Guid
notifier	192.0.2.201	54014	oracle-desk/192.0.2.201-3a43be2d:136c6d640a2

30.6 Cache Management - Sites Cache

Sites Cache Cache management.

This section contains the following topics:

- [Section 30.6.1, "Summary"](#)
- [Section 30.6.2, "Page Cache"](#)
- [Section 30.6.3, "Dependency"](#)
- [Section 30.6.4, "Asset Cache"](#)
- [Section 30.6.5, "URL Cache"](#)
- [Section 30.6.6, "Cluster Info"](#)

30.6.1 Summary

This form lists the following basic data for each cache (values are estimates):

- Total Count (total number of cached elements)

- Memory Store Count (number of cached elements in memory)
- Disk Store Count (number of cached elements on disk)
- Cache Hits (number of cached elements retrieved from both memory and disk)
- In Memory Hits (number of cached elements retrieved from memory)
- On Disk Hits (number of cached elements retrieved from disk)
- Cache Misses (number of times elements were not found in the cache)
- Eviction Count (number of elements evicted from the cache)
- Average Get Time (average number of milliseconds for retrieving an element)
- Max Elements In Memory (maximum number of elements allowed in memory)
- Max Elements On Disk (maximum number of elements allowed on disk)
- Overflow to Disk (whether elements are written to disk when memory is full)

This form also provides the following options. They should be used with caution, especially on a delivery system, as they can impact the performance of your site.

- Writing a cache's entire memory store to disk – This option can be particularly useful if WebCenter Sites is not set up to automatically write a cache's memory store to disk before shutting down.
- Flushing a cache's entire memory and disk stores

Note: If legacy page caching is enabled, cached page data is stored in the `SystemPageCache` table and information about the stored data is not displayed in the Summary window. In addition, this window displays the **Clear Page Cache** button which allows you to flush the stored data in System Page Cache and Co-Res SS Cache.

As System Page Cache is legacy, it must be enabled for use. This is done by disabling ehcache by including the following as a JVM parameter of appserver:

```
-Dcs.useEhcache=false
```

To view and work with summary cache information

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Sites Cache** and double-click **Summary**.

The Summary page displays.

Figure 30–20 Sites Cache Summary**Summary**

Node:http://localhost:8380 on Monday, February 25, 2013 10:29:16 AM PST

Name	Page Cache	Cache Dependencies	Asset Cache	URL Cache
Total Count	1	118	7	0
Memory Store Count	1	9	7	0
Disk Store Count	1	118	7	0
Cache Hits	1	32	12	0
In Memory Hits	1	23	12	0
On Disk Hits	0	9	0	0
Cache Misses	2	0	7	0
Eviction Count	0	0	0	0
Average Get Time	0.0	0.0	0.0	0.0
Max Elements In Memory	200000	200000	5000	1000
Max Elements On Disk	1000000	1000000	5000	1000
Overflow To Disk	true	true	true	true
Write To Disk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Flush Cache	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Select the checkbox for the **PageCache**, **Cache Dependencies**, **Asset Cache**, or **URL Cache** and click **Write To Disk** to write the selected cache's entire memory store to disk.
3. Select the checkbox for the **PageCache**, **Cache Dependencies**, **Asset Cache**, or **URL Cache** and click **Flush Cache** to flush the memory and disk stores for the selected caches.

If you have enabled System Page Caching, a button labeled **Clear Page Cache** will appear underneath the **Flush Cache** button. Click **Clear Page Cache** to flush the SystemPageCache table and Co-Res SS Cache.

30.6.2 Page Cache

This form lists statistics for each cached page, such as hit count, time of creation/last access/expiration, and dependencies. The table legend identifies which types of information are tracked: invalidated pages, updates to dependencies, and changes to the dependency generation count. The form also provides the following options:

- Searching the cache for specific pages – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing pages from the cache
- Viewing html information for pages
- Looking up page dependencies

The view and work with items in the Page Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Sites Cache** and double-click **Page Cache**.

The Pages form displays.

Figure 30–21 Sites Pages Cache

Pages

Items Per Page Search By

Table Legend

pagename=FirstSiteII/Page Invalidated Page

asset-123+Content_C Updated Dependency

asset-123:Page Superscript⁵ Dependency Generation Count

Number of items in cache = 1 1 of 1 page

	Key	Hit Count	Created Time	Last Access Time	Expire Time	Dependency
<input type="checkbox"/>	pagename=fatwire/wem/ui/Ping&ft_ss=false view page details	2	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:24 PM PST	Thursday, February 23, 2023 08:30:49 AM PST	_NODEP_ ¹⁴ fatwire/wem/ui/Ping ¹⁴

2. At the top of the form is the Search area. Select to search by Key, Expired Time, or by Dependency. If you enter an expired time to search by, use the forms *mm/dd/yyyy* or *mm/dd/yyyy hh:mm:ss* to enter a date and time. Click **Search**.

3. In the displayed items in the cache, click the header title to sort the items by that criteria.

All items are shown when initially loading the Pages form. The Search form limits what is displayed, making it easier to find items in the cache.

4. Underneath each item in the **Key** column, a **view page details** link appears. Click **view page details** to open a new window that displays the page's HTML information.

5. To flush selected assets from the cache, mark the checkbox on the left side of each asset. Mark the checkbox in the header to select all displayed assets. Click **Flush Cache** to flush the memory and stores of the assets.

6. To view the pages for a specific item within the cache, mark the checkbox on the right side of each item. Mark the checkbox in the header to mark all displayed dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note that each item in cache can have multiple checkboxes on the right side, reflecting multiple dependencies.

Note: After WebCenter Sites is restarted, this form does not show page cache information until the cache engine is initialized. The cache engine is initialized when the first page or asset is requested.

30.6.3 Dependency

This form lists the contents of the dependency cache and the generation count for each dependency. In addition, it provides the following options:

- Searching the cache for specific dependencies – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing dependencies from the cache
- Looking up page dependencies

The view and work with items in the Dependency Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Sites Cache** and double-click **Dependency**.

The Dependencies form displays.

Figure 30-22 Sites Cache Dependencies

Dependencies

Items Per Page

Search

Current System Generation Count is 36

Number of items in cache = 118 1 2 3 4 5 6 7 of 12 pages [last](#)

Flush Cache **Page LookUp**

<input type="checkbox"/> Key	Generation Count	Valid	<input type="checkbox"/>
<input type="checkbox"/> FW_View:1283176967296	0	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351718518:ContentDef	0	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1328196048187:AVIArticle	0	true	<input type="checkbox"/>
<input type="checkbox"/> FW_View:1262707329009	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1328196047475:AVIArticle	29	true	<input type="checkbox"/>
<input type="checkbox"/> SitePlan:1346041992371	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351718519:ContentAttribute	0	true	<input type="checkbox"/>
<input type="checkbox"/> Page:1327351719525	0	true	<input type="checkbox"/>
<input type="checkbox"/> FW_Application:1262707329030	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351720065:Template	29	true	<input type="checkbox"/>

2. In the displayed items in the cache, click the header title to sort the items by that criteria.
 All items are shown when initially loading the Dependencies form. The Search form limits what is displayed, making it easier to find items in the cache.
3. To flush selected assets from the cache, mark the checkbox on the left side of each asset. Mark the checkbox in the header to select all displayed assets. Click **Flush Cache** to flush the memory and stores of the assets.
4. To view the pages for a specific item within the cache, mark the checkbox on the right side of each item. Mark the checkbox in the header to mark all displayed

dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note: After WebCenter Sites is restarted, this form does not show dependency cache information until the cache engine is initialized. The cache engine is initialized when the first page or asset is requested.

30.6.4 Asset Cache

This form lists statistics for each cached asset, such as hit count, time of creation/last access, and dependencies. The table legend identifies which types of information are tracked: invalidated assets, updates to dependencies, and changes to the dependency generation count. The form also provides the following options:

- Searching the cache for specific assets – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing assets from the cache
- Looking up asset dependencies

The view and work with items in the Asset Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Sites Cache** and double-click **Asset Cache**.

The Asset Cache form displays.

Figure 30–23 Sites Cache Assets

Assets

Table Legend

Product_C:123:2	Invalidated Page
asset-123:Content_C	Updated Dependency
asset-123:Page Superscript ⁵	Dependency Generation Count

Number of items in cache = 7

1 of 1 page

Flush Cache

Asset Lookup

	Key	Hit Count	Created Time	Last Access Time	Dependency
<input type="checkbox"/>	FW_Application:1283176967302:3	2	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:23 PM PST	FW_Application:1283176967302 ¹⁴
<input type="checkbox"/>	SitePlan:1346041992371:2	3	Monday, February 25, 2013 08:31:08 AM PST	Monday, February 25, 2013 13:21:25 PM PST	SitePlan:1346041992371 ¹⁴
<input type="checkbox"/>	FW_Application:1262707329030:3	5	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:24 PM PST	FW_Application:1262707329030 ¹⁴
<input type="checkbox"/>	FW_Application:1253211458856:3	2	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:23 PM PST	FW_Application:1253211458856 ¹⁴
<input type="checkbox"/>	SitePlan:1343556967882:2	3	Monday, February 25, 2013 08:31:07 AM PST	Monday, February 25, 2013 13:21:25 PM PST	SitePlan:1343556967882 ¹⁴
<input type="checkbox"/>	FW_View:1262707329009:3	2	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:24 PM PST	FW_View:1262707329009 ¹⁴
<input type="checkbox"/>	SitePlan:1346041992593:2	3	Monday, February 25, 2013 08:31:08 AM PST	Monday, February 25, 2013 13:21:25 PM PST	SitePlan:1346041992593 ¹⁴

2. At the top of the form is the Search area. Select to search by Key or by Dependency, and to show All or Any assets with the entered search value. Click **Search**.

3. In the displayed items in the cache, click the header title to sort the items by that criteria.

All assets are shown when initially loading the Assets form. The Search form limits what is displayed, making it easier to find items in the cache.

4. To flush selected assets from the cache, mark the checkbox on the left side of each asset. Mark the checkbox in the header to select all displayed assets. Click **Flush Cache** to flush the memory and stores of the assets.
5. To view the assets for a specific dependency within the cache, mark the checkbox on the right side of each asset. Mark the checkbox in the header to mark all displayed assets. Click **Asset Lookup** to show the dependencies.

Note: After WebCenter Sites is restarted, this form does not show asset cache information until the cache engine is initialized. The cache engine is initialized when the first page or asset is requested.

30.6.5 URL Cache

All the Vanity URLs are cached after the resolution is done in a web page in inCache. The cache can be looked up and managed in the **Cache Management** node under **System Tools**.

The view and work with items in the URL Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Sites Cache** and double-click **URL Cache**.

The URLs form displays.

Figure 30–24 URLs form, URL Cache

The screenshot shows the URL Cache interface. At the top, there are search controls: 'Items Per Page' set to 10 and 'Search By' set to 'Key'. Below this is a search input field and a 'Search' button. To the right is a 'Table Legend' with three entries: 'http://localhost:8080/cs/PSIL/home' (Invalidated Page), 'asset-123+Content_C' (Updated Dependency), and 'asset-123:Page Superscript' (Dependency Generation Count). Below the search area, it says 'Number of items in cache = 2' and '1 of 1 page'. There are 'Flush Cache' and 'Asset Lookup' buttons. The main table has columns for Key, Hit Count, Created Time, Last Access Time, and Dependency. Two items are listed:

Key	Hit Count	Created Time	Last Access Time	Dependency
<input type="checkbox"/> http://localhost:8080/ps7/PSI/55555	0	Tuesday, February 19, 2013 21:53:20 PM UTC	Tuesday, February 19, 2013 21:53:20 PM UTC	asset-1155737750447:Page ¹¹
<input type="checkbox"/> http://localhost:8080/ps7/Avi/TouchHome	0	Tuesday, February 19, 2013 22:35:15 PM UTC	Tuesday, February 19, 2013 22:35:15 PM UTC	asset-1327351719456:Page ¹³

2. The Search area is at the top of the form. Select to search by Key or by Dependency with the entered search value. Click **Search**.
3. In the displayed items in the cache, click the header title to sort the items by that criteria.

All items are shown when initially loading the URL form. The Search form limits what is displayed, making it easier to find items in the cache.

4. To flush selected URLs from the cache, mark the checkbox on the left side of each URL. Mark the checkbox in the header to select all displayed URLs. Click **Flush Cache** to flush the memory and stores of the URLs.
5. To view the URL for a specific dependency within the cache, mark the checkbox on the right side of each URL. Mark the checkbox in the header to mark all displayed URLs. Click **Asset Lookup** to show the dependencies.

30.6.6 Cluster Info

If Sigar is configured, this form lists the name, URL, port, and GUID of each additional WebCenter Sites cluster member.

To view the cluster information, expand **System Tools** on the **Admin** tab, then select **Cache Management**, then expand **Sites Cache** and double-click **Cluster Info**.

Figure 30–25 Cluster Info Form

Cluster Info

Number of remote cluster peer = **1**

Name	Url	Port	Guid
notifier	192.0.2.201	54014	oracle-desk/192.0.2.201–3a43be2d:136c6d640a2

30.7 Cache Management - Co-Res Cache

Co-Res Cache Cache Management

This section includes the following topics:

- [Section 30.7.1, "Summary"](#)
- [Section 30.7.2, "Page Cache"](#)
- [Section 30.7.3, "Dependency"](#)
- [Section 30.7.4, "URL Cache"](#)
- [Section 30.7.5, "Cluster Info"](#)

30.7.1 Summary

This form lists the following basic data for each cache (values are estimates):

- Total Count (total number of cached elements)
- Memory Store Count (number of cached elements in memory)
- Disk Store Count (number of cached elements on disk)
- Cache Hits (number of cached elements retrieved from both memory and disk)
- In Memory Hits (number of cached elements retrieved from memory)
- On Disk Hits (number of cached elements retrieved from disk)
- Cache Misses (number of times elements were not found in the cache)

- Eviction Count (number of elements evicted from the cache)
- Average Get Time (average number of milliseconds for retrieving an element)
- Max Elements In Memory (maximum number of elements allowed in memory)
- Max Elements On Disk (maximum number of elements allowed on disk)
- Overflow to Disk (whether elements are written to disk when memory is full)

This form also provides the following options. They should be used with caution, especially on a delivery system, as they can impact the performance of your site.

- Writing a cache's entire memory store to disk – This option can be particularly useful if WebCenter Sites is not set up to automatically write a cache's memory store to disk before shutting down.
- Flushing a cache's entire memory and disk stores

Note: If legacy page caching is enabled, cached page data is stored in the `SystemPageCache` table and information about the stored data is not displayed in the Summary window.

To view and work with summary cache information

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Co-Res SS Cache** and double-click **Summary**.

The Summary page displays.

Figure 30–26 Co-Res SS Cache Summary**Summary**

Node:http://localhost:8380 on Monday, February 25, 2013 10:30:15 AM PST

Name	Page Cache	Cache Dependencies	URL Cache
Total Count	0	0	0
Memory Store Count	0	0	0
Disk Store Count	0	0	0
Cache Hits	0	0	0
In Memory Hits	0	0	0
On Disk Hits	0	0	0
Cache Misses	0	0	0
Eviction Count	0	0	0
Average Get Time	0.0	0.0	0.0
Max Elements In Memory	1000	1000	1000
Max Elements On Disk	1000	1000	1000
Overflow To Disk	true	true	true
Write To Disk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Flush Cache	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Select the checkbox for the **PageCache**, **Cache Dependencies**, or **URL Cache** and click **Write To Disk** to write the selected cache's entire memory store to disk.
3. Select the checkbox for the **PageCache**, **Cache Dependencies**, or **URL Cache** and click **Flush Cache** to flush the memory and disk stores for the selected caches.

30.7.2 Page Cache

This form lists statistics for each cached page, such as hit count, time of creation/last access/expiration, and dependencies. The table legend identifies which types of information are tracked: invalidated pages, updates to dependencies, and changes to the dependency generation count. The form also provides the following options:

- Searching the cache for specific pages – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing pages from the cache
- Viewing html information for pages
- Looking up page dependencies

The view and work with items in the Page Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Co-Res SS Cache** and double-click **Page Cache**.

The Pages form displays.

Figure 30–27 Co-Res SS Pages Cache

Pages

Items Per Page: 10 Search By: Key

Search

Number of items in cache = 1

Flush Cache

1 of 1 page Page Lookup

Key	Hit Count	Created Time	Last Access Time	Expire Time	Dependency
<input type="checkbox"/> <p>pagename=fatwire/wem/ui/Ping&ft_ss=false</p> view page details	2	Monday, February 25, 2013 08:30:50 AM PST	Monday, February 25, 2013 13:21:24 PM PST	Thursday, February 23, 2023 08:30:49 AM PST	<input type="checkbox"/> _NODEP_ ¹⁴ <input type="checkbox"/> fatwire/wem/ui/Ping ¹⁴

Table Legend

- pagename=FirstSiteII/Page Invalidated Page
- asset-123:Content_C Updated Dependency
- asset-123:Page Superscript⁵ Dependency Generation Count

2. At the top of the form is the Search area. Select to search by Key, Expired Time, or by Dependency. If you enter an expired time to search by, use the forms *mm/dd/yyyy* or *mm/dd/yyyy hh:mm:ss* to enter a date and time. Click **Search**.

3. In the displayed items in the cache, click the header title to sort the items by that criteria.

All items are shown when initially loading the Pages form. The Search form limits what is displayed, making it easier to find items in the cache.

4. Underneath each item in the **Key** column, a **view page details** link appears. Click **view page details** to open a new window that displays the page's HTML information.
5. To flush selected assets from the cache, mark the checkbox on the left side of each asset. Mark the checkbox in the header to select all displayed assets. Click **Flush Cache** to flush the memory and stores of the assets.
6. To view the pages for a specific item within the cache, mark the checkbox on the right side of each item. Mark the checkbox in the header to mark all displayed dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note that each item in cache can have multiple checkboxes on the right side, reflecting multiple dependencies.

Note: After WebCenter Sites is restarted, this form does not show page cache information until the cache engine is initialized. The cache engine is initialized when the first page or asset is requested.

30.7.3 Dependency

This form lists the contents of the dependency cache and the generation count for each dependency. In addition, it provides the following options:

- Searching the cache for specific dependencies – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing dependencies from the cache
- Looking up page dependencies

The view and work with items in the Dependency Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Co-Res SS Cache** and double-click **Dependency**.

The Dependencies form displays.

Figure 30–28 Co-Res SS Cache Dependencies

Dependencies

Items Per Page

Current System Generation Count is 36

Number of items in cache = 118 1 2 3 4 5 6 7 of 12 pages [last](#)

<input type="checkbox"/> Key	Generation Count	Valid	<input type="checkbox"/>
<input type="checkbox"/> FW_View:1283176967296	0	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351718518:ContentDef	0	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1328196048187:AVIArticle	0	true	<input type="checkbox"/>
<input type="checkbox"/> FW_View:1262707329009	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1328196047475:AVIArticle	29	true	<input type="checkbox"/>
<input type="checkbox"/> SitePlan:1346041992371	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351718519:ContentAttribute	0	true	<input type="checkbox"/>
<input type="checkbox"/> Page:1327351719525	0	true	<input type="checkbox"/>
<input type="checkbox"/> FW_Application:1262707329030	14	true	<input type="checkbox"/>
<input type="checkbox"/> asset-1327351720065:Template	29	true	<input type="checkbox"/>

2. In the displayed items in the cache, click the header title to sort the items by that criteria.

All items are shown when initially loading the Dependencies form. The Search form limits what is displayed, making it easier to find items in the cache.

3. To flush selected assets from the cache, mark the checkbox on the left side of each asset. Mark the checkbox in the header to select all displayed assets. Click **Flush Cache** to flush the memory and stores of the assets.
4. To view the pages for a specific item within the cache, mark the checkbox on the right side of each item. Mark the checkbox in the header to mark all displayed dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note: After WebCenter Sites is restarted, this form does not show dependency cache information until the cache engine is initialized. The cache engine is initialized when the first page or asset is requested.

30.7.4 URL Cache

All the Vanity URLs are cached after the resolution is done in a web page in inCache. The cache can be looked up and managed in the **Cache Management** node under **System Tools**.

The view and work with items in the URL Cache

1. On the **Admin** tab, expand **System Tools**, then select **Cache Management**, then expand **Co-Res SS Cache** and double-click **URL Cache**.

The URLs form displays.

Figure 30–29 URLs form, URL Cache

URLs

Items Per Page: 10 Search By: Key

Search

Number of items in cache = 2

Flush Cache

Asset Lookup

Key	Hit Count	Created Time	Last Access Time	Dependency
<input type="checkbox"/> http://localhost:8080:/ps7/PSI1/55555	0	Tuesday, February 19, 2013 21:53:20 PM UTC	Tuesday, February 19, 2013 21:53:20 PM UTC	asset-1155737750447:Page ¹¹
<input type="checkbox"/> http://localhost:8080:/ps7/Avi/TouchHome	0	Tuesday, February 19, 2013 22:35:15 PM UTC	Tuesday, February 19, 2013 22:35:15 PM UTC	asset-1327351719456:Page ¹⁹

Table Legend

- Invalidated Page
- Updated Dependency
- Dependency Generation Count

1 of 1 page

2. At the top of the form is the Search area. Select to search by Key or by Dependency with the entered search value. Click **Search**.
3. In the displayed items in the cache, click the header title to sort the items by that criteria.

All items are shown when initially loading the URLs form. The Search form limits what is displayed, making it easier to find items in the cache.

4. To flush selected URLs from the cache, mark the checkbox on the left side of each URL. Mark the checkbox in the header to select all displayed URLs. Click **Flush Cache** to flush the memory and stores of the URLs.
5. To view the URLs for a specific dependency within the cache, mark the checkbox on the right side of each URL. Mark the checkbox in the header to mark all displayed URLs. Click **Asset Lookup** to show the dependencies.

30.7.5 Cluster Info

If Sigar is configured, this form lists the name, URL, port, and GUID of each additional WebCenter Sites cluster member.

To view the cluster information, expand **System Tools** on the **Admin** tab, then select **Cache Management**, then expand **Co-Res SS Cache** and double-click **Cluster Info**.

Figure 30–30 Cluster Info Form

Cluster Info

Number of remote cluster peer = 1

Name	Url	Port	Guid
notifier	192.0.2.201	54014	oracle-desk/192.0.2.201-3a43be2d:136c6d640a2

30.8 Log Viewer

The Log Viewer tool enables you to view, tail, download, and search the contents of the WebCenter Sites log specified in the log4j.properties file directly from the WebCenter Sites administrator's interface. Since the total size of the log affects the tool's performance, if the log size exceeds 100 megabytes, you are informed and given the choice to proceed or cancel.

This section contains the following topics:

- [Section 30.8.1, "Viewing the Log"](#)
- [Section 30.8.2, "Tailing the Log"](#)
- [Section 30.8.3, "Downloading the Log"](#)
- [Section 30.8.4, "Searching the Log"](#)

Note: The Log Viewer tool is available only if WebCenter Sites is set up to use the Apache log4j logging system and the log4j.properties file includes the file appender `FWDefaultAppender`. For information about setting up log4j, see [Section 30.1.1, "Setting Up Apache log4j."](#)

30.8.1 Viewing the Log

You can browse the entire contents of the WebCenter Sites log, starting with statements written most recently or written at a specific time.

To view the contents of the log

1. On the **Admin** tab, expand **System Tools** and double-click **Log Viewer**.

The Log Viewer form shows the most recent 100 lines in the log.

Figure 30–31 Log Viewers View Form



2. To jump to statements written at a specific time, do the following:
 - a. Click the calendar icon and select a date and time, or enter the date and time manually in the "yyyy-mm-dd hh:mm:ss" format.

Note: This feature takes into consideration only log statements that were written with the timestamping pattern currently specified in the log4j.properties file.

- b. Click **View Logs**.

The Log Viewer form shows the statements written at and/or after the specified date and time.

30.8.2 Tailing the Log

You can tail the WebCenter Sites log in a separate window. This option is particularly useful when you want to keep track of the statements written to the log while executing an action that causes an exception.

To tail the contents of the log

1. On the **Admin** tab, expand **System Tools** and double-click **Log Viewer**.
2. Click **Tail Logs**.

The most recent 100 lines in the log are displayed in a new window and are refreshed every five seconds.

30.8.3 Downloading the Log

You can download a compressed file containing the entire contents of the WebCenter Sites log.

To download the contents of the log

1. On the **Admin** tab, expand **System Tools** and double-click **Log Viewer**.
2. Click **Download Logs**.
3. Open or save the compressed file containing the WebCenter Sites log. If the log is spread over several files, each file is included.

30.8.4 Searching the Log

You can search the contents of the WebCenter Sites log for specific information.

To search the contents of the log

1. On the **Admin** tab, expand **System Tools** and double-click **Log Viewer**.

The Log Viewer form shows the most recent 100 lines in the log.

Figure 30-32 Log Viewer Form

Log Viewer: View, Search and Export Oracle WebCenter Sites Logs

Move to timestamp  Lines per page

Type search expression here...
 Case Sensitive 
 Regular Expression 

Results per page

Log Read At: Tue Mar 06 19:33:53 PST 2012.

[← Previous](#)[Scroll to Bottom](#)[Next →](#)

```
[2012-03-02 11:00:17,936] [INFO] [main] [springframework.web.context.ContextLoader] Root
[2012-03-02 11:00:17,936] [INFO] [main] [fatwire.logging.cs.event] CSShutdownListener is
[2012-03-02 11:00:17,983] [INFO] [main] [springframework.context.support.ClassPathXmlAppl
[2012-03-02 11:00:17,983] [INFO] [main] [beans.factory.xml.XmlBeanDefinitionReader] Loadi
[2012-03-02 11:00:18,576] [INFO] [main] [springframework.context.support.ClassPathXmlAppl
[2012-03-02 11:00:18,592] [INFO] [main] [beans.factory.support.DefaultListableBeanFactory
[2012-03-02 11:00:22,544] [INFO] [main] [fatwire.logging.cs.filelock] ftFileLock using ni
[2012-03-02 11:00:26,137] [INFO] [main] [beans.factory.xml.XmlBeanDefinitionReader] Loadi
[2012-03-02 11:00:27,543] [INFO] [main] [beans.factory.xml.XmlBeanDefinitionReader] Loadi
[2012-03-02 11:00:27,606] [INFO] [main] [beans.factory.xml.XmlBeanDefinitionReader] Loadi
[2012-03-02 11:00:30,527] [INFO] [main] [logging.cs.satellite.host] Remote Host: http://1
```

2. If you want to enable case-sensitive matching, select the **Case Sensitive** check box. By default, case-sensitivity covers only US-ASCII characters. If you want case-sensitivity to apply to Unicode characters, select the **Unicode Case** check box as well. The Unicode Case check box appears once the Case Sensitive check box is selected.
3. If you want the search to allow the usage of Java regular expressions, select the **Regular Expression** check box. Once the Regular Expression check box is selected, the Dot All and Multi Line check boxes appear. Use these to refine the regular expression settings:
 - If you want a dot (.) to match any characters and line terminators, select the **Dot All** check box. If this check box is not selected, a dot matches only characters.
 - If you want a caret (^) to match the beginning of a line and a dollar sign (\$) to match the end of a line, select the **Multi Line** check box.

Note: If you search with regular expressions, a single search result is restricted to a maximum of 40 lines of the log.

4. In the search field, enter the expression that you want to find in the log.
5. In the **Results per page** field, select the number of results that you want to list on each page.

6. Click **Search Logs**.

The Log Viewer form shows the lines in the log that match your search.

Figure 30–33 Log Viewer Form, After Search



7. To view a search result in its larger context, click **View log** below the result line.

100 lines of the log, starting with the line matching the search, are displayed in a new window.

30.9 File System Test

The File System Test tool enables you to test the local application server directory and the central directories in the shared file system (<cs_install_dir>/Shared) in order to compare their performance with provided benchmarks.

Table 30–3 Tested File Systems

FileSystem	Description
spc	Shared page cache directory for all WebCenter Sites cluster members. Sample path: \cs76\Shared\SystemPageCache\
local	Temporary directory for servlet context information. Sample path: \apache-tomcat-6.0.16\work\Catalina\localhost\cs
data	Shared blob data directory for all WebCenter Sites cluster members. Sample path: \cs76\Shared\ccurl

Table 30–3 (Cont.) Tested File Systems

FileSystem	Description
sync	Shared synchronization directory for all WebCenter Sites cluster members. Sample path: <code>\cs76\Shared\clustersync</code>

A test suite is composed of a number of unique tests which are performed on the local application server directory and the shared file system directories. Each of these tests executes multiple threads. Each thread creates an empty file and then writes to, reads, and deletes the file. The following test suites are available:

- **Short Test** – runs a total of 16 tests (4 tests are performed on each of the directories listed in [Table 30–3](#)).
- **Medium Test** – runs a total of 64 tests (16 tests are performed on each of the directories listed in [Table 30–3](#)).
- **Extensive Test** – runs a total of 384 tests (96 tests are performed on each of the directories listed in [Table 30–3](#)).

To test the file system

1. On the **Admin** tab, expand **System Tools** and double-click **File System Test**.
2. In the File System Test form, select the type of test you want to perform and click **Start Test**.

Figure 30–34 File System Test Form

File System Test

This tool tests the performance of the file systems and compares the results with benchmark data. The graph refreshes every 10 tests (20 for IE running an extensive test).



The test is executed on the directories named in the Performance Summary table shown in step 3. You can follow the results of the file system test as it progresses.

Note: If the test takes too long or affects performance more than is acceptable, you can click Stop Test. The results of a file system test can be compared with the provided benchmark only when the test is completed.


3. Once the test is complete, review the overall results in the Performance Summary table.

Figure 30–35 File System Test Results

File System Test

This tool tests the performance of the file systems and compares the results with benchmark data. The graph refreshes every 10 tests (20 for IE running an extensive test).

Short Test
 Medium Test
 Extensive Test

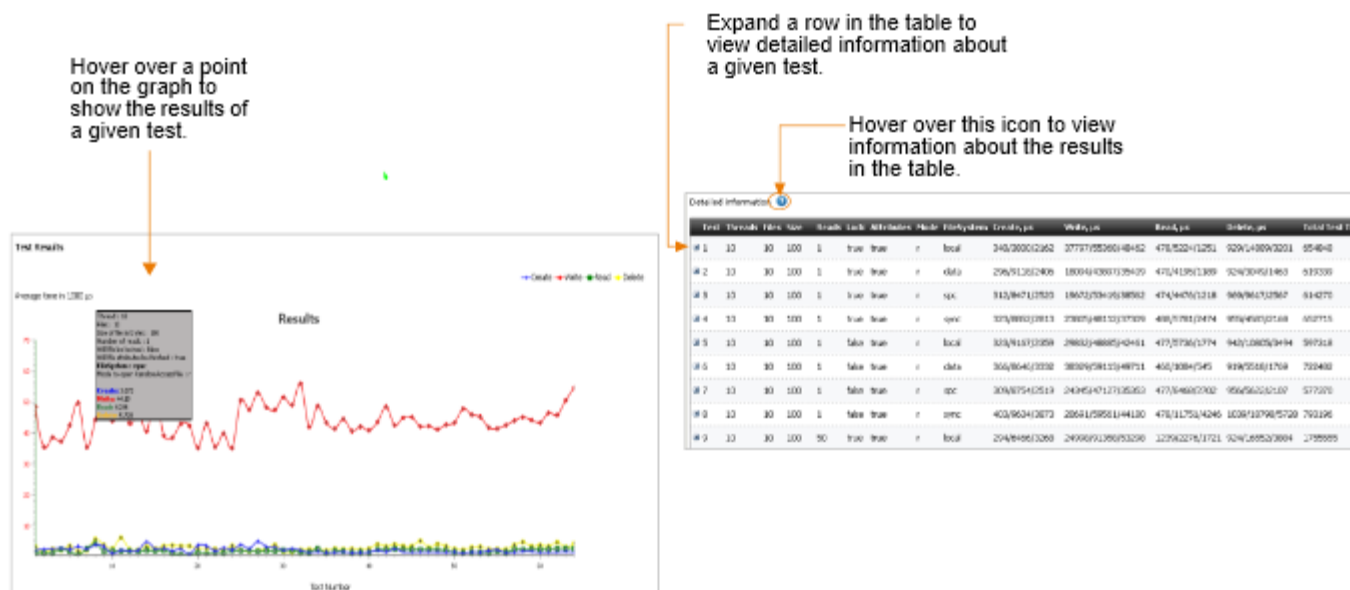
Performance Summary 

FileSystem	Path	Create, μ s	Write, μ s	Read, μ s	Delete, μ s
spc - SystemPageCache	C:\FatWire\JSK\JSK_030212-FSIIAVI-8280\ContentServer\11.1.1.6.0\Shared\SystemPageCache\	2172	45547	1800	2845
		1141	5207	1202	1017
local - Servlet Context Temp Folder	C:\FatWire\JSK\JSK_030212-FSIIAVI-8280\App_Server\apache-tomcat-6.0.32\work\Catalina\localhost\cs	2185	45287	1792	3236
		48	73	34	54
data - MungoBlobs or ccurl	C:\FatWire\JSK\JSK_030212-FSIIAVI-8280\ContentServer\11.1.1.6.0\Shared\ccurl\	2105	43938	1773	2915
		1113	5250	1159	1220
sync - Usedisksync property (futuretense.ini)	C:\FatWire\JSK\JSK_030212-FSIIAVI-8280\ContentServer\11.1.1.6.0\Shared\clustersync	2144	43308	2097	3226
		1139	5225	1172	958

The following information is shown for each directory:

- Numbers displayed in bold are benchmarks. The benchmarks were calculated by running standard tests on an Oracle Secure Linux 5.4 system connected to a 10K RPM SAS300 HD via NFS version 4. Both the client and server were on the same network switch and there was no additional traffic at the time of the test.
 - **FileSystem** specifies the directory on which the test was performed.
 - **Path** specifies the path to the directory on which a file system test was performed.
 - **Create** specifies the average number of microseconds in which a create operation was performed.
 - **Write** specifies the average number of microseconds in which a write operation was performed.
 - **Read** specifies the average number of microseconds in which a read operation was performed.
 - **Delete** specifies the average number of microseconds in which a delete operation was performed.
4. For detailed information about each test, review the graph (located under the Performance Summary table) and the test results table (located under the graph).
 - a. To view the results of a single test, either hover over a point on the graph or navigate to the desired row in the test results table.

Figure 30–36 Test Results Graph and Table



Note: Each point on the graph and row in the table shows information about a single test. However, to view the details of a single test when you hover over a point in the graph, you must be using a browser other than Internet Explorer.

The following information is shown in the table for each test:

- **Test** specifies a serial number for each test.
- **Threads** specifies the number of threads created during the test.
- **Files** specifies the number of files created per thread.
- **Size** specifies the number of bytes per file.
- **Reads** specifies the number of times each file was read by a test.
- **Lock** specifies whether each file was locked (random access lock operation).
- **Attributes** specifies whether each file's attributes were read during the test.
- **Mode** specifies the Java I/O random access mode for accessing the files in the thread.
- **FileSystem** specifies the directory on which the test was performed.
- **Create** specifies the minimum, maximum, and average number of microseconds in which the test performed a create operation.
- **Write** specifies the minimum, maximum, and average number of microseconds in which the test performed a write operation.
- **Read** specifies the minimum, maximum, and average number of microseconds in which the test performed a read operation.

- **Delete** specifies the minimum, maximum, and average number of microseconds in which the test performed a delete operation.
 - **Total Test Time** specifies the total number of microseconds in which a test was completed.
- b. In the test results table, expand a row to view detailed information about each thread that was created during a specific test.

Figure 30–37 Test Results Table

Slash separated values in the main table specify the minimum, maximum, and average values for the test.

Test	Threads	Files	Size	Reads	Lock	Attributes	Mode	FileSystem	Create, µs	Write, µs	Read, µs	Delete, µs	Total Test Time, µs
1	10	10	100	1	true	true	r	local	340/3830/2162	37797/55368/4845	478/522/4/1251	929/14886/3001	654848
2	10	10	100	1	true	true	r	date	294/2118/2-106	1000-1/13097/38-100	178/1196/1180	52-1/30-19/1163	610009

Thread	Create, µs	Write, µs	Read, µs	Delete, µs
1	1018	40205	484	1170
2	1030	29342	1205	937
3	296	18094	4195	926
4	917	43190	481	959
5	9118	35323	1902	3049
6	4326	33712	470	1647
7	1203	38529	471	524
8	3465	37584	477	1636
9	2338	33819	1715	2438
10	395	43697	490	947

The following information is shown in the thread detail table:

- **Thread** specifies a serial number for each thread.
 - **Create** specifies the total number of microseconds in which the thread created an empty file.
 - **Write** specifies the total number of microseconds in which the thread finished writing to the file.
 - **Read** specifies the total number of microseconds in which the thread read the file it created.
 - **Delete** specifies the total number of microseconds in which the thread deleted the file.
5. To download the test results as a spreadsheet, go to the top of the File System Test form and click **Download**.

30.10 Slots

Slots assets are used to store presentation changes made when editing a web site in context. Slots assets are managed in a transparent way, that is, they are not available in the user interfaces. The Slots node in the System Tools is the location to void or delete existing slot data.

A deleted slot asset has its rows removed from the database table. A voided slot asset is an asset that has had its status changed to VO. It is not removed from the database, however, it will also not show up in any searches and similar inquiries. Voided assets can be physically removed in a purge. For more information on purging voided assets, see [Section 30.12, "Purge Voided Assets."](#)

To Void Assets or Delete Assets

1. On the **Admin** tab, expand the **System Tools** node. Double-click the **Slots** node. The Site Assets Utility form is displayed.
2. The Site Assets Utility form displays the slot assets. Click one of the table headings to re-organize based on that criteria (that is, by **Slot Name**, by **Site**, by **Context**, by **Template**, or by **Primary Asset Id**). Alternately, you can search by selecting one of those criteria as a search parameter and entering a value to search.

Figure 30–38 Site Assets Utility

Slot Assets Utility

Use this utility to view and delete slot assets in the system

Search Criteria

<input type="checkbox"/>	Slot Name	Site	Context	Template	Primary Asset Id
<input type="checkbox"/>	SectionLayoutSlotLeft0	avisports	SectionLayout:1327351719380	Summary	1327351719380
<input type="checkbox"/>	SectionLayoutSlotLeft0	avisports	SectionLayout:1327351719380:green	Summary	1327351719380
<input type="checkbox"/>	SectionLayoutSlotLeft0	avisports	SectionLayout:1329326782720	Summary/Feature	1329326782720
<input type="checkbox"/>	SectionLayoutSlotLeft0	avisports	SectionLayout:1327351719525	Summary	1327351719525
<input type="checkbox"/>	SectionLayoutSlotLeft1	avisports	SectionLayout:1329326782720	Summary/Feature	1329326782720
<input type="checkbox"/>	SectionLayoutSlotLeft2	avisports	SectionLayout:1329326782720	Summary/Feature	1329326782720
<input type="checkbox"/>	SectionLayoutSlotLeft3	avisports	SectionLayout:1329326782720	Summary/Feature	1329326782720
<input type="checkbox"/>	SectionLayoutSlotRight0	avisports	SectionLayout:1327351719467	Summary/Feature	1327351719467
<input type="checkbox"/>	SectionLayoutSlotRight0	avisports	SectionLayout:1327351719525	Summary/Large	1327351719525
<input type="checkbox"/>	SectionLayoutSlotRight0	avisports	SectionLayout:1327351719380	Summary/Feature	1327351719380
<input type="checkbox"/>	SectionLayoutSlotRight0	avisports	SectionLayout:1329326782720	Summary/Highlight	1329326782720
<input type="checkbox"/>	SectionLayoutSlotRight1	avisports	SectionLayout:1329326782720	Summary	1329326782720
<input type="checkbox"/>	SectionLayoutSlotRight2	avisports	SectionLayout:1329326782720	Summary	1329326782720
<input type="checkbox"/>	SectionLayoutSlotRight3	avisports	SectionLayout:1329326782720	Summary	1329326782720
<input type="checkbox"/>	SectionLayoutSlotRight4	avisports	SectionLayout:1329326782720	Summary	1329326782720

3. Select one or more assets by marking the checkboxes.
4. Click **Void Asset** to void the selected assets (for a later purge), or click **Delete** to delete the assets.

Note: When either deleting or voiding the slot assets, no confirmation message is displayed. The list of remaining assets will redisplay.

Once an asset is voided, it doesn't appear in any lists of assets or searches for assets. However, it is still taking space in the database. In some cases, the space taken up by voided assets must be reclaimed; this is handled through purging voided assets. For information on how to purge voided assets, see [Section 30.12, "Purge Voided Assets."](#)

30.11 URL

The URL node allows you to search, view, and delete custom URLs, also known as vanity URLs. These URLs are domains used to redirect to a certain section of your website, which may or may not be accessible to the rest of the hosted website. Often these vanity URLs are used to prevent a long URL that has coded strings within it, making it easier for the web user to view your page.

To view and manage vanity URLs

1. On the **Admin** tab, expand **System Tools** and double-click **URL**.

The URL Utility form will display, showing all vanity URLs including auto-generated URLs.

2. You can limit the list by entering the search criteria for Host, URL, Template, Wrapper, or HTTP Status, then click **Search**.

Figure 30–39 URL Utility Form

URL Utility

Use this utility to view and manage URLs

Results per page Search Criteria

Page 1 of 1

<input type="checkbox"/>	Host	URL	Template	Wrapper	HTTP Status
<input type="checkbox"/>	Avi	SkiBrampton	avisports/Page/SectionLayout		200
<input type="checkbox"/>	Avi	SkiBC	avisports/Page/SectionLayoutGreen		200
<input type="checkbox"/>	Avi	m	avisports/Page/SectionLayout		200

3. The displayed vanity URLs can be sorted by clicking the column headers - Host, URL, Template, Wrapper, and HTTP Status.
4. To delete a vanity URL, mark the checkbox and click **Delete**.

30.12 Purge Voided Assets

The Purge Voided Assets node allows you to remove those assets which have been marked void. You can choose between purging all voided assets or purging those voided assets of a specific asset type.

To purge voided assets

1. Expand the **System Tools** node, then double-click **Purge Voided Assets**.

The Purge Voided Assets page opens.

2. In the Purge Voided Assets page, select the asset type, or all asset types, to purge the voided assets.

Figure 30–40 Purge Voided Assets

Purge Voided Assets

Use this utility to Purge voided assets

3. Click Purge.

If assets were purged, a confirmation message will display:

Purged *n* assets of type *asset type*

If no assets were purged, a message will display:

No voided Assets of type *asset type* found

If **All** was selected, then a message will display for each type, stating if assets were purged or if no assets were available to purge for that type.

This chapter lists and describes the WebCenter Sites system defaults that you will be using routinely.

This chapter contains the following sections:

- [Section 31.1, "ACLs"](#)
- [Section 31.2, "System ACLs"](#)
- [Section 31.3, "ACLs of Default Users"](#)
- [Section 31.4, "Required ACLs for Custom Users"](#)
- [Section 31.5, "System Roles"](#)
- [Section 31.6, "System Asset Types"](#)
- [Section 31.7, "Default Tree Tabs"](#)

31.1 ACLs

WebCenter Sites and its applications use several default ACLs to control user access to their features and functions. This section summarizes the permissions that can be specified in an ACL, and describes WebCenter Sites' default system ACLs.

This section contains the following topics:

- [Section 31.1.1, "Permissions"](#)
- [Section 31.1.2, "Accessing ACLs"](#)

31.1.1 Permissions

An ACL specifies a set of permissions. When an ACL is assigned to a database table, only the permissions specified in the ACL can be exercised on the database table. Only a user with the same ACL as the table can exercise those permissions.

[Table 31–1, "Permissions Supported by WebCenter Sites"](#) lists all the permissions that can be specified in an ACL.

Table 31–1 *Permissions Supported by WebCenter Sites*

Permissions	Bit Mask ¹	Action
Read	1	Read data from a table.
Write	2	Write information to a table.

Note: To add a row to a table, the user must have Create permission.

Table 31–1 (Cont.) Permissions Supported by WebCenter Sites

Permissions	Bit Mask ¹	Action
Create	4	Create a table.
Delete	8	Delete information from a table.
Retrieve	16	Retrieve the contents of a URL column, also known as an upload field. For information about URL columns, see the <i>Oracle Fusion Middleware WebCenter Sites Developer's Guide</i> .
Revision Tracking Audit	32	Access all the revision tracking information for the rows (records) in a tracked table.
Revision Tracking Admin	64	Assign or remove revision tracking on a table.

¹ When an ACL is created, the bit mask numbers for each permissions assigned to an ACL are added together and the totals are listed with the ACL in the `SystemACL` table.

31.1.2 Accessing ACLs

WebCenter Sites' ACLs and their permissions are accessible as either a listing or an individual entry.

- To obtain the list of ACLs and their permissions, open the `SystemACL` table directly.
- To obtain an individual ACL and its permissions, use the administrator's interface:
 1. In the **Admin** tab, expand **User Access Management** and double-click **ACLs**.
 2. In the drop-down list, select the ACL you want to work with.
 3. Select **Modify ACL** and click **OK**.

Caution: Never modify a default system ACL. Never modify the SCLs assigned to any of the system tables.

For descriptions of the system ACLs, see [Section 31.2, "System ACLs."](#)

31.2 System ACLs

[Table 31–2, "System ACLs and Their Permissions"](#) lists the system ACLs and their permissions. Each system ACL exists in order to control access to specific parts of the database tables, and subsequently, the product features that use those tables. Although several of the default ACLs have the same set of permissions, the ACLs are all necessary because they are assigned to different tables.

[Table 31–3, "System ACLs and Their Descriptions"](#) describes the functions of each ACL and how each ACL is used by WebCenter Sites and the Sites content applications.

Table 31–2 System ACLs and Their Permissions

ACL Name	Read	Retrieve	Write	Create	Delete	Rev. Track Audit	Rev. Track Admin
Browser	Yes	No	No	No	No	No	No
ContentEditor	Yes	Yes	Yes	Yes	Yes	Yes	No
ElementEditor	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 31–2 (Cont.) System ACLs and Their Permissions

ACL Name	Read	Retrieve	Write	Create	Delete	Rev. Track Audit	Rev. Track Admin
ElementReader	Yes	No	No	No	No	No	No
PageEditor	Yes	Yes	Yes	Yes	Yes	Yes	No
PageReader	Yes	No	No	No	No	No	No
RemoteClient	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SiteGod	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TableEditor	Yes	Yes	Yes	Yes	Yes	Yes	No
UserEditor	Yes	Yes	Yes	Yes	Yes	Yes	No
UserReader	Yes	No	No	No	No	No	No
Visitor	Yes	Yes	Yes	Yes	Yes	Yes	No
VisitorAdmin	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WsAdmin	No	No	No	No	No	No	No
WSEditor	No	No	No	No	No	No	No
WSUser	No	No	No	No	No	No	No
xceladmin	Yes	Yes	Yes	Yes	Yes	Yes	Yes
xceleditor	Yes	Yes	Yes	Yes	Yes	Yes	No
xcelpublish	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 31–3 System ACLs and Their Descriptions

ACL Name	Description
Browser	<p>Allows read-only access to the content in the WebCenter Sites database. It is assigned to most of the system default and sample site users.</p> <p>WebCenter Sites requires that all visitors to an online site that it manages have user accounts. For this reason, WebCenter Sites is delivered with a default user account, named DefaultReader, that it assigns to all non-authenticated visitors, that is, those who do not have a user account of their own.</p> <p>The Browser ACL is assigned to the DefaultReader user account, which gives non-authenticated visitors read-only access rights to the content in the WebCenter Sites database.</p>
ContentEditor	<p>Used in a sample WebCenter Sites site.</p> <p>This ACL is assigned to the tables that support the sample site.</p>
ElementEditor	<p>Allows users to write data to the ElementCatalog and SystemSQL tables.</p> <p>Site designers and anyone who creates templates, CSElement, and SiteEntry assets need this ACL.</p>
ElementReader	<p>Allows users to read data in the ElementCatalog and SystemSQL tables.</p> <p>WebCenter Sites users need this ACL so they can inspect the templates assigned to their assets.</p>

Table 31-3 (Cont.) System ACLs and Their Descriptions

ACL Name	Description
PageEditor	Allows users to create page entries in the SiteCatalog table. Site designers and anyone who creates a template, CSElement, or SiteEntry asset need this ACL.
PageReader	Allows users to read page entries from the SiteCatalog table. WebCenter Sites users need this ACL so they can inspect the templates assigned to their assets.
RemoteClient	Grants users the ability to log in to the Sites management system through a remote client like Sites Desktop. All Sites Desktop users need this ACL.
SiteGod	Enables complete access to all the tables in the WebCenter Sites database. At least one user of the management system, typically an administrator, must have the SiteGod ACL.
TableEditor	Allows users to create and delete tables in the WebCenter Sites database. Site designers who create database tables or who create new asset types (which causes new tables to be created) need this ACL. Administrators or anyone else who will use the Initialize Mirror Destination feature also needs this ACL.
UserEditor	Allows users to manage user accounts. Administrators need this ACL.
UserReader	Allows user account information to be recognized by WebCenter Sites. WebCenter Sites uses this ACL to determine which users have which roles on which sites. All users need this ACL to be able to access the content management sites to which they are assigned.
Visitor	Grants users the ability to write data to the Oracle WebCenter Sites: Engage tables that store visitor data, and to create recommendation assets. <ul style="list-style-type: none"> ■ Any Engage user who needs to create Recommendation assets needs this ACL. ■ Any authorized visitor (of Engage assets) whose data you are collecting on the delivery system must have this ACL assigned to their user account. ■ All unauthorized visitors of the online site are automatically assigned this ACL (in the DefaultReader user account). For more information about the DefaultReader account, see Section 6.1.2, "DefaultReader, secure.CatalogManager, and secure.TreeManager".
VisitorAdmin	Grants users the ability to create visitor attributes, history attributes, and history types. Any Engage user who needs to create assets of those types needs this ACL.
WSUser	Assigned to SiteCatalog page entries for the Web Services feature. Grants users the ability to access WebCenter Sites through the WebCenter Sites web services.
WSEditor	Assigned to SiteCatalog page entries for the Web Services feature. Grants users the ability to access WebCenter Sites through the WebCenter Sites web services.

Table 31–3 (Cont.) System ACLs and Their Descriptions

ACL Name	Description
WSAdmin	Assigned to SiteCatalog page entries for the Web Services feature. Grants users the ability to access WebCenter Sites through the WebCenter Sites web services.
xceladmin	Grants users the ability to create user profiles, roles, sites, asset types, and so on—that is, to use all the functions in the Admin, Site Admin, and Workflow tabs. System, site, and workflow administrators need this ACL. Also, because the Admin tab has both administrative and site design functions, site designers also need this ACL.
xceleditor	Grants users the ability to log in to the Sites content applications. The log in request code verifies whether or not a user has the ACL. All users of the management system need this ACL.
xcelpublish	Grants users the ability to view the Publish Console.

31.3 ACLs of Default Users

Table 31–4 Default Users and Their ACLs

User Name	Browser	Description
fwadmin	Browser	Basic administrator user that WebCenter Sites creates so that you can begin configuring your Sites content applications.
	ElementEditor	
	PageEditor	Do not delete this user unless another user with identical ACLs already exists.
	PageReader	
	RemoteClient	
	TableEditor	
	UserEditor	
	UserReader	
	Visitor	
	VisitorAdmin	
	xceladmin	
	xceleditor	
	xcelpublish	
	wsadmin	
wseditor		
wsuser		

Table 31–4 (Cont.) Default Users and Their ACLs

User Name	Browser	Description
WebCenter Sites (the installation's user account)	Browser	User account that the installation program creates during the installation of the products.
	ContentEditor	The name of this account is whatever the installers chose for it.
	ElementEditor	
	ElementReader	
	PageEditor	
	PageReader	
	SiteGod	
	TableEditor	
	UserEditor	
	UserReader	
DefaultReader	Browser	
	Visitor	The Visitor ACL is also automatically assigned.

31.4 Required ACLs for Custom Users

Table 31–5 System ACLs Required by Users

User	Required ACLs
All users	Browser, Element Reader, PageReader, UserReader, xceleditor
Workflow Administrator	xceladmin
Site Administrator	
General Administrator	xceladmin, TableEditor, UserEditor, VisitorAdmin (for Engage)
Site Designer	xceladmin, ElementEditor, PageEditor, TableEditor, Visitor (for Engage), Visitor Admin (for Engage)
Engage Users	Visitor
Users of Web Mode in the Contributor interface	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor (for Engage), xceleditor
Sites Desktop	RemoteClient, Visitor (for Engage)
Sites DocLink	

31.5 System Roles

Table 31–6 System Roles

Role	Description
GeneralAdmin	<p>Default system role for global WebCenter Sites administrators.</p> <p>Required for users who need access to the Admin tab (and all other possible functions) in the tree.</p> <p>Note: A user with the GeneralAdmin role must also have the xceladmin ACL in order to use any of the functions in the Admin tab.</p>

Table 31–6 (Cont.) System Roles

Role	Description
SiteAdmin	<p>Default system role for site administrators.</p> <p>Required for users who are administrators of selected sites and therefore need access to the Site Admin tab (which displays a subset of the functions in the Admin tab).</p> <p>Assign the SiteAdmin role to users who will manage, but not create, other site users.</p> <p>Note: A site user with the SiteAdmin role must also have the xceladmin ACL in order to use any of the functions on the Site Admin tab.</p>
WorkflowAdmin	<p>Default system role for workflow administrators.</p> <p>Required for users who need access to the Workflow tab in the tree.</p> <p>Note: A user with the WorkflowAdmin role must also have the xceladmin ACL in order to use any of the functions on the Workflow tab.</p>
AdvancedUser	Grants WebCenter Sites users access to the administrator's interface.
SitesUser	Grants WebCenter Sites users access to the content contributor's interface.

31.6 System Asset Types

Table 31–7 lists the default asset types. Unlike custom asset types, system asset types cannot be deleted.

Table 31–7 System Asset Types

Asset Type	Description
Attribute Editor	An attribute editor specifies how data is entered for a flex attribute when that attribute is displayed on a New or Edit form for a flex asset or a flex parent asset. It is similar to a template asset. However, unlike a template asset, you use it to identify the code that you want WebCenter Sites to use when it displays an attribute in the Sites interface—not when it displays the value of an attribute on your online site.
CSElement	Stores code (XML or JSP and Java) does not render assets. Typically, you use CSElements for common code that you want to call from more than one template (a banner perhaps). You also use CSElements to provide the queries that are needed to create DynamicList recommendations in Engage.
Collection	Stores an ordered list of assets of one type. You build collections by running one or more queries, selecting items from their resultsets, and then ranking (ordering) the items that you selected. This ranked, ordered list is the collection. For example, you could rank a collection of articles about politics so that the article about last night's election results is number one.
Dimension	Represents a locale in a site. You must create a Dimension asset for each locale you want to enable on the management system. To enable publishing of content in a given locale, you must publish the corresponding Dimension asset to the delivery system, and enable the locale in the site's dimension set.

Table 31–7 (Cont.) System Asset Types

Asset Type	Description
Dimension Set	Defines which locales and locale filter are enabled on the online site. For locale filtering to work on the delivery site, you must create and publish to the delivery system at least one DimensionSet asset. Has no effect on the management system.
History Attribute	Individual information types that you group together to create a vector of information that Engage treats as a single record. This vector of data is the history definition. For example, a history type called Purchases can consist of the history attributes SKU, itemname, quantity, and price. Available in Engage.
History Definition	The vector of data in a History Attribute. This vector of data is the history definition. For example, a history type called Purchases can consist of the history attributes SKU, itemname, quantity, and price. Available in Engage.
Link	Stores a URL to an external web site. You use this asset to embed an external link within another asset.
Page	Stores references to other assets. Arranging and designing page assets is how you represent the organization or design of your site. You design page assets by selecting the appropriate collections, articles, imagefiles, queries, and so on for them. Then, you position your page assets on the Site Plan tab that represents your site in the tree on the left side of the WebCenter Sites interfaces.
Promotion	Is a merchandising asset that offers some type of value or discount to your site visitors based on the flex assets (for example, products) that the visitor is buying and the segments that the visitor qualifies for. Available in Engage.
Query	Stores queries that retrieve a list of assets based on selected parameters or criteria. You use query assets in page assets, collections, and recommendations. The database query can be either written directly in the New or Edit form for the query asset as a SQL query, or written in an element (with WebCenter Sites query tags or a as a search engine query) that is identified in the New or Edit form.
Recommendation	This is like an advanced collection. It collects, assesses, and sorts flex assets (products or articles, perhaps) and then recommends the most appropriate ones for the current visitor, based on the segments that visitor belongs to. Available in Engage.
Segment	Assets that divide visitors into groups based on common characteristics (visitor attributes and history types). You build segments by determining which visitor data assets to base them on and then setting qualifying values for those criteria. For example, a segment could define people who live in Alaska and own fly fishing gear, or it could define people who bought a personal computer in the past six months, and so on. Available in Engage.
SiteEntry	Represents a WebCenter Sites page or pagelet and has a CSElement assigned as the root element that generates the page. Template assets do not have associated SiteEntry assets because they represent both an element and a WebCenter Sites page.
Template	Stores code (XML or JSP and Java) that renders other assets into WebCenter Sites pages and pagelets. Developers code a standard set of templates for each asset type (other than CSElement and SiteEntry) so that all assets of the same type are formatted in the same way. Content providers can select templates for previewing their content assets without having access to the code itself or being required to code.

Table 31–7 (Cont.) System Asset Types

Asset Type	Description
Visitor Attribute	Holds types of information that specify one characteristic only (scalar values). For example, you can create visitor attributes named Years of Experience, Job Title, or Number of Children. Available in Engage.

31.7 Default Tree Tabs

[Table 31–8](#) lists the default tabs in WebCenter Sites' tree. These tabs are critical to WebCenter Sites. All features which stem from WebCenter Sites can be accessed through these tabs; they are automatically created upon installation.

Table 31–8 Default Tabs in WebCenter Sites

Tab	Description
Active List	Displays items that are in the process of being created or edited in WebCenter Sites.
Admin	Displays the administrative functions that affect all of the CM sites in the system. By default, only users with the default system role named GeneralAdmin have access to this tab.
Design	A source for creating pages on your site. Some of these sources are: Templates, Product Definition, Content Definition, and other sources for the creating pages.
History	Displays the assets that you worked with during the current session. All users see this tab as soon as they create, inspect, edit, or copy their first asset.
Query	Enables a user to query for certain types of articles and organize them in that fashion. This is accomplished using a SQL query.
Site Admin	Holds a subset of the system-wide administrative functions. The subset applies only to the CM site that the SiteAdmin is logged in to. By default, only users with the default system role named SiteAdmin have access to this tab. This tab is useful if to individuals who manage access to individual CM sites, but who do not need to create new users or new sites.
Site Plan	Represents the layout and overview of the site. This tab shows each site that is controlled by WebCenter Sites. It lists the placed pages and the unplaced pages. The placed pages are pages which are created and have been integrated into the live site. Unplaced Pages are pages which are finished but are not integrated into the live site.
Workflow	Lists the workflow configuration functions. By default, only users with the Workflow Admin role have access to this tab.

System Data: WebCenter Sites Database

This chapter contains information about the dynamic tables in the WebCenter Sites database and how they grow. DBAs can use this information to determine how to size the WebCenter Sites database appropriately.

This chapter contains the following sections:

- [Section 32.1, "Cache Management Tables"](#)
- [Section 32.2, "Approval System Tables"](#)
- [Section 32.3, "Publishing System Tables"](#)
- [Section 32.4, "Workflow Tables"](#)
- [Section 32.5, "Basic Asset Tables"](#)
- [Section 32.6, "Flex Asset Tables"](#)
- [Section 32.7, "Visitor Tables \(Oracle WebCenter Sites: Engage\)"](#)

This chapter also contains information about how to purge inactive data from the visitor tables.

32.1 Cache Management Tables

WebCenter Sites delivers the CacheManager, a page caching utility that manages both the WebCenter Sites page cache and the Satellite Server caches.

Because cached pages need to expire both when their freshness date expires and when an asset that the page refers to in some way is changed, the CacheManager keeps track of expiration times as well as the dependencies that exist between the pages and pagelets stored in the cache. It stores this information in the `SystemPageCache` and `SystemItemCache` tables.

Every Sites system uses a CacheManager, which means that these tables grow dynamically on any system—development, management, or delivery. The cache-tracking tables grow at the following rate:

Table	Number of Rows
<code>SystemPageCache</code>	One row for every cached page. When a page expires, the row is removed.
<code>SystemItemCache</code>	One row for each asset, pagelet, or other item that is referenced by a cached page in the <code>SystemPageCache</code> table. For example, if a cached page was created from a page asset that has associations to three article assets, there would be four rows for that cached page.

32.2 Approval System Tables

The WebCenter Sites approval system keeps track of each asset that has been approved, the dependencies that approved assets have on other assets, and the targets for which assets are approved. It stores this information in the `ApprovedAssets` and `ApprovedAssetDeps` tables.

These tables have the potential to grow very large on a management system, but are not used on a delivery system. The approval system tables grow at the following rate:

Table	Number of Rows
<code>ApprovedAssets</code>	One row for each asset that has been approved, for each target destination. That is, if an asset has been approved for two destinations, that asset has two rows in this table.
<code>ApprovedAssetDeps</code>	One row for each asset dependency for each approved asset, for each target destination. For example, if an approved asset is dependent on four other assets, it has four rows in this table. If that same asset is approved to two destinations, it has one row each for each dependency for each destination.

32.3 Publishing System Tables

The WebCenter Sites publishing system keeps track of when assets were published, and where they were published to. It stores this information in the `PubKey` and the `PublishedAssets` tables.

As the number of assets in your Sites management system increases, so does the number of rows in these tables. These tables grow at the following rate:

Table	Number of Rows
<code>PubKey</code>	<ul style="list-style-type: none"> ■ For mirror publishing: one row for every asset that is mirrored, for each target destination. ■ For export publishing: one row for each page that is created during the export. That is, if 14 assets are rendered into 1 page, the table holds 1 row—not 14—for the entire group of assets.
<code>PublishedAssets</code>	<ul style="list-style-type: none"> ■ For mirror publishing: one row for every asset that is mirrored, for each target destination. ■ For export publishing: one row for each asset that is exported. To continue the preceding example, if 14 assets are rendered onto 1 page, the table holds 14 rows (one for each asset) for that page.

32.4 Workflow Tables

The workflow system keeps track of all the assets that are involved in a workflow process at any given time. It stores this information in the `Assignment` and `WorkflowObject` tables.

As the number of assets that are placed in a workflow process increases, so does the number of rows in these tables. These tables grow at the following rate:

Table	Number of Rows
Assignment	One row for each workflow assignment. For example, if an asset is assigned to four users during the course of a workflow process, that asset has four rows in the table. These rows are not deleted when the workflow process is completed for the asset.
WorkflowObjects	One row for each asset in workflow. When an asset leaves workflow, the row is deleted.

32.5 Basic Asset Tables

A basic asset type has one primary storage table. For example, the primary storage table for the article asset type is named `Article`; the primary storage table for the `HelloArticle` asset type is named `HelloArticle`.

As the number of assets of a single type increases, so does the size of the table that holds assets of that type. The primary storage table for a basic asset has one row for each asset of that type.

32.6 Flex Asset Tables

Each asset type in a flex family has several database tables. The three types of tables in any flex family that can potentially grow quite large are as follows:

- The primary storage table for the **flex asset** type. For example, the primary storage table for the Avisports sample site asset type named `article` is `AVIArticle`.
- The `_AMap` tables for flex asset or flex parent asset types. (For example, `AVIArticle_AMap`.)
- The `_Group` tables for flex parent asset types. (For example, `ArticleCategory_Group`.)
- The `_Mungo` table for the **flex asset** type. (For example, `AVIArticle_Mungo`.)
- The `_Mungo` table for the flex **parent** asset type. (For example, `ArticleCategory_Mungo`.)
- The `Mungo_Blobs` table

These types of tables grow at the following rate:

Table	Number of Rows
<i>FlexAssetType</i> (for example: <code>AVIArticle</code>)	One row for every asset of this type.
<i>FlexAssetType_AMap</i> (for example: <code>AVIArticle_AMap</code>)	One row per attribute value—whether the attribute value is inherited or directly assigned—for each of the assets of that type.
<i>FlexAssetType_Group</i> (for example: <code>ArticleCategory_Group</code>)	One row per ancestor relationship between flex parent asset and flex asset—includes rows for grandparent, great-grandparent, and so on, relationships.
<i>FlexAssetType_Mungo</i> (for example: <code>AVIArticle_Mungo</code>)	One row for every attribute value for each asset of this type. In some cases, this equation can result in more than 10 million rows.

Table	Number of Rows
FlexParent_Mungo(for example: ArticleCategory_Mungo)	One row for every attribute value for every parent asset of this type.
Mungo_Blobs	One row for every attribute value saved for an attribute of type blob.

32.7 Visitor Tables (Oracle WebCenter Sites: Engage)

Oracle WebCenter Sites: Engage captures visitor information and stores it in the visitor data tables. These tables store information such as session IDs for visitors so that they can be linked with their previous sessions and values for the attributes that represent the data you are collecting.

As the number of visitors who visit your online site increases, so do the rows in these tables. These tables grow at the following rate:

Table	Number of Rows
scratch	One row for each visitor context session object that is created for a visitor. Visitor context session objects are things like promotion lists, segment lists, shopping carts, and so on. There are at least five rows added to this table for each visitor in each session.
VMVISITOR	One row for each visitor for each browser session. Engage creates a unique visitor ID for each visitor for each session.
VMVISITORALIAS	In database-centric visitor tracking, one row is created for each visitor for each browser session, even for returning visitors. This method creates <i>one row per visit</i> . Memory-centric visitor tracking creates <i>one row per visitor</i> , regardless of the number of returning visits. Note: A row holds the name/value pair of an alias and the visitor ID (also listed in VMVISITOR) that marks a session.
VMVISITORSCALARVALUE	One row for each visitor attribute value (except for attributes of type binary) that is saved. Note: This table is not populated when memory-based visitor tracking is enabled.
VMVISITORSCALARBLOB	One row for each visitor attribute value of type binary (also referred to as scalar objects) that is saved. Note: This table is not populated when memory-based visitor tracking is enabled.
VMz -----	These are dynamically generated tables that are created when values for a history attribute are saved. Engage creates one table for each history type and adds a row to the table each time a record of that type is saved.

This section contains the following topics:

- [Section 32.7.1, "Managing the Attribute Tables"](#)
- [Section 32.7.2, "Managing the Session Objects Table \(scratch\)"](#)
- [Section 32.7.3, "Deleting Unnecessary .class Files"](#)

32.7.1 Managing the Attribute Tables

Because the tables that hold attribute values can grow very quickly, you should purge inactive data from them regularly.

You use the following Engage XML object method or its JSP equivalent to delete inactive data from these tables:

```
<VDM.FLUSHINACTIVE STARTDATE="cutoffDate" />
```

Inactive visitor data is data marked with a visitor ID that is not connected through an alias to data that you consider current. You set a cutoff date (*STARTDATE*) for Engage to use. All visitor data recorded before that date is deleted from the previously listed visitor tables unless it is linked through an alias with data recorded after that date.

There are several ways to use the *VDM.FLUSHINACTIVE* tag. For example:

- You can create an administrative element that invokes the tag and prompts you to enter the cutoff date.
- You can provide it with an equation that calculates a cutoff date based on some parameter so that you can set it up to run as an automatic event at a regularly scheduled time. To set it up as an automatic event, use the WebCenter Sites *APPEVENT* tag (which functions like a *kron* job). For information about this tag, see the *Oracle Fusion Middleware WebCenter Sites: Tag Reference*, and the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Note that the value that you pass to the *STARTDATE* parameter must be in epoch time. You can use the WebCenter Sites *DATE.CONVERT* tag to obtain an epoch value for the date that you want to use. For example:

```
<DATE.CONVERT VARNAME="flushtime"
  YEAR="four digit year" MONTH="number in the range 1-12"
  DAY="number in the range 1-31"
  [HOURL="number in the range 0-11 where 0 is midnight" AMPM="am
  or pm" MINUTE="number in the range 0-59" TIMEZONE="timezone"]/>
<VDM.FLUSHINACTIVE STARTDATE="Variables.flushtime" />
```

For more information about these tags, see the *Oracle Fusion Middleware WebCenter Sites: Tag Reference*.

32.7.2 Managing the Session Objects Table (scratch)

The *scratch* table can grow quickly because there are at least five session objects stored for each visitor in each session. Each object has a timestamp; you should purge old objects regularly, based on their timestamps.

You use the following Engage XML object method or its JSP equivalent to delete old session objects from the *scratch* table:

```
<SESSIONOBJECTS.FLUSH TIMESTAMP="cutoffTime" />
```

Because session objects includes carts, you must set the cutoff time to represent the point at which you consider a cart to be abandoned.

Note: This object method does not affect the *VMSCALARBLOB* table, which means that it does not delete carts that you have stored as a visitor attribute of type binary (a scalar object).

Note that the value that you pass to the `STARTDATE` parameter must be in epoch time. You can use the WebCenter Sites `DATE.CONVERT` tag to obtain an epoch value for the date that you want to use. For an example, see the code example for the `VDM.FLUSHINACTIVE` tag provided in the preceding section.

For more information about these tags, see the *Oracle Fusion Middleware WebCenter Sites: Tag Reference*.

32.7.3 Deleting Unnecessary .class Files

Oracle WebCenter Sites: Engage is a Java-based application and it generates Java `.class` files each time one of the following events occurs:

- A segment, recommendation, or promotion is created.
- A product is configured for a related items recommendation.
- A segment, recommendation, or promotion is calculated or invoked by Engage.

Typically, old `.class` files are deleted when a segment, recommendation, promotion, or product is updated and are then replaced with new `.class` files. However, if the segment, recommendation, product, or promotion is in use when an updated version is published, Engage cannot delete the old `.class` file because it is locked.

The old `.class` files can build up, filling up the disk and using memory. Therefore, depending on how much development work you are doing and how frequently you publish to the delivery system, you must manually delete these `.class` files at a regularly scheduled time.

Complete the following steps to delete old `.class` files:

1. Use the Property Editor to examine the `vis.genclasspath` property and note the directory name designated by that parameter. This is the directory where Engage stores `.class` files.
2. During a quiet time on your site, shut down and restart each instance of JRE runtime that is running. This process releases any old `.class` files that the JRE runtime has locked.
3. Using any file management tool, navigate to the directory that holds the `.class` files and delete the contents of this directory.

Engage regenerates any `.class` files that it needs when it needs them.

Managing Users, Sites, and Roles in LDAP-Integrated Sites Systems

This chapter is for WebCenter Sites administrators who are managing users, ACLs, sites and roles from the administrator's interface of an LDAP-integrated Sites system and need to know how their operations affect both the Sites database and the LDAP server. This chapter summarizes:

- Which operations are propagated to the LDAP server (given the type of LDAP server, the operation, and whether you are connected as user with read-only or read-write permissions)
- Which operations produce errors
- How to correct errors

This chapter contains the following sections:

- [Section 33.1, "Overview"](#)
- [Section 33.2, "User Management Operations"](#)
- [Section 33.3, "Site and Role Management Operations"](#)

33.1 Overview

WebCenter Sites can be integrated with LDAP servers that use the LDAP-2 protocol or any other protocol (although, in the latter case, write operations from the administrator's interface to the LDAP server are not supported).

This overview summarizes

- the types of LDAP schema that WebCenter Sites supports,
- operations that can be performed from the WebCenter Sites interface when each type of schema is deployed, and
- the outcome of the operations, given the permissions of the WebCenter Sites administrator to the LDAP server and the nature of the LDAP server.

This section contains the following topics:

- [Section 33.1.1, "LDAP Schema"](#)
- [Section 33.1.2, "LDAP Connectivity for Site and Role Management"](#)
- [Section 33.1.3, "LDAP Users and Their Permissions to LDAP Servers"](#)
- [Section 33.1.4, "LDAP-Integrated Operations"](#)

33.1.1 LDAP Schema

The following LDAP-schema scenarios are covered in this chapter: flat- and hierarchical -schema LDAP. Each schema supports selected operations and installations:

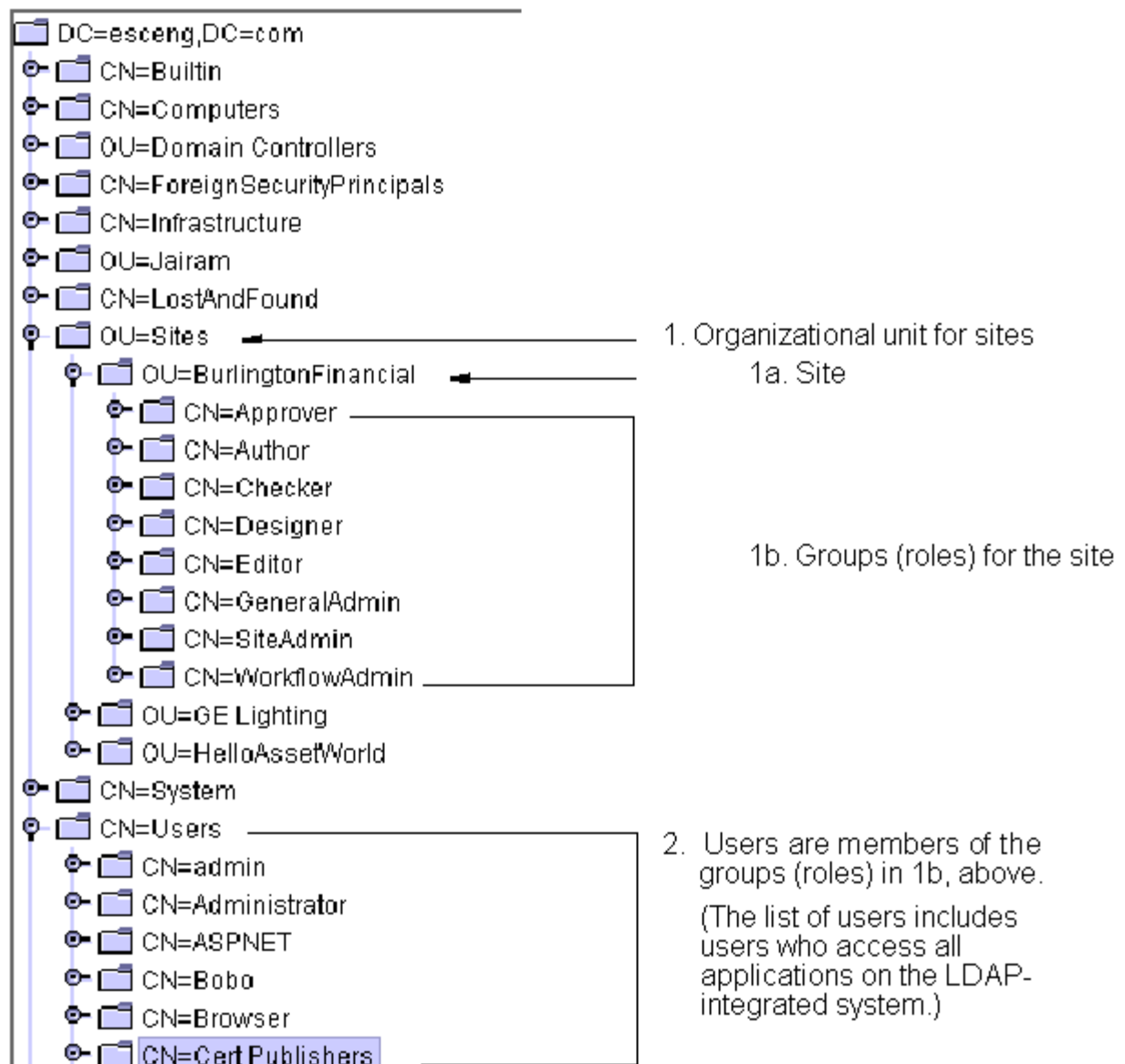
- Both flat- and hierarchical-schema LDAP support operations on ACLs and users (user accounts, user profiles, and user attributes) in WebCenter Sites.
- Flat-schema LDAP supports operations on sites and roles in WebCenter Sites and levels the sites and roles so that no hierarchy exists among them.
- Hierarchical-schema LDAP supports operations on sites and roles in WebCenter Sites. This schema requires LDAP users to define a site organizational unit in the LDAP server in which roles must be subordinated to their relevant sites (for an example, see [Figure 33–1, "LDAP Hierarchies"](#)).

[Table 33–1, "LDAP Integration Scenarios"](#) summarizes LDAP schema and the possible operations. Using the correct schema, the WebCenter Sites administrator can perform operations in the WebCenter Sites interface and have them propagate to the WebCenter Sites database, or the LDAP server, or both, depending also on the conditions that are outlined in [Section 33.1.2, "LDAP Connectivity for Site and Role Management."](#)

Table 33–1 LDAP Integration Scenarios

Create/Modify/Delete Operations on...	Flat-Schema LDAP	Hierarchical-Schema LDAP
ACLs, Users, User Profiles, User Attributes	Table 33–2	Table 33–2
Sites and Roles	Table 33–3	Table 33–4

Figure 33-1 LDAP Hierarchies



33.1.2 LDAP Connectivity for Site and Role Management

For operations on roles and sites to work as described in this chapter, LDAP connectivity must be enabled by setting the values of two properties in the `futuretense_xcel.ini` file:

```
xcelerate.usermanagerclass
xcelerate.rolemanagerclass
```

These properties allow you to manage sites and roles directly in the LDAP server (as shown in this appendix) or exclusively in the WebCenter Sites database.

- If the property values specify LDAP, then LDAP connectivity is established. Sites and roles can be managed in the LDAP server (and in the WebCenter Sites database).

- For hierarchical-schema LDAP, only the `xcelerate.usermanagerclass` property must specify LDAP. The `xcelerate.rolemanagerclass` property uses the default value. For example:

```
xcelerate.usermanagerclass=com.openmarket.xcelerate.user.LDAPSchemaUserManager
xcelerate.rolemanagerclass=com.openmarket.xcelerate.roles.RoleManager
```

- For flat-schema LDAP, both properties must specify LDAP. For example:

```
xcelerate.usermanagerclass=com.openmarket.xcelerate.user.FlatLDAPSchemaUserManager
xcelerate.rolemanagerclass=com.openmarket.xcelerate.roles.FlatLDAPSchemaRoleManager
```

- If the property values do not specify LDAP (for example, `xcelerate.usermanagerclass=com.openmarket.xcelerate.user.usermanager`) then sites and roles can be managed only in the WebCenter Sites database.

33.1.3 LDAP Users and Their Permissions to LDAP Servers

In an LDAP-integrated Sites system, the administrator of the WebCenter Sites system may or may not be an administrative user of the LDAP server, depending on the value of the `jndi.connectAsUser` property in the `dir.ini` file. The value determines how the WebCenter Sites administrator is connected to the LDAP server, and therefore defines the LDAP user:

- If `jndi.connectAsUser` is set to `true`, then WebCenter Sites defines the LDAP user to be the same one that is logged in to WebCenter Sites and connects that user to the LDAP server.
- If `jndi.connectAsUser` is set to `false`, then WebCenter Sites defines the LDAP user to be the one that is specified in the `jndi.login` property (in `dir.ini`) and connects that user to the LDAP server.

For the connection to take place, the user and his permissions must also be defined in the LDAP server. If the user has read-only permissions to the LDAP server, he is not an administrative LDAP user. If the user has read and write permissions to the LDAP server, he is an administrative LDAP user (or simply, an LDAP administrator).

Note: For more information about LDAP-related properties, see the *Oracle Fusion Middleware WebCenter Sites Property Files Reference*.

33.1.4 LDAP-Integrated Operations

For the WebCenter Sites administrator to successfully perform an operation (such as creating an ACL in both the WebCenter Sites database and the LDAP server), it is critical for WebCenter Sites to be properly integrated with the LDAP server. Barring integration issues, the outcome of an operation depends on the following factors:

1. The LDAP user, as defined by the `jndi.connectAsUser` property (see the previous section).
2. The LDAP user's permissions, as defined in the LDAP server.

If the WebCenter Sites administrator is connected to the LDAP server as a user without administrative rights, his operations (such as deleting an ACL from the WebCenter Sites database) cannot be written to the LDAP server. The system responds by either writing the operations to the WebCenter Sites database, or not

at all. The operations must then be repeated in the administrator's interface and performed manually in the LDAP server by an LDAP administrator.

3. The nature of the operation.

Certain operations that are performed in the administrator's interface (such as editing an ACL) are not written to the LDAP server, even when the WebCenter Sites administrator is connected with write permissions.

4. Whether the LDAP server supports the LDAP-2 protocol.

If the LDAP server does not support the LDAP-2 protocol, then write operations from the administrator's interface to the LDAP server are not supported, and the result of an operation cannot always be predicted.

Table 33–2 through Table 33–4 summarize the results of operations that can be performed by a WebCenter Sites administrator who is connected to the LDAP server as a user with read-only and read/write permissions.

33.2 User Management Operations

Table 33–2, "System Response to User Management Operations in WebCenter Sites" applies to flat- and hierarchical-schema LDAP web environments and summarizes system response to user management operations. The operations are performed by a WebCenter Sites administrator using the interfaces of three WebCenter Sites systems, each integrated with one of the following LDAP options:

- LDAP-2 server with read-only permissions for the LDAP user (defined on page 33-3)
- LDAP-2 server with write permissions for the LDAP user
- LDAP server other than LDAP-2, in which case write operations are not supported

The results of each user management operation are described on the pages that are noted in the left-hand column of Table 33–2.

Note: The term "user management" in this appendix includes the management of ACLs, user accounts, user profiles, and user attributes.

Table 33–2 System Response to User Management Operations in WebCenter Sites

Operation in WebCenter Sites	Result	System Response with LDAP-2: User has Read-Only Permissions ¹	System Response with LDAP-2: User has Write Permissions ²	System Response with Other LDAP Server: Write not Supported ³
Creating an ACL (on page 33-7)	<i>ACL is created in:</i>	Returns error ⁴	LDAP + Sites database	Unpredictable
Editing an ACL (on page 33-7)	<i>ACL is modified in:</i>	Sites database	Sites database	Sites database
Deleting an ACL (on page 33-7)	<i>ACL is deleted in:</i>	Returns error	LDAP + Sites database	Unpredictable
Assigning ACLs to Tables (on page 33-8)	<i>ACL is assigned in:</i>	Sites database	Sites database	Sites database

Table 33–2 (Cont.) System Response to User Management Operations in WebCenter Sites

Operation in WebCenter Sites	Result	System Response with LDAP-2: User has Read-Only Permissions ¹	System Response with LDAP-2: User has Write Permissions ²	System Response with Other LDAP Server: Write not Supported ³
Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries) (on page 33-8)	<i>ACL is assigned in:</i>	Sites database	Sites database	Sites database
Creating a User (Granting ACLs for Access Privileges) (on page 33-8)	<i>User is created in:</i>	Returns error	LDAP	Unpredictable
Editing a User (on page 33-9)	<i>User is modified in:</i>	Returns error	LDAP	Unpredictable
Deleting a User (on page 33-9)	<i>User is deleted from:</i>	Returns error	LDAP	Unpredictable
Creating/Editing a User Profile (on page 33-10)	<i>User profile is created/edited in:</i>	Returns error	LDAP	Unpredictable
Deleting a User Profile (on page 33-10)	<i>User profile is deleted from:</i>	Returns error	LDAP	Unpredictable
Creating User Attributes (on page 33-10)	<i>User profile is created in:</i>	Returns error	LDAP	Unpredictable
Editing/Deleting User Attributes (on page 33-11)	<i>User profile is deleted from:</i>	Returns error	LDAP	Unpredictable

¹ ACLs and users cannot be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user without write permissions).

² ACLs and users can be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user with write permissions).

³ The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP servers, write operations from the administrator's interface to the LDAP server are not supported.

⁴ "Returns error" means that the operation is not performed (the system returns an error).

This section contains the following topics:

- [Section 33.2.1, "Creating an ACL"](#)
- [Section 33.2.2, "Editing an ACL"](#)
- [Section 33.2.3, "Deleting an ACL"](#)
- [Section 33.2.4, "Assigning ACLs to Tables"](#)
- [Section 33.2.5, "Assigning ACLs to WebCenter Sites Pages \(Site Catalog Page Entries\)"](#)
- [Section 33.2.6, "Creating a User \(Granting ACLs for Access Privileges\)"](#)
- [Section 33.2.7, "Editing a User"](#)
- [Section 33.2.8, "Deleting a User"](#)
- [Section 33.2.9, "Creating/Editing a User Profile"](#)
- [Section 33.2.10, "Deleting a User Profile"](#)
- [Section 33.2.11, "Creating User Attributes"](#)

- [Section 33.2.12, "Editing/Deleting User Attributes"](#)

33.2.1 Creating an ACL

Operation

The WebCenter Sites administrator attempts to create an ACL by using the **User Access Management** node on the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot create the ACL. The LDAP administrator must first create the ACL manually in the LDAP server and then in the WebCenter Sites database.
2. LDAP-2 Server with Write Permissions—The system creates the ACL in both the LDAP server and the WebCenter Sites database.
3. Not LDAP-2: Write not Supported—Response is unpredictable. The LDAP administrator must first create the ACL in the LDAP server and then in WebCenter Sites.

33.2.2 Editing an ACL

Operation

The WebCenter Sites administrator attempts to modify an ACL by using the **User Access Management** node on the **Admin** tab.

Note: The WebCenter Sites administrator can modify either the description of the ACL, or the permission associated with the ACL, but not the ACL's name.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system modifies the ACL in the WebCenter Sites database. The modification is stored only in the database, even with LDAP integration.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.2.3 Deleting an ACL

Operation

The WebCenter Sites administrator attempts to delete an ACL by using the **User Access Management** node on the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot delete the ACL. The LDAP administrator must first delete the ACL from the LDAP server and then from the WebCenter Sites database.
2. LDAP-2 Server with Write Permissions—The system deletes the ACL from the LDAP server and from the WebCenter Sites database.

3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must first delete the ACL from the LDAP server before attempting to delete the ACL from WebCenter Sites.

33.2.4 Assigning ACLs to Tables

Operation

The WebCenter Sites administrator attempts to assign an ACL to system tables.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system assigns the ACLs to the database tables. If the ACLs cannot be assigned, the system returns an error message.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.2.5 Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries)

Operation

The WebCenter Sites administrator attempts to assign an ACL to WebCenter Sites Pages.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system adds the selected ACLs for the specified WebCenter Sites Page. If the ACLs cannot be assigned, the system returns an error message.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.2.6 Creating a User (Granting ACLs for Access Privileges)

Operation

The WebCenter Sites administrator attempts to create a user by using the **User Access Management** node on the **Admin** tab.

Note: WebCenter Sites displays only the ACLs that are stored in its database.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot add the user and the user's assigned ACLs. The LDAP administrator must use the LDAP server to manually add the user to the LDAP server and assign ACLs to the user.
2. LDAP-2 Server with Write Permissions—The system creates the user in the LDAP server and assigns ACLs to the user.

3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must use the LDAP server to manually add the user to the LDAP server and assign ACLs to the user.

33.2.7 Editing a User

Operation

The WebCenter Sites administrator attempts to modify the ACLs that are assigned to a user or to change the user's password by using the **User Access Management** node on the **Admin** tab.

Note: The Modify user operation allows you to change only the user password and the ACLs that are associated with the user.

The Modify user attributes operation allows you to change all attributes that are stored in the database or in the LDAP server.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot modify the user's password and/or ACLs. The LDAP administrator must manually modify the user in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system modifies the user's password and/or assigned ACLs in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually modify the user in the LDAP server.

33.2.8 Deleting a User

Operation

The WebCenter Sites administrator attempts to delete a user by using the **User Access Management** node on the **Admin** tab.

Before deleting a user, the administrator must delete the user profile. If a user profile exists, an error message is triggered. The behavior described below is expected when no user profile exists for the user being deleted.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot delete the user. The LDAP administrator must manually delete the user from LDAP server.
2. LDAP-2 Server with Write Permissions—The system deletes the user from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually delete the user from the LDAP server.

33.2.9 Creating/Editing a User Profile

Operation

The WebCenter Sites administrator attempts to create or modify a user profile by using the **User Profiles** node on the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot create/modify the user's profile and/or Locale Preference. The LDAP administrator must manually create/modify the user profile in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system creates/modifies the user's profile and/or locale preference in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually create/modify the user profile in the LDAP server.

33.2.10 Deleting a User Profile

Operation

The WebCenter Sites administrator attempts to delete a user profile by using the **User Profiles** node on the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot delete the user's profile. The LDAP administrator must manually delete the user profile in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system deletes the user's profile from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually delete the user profile from the LDAP server.

33.2.11 Creating User Attributes

Note: User attributes are stored in only one place—either in the database or in LDAP.

Operation

The WebCenter Sites administrator attempts to create new attributes for the user from the **User Access Management** node of the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot add the new attributes to the LDAP server. The LDAP administrator must manually add user attributes to the LDAP server.
2. LDAP-2 Server with Write Permissions—The system adds the user's new attributes to the LDAP server.

3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually add user attributes to the LDAP server.

33.2.12 Editing/Deleting User Attributes

Operation

The WebCenter Sites administrator attempts to modify the user attributes from the **User Access Management** node of the **Admin** tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot modify or delete the user's attributes from the LDAP server. The LDAP administrator must manually modify user attributes in the LDAP server and delete user attributes from the LDAP server.
2. LDAP-2 Server with Write Permissions—The system modifies the user attributes in the LDAP server and deletes them from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually modify user attributes in the LDAP server, and manually delete user attributes from the LDAP server.

33.3 Site and Role Management Operations

This section covers site and role management operations with flat- and hierarchical-schema LDAP servers.

This section contains the following topics:

- [Section 33.3.1, "Operations with Flat-Schema LDAP Servers"](#)
- [Section 33.3.2, "Operations with Hierarchical-Schema LDAP Servers"](#)
- [Section 33.3.3, "Creating a Site"](#)
- [Section 33.3.4, "Editing a Site"](#)
- [Section 33.3.5, "Deleting a Site"](#)
- [Section 33.3.6, "Granting Users Access to Sites"](#)
- [Section 33.3.7, "Removing Users' Access to Sites"](#)
- [Section 33.3.8, "Creating a Role"](#)
- [Section 33.3.9, "Editing a Role"](#)
- [Section 33.3.10, "Deleting a Role"](#)

33.3.1 Operations with Flat-Schema LDAP Servers

[Table 33–3, "System Response to Site and Role Management Operations with Flat-Schema LDAP"](#) summarizes system response to site and role management operations. The operations are performed by a WebCenter Sites administrator using the interfaces of three WebCenter Sites systems, each integrated with one of the following LDAP options:

- An LDAP-2 server with read-only permissions for the LDAP user (defined on page 33-3)
- An LDAP-2 server with write permissions for the LDAP user

- An LDAP server other than LDAP-2, in which case write operations are not supported

The results of each operation are described on the pages that are noted in the left-hand column of [Table 33-3](#).

Table 33-3 System Response to Site and Role Management Operations with Flat-Schema LDAP

Operation in WebCenter Sites	Results	System Response with Flat-Schema LDAP-2: User has Read Only Permissions ¹	System Response with Flat-Schema LDAP-2: User has Write Permissions ²	System Response with Flat-Schema LDAP Server (Other Than LDAP-2): Write Not Supported ³
Creating a Site (on page 33-13)	<i>Site is created in:</i>	Sites database	Sites database	Sites database
Editing a Site (on page 33-14)	<i>Site is edited in:</i>	Sites database	Sites database	Sites database
Deleting a Site (on page 33-14).	<i>Site is deleted from:</i>	Sites database	Sites database	Sites database
Creating a Role (on page 33-16)	<i>Role is created in:</i>	Returns error ⁴	LDAP + Sites database ⁵	Unpredictable
Editing a Role (on page 33-16)	<i>Role is edited in:</i>	Sites database	Sites database	Sites database
Deleting a Role (on page 33-17)	<i>Role is deleted from:</i>	Returns error	LDAP + Sites database	Unpredictable
Granting Users Access to Sites (on page 33-15)	<i>Access is granted in:</i>	Returns error	LDAP	Unpredictable
Removing Users' Access to Sites (on page 33-15)	<i>Access is removed from:</i>	Returns error	LDAP	Unpredictable

¹ Sites and roles cannot be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user without write permissions).

² Sites and roles can be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user with write permissions).

³ The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP server, write operations from the administrator's interface to the LDAP server are not supported.

⁴ "Returns error" means that the operation is not performed (the system returns an error).

⁵ For a role to be created in both the LDAP server and the Sites database, at least one site must exist in the LDAP server. If no site exists, the role is created only in the Sites database.

33.3.2 Operations with Hierarchical-Schema LDAP Servers

[Table 33-4, " System Response to Site and Role Management Operations with Hierarchical-Schema LDAP \(WebCenter Sites Web Applications\)"](#) summarizes system response to site and role management operations. The operations are performed by a WebCenter Sites administrator using the interfaces of three WebCenter Sites systems, each integrated with one of the following LDAP options:

- An LDAP-2 server with read-only permissions for the LDAP user (defined on page 33-3)
- An LDAP-2 server with write permissions for the LDAP user
- An LDAP server other than LDAP-2, in which case write operations are not supported

The results of each operation are described on the pages that are noted in the left-hand column of [Table 33-4](#).

Table 33-4 System Response to Site and Role Management Operations with Hierarchical-Schema LDAP (WebCenter Sites Web Applications)

Operation in WebCenter Sites	Results	System Response with Hierarchical-Schema LDAP-2: User has Read-Only Permissions ¹	System Response with Hierarchical-Schema LDAP-2: User has Write Permissions ²	System Response with Hierarchical-Schema LDAP Server (Other Than LDAP-2): Write Not Supported ³
Creating a Site (on page 33-13)	<i>Site is created in:</i>	Sites database	Sites database	Sites database
Editing a Site (on page 33-14)	<i>Site is edited in:</i>	Sites database	Sites database	Sites database
Deleting a Site (on page 33-14)	<i>Site is deleted in:</i>	Sites database	Sites database	Sites database
Creating a Role (on page 33-16)	<i>Role is created in:</i>	Sites database	Sites database	Sites database
Editing a Role (on page 33-16)	<i>Role is edited in:</i>	Sites database	Sites database	Sites database
Deleting a Role (on page 33-17)	<i>Role is deleted in:</i>	Sites database	Sites database	Sites database
Granting Users Access to Sites (on page 33-15)	<i>Access is granted in:</i>	Returns error ⁴	LDAP	Unpredictable
Removing Users' Access to Sites (on page 33-15)	<i>Access is removed from:</i>	neither the Sites database nor the LDAP server	neither the Sites database nor the LDAP server	neither the Sites database nor the LDAP server

¹ Sites and roles cannot be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user without write permissions).

² Sites and roles can be stored in the LDAP server (the WebCenter Sites administrator is connected to the LDAP server as a user with write permissions).

³ The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP servers, write operations from the administrator's interface to the LDAP server are not supported.

⁴ "Returns error" means that the operation is not performed (the system returns an error).

33.3.3 Creating a Site

Operation

The WebCenter Sites administrator attempts to create a site by using the **Sites** node on the **Admin** tab.

Note: If you are manually creating a site in an LDAP-integrated system, you must ensure that the database and the LDAP server are synchronized. Otherwise, sites will not be properly listed in the WebCenter Sites interfaces.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system creates the site in the WebCenter Sites database. The LDAP administrator must create the site entry in the LDAP server.

2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP server with Read-Only Permissions—The system creates the site in the WebCenter Sites database. The administrator must create an entry for all available roles by prefixing *each* site name to the role name in the LDAP server.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.3.4 Editing a Site

Operation

The WebCenter Sites administrator attempts to edit the description of a site by using the **Site Edit** option on the **Admin** tab.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the site description in the WebCenter Sites database. (The site description is stored only in the WebCenter Sites database, even with LDAP integration.)
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the site description in the WebCenter Sites database. (The site description is stored only in the WebCenter Sites database, even with LDAP integration.)
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.3.5 Deleting a Site

Operation

The WebCenter Sites administrator attempts to delete the description of the site by using the **Site Edit** node on the **Admin** tab.

Note: If you are manually deleting a site with LDAP integration, you must ensure that the database and the LDAP server are synchronized. Otherwise, sites will not be properly listed in the WebCenter Sites interfaces.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the site from the WebCenter Sites database. The LDAP administrator must manually delete the site entry from the LDAP server.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).

3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the site from the WebCenter Sites database. The LDAP administrator must manually delete the site entry from the LDAP server.
2. LDAP-2 Server with Write permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.3.6 Granting Users Access to Sites

Operation

The WebCenter Sites administrator attempts to grant users access to a site by using the **Users** node on the **Admin** tab.

Note: WebCenter Sites displays only the roles that are stored in its database.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message because, in the LDAP server, it cannot assign any roles to the specified users of the selected site. The LDAP administrator must manually assign the roles.
2. LDAP-2 Server with Write Permissions—The system assigns roles, within the LDAP server, to the specified users of the selected site.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message because, in the LDAP server, it cannot assign any roles to the specified users of the selected site. The LDAP administrator must manually assign the roles.
2. LDAP-2 Server with Write Permissions—The system assigns roles, within the LDAP server, to the specified users of the selected site.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

33.3.7 Removing Users' Access to Sites

Operation

The WebCenter Sites administrator attempts to remove user access to sites by using the **Users** node on the **Admin** tab.

Note: WebCenter Sites displays only the sites that are stored in its interface.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The LDAP administrator must manually remove the user's permissions to the sites, directly in the LDAP server.

2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message. The LDAP administrator must manually remove the user's permissions to the sites, directly in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system removes the user's permissions to the sites, directly in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

33.3.8 Creating a Role

Operation

The WebCenter Sites administrator attempts to create a role by using the **Roles** node on the **Admin** tab.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system creates the role in the WebCenter Sites database. The LDAP administrator must manually create the role in the LDAP server.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot create the role in the LDAP server or in the WebCenter Sites database. The LDAP administrator must create the role in the LDAP server. The same role must be re-created in the WebCenter Sites database.
2. LDAP-2 Server with Write Permissions—The system creates the role in the WebCenter Sites database. The system also creates the role for all available sites in the LDAP server by pre-fixing the name of each existing site to the name of the role (*SiteA-Role*, *SiteB-Role*, and so on).

Note: For a role to be created in both the LDAP server and the WebCenter Sites database, at least one site must exist in the LDAP server. If no sites exist, the role is created only in the WebCenter Sites database.

3. Not LDAP-2: Write not Supported—Behavior is unpredictable. An error message can be returned.

33.3.9 Editing a Role

Operation

The WebCenter Sites administrator attempts to modify the description of a role by using the **Roles** node on the **Admin** tab.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the description of the role in the WebCenter Sites database. (The role description is stored only in the WebCenter Sites database, even with LDAP integration.)
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the description of the role in the WebCenter Sites database. (The role description is stored only in the WebCenter Sites database, even with LDAP integration.)
2. LDAP-2 Server with Read-Only Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

33.3.10 Deleting a Role

Operation

The WebCenter Sites administrator attempts to delete a role by using the **Roles** node on the **Admin** tab.

System Response - Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the role from the WebCenter Sites database. The LDAP administrator must manually delete this role from the LDAP server.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Responds as above (scenario 1).

System Response - Flat Schema

1. LDAP server with Read-Only Permissions—The system returns an error message; it cannot delete the role from either the LDAP server or the WebCenter Sites database. The LDAP administrator must create the sites and roles in the LDAP server. The same sites and roles must be re-created in the WebCenter Sites database.
2. LDAP-2 Server with Read-Only Permissions—The system deletes the role from both the WebCenter Sites database and the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The system can return an error message.

Part II

Administering the Web Experience Management Framework

Oracle WebCenter Sites: Web Experience Management (WEM) Framework provides the technology for developing applications and integrating them with Oracle WebCenter Sites. The chapters in this part of the documentation will illustrate the administration of the WEM framework through the WEM Admin, which supports centralized application management and user authorization.

Part II contains the following chapters:

- [Chapter 34, "Web Experience Management Framework"](#)
- [Chapter 35, "Getting Started"](#)
- [Chapter 36, "Creating and Authorizing Users"](#)
- [Chapter 37, "Configuring REST Security"](#)
- [Chapter 38, "Working with Sites"](#)
- [Chapter 39, "WEM Admin Interface Quick Reference"](#)

Web Experience Management Framework

This chapter provides an overview of the Oracle WebCenter Sites: Web Experience Management (WEM) Framework and its administrators.

This chapter includes the following sections:

- [Section 34.1, "Overview"](#)
- [Section 34.2, "Administrative Roles and Permissions"](#)
- [Section 34.3, "WebCenter Sites Applications"](#)
- [Section 34.4, "Sample Sites"](#)

34.1 Overview

Oracle WebCenter Sites: Web Experience Management (WEM) Framework provides the technology for developing applications and integrating them with Oracle WebCenter Sites. A single administrative interface, WEM Admin, supports centralized application management and user authorization. Single sign-on enables users to log in once and gain access to all applications allowed to them during the session.

The WEM Framework requires a content management platform. In this release, the Framework runs on Oracle WebCenter Sites and ships with the WebCenter Sites Representational State Transfer (REST) API. Objects in the WebCenter Sites database, such as sites, users, and data model map to REST resources in the WEM Framework.

When implemented on the WEM Framework, applications communicate with the WebCenter Sites database through REST services ([Figure 34-1](#)). The applications appear in WEM Admin as list items on the Apps page ([Figure 34-2](#)). Administrators authorize users, which involves configuring access to applications and their resources. To this end, the WEM Admin interface exposes authorization items (along with applications) through links to Sites, Apps Users and Roles on the menu bar.

Figure 34–1 WEM Framework

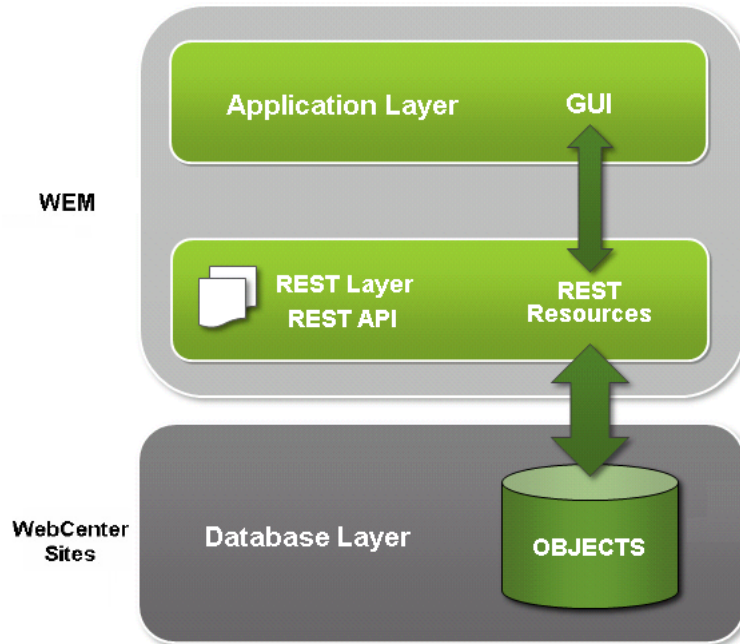
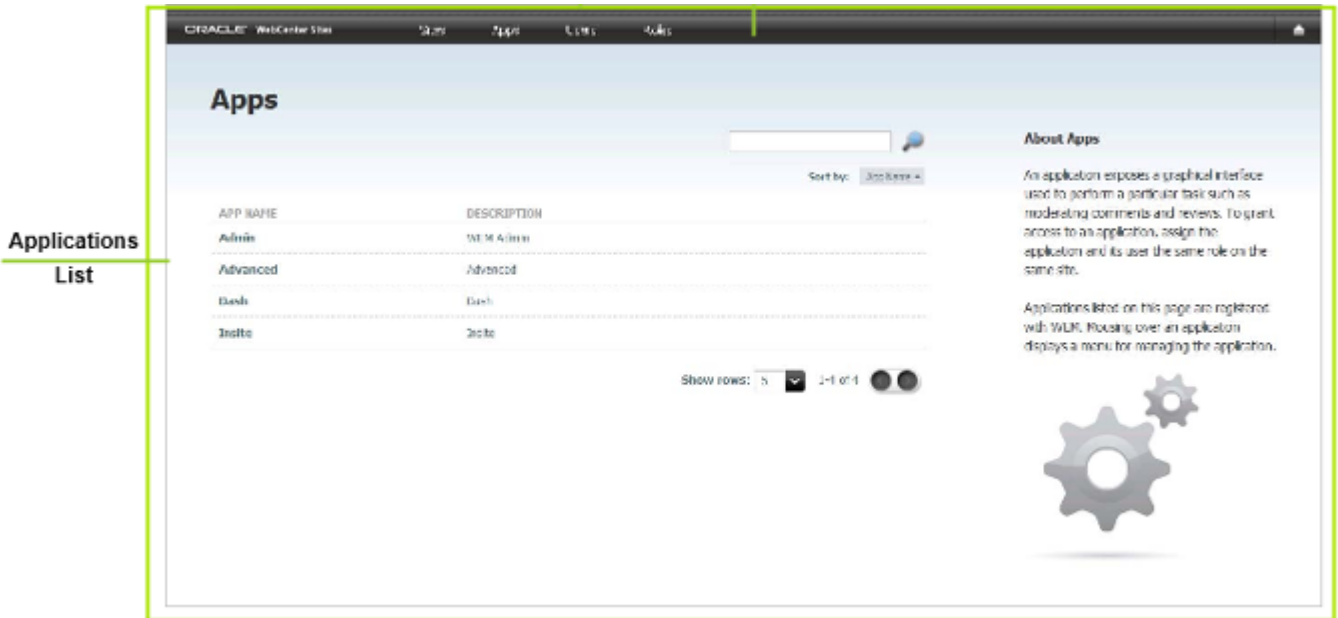
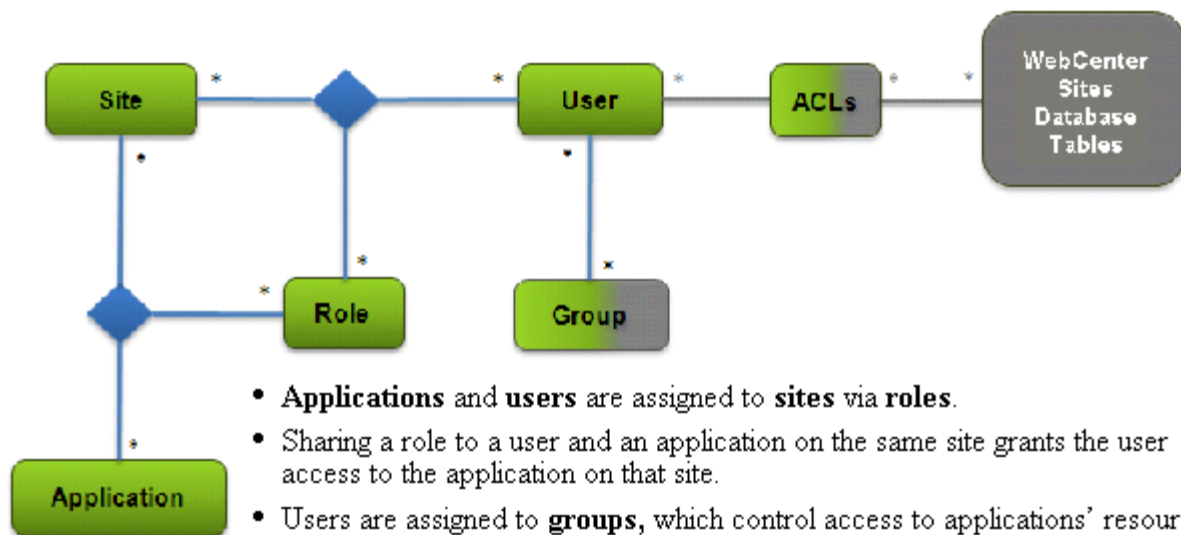


Figure 34–2 Apps Page, WEM Admin



Coupling the items as shown in Figure 34–3 enables applications for users.

Figure 34–3 Authorization Model

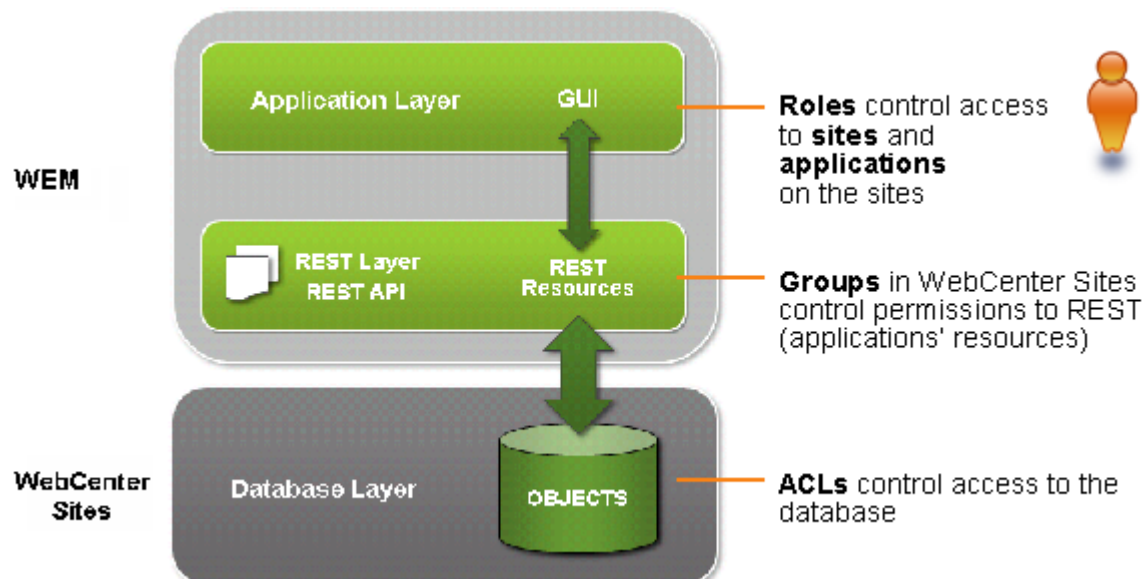


- **Applications** and **users** are assigned to **sites** via **roles**.
- Sharing a role to a user and an application on the same site grants the user access to the application on that site.
- Users are assigned to **groups**, which control access to applications' resources (REST layer).
- **ACLs** are assigned to users, providing them with access to the system.

Using WEM Admin, general administrators can create and otherwise manage sites, applications, users, and roles. Groups and ACLs must be configured in the WebCenter Sites Admin interface. They are exposed in WEM Admin, in user accounts.

Once the coupling is complete, users are authorized at the database, REST, and application levels.

Figure 34–4 WEM Framework



Experienced WebCenter Sites administrators will recognize that the WEM Admin interface extends the use of sites and roles to control access to applications. Roles can also be used within applications to protect interface functions (as in WebCenter Sites) and therefore regulate access to the applications' content.

Unlike WebCenter Sites, WEM Admin does not expose the data model. The REST API does. In this respect, WEM Admin can be thought of as strictly an authorization interface, supported by the WebCenter Sites Admin interface (for configuring ACLs and groups). The rest of this guide provides instructions for creating and authorizing users, as well as guidelines for managing sites.

34.2 Administrative Roles and Permissions

The WEM Framework supports two types of administrators: general and site administrators.

This section contains the following topics:

- [Section 34.2.1, "General Administrator"](#)
- [Section 34.2.2, "Site Administrator"](#)
- [Section 34.2.3, "All Administrators"](#)

34.2.1 General Administrator

A general administrator has complete control of the system and full permissions to the WEM Admin interface: Sites, Apps, Users, and Roles. Using WEM Admin, general administrators can add and delete sites, applications, users, and roles; modify their details; and perform authorization tasks.

A general administrator in the WEM Framework is a WebCenter Sites general administrator who is also specially configured for the WEM Framework. During the WebCenter Sites installation process, the default general administrator (`fwadmin`) was automatically assigned to the `RestAdmin` group for unrestricted access to REST services, and enabled on `AdminSite` where the WEM Admin application runs by default. The default general administrator must not be deleted.

You can create equivalent general administrators, as many as necessary, and you must modify pre-existing general administrators by adding them to the `RestAdmin` group and `AdminSite` (via the `GeneralAdmin` role). Instructions are available in [Chapter 36, "Creating and Authorizing Users."](#) For more information about WEM-related changes to WebCenter Sites, see the *Oracle Fusion Middleware WebCenter Sites Installation Guide*.

34.2.2 Site Administrator

Site administrators are assigned by general administrators to selected sites, where they manage site users and applications. When users are assigned the `SiteAdmin` role in a site other than `AdminSite` they are implicitly assigned the `SiteAdmin` role in `AdminSite`. Users cannot be assigned the `SiteAdmin` role in only `AdminSite`. In the WEM Admin interface, site administrators can access only the Sites form. They can perform the following operations on the sites to which they are assigned:

- Assign and remove users to and from sites
- Assign and remove applications to and from sites
- Modify the role assignments of site users
- Modify the role assignments of applications on the sites

A site admin cannot create, modify, or delete sites, users, and roles. (Your permissions determine which forms and interface functions WEM Admin displays to you.)

Site administrators on WebCenter Sites systems running the WEM Framework must be specially configured for WEM. They must be assigned to the `SiteAdmin_AdminSite`

group, a default REST security group configured in the WebCenter Sites Admin interface. Instructions are available in [Chapter 36.1, "Creating Users."](#)

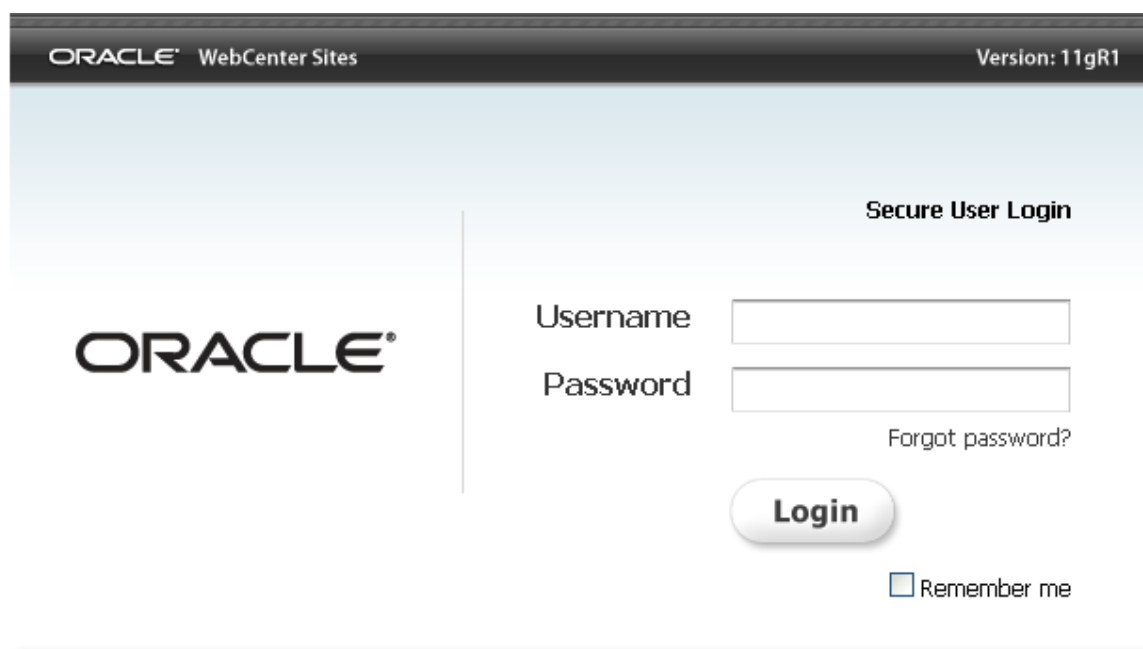
34.2.3 All Administrators

All WEM Framework administrators must be experienced WebCenter Sites users.

34.3 WebCenter Sites Applications

The WebCenter Sites Contributor and Admin interfaces are registered applications. They are listed on the **Apps** page of the WEM Admin interface. WebCenter Sites users can access the applications from the WEM Framework once they are authorized at the application level. They can also continue accessing the WebCenter Sites applications directly, at the usual URL. The login dialog is the following:

Figure 34–5 WebCenter Sites Application Login



The screenshot shows a web browser window titled "ORACLE WebCenter Sites" with "Version: 11gR1" in the top right corner. The main content area is light blue and contains the Oracle logo on the left. On the right, there is a "Secure User Login" section. It includes a "Username" label and an input field, a "Password" label and an input field, a "Forgot password?" link, a "Login" button, and a "Remember me" checkbox.

34.4 Sample Sites

Your WebCenter Sites system may have been installed with sample sites, in which case they are listed in the WEM Admin interface, along with your custom sites. The sample sites are:

- FirstSiteII
- avisports

This chapter provides instructions for logging in to and navigating the WEM Admin interface.

This chapter contains the following sections:

- [Section 35.1, "Logging In and 'Quick Tour'"](#)
- [Section 35.2, "Next Steps"](#)

35.1 Logging In and 'Quick Tour'

In this section, you are going to access WEM Admin by logging in to Oracle WebCenter Sites. When you sign in, the WEM Framework determines your status as a general administrator or site administrator and enables only the forms and interface functions that you need to work with.

Note: This guide is for general administrators. If you are not a general administrator, then certain sections of this guide (such as creating users) do not apply to you.

To access WEM Admin

1. Access Oracle WebCenter Sites using the following URL:

```
http://<server>:<port>/<context>/login
```

where <server> is the host name or IP address of the server running WebCenter Sites, <port> is the port number, and <context> is the name of the web application which was deployed on that server. Depending on how the system was set up, you may also need to include the port number.

2. Log in as the general administrator that was used during the WebCenter Sites installation process.

This guide uses the general administrator's default credentials:

User: fwadmin

Password: xceladmin

Figure 35-1 WebCenter Site Login



3. Click **Login**.
4. If you are logging in for the first time or in to a site that you have never accessed before, the following dialog is displayed:

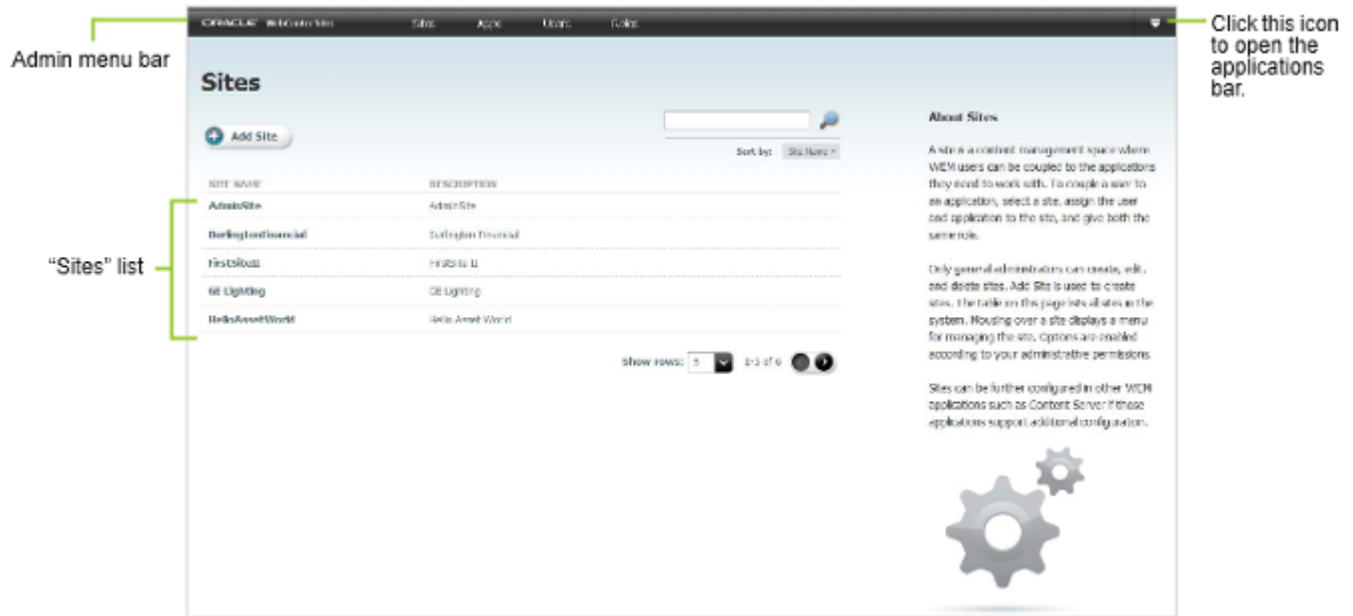
Figure 35-2 WebCenter Sites Login



Select **AdminSite** and the **WEM Admin** application icon to open the WEM Admin interface.

The first form you see is the Sites form.

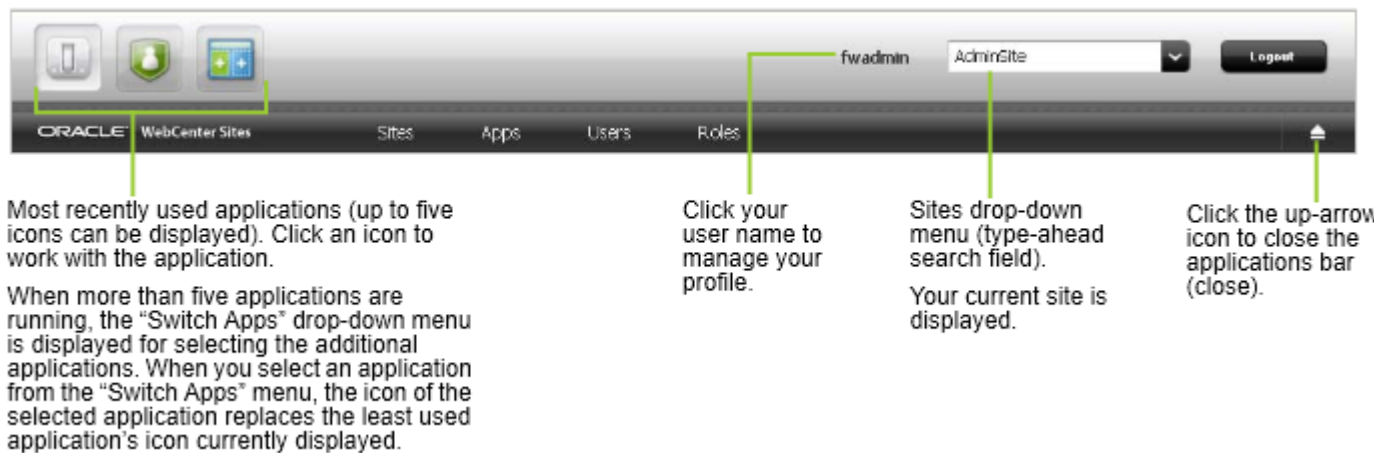
Figure 35–3 WEM Admin Interface Sites Form



The Sites form lists all the sites in the system. Site administrators will see only the sites in which they are assigned the SiteAdmin role. If you logged in as a general administrator you also have access to the Apps, Users, and Roles forms.

5. Open the applications bar by mousing over the **down-arrow** at the extreme right of the menu bar. The bar displays the icons of applications that are available to you on the current site, a link to your profile, a drop-down menu of sites accessible to you, the logout button, and the pin icon.

Figure 35–4 Applications bar



35.2 Next Steps

The next chapter shows you how to create users in the WEM Admin interface and authorize them to work with applications.

Creating and Authorizing Users

This chapter provides information and instructions about creating a user in the WEM Admin interface, and authorizing that user to manage a site and the application(s) available on that site.

This chapter includes the following sections:

- [Section 36.1, "Creating Users"](#)
- [Section 36.2, "Authorizing Users to Work with Applications"](#)
- [Section 36.3, "Authorizing a Predefined User"](#)
- [Section 36.4, "Authorizing Developers to Register Applications"](#)
- [Section 36.5, "Ask Your Developers"](#)

36.1 Creating Users

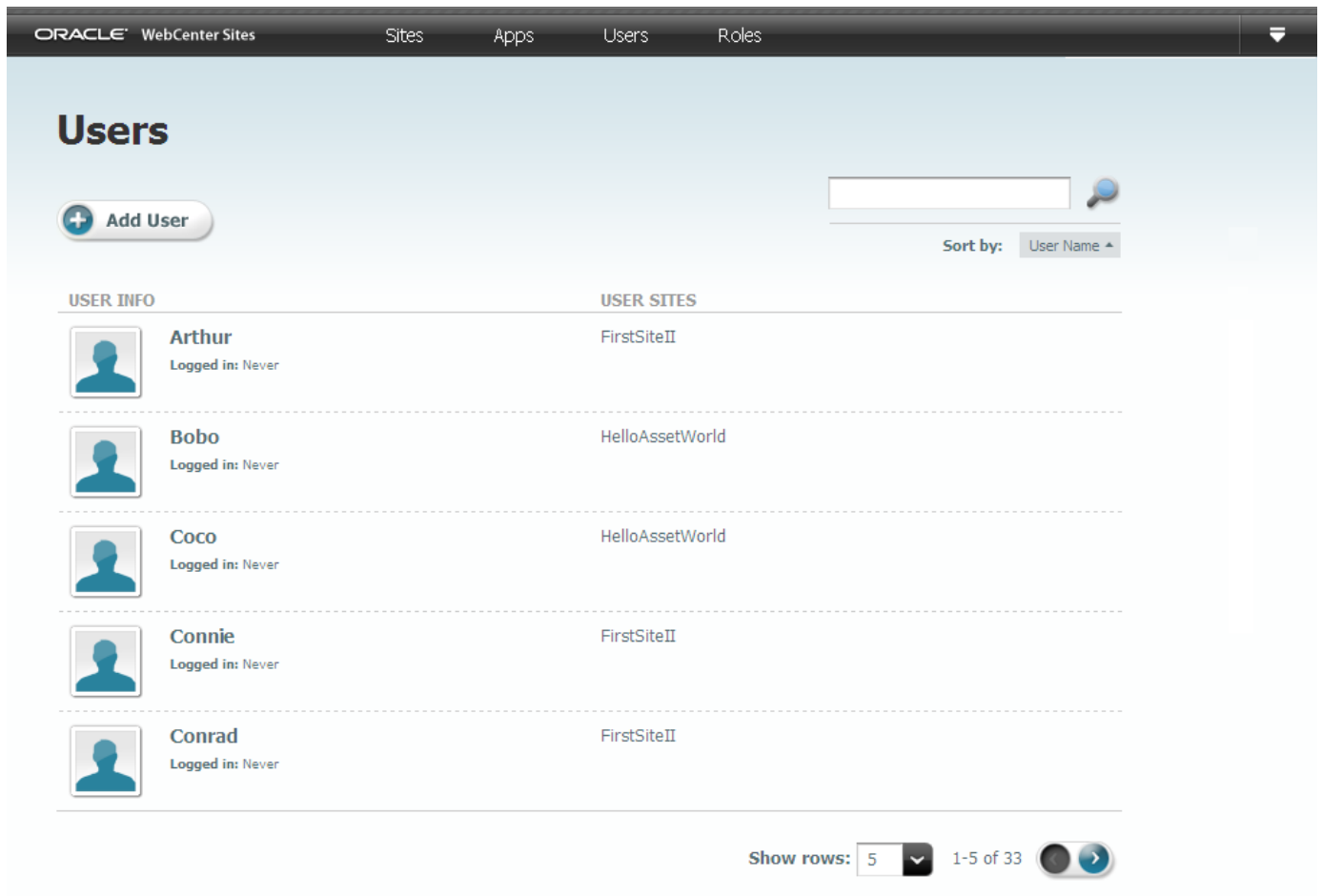
The types of users you can create are general administrators, site administrators, and regular users.

Note: Only general administrators can create users. For information about predefined users, see [Section 36.3, "Authorizing a Predefined User."](#)

To create the user

1. Log in to the WEM Admin interface as the general administrator that was used during the WebCenter Sites installation process.
2. In the Admin menu bar, click **Users**. The Users form is displayed.

Figure 36-1 Users Form




3. In the Users form, click the **Add User** button to open the Add User form.

Figure 36-2 Add User Form

Add User

USER IMAGE

Image Preview  Browse Remove

USER DATA

*** Name** This value is required.

Email

Locale

Time Zone

*** ACLs** Filter List

<p>Available (19)</p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <ul style="list-style-type: none"> Browser ContentEditor ElementEditor ElementReader PageEditor PageReader RemoteClient SiteGod TableEditor </div>	<div style="margin: 5px 0;">→</div> <div style="margin: 5px 0;">←</div>	<p>Selected (0)</p> <div style="border: 1px solid #ccc; height: 100px;"></div>
---	---	---

*** Groups** Filter List

<p>Available (2)</p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> <ul style="list-style-type: none"> RestAdmin SiteAdmin_AdminSite </div>	<div style="margin: 5px 0;">→</div> <div style="margin: 5px 0;">←</div>	<p>Selected (0)</p> <div style="border: 1px solid #ccc; height: 100px;"></div>
--	---	---

PASSWORD

*** New Password** This value is required.

*** Confirm Password** The value entered is not valid.

Save and Close
Save
Cancel

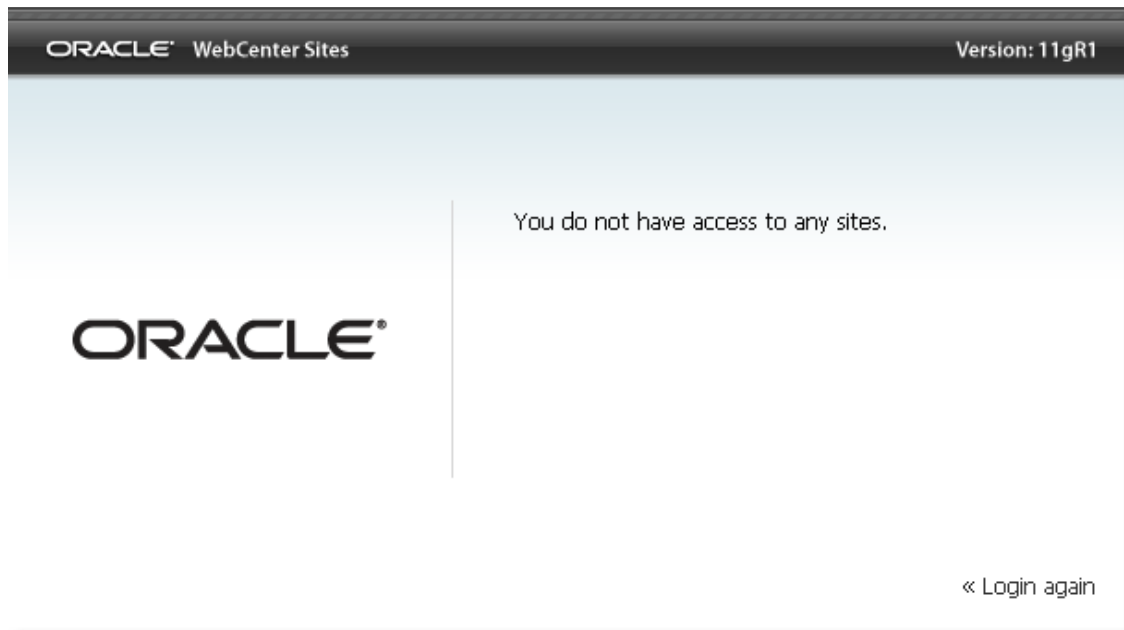
Fill in the following fields:

- **Image Preview** – (Optional) Use the **Browse** button to associate a picture with the new user.
- **Name** – Enter a name that the user will use to log in.
- **Email** – (Optional) Enter a valid, unique email address.
- **Locale** – (Optional) Select the user's language preference. If you do not specify a preference, the WEM Framework uses the default locale that is set for the user's browser.
- **Time Zone** – (Optional) Select the user's time zone preference from the dropdown menu. If you do not specify a time zone preference, the system will automatically detect the time zone of the user from the user's browser.
- **ACLs** – ACLs regulate the user's access to the database tables. All users require Browser, ElementReader, PageReader, UserReader, and xceleditor. General and site administrators also require xceladmin. General administrators further require TableEditor and UserEditor (and VisitorAdmin, if they use WebCenter Sites: Engage).
- **Groups** – Groups provide access to REST. They are used to control access to applications' resources.
 - If you are creating a general administrator, assign the user to the RestAdmin group (a default group, configured in the WebCenter Sites Admin interface). This group has unrestricted permissions to REST resources.
 - If you are creating a site administrator, assign the user to the SiteAdmin_AdminSite group (a default group, configured in the WebCenter Sites Admin interface).

Note: Security configurations for groups are available in the WebCenter Sites Admin interface. See [Section 37.2.1, "Viewing REST Security Configurations."](#)

- If you are creating a regular user, skip this step, for now. You will assign the user to a group (or groups) in step 4, as part of the authorization process ([Section 36.2, "Authorizing Users to Work with Applications"](#)).
 - **New Password** – Enter a password that is at least 6 characters long.
 - **Confirm Password** – Re-type the password you just entered.
4. Click **Save and Close**.

At this point the user can log in, but a message will be displayed indicating that the user does not have access to any sites.

Figure 36–3 WebCenter Sites Login - Access Denied**To enable the user as an administrator or regular user**

1. Assign the user to a site:
 - a. From the Users form, mouse over the user, select **Manage User**, and click **Assign to Sites**.
 - If you are creating a general administrator, assign the user to AdminSite.
 - If you are creating a site administrator or regular user, assign the user to a site other than AdminSite.
 - b. Assign roles to the user on the site:
 - If you are creating a general administrator, assign the GeneralAdmin role, which grants the user access to the system.

The WEM Admin application is now available to the user on AdminSite:

Figure 36-4 WebCenter Sites Login Showing WEM Admin App Icon



From this menu, the user that you created has access to AdminSite and the site in which he is assigned the SiteAdmin role. If the SiteAdmin role does not authorize the user to access any applications on the site in which he is assigned the SiteAdmin role, no application icons are displayed below the menu when the user selects the

- If you are creating a site administrator, assign the SiteAdmin role.

A user who is assigned the SiteAdmin role on a site other than AdminSite is implicitly assigned to AdminSite and gains access to the WEM Admin application on AdminSite. In the WEM Admin application, the user can access only the Sites form, which lists only the site(s) in which he is assigned the SiteAdmin role.

Figure 36-5 WebCenter Sites Login Showing WEM Admin App Icon



From this menu, the user that you created has access to AdminSite and the site in which he is assigned the SiteAdmin role. If the SiteAdmin role does not authorize the user to access any applications on the site in which he is assigned the SiteAdmin role, no application icons are displayed below the menu when the user selects the

- If you are creating a regular user, assign the user roles that are not GeneralAdmin or SiteAdmin.

The user now has access to the site (listed in the menu), but if the user's roles do not authorize access to any applications on that site, no application icons are displayed below the menu.

Figure 36–6 WebCenter Sites Login



2. To authorize a user to work with applications, continue to the next [Section 36.2, "Authorizing Users to Work with Applications."](#)

36.2 Authorizing Users to Work with Applications

Users require authorization to work with applications, such as the WebCenter Sites Contributor interface.

This procedure shows you how to authorize a user, which involves:

1. Selecting or creating a site
2. Assigning an application to the site
3. Assigning a user to the same site and coupling the user to the application
4. Assigning the user to a group to enable the user's permissions to REST (applications' resources).

Note: ■Both general and site administrators can authorize users.

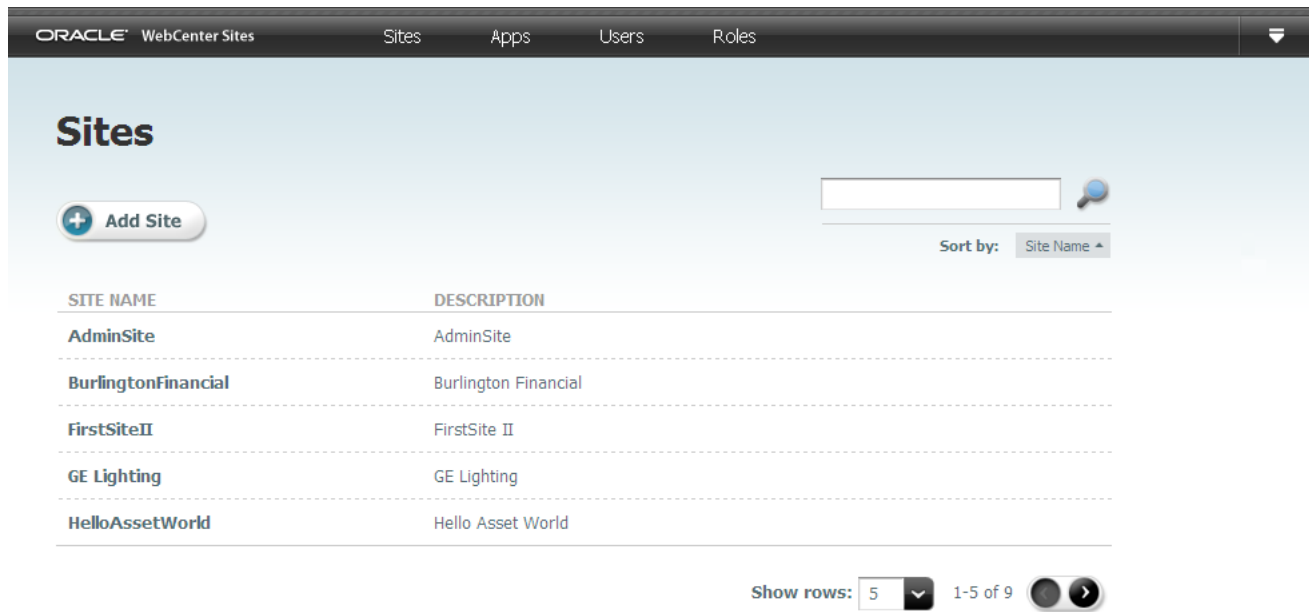
- If you need access to an application on a given site, authorize yourself to access the application on the site.
 - In the steps below, you can select multiple applications and multiple users. For simplicity, instructions specify a single application and user.
 - In this procedure, we assume the user you are authorizing will be working with applications that do not specify a predefined user. For information about predefined users, see [Section 36.3, "Authorizing a Predefined User."](#)
-
-

To authorize a user

1. Select or create the site:

From the WEM Admin interface, click **Sites** on the Admin menu bar.

Figure 36–7 WEM Admin Sites Form

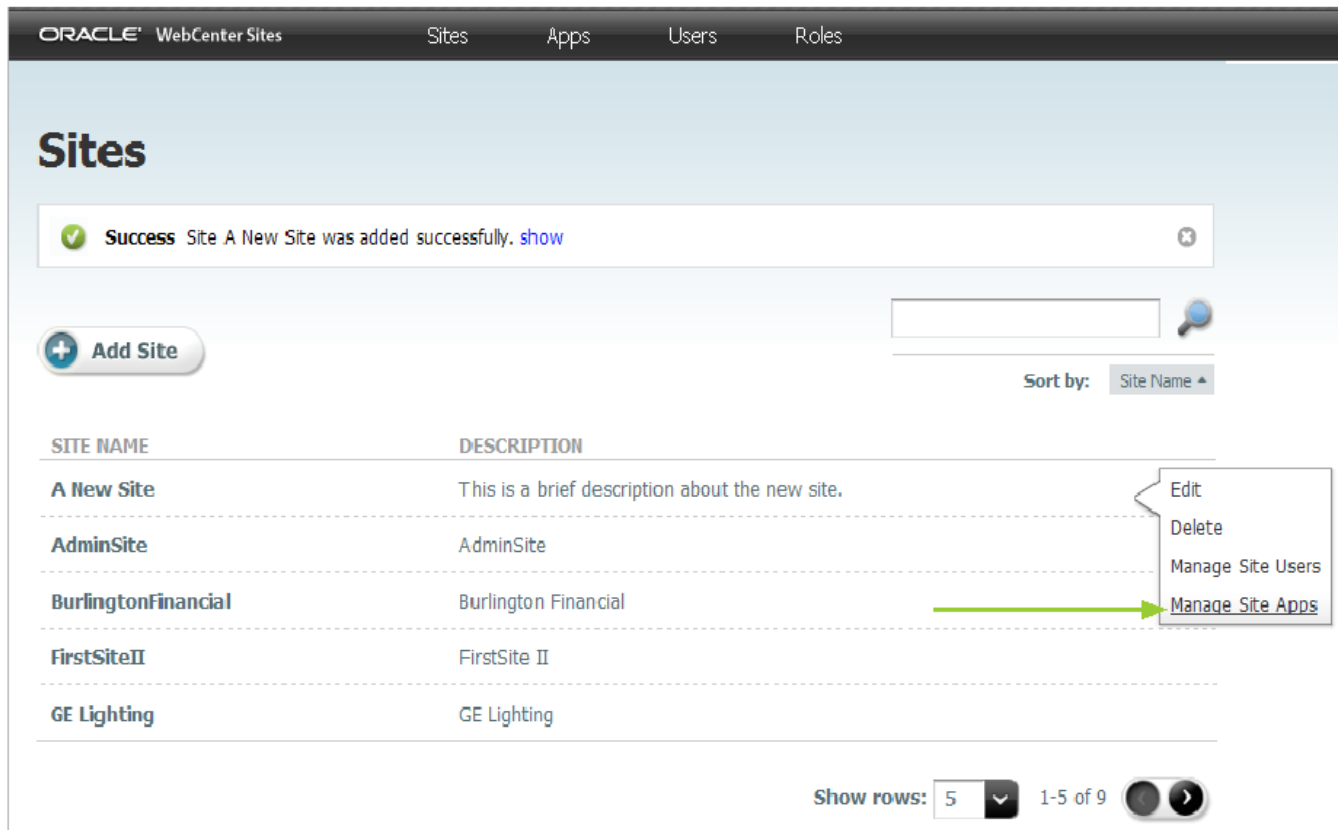


If you are a general administrator you can select a site or add a site (click **Add Site**). If you are a site administrator you can select a site. Your Sites form lists only the sites you are allowed to manage.

2. Assign an application to the site:

- a. In the Sites form, mouse over the site's name and click **Manage Site Apps**.

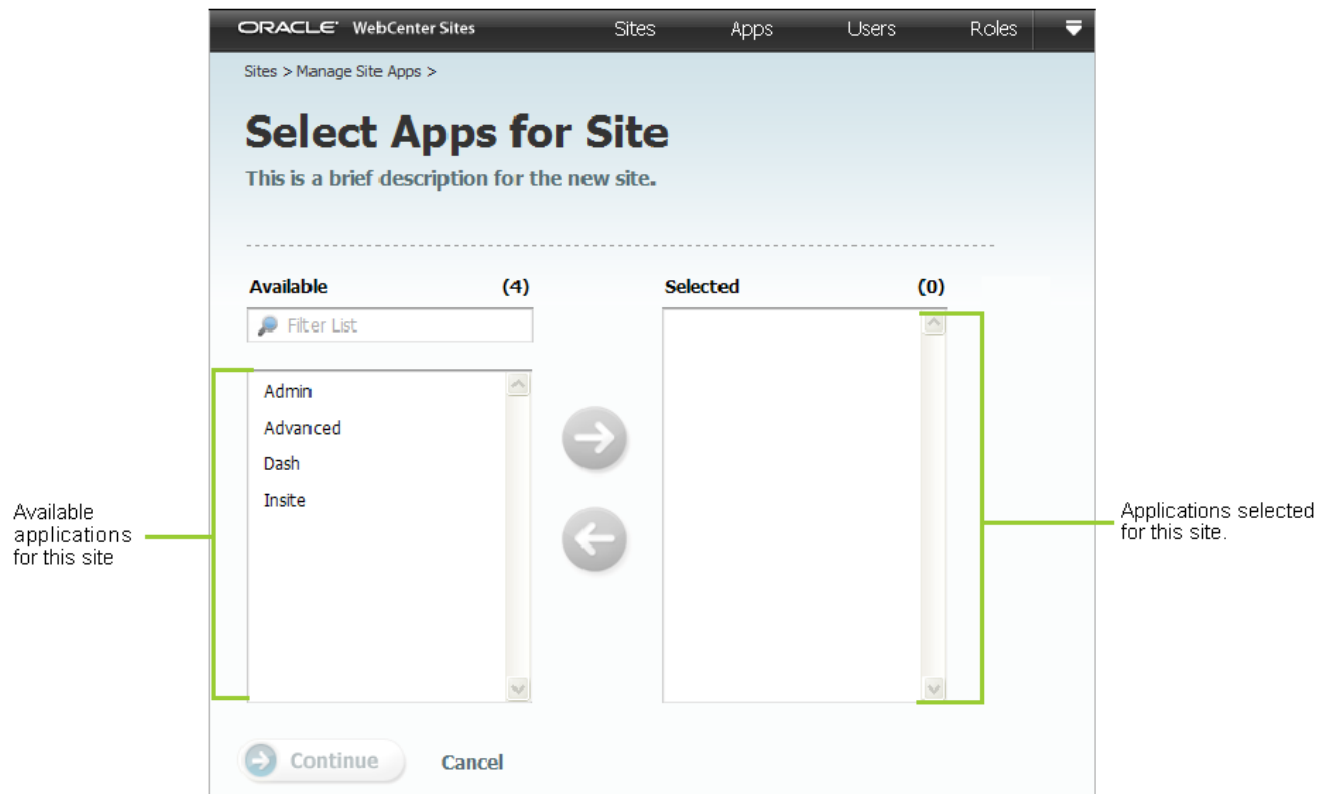
Figure 36–8 Sites Form



- b. Click **Assign Apps**.

Note: The **Assign Apps** button is dimmed if no applications are registered with the WEM Framework.

Figure 36–9 Select Apps for Site Form



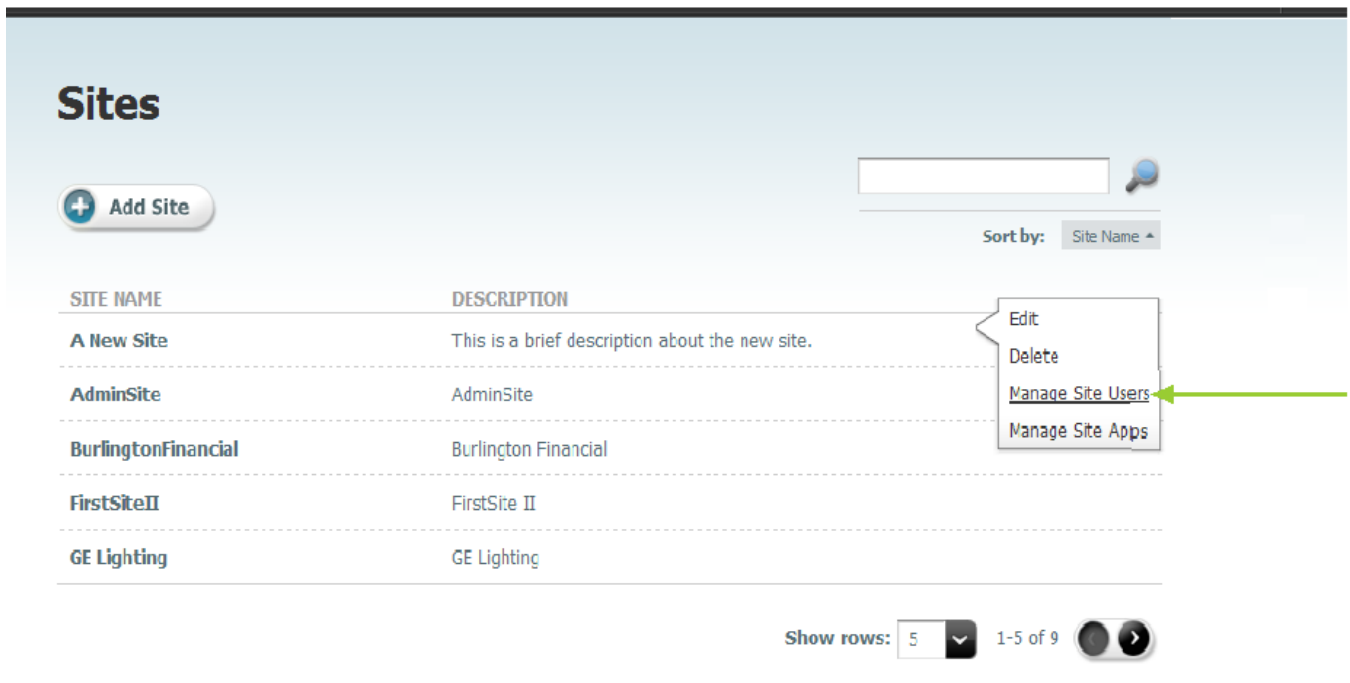
- c. Select the application you wish to assign to the site and move it to the **Selected** list box. (To search for an application, type its name in the **Filter List** field. The results appear in the **Available** list box).
- d. Click **Continue** to assign roles to the application.
- e. In the Assign Roles to Apps form, select roles for the application and move them to the Selected list box.

Note: if the application is the WebCenter Sites Admin interface, assign it the AdvancedUser role. If the application is the WebCenter Sites Contributor interface, assign it the SitesUser role.

Take note of the roles you are assigning. You will assign at least one of those roles to the user on the site to grant the user access to the application.

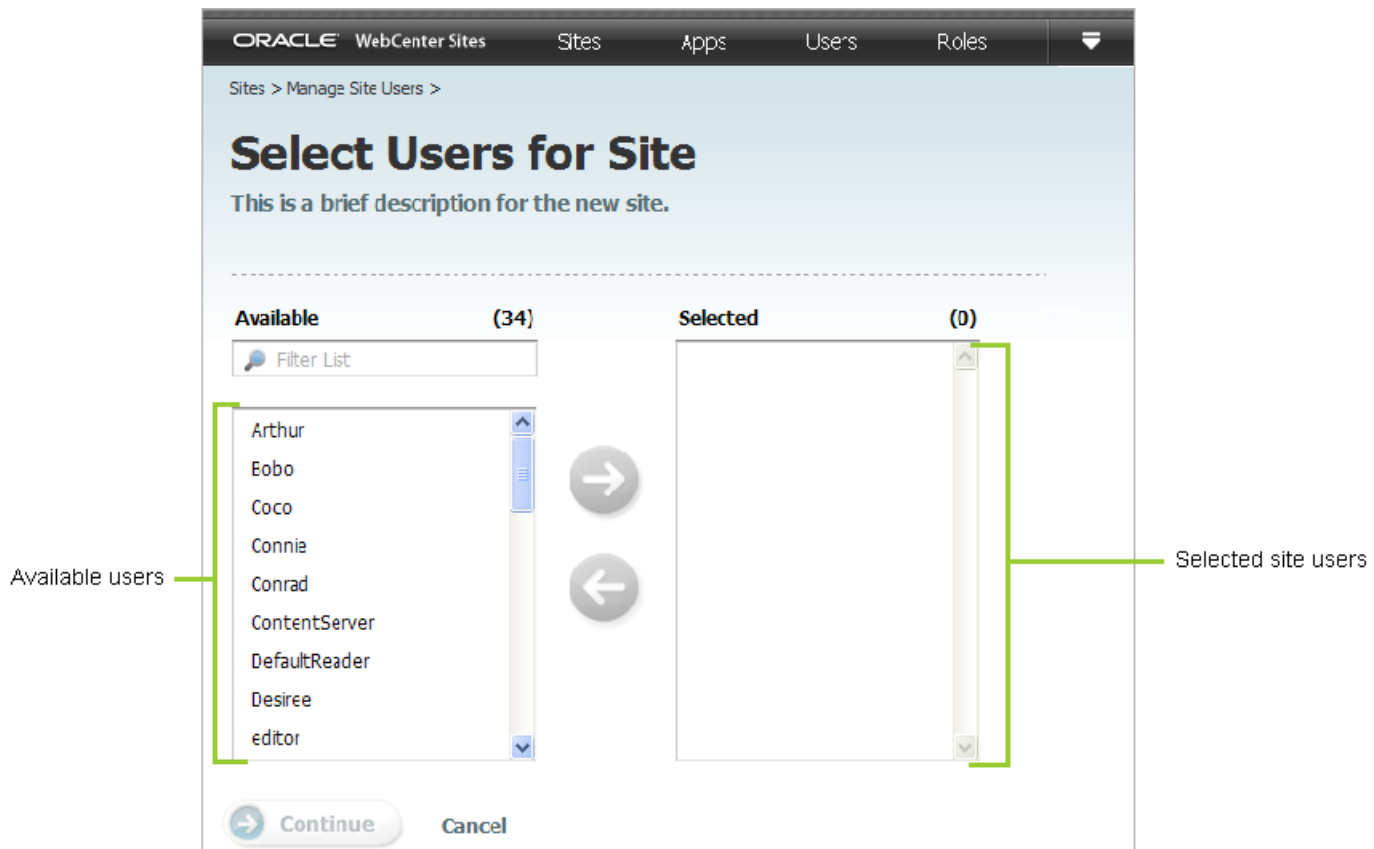
- f. Click **Save and Close**.
3. **Assign a user to the site:**
- a. In the Admin menu bar, click **Sites**.
 - b. Mouse over the new site's name and click **Manage Site Users**.

Figure 36–10 Manage User Sites Link on Sites Form



c. Click Assign Users.

Figure 36–11 Assign Users Form



- d. In the Select Users for Site form, select the user you wish to assign to the site and move the user to the Selected list box.
- e. Click **Continue** to assign roles to the user.
- f. **Couple the user to the application (application-level authorization):**
In the Assign Roles to Users form, assign the user at least one of the roles that you assigned to the application in step e.

Note: ■ **For all applications.** Sharing a role to a user and an application on a site grants the user access to the application on that site. If the application is the WebCenter Sites Admin interface, you must assign the user the AdvancedUser role. If the application is the WebCenter Sites Contributor interface, you must assign the user the SitesUser role.

- **For applications other than WebCenter Sites.** If the application has role-protected interface functions (such as Edit), configure access to each function by assigning the user at least one of the function's roles (specifications are available from application developers). The user is then fully authorized at the application level. However, the user will not be able to work with the application's resources until you authorize the user at the REST level. Click **Save and Close** and continue to step 5.
 - **For WebCenter Sites applications and users.** WebCenter Sites has role-protected interface functions. The roles of users configured directly in WebCenter Sites are preserved in the WEM Framework. They are listed in the WEM Admin interface, site by site. Also, the application REST service authorizes WebCenter Sites users at the REST level (eliminating step 4 for administrators). Click **Save and Close** and skip to step 5.
-

4. Authorize the user at the REST level:

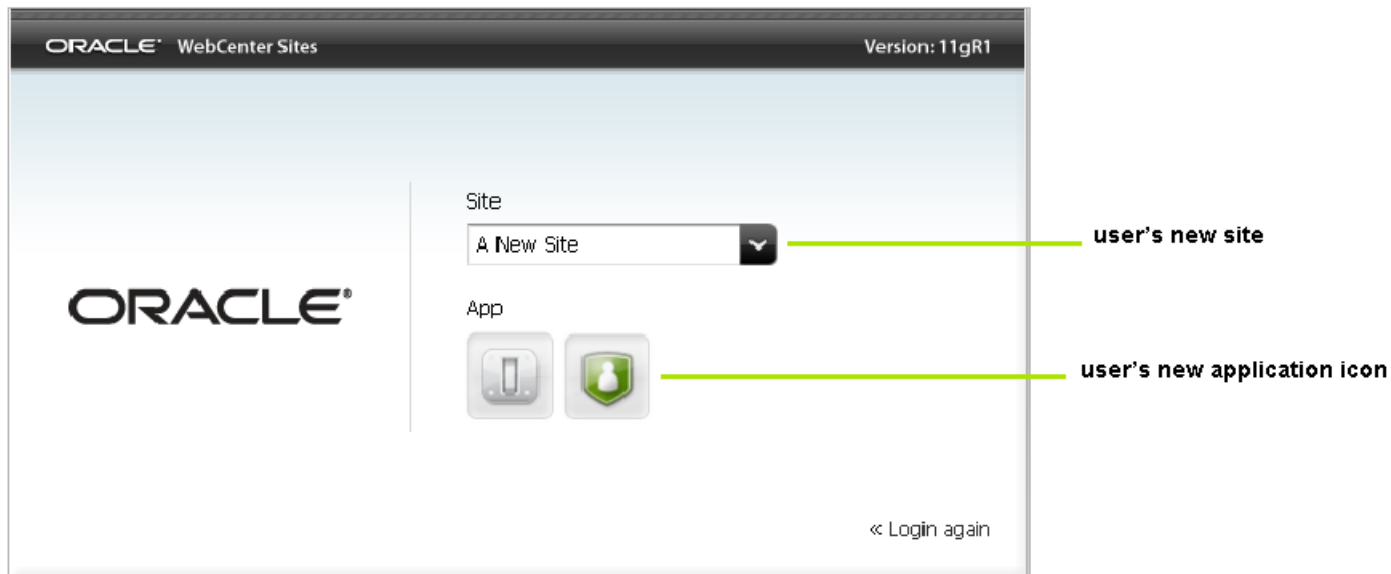
This step grants the user permissions to operate on resources that are used by the application (selected in step 2).

Note: As noted above, skip this step if you are authorizing dedicated WebCenter Sites users to access the WebCenter Sites applications from the WEM Framework. Continue to step 5.

- a. In the **Admin** menu bar, click **Users**.
 - b. In the Users form, mouse over the user you wish to authorize and click **Edit**.
 - c. In the Edit User form, select group(s) for the user. Each group is configured with specific permissions to operate on specific objects (such as asset types and assets), which map to REST resources used by the application. To determine the permissions of the listed REST groups, or to create groups and configure their privileges, see [Section 37.2, "Authorizing Users to Access Application Resources."](#)
 - d. Click **Save and Close**.
5. Verify the user's ability to access the new application.

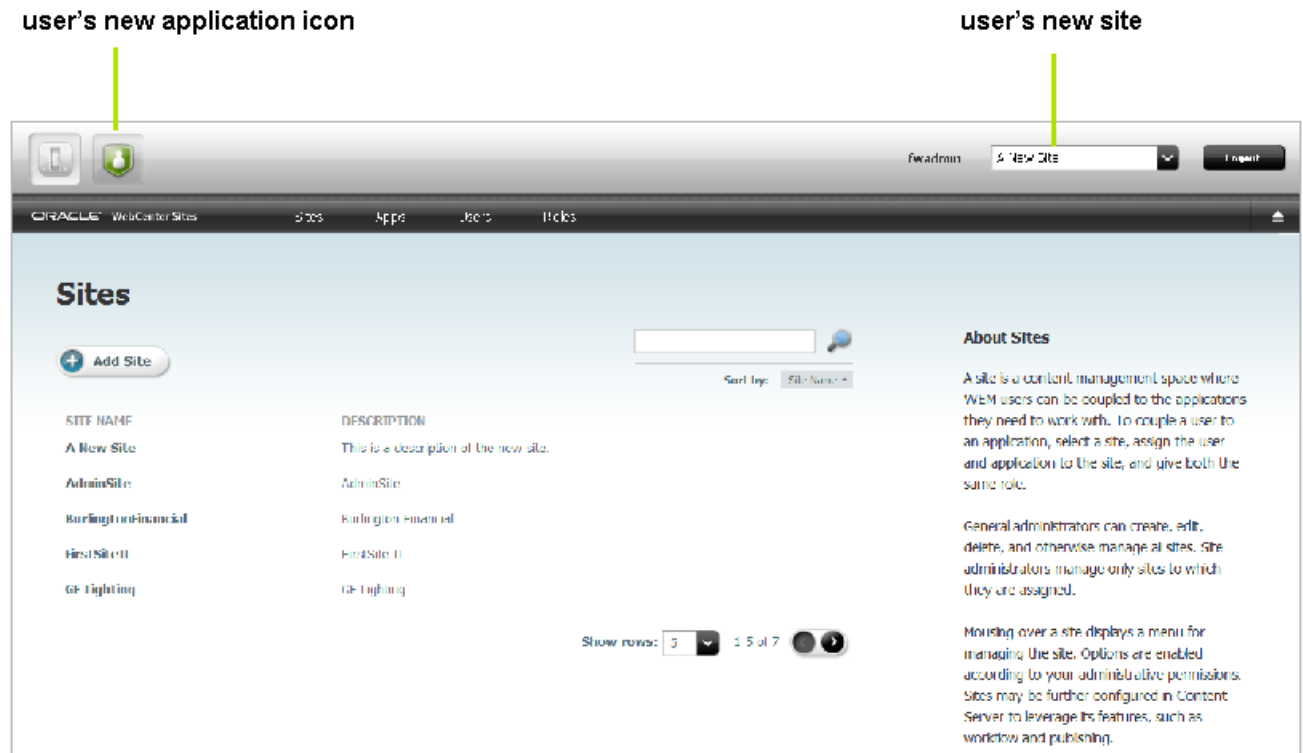
The login dialog lists the user's new site (in the **Site** drop-down menu) and displays the application icon below the menu.

Figure 36–12 WebCenter Sites Login Dialog



The new site is also listed in the drop-down menu next to the name of the logged-in user, and the application icon is displayed in the upper left-hand corner.

Figure 36–13 WEM Admin Sites Form



6. As a reminder, if you have not yet authorized the user with permissions to REST, complete the steps in [Chapter 37, "Configuring REST Security."](#)

36.3 Authorizing a Predefined User

Developers specify predefined users in their applications to simplify administrators' authorization processes. Instead of authorizing each user individually at the REST level, you will authorize the predefined user. Logged-in users with access to the application will gain access to the application's resources through the predefined user's membership in REST groups.

If an application is configured with a predefined user, complete the following steps in the WEM Admin application.

1. Create the predefined user. Have the following information ready:
 - Login name. This name must exactly match the predefined user's name, as specified in the application.
 - Password. The password must exactly match the predefined user's password, as specified in the application.
 - ACLs, which regulate the user's access to the database tables. The predefined user must be assigned the ACLs of the logged-in users who will access the application. All users require Browser, ElementReader, PageReader, UserReader, and xceleeditor. General and site admins also require xceladmin. General admins further require TableEditor and UserEditor (and VisitorAdmin, if they use WebCenter Sites: Engage).
 - Group assignment, which authorizes the user at the REST level (to manage application resources). The predefined user must be assigned to a group with the security privileges that you would otherwise grant to the application users. For information about configuring REST security, see [Chapter 37, "Configuring REST Security."](#)

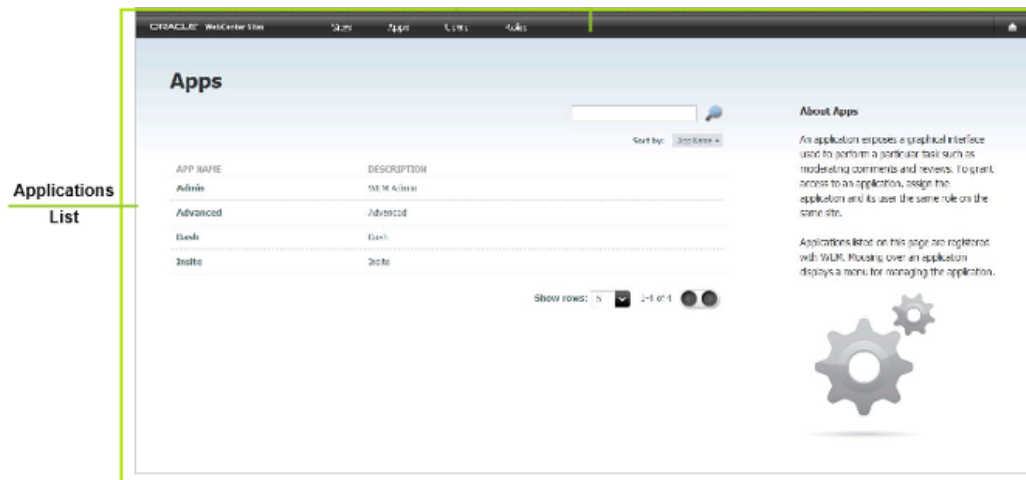
For instructions on creating the user, see [Section 36.1, "Creating Users."](#)

2. Assign the predefined user to the application. For instructions, see [Section 36.2, "Authorizing Users to Work with Applications."](#)
3. Assign users to the application (using the procedure in [Section 36.2, "Authorizing Users to Work with Applications"](#)), but skip their assignment to groups (step 4).

36.4 Authorizing Developers to Register Applications

For applications to be exposed in the WEM Framework, they must be registered – that is, created as assets – so they can be displayed through REST services on the Apps page in the WEM Admin interface. Administrators can then authorize users to work with the applications.

Figure 36–14 WEM Admin Apps Form



Typically, it is developers who register the applications they create. The preferred method is programmatic. If developers choose to register applications manually they must use the WebCenter Sites Admin interface to create assets of type `FW_Application` and `FW_View`. The asset types are enabled on AdminSite. (For more information about registering applications, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.)

To authorize a developer, ensure the developer is a general administrator (i.e., has complete permissions to the system, including REST services). For instructions on creating a general administrator, see [Section 36.1, "Creating Users."](#)

36.5 Ask Your Developers

To ensure your effectiveness in managing applications and users, you will need information from your developers about the applications they have created for the WEM Framework.

This section contains the following topics:

- [Chapter 36.5.1, "Resources and Applications."](#)
- [Chapter 36.5.2, "Roles and Applications."](#)
- [Chapter 36.5.3, "Predefined Users."](#)

36.5.1 Resources and Applications

Ask your developers about the resources that are used by custom-built applications.

Once you know which asset types, assets, and other resources users will be working with, you can determine which privileges (such as create, update) the users must be given to those resources and assign the users to groups that have those privileges. Information about configuring groups and assigning users is available in [Chapter 37, "Configuring REST Security."](#)

36.5.2 Roles and Applications

Ask your developers whether applications' interface functions are role-protected.

In WEM, roles are used to manage access to applications. Sharing a role to a user and an application on the same site grants the user access to the application on that site. Roles can also be used in application code to protect interface functions, such as Edit. When an application specifies role-protected functions, application users must share at least one role with each interface function. To ensure proper authorization, see [Section 36.2, "Authorizing Users to Work with Applications."](#)

36.5.3 Predefined Users

Ask your developers whether predefined users are configured in the applications.

If an application specifies a predefined user, you must authorize the predefined user at the REST level, instead of authorizing all application users individually. Security privileges granted to the predefined user by membership in groups will be passed to logged-in users when they access the application. For instructions on authorizing a predefined user, see [Section 36.3, "Authorizing a Predefined User."](#)

Configuring REST Security

This chapter provides information and instructions about configuring REST security, and contains the following sections:

- [Section 37.1, "REST Authorization"](#)
- [Section 37.2, "Authorizing Users to Access Application Resources"](#)
- [Section 37.3, "REST Security Configuration Reference"](#)

37.1 REST Authorization

REST authorization is the process of granting privileges to perform REST operations on applications' resources, which map to objects in WebCenter Sites. REST authorization uses the "deny everything by default" model. If a privilege is not explicitly granted to a particular group, that privilege is denied. General administrators are responsible for authorizing users once the application is deployed and registered with the WEM Framework.

This section contains the following topics:

- [Section 37.1.1, "Security Model"](#)
- [Section 37.1.2, "Configuring REST Security"](#)
- [Section 37.1.3, "Privilege Resolution Algorithm"](#)

37.1.1 Security Model

The WEM security model is based on objects and groups, which are predefined in WebCenter Sites, and actions, which you create in WebCenter Sites (objects in WebCenter Sites map to REST resources in the WEM Framework). Security must be configured per object type in the WebCenter Sites Admin interface:

Figure 37–1 Add New Security Configuration Form

Add New Security Configuration

Objects of a given type are accessible to a user only if the user belongs to at least one group with privileges to perform specified actions on objects of the given type.

*Type:

*Name:

*Groups:

*Action:

- **Object** is a generic term that refers to any entity such as a site, a user, or an asset. Protected objects are of the following types:
 - Asset Type
 - Site
 - User Locale
 - Application
 - Asset
 - Role
 - ACL
 - Index
 - User
 - Group
- **Security groups** are used to gather users for the purpose of managing their permissions (to operate on objects) simultaneously.
- An action is a security privilege: LIST, HEAD, READ, UPDATE, CREATE, DELETE. Groups are assigned privileges to operate on the objects allowed to the groups. Some objects, such as ACLs, are list-only (they can be created directly in WebCenter Sites, but not over REST).

A security configuration is an array, such as shown above, that specifies:

- The protected object type and object(s)
- Groups that are able to access the objects
- Actions that groups (and their members) can perform on the objects

37.1.2 Configuring REST Security

Procedures for configuring REST security are available in [Section 37.2, "Authorizing Users to Access Application Resources."](#)

37.1.3 Privilege Resolution Algorithm

When configuring a security privilege, you can specify that the privilege applies to all objects of a certain type or a single object of a certain type. For example, granting the privilege to UPDATE (POST) any site allows users in the group to modify the details of all sites in the WEM Framework. Granting the privilege to UPDATE (POST) the FirstSiteII sample site allows users in the group to modify this site's details in WEM.

The Asset object type requires you to specify the site to which the security setting applies, as assets are always accessed from a particular site. The AssetType object type can be extended by specifying a subtype, which is used to make the security configuration more granular. For example, setting the DELETE privilege on asset type Content_C allows a DELETE request to be performed on the REST resource /types/Content_C (i.e., to delete the Content_C asset type from the system).

Because privileges can be granted only to groups, a user's total privileges are not obvious until they are computed across all of the user's group. The WEM Framework provides a privilege resolution algorithm. Its basic steps are listed below:

1. REST finds the groups in which the user has membership.
2. REST determines which groups can perform which REST operations on which REST resources. If site or subtype is specified, each is taken into account.
3. REST compares the results of steps 1 and 2. If at least one of the groups from step 1 is in the list of groups from step 2, then access is granted. Otherwise, access is denied.

37.2 Authorizing Users to Access Application Resources

Before continuing with this section, read "[Section 37.1, "REST Authorization"](#)" for background information relating to the steps provided below.

- [Section 37.2.1, "Viewing REST Security Configurations"](#)
- [Section 37.2.2, "Creating a Group"](#)
- [Section 37.2.3, "Adding Users to a Group"](#)
- [Section 37.2.4, "Configuring Security for REST Resources"](#).

Note: Configure security for REST resources requires groups. You will be assigning privileges to the groups. To view the list of groups, continue with [Section 37.2.1, "Viewing REST Security Configurations."](#) To create groups, see [Section 37.2.2, "Creating a Group."](#)

37.2.1 Viewing REST Security Configurations

A security configuration identifies which groups have which permissions to which REST resources. WebCenter Sites defines security configurations for two default groups. They are RestAdmin and SiteAdmin_AdminSite.

To view REST security configurations

1. Log in to the WebCenter Sites Admin interface as a general administrator:
2. Select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security** node, and double-click **Configure Security**.

The Security Configurations window is rendered in the main window.

Figure 37–2 Security Configuration Form

Security Configurations

Type	Site	Subtype	Name	Groups	Action
			Any	'RestAdmin'	List
	Any		Any	'RestAdmin'	Create
	Any		Any	'RestAdmin'	Update
	Any		Any	'RestAdmin'	Delete
	Any		Any	'RestAdmin'	Read/Head
	Any		Any	'RestAdmin'	List
	Any		Any	'RestAdmin'	Create
	Any		Any	'RestAdmin'	Update
	Any		Any	'RestAdmin'	Delete
	Any		Any	'RestAdmin'	Read/Head
	Any		Any	'RestAdmin'	List
	Any	_ANY_	Any	'RestAdmin'	Read/Head
			Any	'RestAdmin'	List
			Any	'RestAdmin'	Read/Head
			Any	'RestAdmin'	Create
			Any	'RestAdmin'	Update
			Any	'RestAdmin'	Delete
			Any	'RestAdmin'	Read/Head
			Any	'RestAdmin'	List
			Any	'RestAdmin'	Create
			Any	'RestAdmin'	Update
			Any	'RestAdmin'	Delete
			Any	'RestAdmin'	Read/Head
			Any	'RestAdmin','SiteAdmin_AdminSite'	List
			Any	'RestAdmin'	Create
			Any	'RestAdmin','SiteAdmin_AdminSite'	Update
			Any	'RestAdmin'	Delete
			Any	'RestAdmin','SiteAdmin_AdminSite'	Read/Head
			Any	'RestAdmin','SiteAdmin_AdminSite'	List
			Any	'RestAdmin'	Create
			Any	'RestAdmin'	Update
			Any	'RestAdmin'	Delete
			Any	'RestAdmin'	Read/Head
			Any	'RestAdmin','SiteAdmin_AdminSite'	List
			Any	'RestAdmin'	Read/Head
			Any	'RestAdmin'	List

Add New

3. Depending on your requirements, continue as follows:
 - To create a new group, see [Section 37.2.2, "Creating a Group."](#)
 - To add users to a group, see [Section 37.2.3, "Adding Users to a Group."](#)
 - To configure security privileges for REST resources, see [Section 37.2.4, "Configuring Security for REST Resources."](#)

Note: To complete this step, ensure that the required groups exist. You will be assigning privileges to the groups.

37.2.2 Creating a Group

1. Log in to the WebCenter Sites Admin interface as a general administrator:
2. In the WebCenter Sites tree, select the **Admin** tab. Expand the **User Access Management** node, expand the **REST Security** node, and then expand **Groups**. Double-click **Add New**.
3. In the Add New Group form, enter a name and brief description about the group you are creating.

Figure 37–3 Add New Group Form

Add New Group

*Name:

*Description:

Cancel

Save

4. Click **Save**.
The group you created is now listed under the **Groups** node.
5. Now that you have created a group, you can:
 - Add users to the group. For instructions, see [Section 37.2.3, "Adding Users to a Group."](#)
 - Configure REST security for the group. For instructions, see [Section 37.2.4, "Configuring Security for REST Resources."](#)

37.2.3 Adding Users to a Group

Adding users to a group determines their permissions to operate on REST resources used by the applications the users will access.

To add users to a group

1. Log in to the WebCenter Sites Admin interface as a general administrator:

2. In the WebCenter Sites tree, select the **Admin** tab. Expand the **User Access Management** node, then expand the **REST Security** node, then expand **Assign Users to Groups**. Double-click **Add New**.
3. In the Assign Groups to User form, select users and assign them to any combination of the listed groups.

Figure 37-4 Assign Groups to User Form

Assign Groups to User

The screenshot shows a web form titled "Assign Groups to User". It has two primary input fields. The first is labeled "*User Name:" and is a dropdown menu with a list of user names: Arthur, Connie, Conrad, DefaultReader, and demo. The second is labeled "*Groups:" and is a text area containing the text "RestAdmin" and "SiteAdmin_AdminSite". Below these fields are two buttons: "Cancel" and "Save".

Note: If the user you are looking to assign to the group is not listed, that user is already a member of a group. To assign the user to another group, see step 5.

4. Click **Save**.
The user names you selected are listed under the **Assign Users to Groups** node. When you double-click a user's name you are able to view the groups to which that user is a member.
5. (Optional). If the name of the user you wish to assign to a given group is not displayed in the User Name field, then do the following:
 - a. In the WebCenter Sites tree, select the **Admin** tab. Expand the **User Access Management** node, expand the **REST Security** node then expand **Assign Users to Groups**, and double-click the name of the user you wish to assign to another group.
 - b. In the user's Inspect form, click **Edit** to render the Edit User Groups form.
 - c. In the **Groups** field, select the groups you wish to assign the user to, and then click **Save**.
6. Now that you have added users to a group, you can do the following:
 - Create a new group. For instructions see [Section 37.2.2, "Creating a Group."](#)

- Configure security for a group. For instructions, see [Section 37.2.4, "Configuring Security for REST Resources."](#)

37.2.4 Configuring Security for REST Resources

When configuring security, you will specify which object types and objects must be accessible to groups, and which actions the groups can perform on the objects.

To configure security for REST resources

1. Log in to the WebCenter Sites Admin interface as a general administrator:
2. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.
3. In the Add New Security Configuration form, you can set security for object types and objects.

Figure 37–5 Add New Security Configuration Form

Add New Security Configuration

*Type:

*Name:

*Groups:

*Action:

See the following tables for a summary of possible security configurations.

Table 37–1 Available Actions (Security Privileges)

Action	Description
Create	Create allows groups to create specified resources.
Delete	Allows groups to delete specified resources.
List	Allows groups to retrieve specified resources.

Table 37–1 (Cont.) Available Actions (Security Privileges)

Action	Description
Read/Head	Allows groups to read specified resources. While Read returns the requested resources, Head returns metadata describing the requested resources.
Update	Update allows groups to modify specified resources. Note: Create and Update are each paired with the Read/Head privilege. Assigning one of these privileges to a group automatically assigns the Read/Head privilege to the group.

Table 37–2 Summary of Possible Security Configuration Options

Object Type	Name	Subtype	Site	Possible Actions	See Section ...
ACLs	Any	N/A	N/A	List	Section 37.3.1
Application ¹	Any	N/A	N/A	Create, Update, Delete	Section 37.3.2
Application	<i>AppName</i>	N/A	N/A	Update, Delete	Section 37.3.2
Asset	Any	N/A	Any	List, Read/Head, Create, Update, Delete	Section 37.3.3
Asset	Any	N/A	<i>SiteName</i>	List, Read/Head, Create, Update, Delete	Section 37.3.3
Asset	<i>AssetType</i>	N/A	<i>SiteName</i>	List, Read/Head ² , Create, Update, Delete	Section 37.3.3
Asset	<i>AssetType</i> and <i>AssetName</i>	N/A	<i>SiteName</i>	Read/Head, Update, Delete	Section 37.3.3
AssetType	Any	N/A	N/A	List, Read/Head, Create, Delete	Section 37.3.4
AssetType	<i>AssetType</i>	N/A	N/A	Read/Head, Delete	Section 37.3.4
AssetType	<i>AssetType</i>	Any	N/A	List	Section 37.3.4
AssetType	<i>AssetType</i>	<i>Subtype</i>	N/A	Read/Head	Section 37.3.4
Group	Any	N/A	N/A	List	Section 37.3.6
Group	<i>GroupName</i>	N/A	N/A	Read/Head	Section 37.3.6

Table 37-2 (Cont.) Summary of Possible Security Configuration Options

Object Type	Name	Subtype	Site	Possible Actions	See Section ...
Index	Any	N/A	N/A	List, Read/Head, Create, Update, Delete	Section 37.3.7
Index	<i>IndexName</i>	N/A	N/A	Read/Head, Update, Delete	Section 37.3.7
Role	Any	N/A	N/A	List, Read/Head, Create, Update, Delete	Section 37.3.8
Role	<i>Role</i>	N/A	N/A	Read/Head, Update, Delete	Section 37.3.8
Site	Any	N/A	N/A	List, Read/Head ³ , Create, Update, Delete	Section 37.3.9
Site	<i>SiteName</i>	N/A	N/A	Read/Head, Update, Delete	Section 37.3.9
User	Any	N/A	N/A	List, Read/Head, Create, Update, Delete	Section 37.3.10
User	<i>UserName</i>	N/A	N/A	Read/Head, Update, Delete	Section 37.3.10
UserDef	Any	N/A	N/A	List	Section 37.3.11
UserLocales	Any	N/A	N/A	List	Section 37.3.12

¹ For an example of setting security for applications, see step 3.

² READ allows reading associations on the named site.

³ READ allows reading users and asset types on the named site.

37.3 REST Security Configuration Reference

This reference supports [Table 37-2, "Summary of Possible Security Configuration Options"](#). It provides details of the tabulated security configurations.

This reference contains the following sections:

- [Section 37.3.1, "Configuring REST Security for ACL Resources"](#)
- [Section 37.3.2, "Configuring REST Security for Application Resources"](#)
- [Section 37.3.3, "Configuring REST Security for Asset Resources"](#)
- [Section 37.3.4, "Configuring REST Security for Asset Type Resources"](#)
- [Section 37.3.5, "Configuring REST Security for Engage Resources"](#)

- [Section 37.3.6, "Configuring REST Security for Group Resources"](#)
- [Section 37.3.7, "Configuring REST Security for Indexed Asset Type Resources"](#)
- [Section 37.3.8, "Configuring REST Security for Role Resources"](#)
- [Section 37.3.9, "Configuring REST Security for Site Resources"](#)
- [Section 37.3.10, "Configuring REST Security for User Resources"](#)
- [Section 37.3.11, "Configuring REST Security for UserDef Resources"](#)
- [Section 37.3.12, "Configuring REST Security for UserLocale Resources"](#)
- [Section 37.3.13, "Configuring REST Security for Visitor Resources"](#)

37.3.1 Configuring REST Security for ACL Resources

When assigning groups security privileges to ACLs, you determine which groups will be able to view the ACL resource list.

To configure group security for ACLs

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the **Type** field, select **ACLs**.
3. In the **Name** field, select **Any**.

Figure 37–6 Add New Security Configuration for ACLs

Add New Security Configuration

*Type:

*Name:

*Groups:

*Action:

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–3 Add New Security Configuration Form Field Names

Field	Definition
Name	The only available option to make all ACLs available to the group(s).
Groups	Select the group(s) that will have access to the ACLs.
Action	The only available security privilege you can assign to the group(s) is to view the ACLs resource list.

37.3.2 Configuring REST Security for Application Resources

When assigning groups security privileges to applications, you determine which groups can perform which operations on the specified applications.

To configure group security for applications

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Application**.

- In the Name field, select **Any**.

Figure 37–7 Add New Security Configuration

Add New Security Configuration

*Type:

*Name:

*Groups:

*Action:

- Select the Groups and Actions as needed.
- Click **Save** to save the configuration.

Field Definitions:

Table 37–4 Add New Security Configuration Field Names

Field	Definition
Name	Select the name of the application you wish to make available to the group(s), or select Any to make all applications available to the group(s).
Groups	Select the group(s) that will have privileges to operate on the application(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected Any and Create , members of your selected groups will be able to create the assets which make the applications accessible in WEM.

37.3.3 Configuring REST Security for Asset Resources

When assigning groups security privileges to assets, you determine which groups can perform which operations on the specified assets.

To configure group security for Assets

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Asset**.
3. In the Site field, select the appropriate site.
4. In the Name field, select **Any**.

Figure 37–8 Add New Security Configuration for Assets

Add New Security Configuration

*Type:

Site:

*Name:

*Groups:

*Action:

5. Select the Groups and Actions as needed.
6. Click **Save** to save the configuration.

Field Definitions:

Table 37–5 Add New Security Configuration Form Field Names

Field	Definition
Site	Select the site associated with the asset you wish to make available to the group(s), or select Any to make all assets, system wide, available to the group(s).

Table 37-5 (Cont.) Add New Security Configuration Form Field Names

Field	Definition
Name	Select the asset type associated with the asset you wish to make available to the group(s), or select Any to make all assets available to the group(s). You can also make a specified asset of the selected asset type available to the group(s) by clicking the Browse button.
Groups	Select the group(s) that will have privileges to operate on the asset(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected a specific site, a specific asset type, and List , members of your selected groups will be able to perform searches in the specified site for assets of the specified asset type.

37.3.4 Configuring REST Security for Asset Type Resources

When assigning groups security privileges to asset types, you determine which groups can perform which operations on the specified asset types.

To configure group security for asset types

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security** , then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Asset Type**.
3. In the Name field, select **Any**.

Figure 37–9 Add New Security Configuration Form for Field Names**Add New Security Configuration**

***Type:**

***Name:**

***Groups:**

***Action:**

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:**Table 37–6 Add New Security Configuration Field Names**

Field	Definition
Name	Select the asset type(s) you wish to make available to the group(s), or select Any to make all asset types available to the group(s).
Subtype	(Optional) Select the subtype of the asset type you wish to make available to the group(s). Note: If you selected the Any option in the Name field, then the Subtype field is not displayed.
Groups	Select the group(s) that will have privileges to operate on the asset type(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected Any and Create , members of your selected groups will be able to create asset types.

37.3.5 Configuring REST Security for Engage Resources

When assigning groups security privileges to Engage resources, you determine which groups can perform which operations on Engage.

To configure group security for groups

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security** , then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Engage**.
3. In the Site field, select the appropriate site.
4. In the Name field, select **Any**.

Figure 37–10 Add New Security Definition Form for Engage

Add New Security Configuration

*Type: Engage

Site: Select...

*Name: Select...

*Groups: RestAdmin
SiteAdmin_AdminSite

*Action: Create
Delete
List
Read/Head
Update

Cancel Save

5. Select the Groups and Actions as needed.
6. Click **Save** to save the configuration.

Field Definitions:

Table 37–7 Add New Security Configuration Form Field Names

Field	Definition
Site	Select the site associated with the asset you wish to make available to the group(s), or select Any to make all assets, system wide, available to the group(s).

Table 37-7 (Cont.) Add New Security Configuration Form Field Names

Field	Definition
Name	Select the asset type associated with the asset you wish to make available to the group(s), or select Any to make all assets available to the group(s). You can also make a specified asset of the selected asset type available to the group(s) by clicking the Browse button.
Groups	Select the group(s) that will have privileges to operate on the asset(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected a specific site, a specific asset type, and List , members of your selected groups will be able to perform searches in the specified site for assets of the specified asset type.

37.3.6 Configuring REST Security for Group Resources

When assigning groups security privileges to groups, determine which groups can perform which operations on the specified groups.

To configure group security for groups

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Group**.
3. In the Name field, select **Any**.

Figure 37–11 Add New Security Definition Form for Groups**Add New Security Configuration**

***Type:**

***Name:**

***Groups:**

***Action:**

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:**Table 37–8 Add New Security Configuration Field Names**

Field	Definition
Name	Select the group(s) you wish to make available to the groups, or select Any to make all groups available to the groups.
Groups	Select the group(s) that will have privileges to operate on the groups.
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected Any and List , members of your selected groups will be able to view a listing of the system's groups.

37.3.7 Configuring REST Security for Indexed Asset Type Resources

When assigning groups security privileges to indexed asset types, you determine which groups can perform which operations on the specified indexed asset types.

Note: Before configuring security for indexed asset types, you must enable indexing for the WebCenter Sites "Global Search" and "Asset Type Search." If these search features are not enabled, then you will not be able to configure security for indexed asset types.

When assigning groups security privileges to groups, determine which groups can perform which operations on the specified groups.

To configure group security for indexed asset types

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Group**.
3. In the Name field, select **Any**.

Figure 37–12 Add New Security Configuration Form for Indexed Asset Types

Add New Security Configuration

***Type:**

***Name:**

***Groups:**

***Action:**

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–9 Add New Security Configuration Form Field Names

Field	Definition
Name	Select the name of the indexed asset type you wish to make available to the group(s). Select Any to make all indexed asset types available to the group(s). Select Global to make all indexed asset types associated with the "Global Search" available to the group(s).

Table 37–9 (Cont.) Add New Security Configuration Form Field Names

Field	Definition
Groups	Select the group(s) that will have privileges to operate on the indexed asset type(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected Any and List , members of your selected groups will be able to search for assets of all types that are indexed on the system.

37.3.8 Configuring REST Security for Role Resources

When assigning groups security privileges to roles, you determine which groups can perform which operations on the specified roles.

To configure group security for role resources

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Role**.
3. In the Name field, select **Any**.

Figure 37–13 Add New Security Configuration Form for Role Resources

Add New Security Configuration

*Type:

*Name:

*Groups:

*Action:

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:**Table 37–10 Add New Security Configuration Form Field Names**

Field	Definition
Name	Select the name of the role you wish to make available to the group(s), or select Any to make all roles available to the group(s).
Groups	Select the user group(s) that will have privileges to operate on the role(s).
Action	Assign the security privilege(s) to the group(s). Your options depend on your selections in the previous fields. For example, if you selected Any and Create , members of your selected groups will be able to create roles.

37.3.9 Configuring REST Security for Site Resources

When assigning groups security privileges to sites, you determine which groups can perform which operations on the specified sites.

To configure group security for sites

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Site**.
3. In the Name field, select **Any**.

Figure 37–14 Add New Security Configuration for Site**Add New Security Configuration**

***Type:** Site

***Name:** Select ...

***Groups:** RestAdmin
SiteAdmin_AdminSite

***Action:** Create
Delete
List
Read/Head
Update

Cancel Save

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–11 Add New Security Configuration Form Field Names

Field	Definition
Name	Select the name of the site you wish to make available to groups, or select Any to make all sites available to groups.
Groups	Select the user group(s) that will have privileges to operate on the sites.
Action	Assign the security privilege(s) to the group(s). Your menu options depend on your selections in the previous fields. For example, if you selected Any and Create , members of your selected groups will be able to create sites.

37.3.10 Configuring REST Security for User Resources

When assigning groups security privileges to users, you determine which groups can perform which operations on the specified users.

To configure group security for user resources

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **User**.
3. In the Name field, select **Any**.

Figure 37–15 Add New Security Configuration for User Resources**Add New Security Configuration**

***Type:**

***Name:**

***Groups:**

***Action:**

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:**Table 37–12 Add New Security Definition Field Names**

Field	Definition
Name	Select the name of the user you wish to make available to groups, or select the Any option to make all users available to groups.
Groups	Select the group(s) that will have privileges to operate on the user(s).
Action	Assign the security privilege(s) to the group(s). Your menu options depend on your selections in the previous fields. For example, if you selected Any and Create , members of your selected groups will be able to create users.

37.3.11 Configuring REST Security for UserDef Resources

When assigning groups security privileges to user definitions, you determine which groups can view the system's user definitions.

To configure group security for userdef resources

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **UserDef**.
3. In the Name field, select **Any**.

Figure 37–16 Add New Security Configuration Form for UserDef Resources

Add New Security Configuration

***Type:**

***Name:**

***Groups:**

***Action:**

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–13 Add New Security Configuration Form Field Names

Field	Definition
Name	The only available option is to make all user definitions available to groups.
Groups	Select the group(s) that will have privileges to view user definitions.
Action	The only available security privilege you can assign to the group(s) is Read/Head , which enables the members of your selected groups to view your system's user definitions.

37.3.12 Configuring REST Security for UserLocale Resources

When assigning groups security privileges to user locales, you determine which groups can view the UserLocale resource list.

To configure group security for userlocale resources

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **UserLocales**.
3. In the Name field, select **Any**.

Figure 37–17 Add New Security Configuration Form for User Locale

Add New Security Configuration

*Type: UserLocales

*Name: Select ...

*Groups: RestAdmin
SiteAdmin_AdminSite

*Action: Create
Delete
List
Read/Head
Update

Cancel Save

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–14 Add New Security Configuration Field Names

Field	Definition
Name	The only available option is to make all user locales available to groups.
Groups	Select the group(s) that will have privileges to view user locales.
Action	The only available security privilege you can assign to the group(s) is to view a listing of the system's user locales.

37.3.13 Configuring REST Security for Visitor Resources

When assigning groups security privileges to visitors, you determine which groups can view the Visitor assets.

To configure group security for visitor resources

1. In the WebCenter Sites tree select the **Admin** tab, expand the **User Access Management** node, expand the **REST Security**, then expand **Configure Security**. Double-click **Add New**.

The Add New Security Configuration screen appears.

2. In the Type field, select **Visitor**.
3. In the Name field, select **Any**.

Figure 37–18 Add New Security Configuration Form for Visitors

Add New Security Configuration

*Type:

*Name:

*Groups:

*Action:

4. Select the Groups and Actions as needed.
5. Click **Save** to save the configuration.

Field Definitions:

Table 37–15 Add New Security Configuration Field Names

Field	Definition
Name	The only available option is to make all visitors available to groups.
Groups	Select the group(s) that will have privileges to view user locales.

Table 37-15 (Cont.) Add New Security Configuration Field Names

Field	Definition
Action	The only available security privilege you can assign to the group(s) is to view a listing of the visitors.

Working with Sites

This chapter provides instructions for managing sites that are created in WEM Admin but used in the WebCenter Sites applications, and vice versa.

This chapter contains the following sections:

- [Section 38.1, "Managing WebCenter Sites CM Sites in the WEM Framework"](#)
- [Section 38.2, "Enabling Tree Tabs"](#)

38.1 Managing WebCenter Sites CM Sites in the WEM Framework

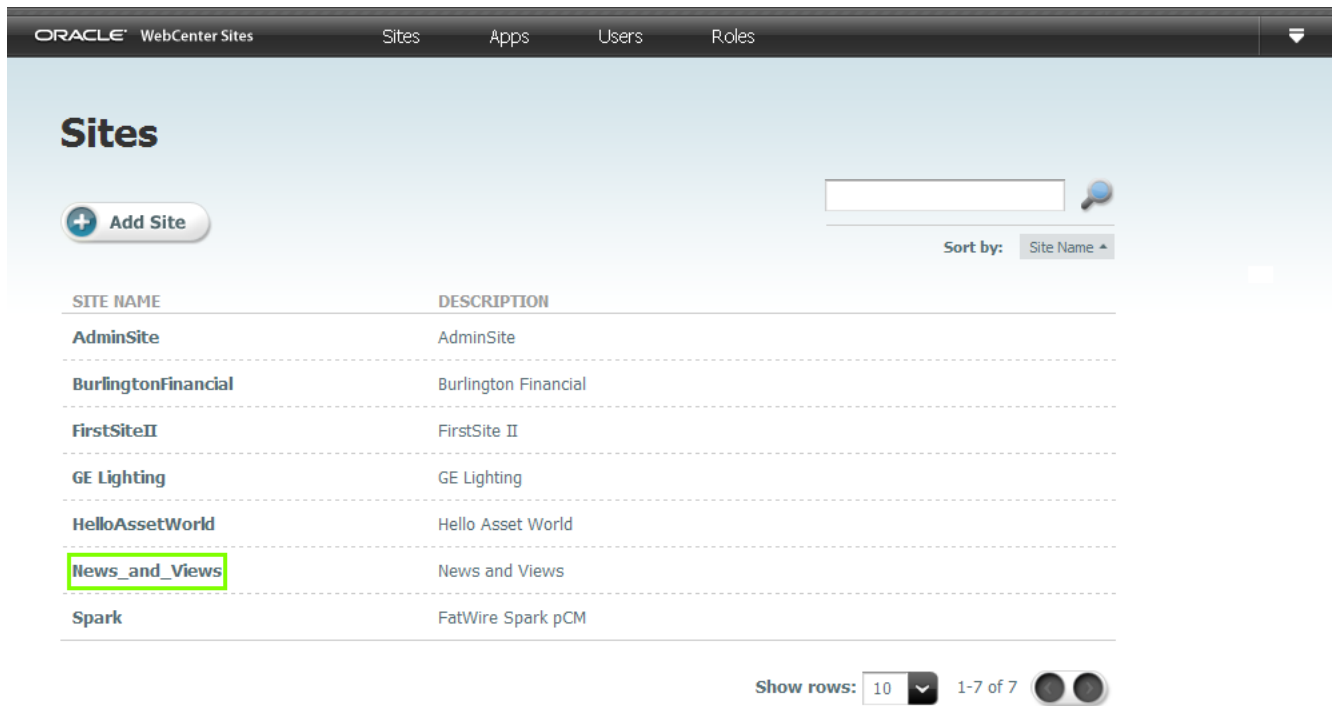
When using the WEM Admin interface to delete or modify sites, ensure they are not active content management sites in the WebCenter Sites platform. Deleting sites from the WEM Admin interface deletes the sites from the system and from any applications where the sites are in use. The same applies to other objects, such as roles, with one difference.

When you attempt to delete a role that is assigned to users and applications, you will be presented with a list of dependencies which you must clear in order to delete the role. When you attempt to delete a site you will be prompted to confirm your decision. It is assumed that you have determined the status of the site.

In WEM Admin you can expose active WebCenter Sites CM sites for quick access by configuring the WebCenter Sites applications to run on those sites and assigning yourself to those sites. For example:

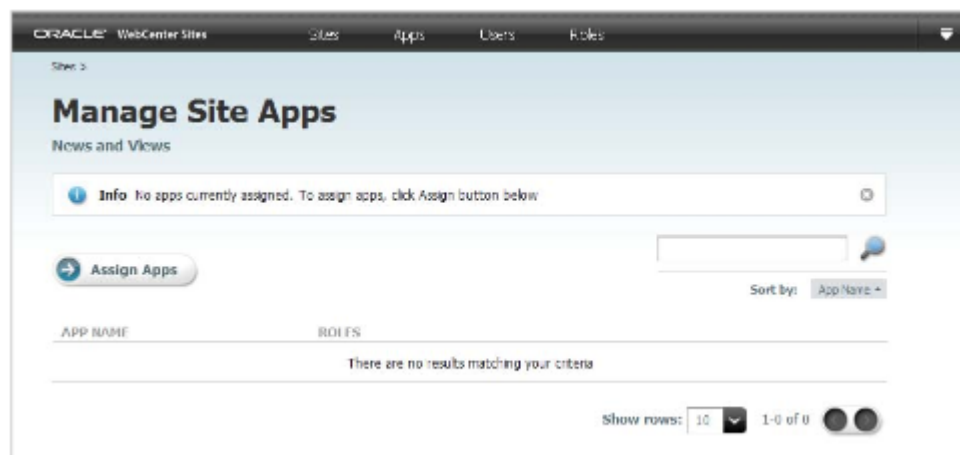
1. If you configure a site named "News_and_Views" in the WebCenter Sites Admin interface, the site is listed on the Sites form in WEM Admin:

Figure 38–1 WEM Admin Sites Form



2. Navigate to **News_and_Views** and click **Manage Site Apps**.

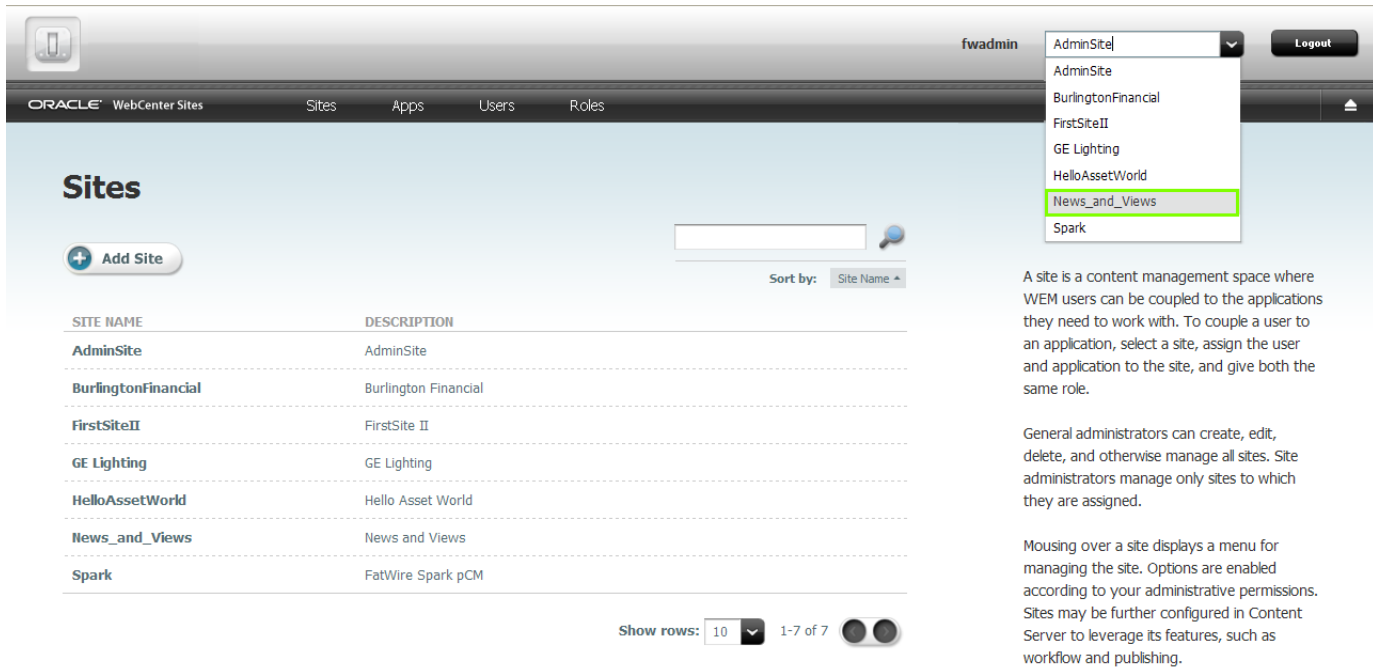
Figure 38–2 Manage Site Apps Form



With no assigned applications, the site may be regarded as inactive.

3. When you assign the WebCenter Sites Admin application and yourself to "News_and_Views," the site is listed as an option in the drop-down menu (refresh the display, if necessary).

Figure 38–3 Sites Form, Showing Pull-Down Menu



4. Selecting **News_and_Views** from the menu displays the login dialog. The site is listed in the drop-down menu; below is the application icon for WebCenter Sites Admin.

Figure 38–4 WebCenter Sites Login Dialog



Once logged in, you will see the WebCenter Sites Admin application icon at the left and the name of the site at the right, in the drop-down menu. Subsequent access does not require login.

Figure 38–5 Admin Form



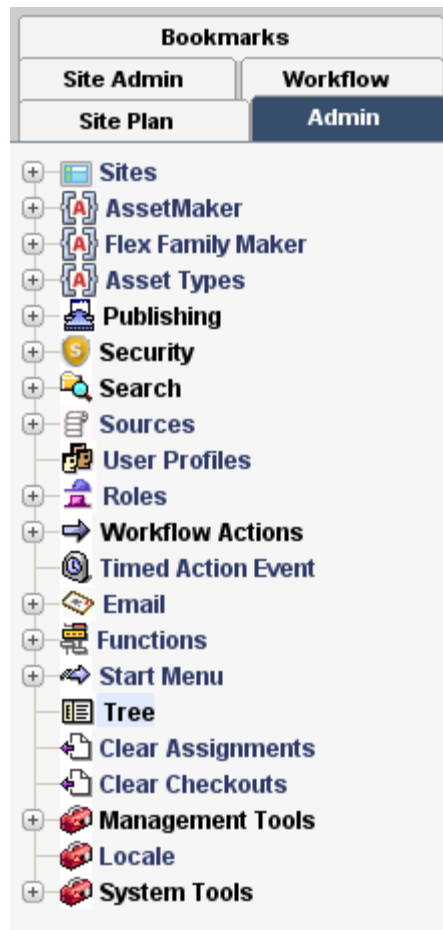
38.2 Enabling Tree Tabs

Sites created in the WEM Admin interface are exposed in the WebCenter Sites Admin interface, but not configured for use in the WebCenter Sites Admin interface. A general administrator must enable operations on each site by enabling the WebCenter Sites tree, tab by tab.

Enabling the WebCenter Sites Admin interface tree and tabs for a WEM site

1. Log in to the WebCenter Sites Admin interface as a general administrator:
 - a. Navigate to the following URL:
`http://<server>:<port>/<cs_context>/Xcelerate/LoginPage.html`
 - b. Enter your user name and password.
 - c. Click **Login**.
2. Enable the relevant WebCenter Sites tree tabs for the site that you created in the WEM Admin interface.
 - a. Navigate to the WebCenter Sites tree and select the **Admin** tab.
 - b. Double-click the **Tree** node.

Figure 38–6 Admin Tab



- c. Individually select each tab you wish to enable. For example, suppose you want to enable the **Site Design** tab:
 - a. From the list of tree tabs, click **Site Design**.
 - b. In the Tree Tab form, click **Edit**.
The Edit form is rendered (for the **Site Design** tab).

Figure 38–7 Edit Tree Tab Site Design Form

Ctrl-click the sites where you wish to enable the Site Design tab

The screenshot shows the 'Edit Tree Tab: Site Design' form. It contains the following elements:

- Title:** Site Design
- Tooltip:** Site Design
- Sites:** A dropdown menu with options: Any, FirstSite, AdminSite, wiposports, **News_and_Videos** (highlighted), and Management Site.
- Required Roles:** A dropdown menu with options: Any, **AdminEditRole** (highlighted), Analytics, Approver, ArticleAuthor, and ArticleEditor.
- Tab Contents:** Two lists. The 'Available' list includes Article, Article Category, Attribute, Collection, **Content** (highlighted), Content Parent, Definition, and Document. The 'Selected' list includes Article. There are 'Add Selected Items' and 'Remove' buttons between the lists.
- Section Name:** History Definition
- Element Name:** OpenMarket%accelerate%AssetType%historyVal
- Buttons:** Add New Section, Edit, Cancel, Save.
- Note:**
 - To create a custom section, enter the section name and element name, click 'Add New Section', then Save. When the tree tab is reloaded, this element will be called.
 - To edit an existing custom section, select the custom section name (in the Selected area above), make the changes, click Edit, then Save.
- ID:** 1339827450446

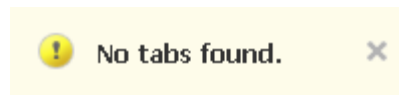
- c. In the **Sites** selection box, select sites on which the tab must be enabled.
 - d. Select roles that are allowed to access the tab.
 - e. Select content for the tab.
 - f. Click **Save**.
 - g. If you wish to enable additional tabs, click **List all Tree Tabs** (at the bottom of the Edit form) and continue as in the steps above.
3. Now that you have enabled the WebCenter Sites tree and tabs on the sites, continue to the next section to make sure the tree and the tabs are rendered properly.

Verify the Tree Tabs are Enabled

The tabs you have enabled in the previous section are accessible only to users who share a role with each tab. Verify site by site that the tree and tabs are enabled.

1. Log in to the WebCenter Sites Admin interface as a user who shares one of the roles assigned to a given tab (or tabs).
2. Select the site. You should see the WebCenter Sites tree, along with the tabs you enabled for the site.
3. If the WebCenter Sites tree is not enabled, you will see the following error message.

Figure 38-8 Error Message



Verify the site and role assignments for the user and the tab.

WEM Admin Interface Quick Reference

This chapter provides you with tips and a quick reference for using the Web Experience Management (WEM) Framework for managing and working with the sites, applications, users, and roles and includes the following sections:

- [Section 39.1, "Quick Tips for Managing the WEM Framework"](#)
- [Section 39.2, "Managing Sites"](#)
- [Section 39.3, "Managing Applications"](#)
- [Section 39.4, "Managing Users"](#)
- [Section 39.5, "Managing Roles"](#)
- [Section 39.6, "Managing Profiles"](#)

39.1 Quick Tips for Managing the WEM Framework

Before following instructions in the rest of this section, take note of a few tips:

- (Optional) If you are experimenting with WEM sites, users, and roles you may want to distinguish them from dedicated WebCenter Sites CM sites, users, and roles. For example, you can add a description for the site/user/role, or prefixing the name with "WEM_" (or a similar qualifier). Note that once a site, user, or role is created, its name cannot be changed.
- ACLs are required for user accounts. ACLs can be created only by general administrators and only in the WebCenter Sites **Admin** tab.
- Sharing a role to a user and an application grants the user access to the application in that site.
- Applications can have role-protected interface functions. Sharing a role to a user and an interface function grants the user access to the interface function.
- Groups provide access to REST. They must be configured in the WebCenter Sites Admin interface.
- Groups are used to control access to applications' resources.

If a custom-built application does not specify a predefined user, authorize application users at the application and REST levels.

If a custom-built application specifies a predefined user, authorize that user at the system, application, and REST levels. Authorize application users at the application level.

The rest of this section provides you with a quick reference to help you create and manage sites, applications, users, and roles in the WEM Admin interface.

39.2 Managing Sites

Only general administrators can create, edit, and delete sites. The Sites form is accessible to general and site administrators.

Table 39–1 Managing Sites

Admin	Action	Path
WEM General Admin	Create a site	WEM Admin interface, then Sites , then Add Site , then update the fields , then Save and Close Note: A site's name can only contain alphanumeric characters (no symbols).
WEM General Admin	Edit a site	WEM Admin interface, then Sites , then mouse over the site you wish to edit, then Edit , then modify the fields, then Save and Close Note: You cannot modify the name of a site.
WEM General Admin	Delete a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site you wish to delete, then Delete 2. In the warning box, click Delete.
WEM General and Site Admins	Add an application to a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site you wish to add an application to, then Manage Site Apps, then Assign Apps 2. In the Select Apps for Site form, select the applications you wish to add to the site and move them to the Selected list box. 3. Click Continue 4. In the Assign Roles to App form, select the roles you wish to assign to the applications and move them to the Selected list box. 5. Click Save and Close. <p>For general admins who want an alternate way to add an application to a site, see Section 39.3, "Managing Applications."</p>
WEM General and Site Admins	Assign users to a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site you wish to assign users to, then Manage Site Users, then Assign Users 2. In the Select Users for Site form, select users to assign to the site and move them to the Selected list box. 3. Click Continue. 4. In the Assign Roles to Users form, select the roles you wish to assign to the users and move them to the Selected list box. 5. Click Save and Close. <p>Note: For general admins who want an alternate way to assign users to a site, see Section 39.4, "Managing Users."</p>

Table 39–1 (Cont.) Managing Sites

Admin	Action	Path
WEM General and Site Admins	Reassign roles to a user	<p>Caution: Reassigning a user's roles on a site in WEM may uncouple the user from certain applications on that site.</p> <p>To reassign roles to a user:</p> <ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site whose user you wish to modify, then Manage Site Users 2. In the Manage Site Users form: mouse over the user whose site roles you wish to modify, then Assign Roles to User 3. In the Assign Roles to User form, assign new roles to the site user, or unassign roles, as necessary. 4. Click Save and Close. <p>Note: For general admins who want an alternate way to modify a user's roles in a site, see Section 39.4, "Managing Users."</p>
WEM General and Site Admins	Reassign roles to an application	<p>Caution: Reassigning an application's roles on a site in WEM may uncouple the application from certain users on that site. To reassign roles to an application:</p> <ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site whose application you wish to modify, then Manage Site Apps 2. In the Manage Site Apps form: mouse over the application whose site roles you wish to modify, then Assign Roles to App 3. In the Assign Roles to App form, assign new roles to the application, or unassign roles, as necessary. 4. Click Save and Close. <p>Note: For general admins who want an alternate way to modify an application's roles in a site, see Section 39.3, "Managing Applications."</p>
WEM General and Site Admins	Remove a user from a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site you wish to remove the user from, then Manage Site Users 2. In the Manage Site Users form: mouse over the user you wish to remove from the site, then Remove 3. In the warning box, click Remove. <p>Note: For general admins who want an alternate way to remove a user from a specific site, see Section 39.4, "Managing Users."</p>
WEM General and Site Admins	Remove an application from a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Sites, then mouse over the site you wish to remove an application from, then Manage Site Apps 2. In the Manage Site Apps form, mouse over the application you wish to remove from the site, then Remove 3. In the warning box, click Remove. <p>Note: For general admins who want an alternate way to remove an application from a site, see Section 39.3, "Managing Applications."</p>

39.3 Managing Applications

Only general administrators can modify applications (their descriptions). The Apps form is accessible only to general administrators.

Table 39–2 *Managing Applications*

Action	Path
Modify an application	<ol style="list-style-type: none"> 1. WEM Admin interface, then Apps, then mouse over the application you wish to modify, then Edit, then update the fields: Name: Cannot be modified. Tooltip: The name that appears when you mouse over an application's icon. Description: Short explanation describing the application. 2. Click Save and Close.
Assign an application to a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Apps, then mouse over the application you wish to assign to a site, then Manage App, then Assign to Sites 2. In the Select Sites for App form, select the sites to which you will assign the application and move them to the Selected list box. 3. Click Continue. 4. In the Assign Roles to App form, select the role(s) you wish to assign to the application and move them to the Selected list box. 5. Click Save and Close. <p>Note: For an alternate way to add an application to a site, see Section 39.2, "Managing Sites."</p>
Reassign roles to an application	<p>Caution: Reassigning an application's roles on a site in WEM may uncouple the application from certain users on that site.</p> <p>To reassign roles to an application:</p> <ol style="list-style-type: none"> 1. WEM Admin interface, then Apps, then mouse over the application whose roles you wish to modify for a select site, then Manage App 2. In the Manage App form: mouse over the site in which you wish to modify the application's roles, then Assign Roles to App 3. In the Assign Roles to App form, add roles to an application, or remove roles, as necessary. 4. Click Save and Close. <p>Note: For an alternate way to modify an application's roles in a site, see Section 39.2, "Managing Sites."</p>
Remove an application from a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Apps, then mouse over the application you wish to remove from the site, then Manage App 2. In the Manage App form, mouse over the site you wish to remove the application from, then Remove 3. In the warning box, click Remove. <p>Note: For an alternate way to remove an application from a site, see Section 39.2, "Managing Sites."</p>

39.4 Managing Users

Only general administrators can create, edit, and delete users. The Users form is accessible only to general administrators.

Table 39–3 *Managing Users*

Action	Path
Create a user	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then Add User, then update the fields: ACLs: ACLs determine the user's access permissions to the database. Note: You cannot create ACLs in the WEM Admin interface. Groups: Groups determine the user's access permissions to REST (applications' resources). Two default groups are configured: RestAdmin (a general administrator group) and SiteAdmin_AdminSite (a site administrator group). 2. Click Save and Close. Note: The user is able to log in. To access sites and applications, the user must be enabled.
Enable a user	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user you wish to enable, then Manage User, then Assign to Sites 2. In the Select Sites for User form: If you are enabling a general administrator, assign the user to AdminSite. If you are enabling a site administrator, assign the user to a site other than AdminSite. If you are enabling a regular user, assign the user to a site other than AdminSite. 3. Click Continue. 4. In the Assign Roles to User form: If you are enabling a general administrator, assign the user the GeneralAdmin role. If you are enabling a site administrator, assign the user the SiteAdmin role. If you are enabling a regular user, assign the user roles that are not GeneralAdmin or SiteAdmin. 5. Click Save and Close.
Assign a user to a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user you wish to assign to the site, then Manage User, then Assign to Sites 2. In the Select Sites for User form, select the site(s) that you wish to assign to the user and move them to the Selected list box. 3. Click Continue. 4. In the Assign Roles to User form, select the roles you wish to assign to the user and move them to the Selected list box. 5. Click Save and Close. For an alternate way to add a user to a site, see Section 39.4, "Managing Users."
Edit a user	<p>WEM Admin interface, then Users, then mouse over the user you wish to modify, then Edit, then update the desired fields, then Save and Close</p> <p>Note: You cannot modify a user's name once the user is saved.</p>
Reassign roles to a user	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user whose roles you wish to modify, then Manage User 2. In the Manage User form: mouse over the site you wish to modify the user's roles in, then Assign Roles to User 3. In the Assign Roles to User form add roles to a user, or remove roles, as necessary. 4. Click Save and Close.

Table 39–3 (Cont.) Managing Users

Action	Path
Assign a user to a group	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user you wish to assign to a group, then Edit 2. In the Groups field, select the appropriate group(s) and move them to the Selected list box. 3. Click Save and Close. <p>Note: Groups provide access to REST. They are used to control access to applications' resources.</p>
Remove a user from a site	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user you wish to remove from a site, then Manage User, then mouse over the site you wish to remove the user from, then Remove 2. In the warning box, click Remove. <p>For an alternate way to remove a user from a site, see Section 39.2, "Managing Sites."</p>
Delete a user from WEM	<ol style="list-style-type: none"> 1. WEM Admin interface, then Users, then mouse over the user you wish to remove from the system, then Delete 2. In the warning box, click Delete.

39.5 Managing Roles

Only general administrators can create, edit, and delete roles. The Roles form is accessible only to general administrators.

Table 39–4 Managing Roles

Action	Path
Add a role	WEM Admin interface, then Roles , then Add Role , then update the fields, then Save and Close
Edit a role	<p>WEM Admin interface, then Roles, then mouse over the role you wish to modify, then Edit, then update the desired fields, then Save and Close</p> <p>Note: You cannot modify a role's name once the role is saved.</p>
Delete a role	<ol style="list-style-type: none"> 1. WEM Admin interface, then Roles, then mouse over the role you wish to remove, then Delete 2. In the warning box, click Delete. <p>Note: If the role is assigned to users and/or applications, it cannot be deleted until you unassign the role from the user and applications. You will be presented with the Role Dependencies form is rendered. Review the table of dependencies. Delete all Dependencies removes the role from its users and applications and deletes the role from the system.</p>

39.6 Managing Profiles

General administrators can modify user profiles.

Table 39–5 Managing Profiles

User	Action	Path
WEM General Admin	Modify a user's profile	WEM Admin interface, then Users , then mouse over the user's name whose profile you wish to modify, then Edit , then modify the desired fields, then Save and Close
All WEM users	Modify your profile	WEM Admin interface, then open the applications bar, then click user name, then update the desired fields, then Save and Close

Part III

Administering Oracle WebCenter Sites: Community-Gadgets

The Oracle WebCenter Sites: Community-Gadgets web application provides the Community and Gadgets Web Experience Management (WEM) Framework applications. The Community WEM application supplies the Community interface and the Gadgets application supplies the Global Gadget Catalog interface and the Gadgets User interface. The chapters in this part provide instructions on enabling the Community and Gadgets interfaces and information about viewing the Community-Gadgets data that is stored on the production and management caches.

This part contains the following chapters:

- [Chapter 40, "Enabling the Community and Gadgets Interfaces"](#)
- [Chapter 41, "Working with the Cache Tool"](#)

Enabling the Community and Gadgets Interfaces

The Oracle WebCenter Sites: Community-Gadgets web application provides two separate Web Experience Management (WEM) applications on the Management WebCenter Sites system – the Community WEM application and the Gadgets WEM application. The Community application supplies the Community interface, and the Gadgets application supplies the Global Gadget Catalog interface and the Gadgets User interface.

This chapter provides information about enabling the Community interface (on sites other than AdminSite), the Global Gadget Catalog interface (on the site specified as the Community-Gadgets application site during the Community-Gadgets installation process), and the Gadgets User interface (on sites other than the site on which the Global Gadget Catalog interface is enabled).

This chapter contains the following sections:

- [Section 40.1, "Enabling the Community Interface"](#)
- [Section 40.2, "Enabling the Gadgets Interfaces"](#)

Note: The following feature is deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): Mirror publishing. This feature is replaced by RealTime publishing.

40.1 Enabling the Community Interface

The Community interface is enabled by administrators assigning the Community application to the desired site and then assigning the `GeneralAdmin`, `SiteAdmin`, `Moderator`, or `Designer` role to the site users who will be working in the interface. The Community interface provides configurable widget tags that can be deployed on any web pages to display Comments, Reviews, Ratings, Poll, and Login Bar widgets to site visitors. For detailed information about the Community interface, and instructions for working with the Community application, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Note: The site on which the Community interface will be enabled must exist on the WebCenter Sites delivery system. For information about mirroring a site, see [Chapter 16, "The Mirror to Server Publishing Process."](#)

This section contains the following topics:

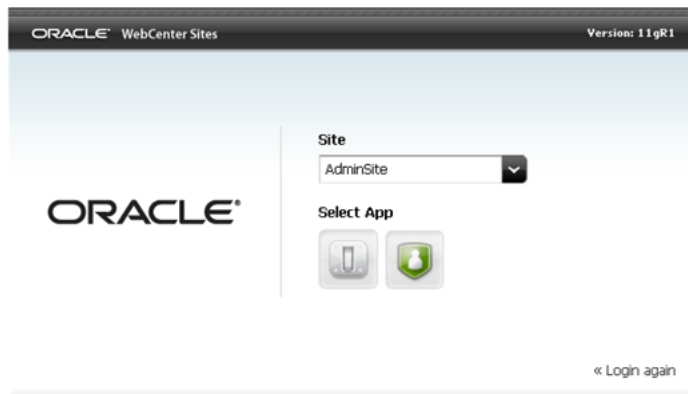
- [Section 40.1.1, "Enabling the Community Interface for Existing Site Users"](#)
- [Section 40.1.2, "Enabling the Community Interface for New Users"](#)

40.1.1 Enabling the Community Interface for Existing Site Users

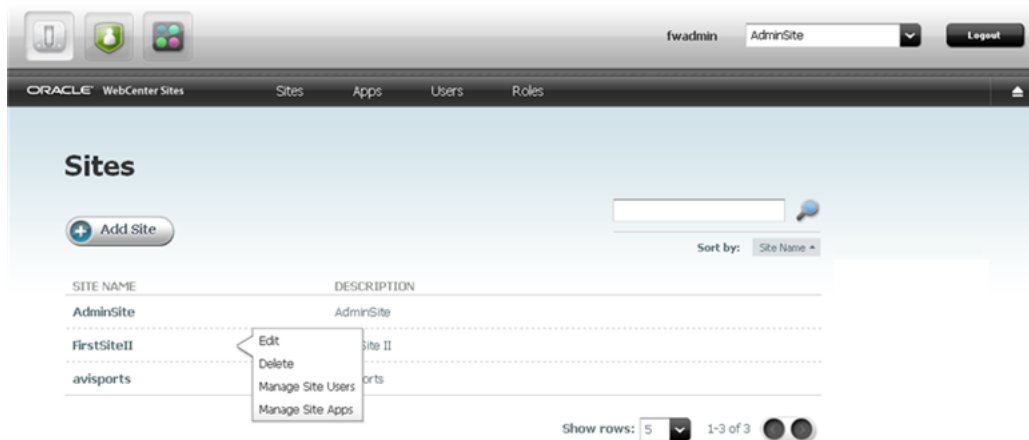
To enable site users to access the Community interface on their site, you must assign them at least one of the roles assigned to the Community application. The roles are: GeneralAdmin, SiteAdmin, Moderator, or Designer.

To enable the Community interface for existing site users

1. If the site on which you wish to enable the Community interface does not exist on the WebCenter Sites delivery system, mirror it. For instructions, see [Chapter 16, "The Mirror to Server Publishing Process."](#)
2. Log in to WebCenter Sites as a general or site administrator, access the **AdminSite**, and then select the **WEM Admin** application icon.



3. In the Sites screen, point to the site on which you wish to enable the Community interface, and click **Manage Site Apps**.



4. In the Manage Site Apps screen, click **Assign Apps**.
5. In the Select Apps for Site form, select the **Community** application from the **Available** list box, and move it to the **Selected** list box.
6. Click **Continue**.

7. In the Assign Roles to App form, select the **GeneralAdmin**, **SiteAdmin**, **Moderator**, and **Designer** roles from the **Available** list box, and move them to the **Selected** list box.

8. Click **Save and Close**.

The Community interface is now available to the site users assigned one of the roles assigned to the Community application (GeneralAdmin, SiteAdmin, Moderator, and Designer).

9. Next steps:

- If you wish to assign users roles to work in the Community interface, see [Section 40.1.2, "Enabling the Community Interface for New Users."](#)
- If you want to start working in the Community interface, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

40.1.2 Enabling the Community Interface for New Users

To enable new users to access a site's Community interface, you must assign the users to the site and then assign them at least one of the roles assigned to the Community application. The roles are GeneralAdmin, SiteAdmin, Moderator, and Designer. For detailed information about the roles assigned to the Community application, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

To enable the Community interface for new users

1. Log in to WebCenter Sites as a general or site administrator, select the **AdminSite**, and then select the **WEM Admin** application icon.
2. In the Sites screen, point to the site on which you enabled the Community interface, and click **Manage Site Users**.
3. In the Manage Site Users screen, click **Assign Users**.
4. In the Select Users for Site form, select the users you wish to assign to the site from the **Available** list box, and move them to the **Selected** list box.
5. Click **Continue**.
6. In the Assign Roles to User form, select either the **GeneralAdmin**, **SiteAdmin**, **Moderator**, or **Designer** role from the **Available** list box, and move the role to the **Selected** list box.
 - Selecting GeneralAdmin or SiteAdmin gives the users administrative access to the Community interface, and all other applications assigned to the site.
 - Selecting Moderator gives the users permissions to moderate the comments, reviews, and ratings that are posted to the pages on which community widget tags are deployed, configure moderation settings and auto-moderation filters for community widgets, and set visitor permissions to community widgets.
 - Selecting Designer gives the users permissions to configure the default appearance of community widgets, create and manage polls, deploy community widget tags, and modify a given community widget tag with custom values. These users can also specify the domain on which community widget tags can be deployed, and configure the language in which deployed community widgets will be displayed.
7. Click **Save and Close**.

The users you assigned to the site are able to access the Community interface on that site. For instructions on working in the Community interface, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

40.2 Enabling the Gadgets Interfaces

The Gadgets application provides the Global Gadget Catalog interface and the Gadgets User interface.

This section contains the following topics:

- [Section 40.2.1, "Enabling the Global Gadget Catalog Interface"](#)
- [Section 40.2.2, "Enabling the Gadgets User Interface"](#)

40.2.1 Enabling the Global Gadget Catalog Interface

The Global Gadget Catalog interface is enabled by general administrators assigning the Gadgets application to the site specified as the Community-Gadgets application site (AdminSite by default) during the installation process. The Global Gadget Catalog interface provides general administrators with the global gadget catalog. Gadgets registered to this catalog can be shared with any site gadget catalog.

Note: The site on which the Global Gadget Catalog interface will be enabled must exist on the WebCenter Sites delivery system. For information about mirroring a site, see [Chapter 16, "The Mirror to Server Publishing Process."](#)

To enable the Global Gadget Catalog interface

1. If the site specified as the Community-Gadgets application site (AdminSite by default) does not exist on the WebCenter Sites delivery system, mirror it. For instructions, see [Chapter 16, "The Mirror to Server Publishing Process."](#)
2. Log in to WebCenter Sites as a general administrator, select the **AdminSite**, and then select the **WEM Admin** application icon.
3. In the Sites screen, point to the site that was specified as the Community-Gadgets application site (AdminSite by default) during the installation process, and click **Manage Site Apps**.
4. In the Manage Site Apps screen, click **Assign Apps**.
5. In the Select Apps for Site form, select the **Gadgets** application from the **Available** list box and move it to the **Selected** list box.
6. Click **Continue**.
7. In the Assign Roles to App form, select the **GeneralAdmin** role from the **Available** list box, and move it to the **Selected** list box.
8. Click **Save and Close**.

The Global Gadget Catalog interface is now accessible to general administrators on the site that was specified as the Community-Gadgets application site (AdminSite by default) during the installation process.

9. Next steps:
 - If you wish to enable the Gadgets User interface on a site (other than the site on which the Global Gadget Catalog interface is enabled), see [Section 40.2.2](#),

["Enabling the Gadgets User Interface."](#)

- If you wish to register gadgets to the Global Gadgets Catalog, see [Chapter 44, "Registering Gadgets."](#) If you wish to manage the gadgets registered to the Global Gadget Catalog, see [Chapter 45, "Managing Gadgets."](#)

40.2.2 Enabling the Gadgets User Interface

The Gadgets User interface is enabled by administrators assigning the Gadgets application to sites other than the site on which the Global Gadget Catalog interface is enabled. A site's Gadgets User interface provides administrators with a site gadget catalog. It also provides all users (assigned the `GeneralAdmin`, `SiteAdmin`, or `Designer` role) with the site gadget catalog's associated dashboard, the **Login Bar** and **Settings** menus, and the screens for single gadget and dashboard deployment.

The site gadget catalog is used to register gadgets directly to the site and to contain gadgets that were shared from the global gadget catalog. Each gadget in the site gadget catalog is associated with its own Single Gadget Deployment screen which is used to display a single instance of that gadget to website visitors. The dashboard and its deployment screen are used to display multiple gadgets to visitors.

Note: The site on which you wish to enable the Gadgets User interface must exist on the WebCenter Sites' delivery system. For information about mirroring a site, see [Chapter 16, "The Mirror to Server Publishing Process."](#)

40.2.2.1 Enabling the Gadgets User Interface for Existing Site Users

To enable site users to access the Gadgets User interface on their site, you must assign them at least one of the roles assigned to the Gadgets application. The roles are `GeneralAdmin`, `SiteAdmin`, or `Designer`.

To enable the Gadgets User interface for existing site users

1. If the site on which you wish to enable the Gadgets User interface does not exist on the WebCenter Sites delivery system, mirror it. For instructions, see [Chapter 16, "The Mirror to Server Publishing Process."](#)
2. Log in to WebCenter Sites as a general or site administrator, select the **AdminSite**, and then select the **WEM Admin** application icon.
3. In the Sites screen, point to the site (other than the site on which the Global Gadget Catalog interface is enabled) on which you wish to enable the Gadgets User interface, and click **Manage Site Apps**.
4. In the Manage Site Apps screen, click **Assign Apps**.
5. In the Select Apps for Site form, select the **Gadgets** application from the **Available** list box and move it to the **Selected** list box.
6. Click **Continue**.
7. In the Assign Roles to App form, select the **GeneralAdmin**, **SiteAdmin**, and **Designer** roles from the **Available** list box, and move them to the **Selected** list box.
8. Click **Save and Close**.

The Gadgets User interface is now available to the site users assigned one of the roles that are assigned to the Gadgets application (`GeneralAdmin`, `SiteAdmin`, and `Designer`).

9. Next steps:

- If you want to assign users roles to work in the site's Gadgets User interface, see [Section 40.2.2.2, "Enabling the Gadgets User Interface for New Users."](#)
- If you want to register gadgets directly to the site's gadget catalog, see [Section 44.4, "Registering Gadgets to a Site Gadget Catalog."](#) For instructions on managing gadgets, see [Section 45.2.2, "Modifying a Gadget's Metadata in a Site Gadget Catalog."](#)
- For instructions about working with the gadgets that are registered to a site gadget catalog, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

40.2.2.2 Enabling the Gadgets User Interface for New Users

To enable new users to access a site's Gadgets User interface, you must assign the users to the site and then assign them at least one of the roles assigned to the Gadgets application. The roles are `GeneralAdmin`, `SiteAdmin`, and `Designer`. For detailed information about the roles assigned to the Gadgets application, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

To enable the Gadgets User interface for new users

1. Log in to WebCenter Sites as a general or site administrator, access the **AdminSite**, and then select the **WEM Admin** application icon.
2. In the Sites screen, point to the site on which you enabled the Gadgets User interface, and click **Manage Site Users**.
3. In the Manage Site Users screen, click **Assign Users**.
4. In the Select Users for Site form, select the users you wish to assign to the site from the **Available** list box, and move them to the **Selected** list box.
5. Click **Continue**.
6. In the Assign Roles to User form, select either the `GeneralAdmin`, `SiteAdmin`, or `Designer` role from the **Available** list box, and move the role(s) to the **Selected** list box.
 - Selecting `GeneralAdmin` or `SiteAdmin` gives the user(s) administrative access to the site's Gadgets User interface, and all the other applications assigned to that site.
 - Selecting `Designer` gives the user(s) access to the site's Gadgets User interface, and enables the user(s) to manage and deploy the site's gadgets and dashboard.
7. Click **Save and Close**.

The users you assigned to the site are able to access the Gadgets User interface on that site.

- For instructions on registering gadgets directly to the site's gadget catalog, see [Section 44.4, "Registering Gadgets to a Site Gadget Catalog."](#) For instructions on managing gadget metadata, see [Section 45.2.2, "Modifying a Gadget's Metadata in a Site Gadget Catalog."](#)
- For instructions on working with the gadgets that are registered to a site gadget catalog, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Working with the Cache Tool

This chapter provides information about using the Cache Tool to view the Community and Gadgets information that is stored in the production and management caches and to verify that those caches contain valid information.

This chapter contains the following sections:

- [Section 41.1, "Overview of the Cache Tool"](#)
- [Section 41.2, "Viewing Cache Information"](#)

41.1 Overview of the Cache Tool

All Community-Gadgets data (for all sites) is stored in and retrieved from the WebCenter Sites database. This includes user generated content (for example, comments, reviews, and user profiles), and settings configured in the Community and Gadget interfaces (for example, permission, appearance, and moderation settings).

When data is requested for the first time, it is read from the WebCenter Sites database and stored in either the management or production cache. A cache is a temporary storage area where data can be accessed quickly. Upon subsequent requests, the data is read directly from the cache in which it is stored (reducing the load on the database). The Cache Tool is used to view information about the data stored in the caches and to validate the data.

41.2 Viewing Cache Information

The first screen you see when you access the Cache Tool is the cache summary screen for either the management or production system. The summary screen contains statistics about the data stored on the management and production systems. Detailed information about the data contained within the caches on each system can be viewed by clicking the name of the desired cache in the summary screen.

To view cache information

1. Access the Cache Tool by doing one of the following:
 - To view information about the data stored in the management caches, enter `<mmmt-protocol>://<mmmt-host>:<mmmt-port>/<mmmt-context>/cache` where `<mmmt-host>:<mmmt-port>` is the host name and port number of the server running the management system.
 - To view information about the data stored in the production caches, enter `<prod-protocol>://<prod-host>:<prod-port>/<prod-context>/cache`

where <prod-host>:<prod-port> is the host name and port number of the server running the production system.

- In the Login screen, log in to the Cache Tool with your assigned credentials.

Note: To log in to the Cache Tool, you must be assigned the xceladmin ACL.

- Click **Login**.

You will see one of the following cache summary screens (Figure 41–1 and Figure 41–2):

Figure 41–1 Summary Management Screen

Cache Name	Total Count	Memory Store Count	Disk Store Count	Cache Hits	In Memory Hits	On Disk Hits	Cache Misses	Eviction Count	Average Get Time, ms	Max Elements In Memory	Max Elements On Disk	Overflow To Disk
Common	712	712	712	3141	3141	0	589	0	15	200000	10000000	true
Dependencies	445	445	445	15372	15372	0	1287	0	8	200000	1000000	true
Shared	20	20	20	552	552	0	874	0	56	200000	10000000	true

Figure 41–2 Summary Production Screen

Cache Name	Total Count	Memory Store Count	Disk Store Count	Cache Hits	In Memory Hits	On Disk Hits	Cache Misses	Eviction Count	Average Get Time, ms	Max Elements In Memory	Max Elements On Disk	Overflow To Disk
Common	178	178	178	747	747	0	226	0	23	200000	10000000	true
Dependencies	68	68	68	5654	5654	0	352	0	8	200000	1000000	true
Shared	71	71	71	580	580	0	1096	0	45	200000	10000000	true

The Summary Management screen contains statistics about the Community-Gadgets data stored in the management cache. This includes user generated content, widget settings, and any other data displayed in the Community and Gadgets interfaces. The Summary Production screen contains statistics about the Community-Gadgets data stored in the production cache. This includes user generated content, visitor profile information, and any other data displayed on the website.

Table 41–1 provides information about the columns in the summary screen. The summary screen also provides the option to flush the caches on the management (or production) system. To flush the data stored in the caches, select the desired cache and click **Flush Cache**, located at the top of the summary screen.

Table 41–1 Overview of Columns in the Summary Screen

Column	Description
Cache Name	The name of the cache where Community-Gadgets data is stored. The available caches are: <ul style="list-style-type: none"> ■ Common ■ Dependencies ■ Shared Click the desired cache name to display detailed information about the data stored in that cache.
Total Count	The number of objects currently stored in the cache.
Memory Store Count	The number of objects stored to memory.
Disk Store Count	The number of objects stored on disk.
Cache Hits	The total number of requested objects served from cache.
In Memory Hits	The total number of requested objects served from memory.
On Disk Hits	The total number of requested objects served from the disk.
Cache Misses	The total number of requested objects that were not stored in cache and so had to be read from the WebCenter Sites database.
Eviction Count	The total number of items that were removed from disk when there was no free disk space.
Average Get Time, ms	The average time (in milliseconds) it took the Community and Gadgets applications to retrieve requested data.
Max Elements In Memory	The maximum number of items allowed in the cache.
Max Elements On Disk	The maximum number of items allowed to be stored on disk.
Overflow to Disk	If set to <code>true</code> , items start to overflow to disk when there are more items in the memory store than are allowed by the <code>maxElementsInMemory</code> parameter.

4. View detailed information about the data contained within the caches on each system. In the **Cache Name** column of the summary screen, click the name of the desired cache. For more information, see:
 - [Viewing Information in the Common Cache](#)
 - [Viewing Information in the Dependencies Cache](#)
 - [Viewing Information in the Shared Cache](#)

41.2.1 Viewing Information in the Common Cache

To access the "Common" cache screen (shown in [Figure 41–3](#)), click **Common** in the "Cache Name" column of the summary screen.

Figure 41–3 Common Cache Screen

Summary >

Common

Delete

Sort by: **Key** ▾

<input type="checkbox"/>	Title	Hit Count	Created Time	Last Access Time	Dependencies
<input type="checkbox"/>	Template (Template) wsdk.ui.textarea/textarea_view.shtml Key: wsdk.ui.textarea/textarea_view.shtml	1	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.ui.textarea/textarea_model.js Key: wsdk.ui.textarea/textarea_model.js	3	Friday, May 17, 2013 4:56:51 PM	Monday, May 20, 2013 1:34:33 PM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.ui.textarea/textarea_action.js Key: wsdk.ui.textarea/textarea_action.js	1	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.ui.input/input_view.shtml Key: wsdk.ui.input/input_view.shtml	1	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.ui.input/input_model.js Key: wsdk.ui.input/input_model.js	1	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.ui.input/input_action.js Key: wsdk.ui.input/input_action.js	1	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template
<input type="checkbox"/>	Template (Template) wsdk.admin_deployment/messages/messages_view.shtml Key: wsdk.admin_deployment/messages/messages_view.shtml	3	Friday, May 17, 2013 4:56:51 PM	Saturday, May 18, 2013 6:41:58 AM	unknowndeps Template

Summary statistics for all entities used on production and management sides to render or manage widgets are shown here. Examples of entities are feeds, records, JavaScript sources, CSS skins, database query results, user profiles, and categories.

The common cache for the management system contains results of data queries necessary for displaying the Community and Gadgets interfaces to the Community and Gadgets applications' users. The common cache for the production system contains results of data queries necessary for displaying widgets and gadgets on the website for site visitors.

Table 41–2 provides information about the columns in this screen. The "Common" cache screen also allows you to view detailed information about an entry listed in the Common cache by pointing to the desired entry and selecting **Details** from the context menu. You can also delete an entry by pointing to it and selecting **Delete**.

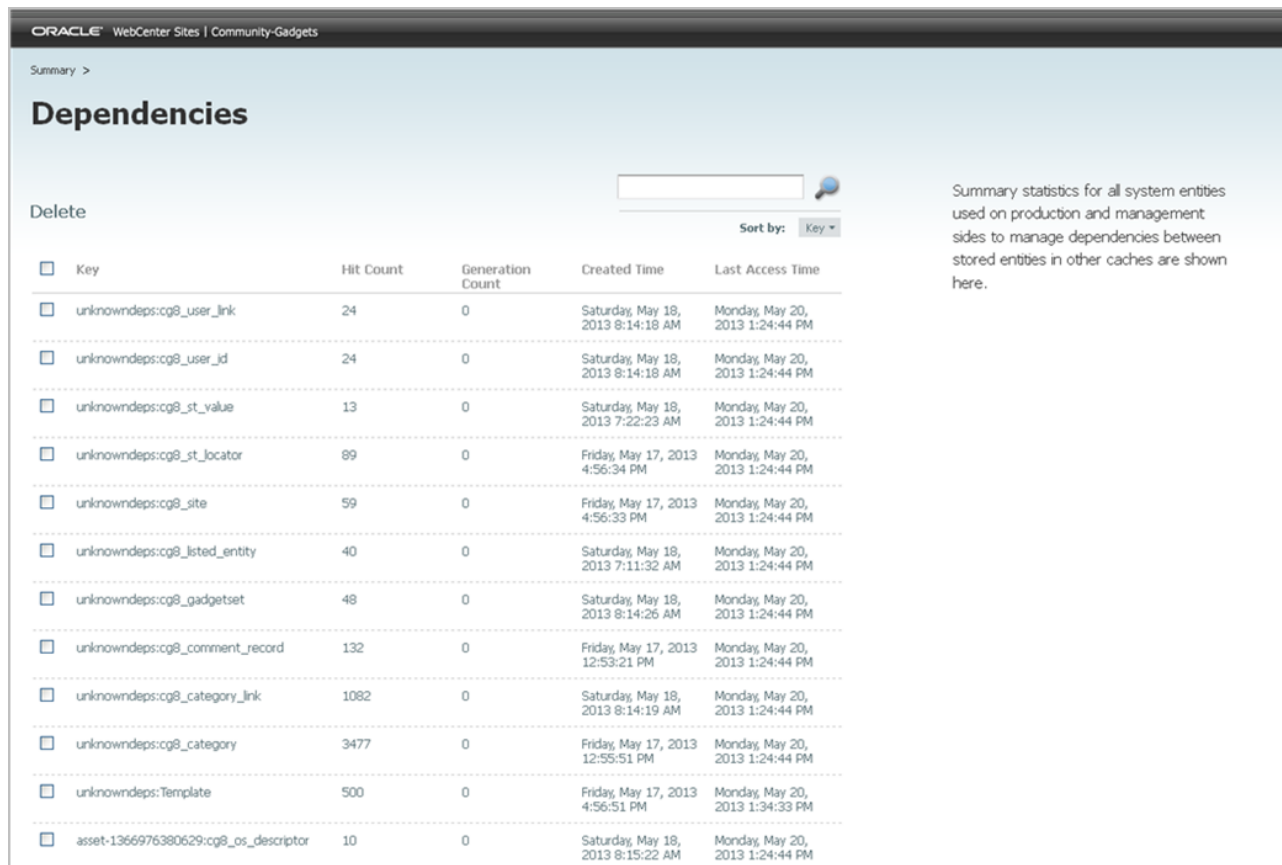
Table 41–2 Overview of Columns in the Common Cache Screen

Column	Description
Title	Uniquely identifies each item in the common cache.
Hit Count	The total number of requested objects that were served by reading the common cache.
Created Time	The date the data was fetched from the WebCenter Sites database and then stored in the cache for future access.
Last Access Time	The most recent date the data was retrieved from the common cache.
Dependencies	Lists the assets in the WebCenter Sites database on which the corresponding data depends. If the data is associated with more than one database table, multiple dependencies will be listed for that entry.

41.2.2 Viewing Information in the Dependencies Cache

To access the "Dependencies" cache screen (shown in [Figure 41-4](#)), click **Dependencies** in the "Cache Name" column of the summary screen.

Figure 41-4 Dependencies Cache Screen



Key	Hit Count	Generation Count	Created Time	Last Access Time
unknowndeps:cg8_user_link	24	0	Saturday May 18, 2013 8:14:18 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_user_id	24	0	Saturday May 18, 2013 8:14:18 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_st_value	13	0	Saturday May 18, 2013 7:22:23 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_st_locator	89	0	Friday May 17, 2013 4:56:34 PM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_site	59	0	Friday May 17, 2013 4:56:33 PM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_listed_entity	40	0	Saturday May 18, 2013 7:11:32 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_gadgetset	48	0	Saturday May 18, 2013 8:14:26 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_comment_record	132	0	Friday May 17, 2013 12:53:21 PM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_category_link	1082	0	Saturday May 18, 2013 8:14:19 AM	Monday May 20, 2013 1:24:44 PM
unknowndeps:cg8_category	3477	0	Friday May 17, 2013 12:55:51 PM	Monday May 20, 2013 1:24:44 PM
unknowndeps:Template	500	0	Friday May 17, 2013 4:56:51 PM	Monday May 20, 2013 1:34:33 PM
asset-1366976380629:cg8_os_descriptor	10	0	Saturday May 18, 2013 8:15:22 AM	Monday May 20, 2013 1:24:44 PM

All links to data in the database tables are stored in the dependencies cache. The "Dependencies" cache screen lists query dependencies that exist on the management (or production) system. [Table 41-3](#) provides information about the columns in this screen.

Table 41-3 Overview of Columns in the Dependencies Cache screen

Column	Description
Key	Uniquely identifies each item in the dependencies cache.
Hit Count	The total number of requested objects that were served by reading the dependencies cache.
Generation Count	The total number of changes made in the database to the corresponding entry since the application was started.
Created time	The date the object was fetched from the WebCenter Sites database and then stored in the cache for future access.
Last Access Time	The most recent date the data was retrieved from the dependencies cache.

41.2.3 Viewing Information in the Shared Cache

To access the "Shared" cache screen (shown in [Figure 41-5](#)), click **Shared** in the Cache Name column of the summary screen.

Figure 41-5 Shared Cache Screen

Title	Hit Count	Created Time	Last Access Time	Dependencies
SSO-ST-78-Rk2a0JULeJj0J0AvzLXK-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-78-Rk2a0JULeJj0J0AvzLXK-cas-webcentersites.ua.oracle.com-1	12	Monday, May 20, 2013 2:52:16 PM	Monday, May 20, 2013 2:56:28 PM	
SSO-ST-77-zadbGlgKt7c7UHTejUy-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-77-zadbGlgKt7c7UHTejUy-cas-webcentersites.ua.oracle.com-1	8	Monday, May 20, 2013 1:31:46 PM	Monday, May 20, 2013 1:36:00 PM	
SSO-ST-76-INdOW4mPlLokCe9CSWaRt-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-76-INdOW4mPlLokCe9CSWaRt-cas-webcentersites.ua.oracle.com-1	30	Monday, May 20, 2013 11:58:53 AM	Monday, May 20, 2013 11:59:02 AM	
SSO-ST-71-7RPudHMM97vUVEfZuic-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-71-7RPudHMM97vUVEfZuic-cas-webcentersites.ua.oracle.com-1	0	Monday, May 20, 2013 8:20:01 AM	Monday, May 20, 2013 8:20:01 AM	
SSO-ST-59-c7wawmkIGqcSMYECfdCS-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-59-c7wawmkIGqcSMYECfdCS-cas-webcentersites.ua.oracle.com-1	22	Monday, May 20, 2013 4:20:32 AM	Monday, May 20, 2013 4:27:44 AM	
SSO-ST-47-kDWAhKGLsStU9EgtVWh-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-47-kDWAhKGLsStU9EgtVWh-cas-webcentersites.ua.oracle.com-1	132	Saturday, May 18, 2013 8:14:18 AM	Saturday, May 18, 2013 8:19:00 AM	
SSO-ST-42-dJxglFH6r2OWCreTpfPX-cas-webcentersites.ua.oracle.com-1 Key: SSO-ST-42-dJxglFH6r2OWCreTpfPX-cas-webcentersites.ua.oracle.com-1	136	Saturday, May 18, 2013 7:11:19 AM	Saturday, May 18, 2013 7:40:38 AM	

The shared cache stores data that is shared among cluster members and enables the synchronization of cluster members. The data stored in the shared cache includes feed identities, site identities, captcha verification data, single sign-on ticket states, restricted words, and authorized domains. The "Shared" cache screen enables you to view the data stored in the shared cache.

[Table 41-4](#) provides information about the columns in the Shared cache screen. The "Shared" cache screen also allows you to view detailed information about an entry listed in this screen by pointing to the desired entry and selecting **Details** from the context menu. You can also delete an entry by pointing to it and selecting **Delete**.

Table 41-4 Overview of Columns in the Shared Cache screen

Column	Description
Title	Uniquely identifies each item in the shared cache.
Hit Count	The total number of requested objects that were served by reading the shared cache.
Created time	The date the object was fetched from the WebCenter Sites database and then stored in the cache for future access.

Table 41-4 (Cont.) Overview of Columns in the Shared Cache screen

Column	Description
Last Access Time	The most recent date the data was retrieved from the shared cache.
Dependencies	Lists the assets in the WebCenter Sites database on which the corresponding data depends. If the data is associated with more than one database table, multiple dependencies will be listed for that entry.

Part IV

Administering the Oracle WebCenter Sites: Gadgets Application

The Oracle WebCenter Sites: Gadgets application supplies the Global Gadget Catalog interface and the Gadgets User interface. The Global Gadget Catalog interface provides the global catalog and the Gadgets User interface provides the site gadget catalog. This part contains information about registering gadgets to the global catalog and site gadget catalogs as well as instructions for managing registered gadgets.

This part contains the following chapters:

- [Chapter 42, "Overview of the WebCenter Sites: Gadgets Application"](#)
- [Chapter 43, "Getting Started with the Gadgets Application"](#)
- [Chapter 44, "Registering Gadgets"](#)
- [Chapter 45, "Managing Gadgets"](#)

Overview of the WebCenter Sites: Gadgets Application

This chapter provides an overview of the Oracle WebCenter Sites: Gadgets application. It also provides information about the roles that can be assigned to the Gadgets application's users to grant them permissions to the Gadgets application's interface functions.

This chapter contains the following sections:

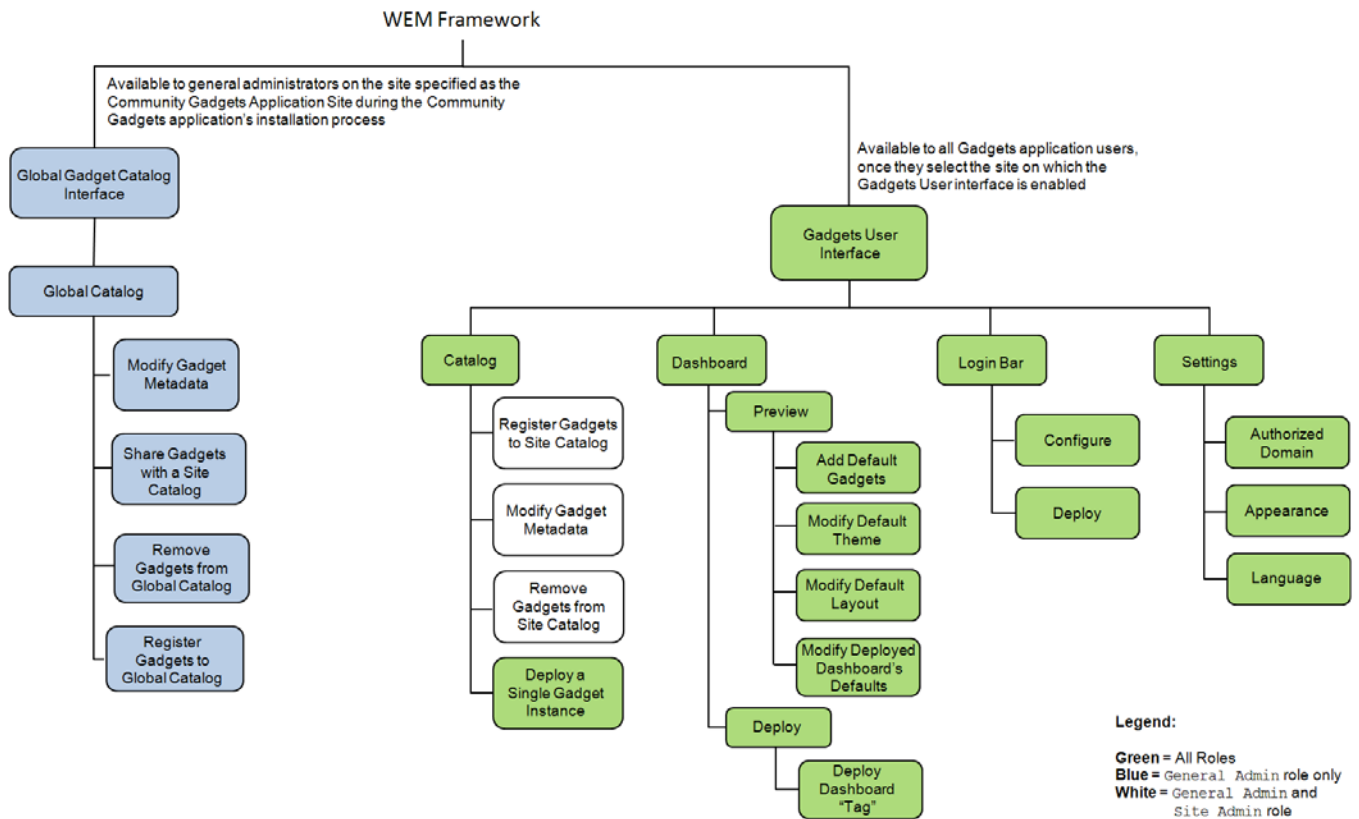
- [Section 42.1, "Overview of the Gadgets Application Interfaces"](#)
- [Section 42.2, "Gadgets Application Roles and Permissions"](#)
- [Section 42.3, "Making Gadgets Available for Deployment"](#)

42.1 Overview of the Gadgets Application Interfaces

The Gadgets application provides two interfaces: the Global Gadget Catalog interface and the Gadgets User interface. The Global Gadget Catalog interface is displayed when the Gadgets application is assigned to the site specified as the Community-Gadgets application site (AdminSite by default) during the Gadgets application's installation process. The Gadgets User interface is displayed when the Gadgets application is assigned to a site other than the site on which the Global Gadget Catalog interface is enabled.

Note: For instructions on enabling the Global Gadget Catalog interface and the Gadgets User interface, see [Chapter 40, "Enabling the Community and Gadgets Interfaces."](#)

[Figure 42-1](#) illustrates the functions associated with both of the Gadgets application's interfaces and identifies the roles that have permissions to those interfaces functions.

Figure 42–1 Gadgets Application Interfaces

This part contains information about the interface functions specific to general and site administrators. For information about the permissions granted to users assigned the Designer role, and a detailed overview of the Gadgets User interface, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

42.2 Gadgets Application Roles and Permissions

Once the Gadgets application is installed on the WEM Framework and assigned to the site specified as the Community-Gadgets application site (AdminSite by default), and at least one other CM site, general administrators can assign selected users roles to work with the application. The roles are *GeneralAdmin*, *SiteAdmin*, and *Designer*.

Table 42–1 provides an overview of the permissions granted to each role.

Note: Certain permissions are shared by the roles. For example, all users of the Gadgets application are able to add default gadgets to the dashboard panel. This part contains information pertaining only to users assigned the GeneralAdmin and SiteAdmin roles. For detailed information about the permissions granted to the Designer role, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Table 42–1 Gadgets application's roles and permissions

Permissions	GeneralAdmin Role	SiteAdmin Role	Designer Role	For Information, See...
Enable the Global Gadget Catalog Interface	Yes	No	No	Section 40.2.1, "Enabling the Global Gadget Catalog Interface"
Enable a Gadgets User Interface	Yes	Yes	No	Section 40.2.2, "Enabling the Gadgets User Interface"
Register Gadgets to the Global Gadget Catalog	Yes	No	No	Section 44.3, "Registering Gadgets to the Global Catalog"
Register Gadgets to a Site Gadget Catalog	Yes	Yes	No	Section 44.4, "Registering Gadgets to a Site Gadget Catalog"
Remove Gadgets from the Global Gadget Catalog	Yes	No	No	Section 44.5, "Deleting Gadgets from the Global Catalog or Site Gadget Catalog"
Remove Gadgets from a Site Gadget Catalog	Yes	Yes	No	Section 44.5, "Deleting Gadgets from the Global Catalog or Site Gadget Catalog"
Share Gadgets from the Global Gadget Catalog to a Site Gadget Catalog	Yes	No	No	Section 45.1, "Sharing Gadgets from the Global Catalog"
Modify Gadget Metadata	Yes	Yes	No	Section 45.2, "Modifying a Gadget's Metadata"
Add Gadgets to the Dashboard as Defaults	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Modify the Dashboard's Default Layout	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Modify the Dashboard's Default Color Theme	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Configure and Deploy the Login Bar Widget Tag	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Authorize a Domain	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Configure Appearance Settings for the Gadgets Application	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>

Table 42–1 (Cont.) Gadgets application's roles and permissions

Permissions	GeneralAdmin Role	SiteAdmin Role	Designer Role	For Information, See...
Configure a Language Preference for Deployed Gadgets	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Deploy the Dashboard "Tag"	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Deploy an Individual Gadget's "Tag"	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>
Modify the Deployed Dashboard's Defaults	Yes	Yes	Yes	<i>Oracle Fusion Middleware WebCenter Sites User's Guide</i>

GeneralAdmin

Users assigned the `GeneralAdmin` role are administrators of the entire WebCenter Sites system. These users can enable and manage the Global Gadget Catalog interface. They can also enable the Gadgets User interface on sites other than the site on which the Global Gadget Catalog interface is enabled. These users manage the global catalog, which is contained in the Global Gadget Catalog interface.

SiteAdmin

Users assigned the `SiteAdmin` role are administrators of their assigned content management sites and its applications, including the Gadgets application. These users have permissions to all Gadgets User interface functions. However, their main purpose is to manage the CM site's gadgets in the site gadget catalog.

Designer

Users assigned the `Designer` role can manage the appearance of the default dashboard, deploy the dashboard tag and single gadget tags on web pages, and configure and deploy the Login Bar widget tag. These users also have permissions to specify the language in which deployed gadgets and the dashboard are displayed, specify a domain on which the gadgets and dashboard tags can be deployed, and configure the appearance settings of the Gadgets User interface, the dashboard tag, and the single gadget tags. For more information about the permissions granted to the `Designer` role, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

42.3 Making Gadgets Available for Deployment

Users can deploy gadgets only when the gadgets are registered to a site gadget catalog or shared to a site gadget catalog from the global catalog. If you are a general administrator, you can either register gadgets to the global catalog or a site gadget catalog. If you are a site administrator, you will register gadgets directly to the site gadget catalog.

Note: The catalogs are provided in the Gadgets application's interfaces – the Global Gadget Catalog interface and the Gadgets User interface (as shown in [Figure 42-1](#)).

- The global catalog is accessible from the Global Gadget Catalog interface and is used by general administrators to distribute gadgets to as many site gadget catalogs as necessary. Which in turn makes the gadgets available for single gadget and dashboard deployment. When a gadget is registered to the global catalog, general administrators can share the gadget to multiple site gadget catalogs, without registering the gadget separately on each desired site gadget catalog.
- Site gadget catalogs are part of the Gadgets User interface. They are local and specific to each site on which the Gadgets User interface is enabled. The Gadgets User interface also provides a dashboard and screens for single gadget deployment, which are dynamically associated with the site gadget catalog. When a general or site administrator registers a gadget to a site gadget catalog, that gadget is made available on the dashboard (as a thumbnail, on the gadget selection panel) and for individual deployment.

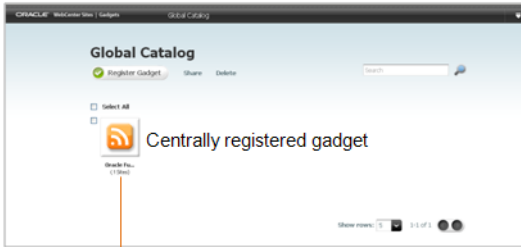
Note: For more information about single gadget deployment and the dashboard, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

[Figure 42-2](#) summarizes the process of how administrators make gadgets available for single gadget and dashboard deployment. Gadgets can be registered to the site gadget catalog either directly (from the User interface) or shared from the global catalog (in the Global Gadget Catalog interface).

Figure 42–2 Gadgets available for single and dashboard deployment

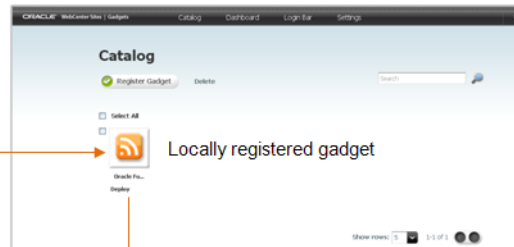
A. Global Gadget Catalog

Global Gadget Catalog interface (on the site specified as the Community Gadgets Application Site during the Community Gadgets application's installation process)



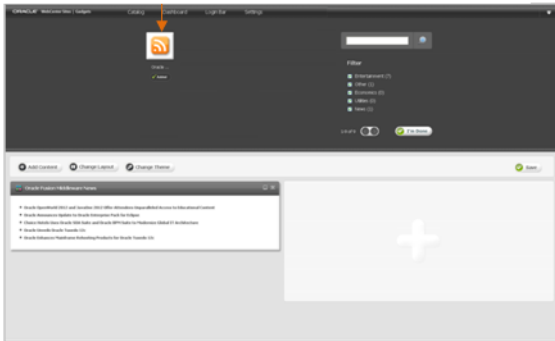
B. Site Gadget Catalog

Gadgets User interface (on site other than the site on which the Global Gadget Catalog interface is enabled).



A locally registered gadget is available in two places:

On the gadget selection panel, as a thumbnail for dashboard deployment



In the "Single Gadget Deployment" screen for individual deployment



Getting Started with the Gadgets Application

This chapter provides instructions for logging in to the Gadgets interfaces.

This chapter includes the following sections:

- [Section 43.1, "Logging In"](#)
- [Section 43.2, "Sample Gadgets"](#)
- [Section 43.3, "Next Steps"](#)

43.1 Logging In

The Gadgets application is associated with WebCenter Sites. The Global Gadget Catalog interface can be accessed by general administrators when they log in to the site on which this interface is enabled. When assigned to sites other than the site on which the Global Gadget Catalog interface is enabled, the Gadgets application displays its User interface to general and site administrators (and users assigned the Designer role).

Note: For information about the permissions associated with the Designer role, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

To log in to the Global Gadget Catalog interface and the Gadgets User Interface

1. Access the WEM Framework at the following URL:

```
http://<server>:<port>/<application context>/login
```

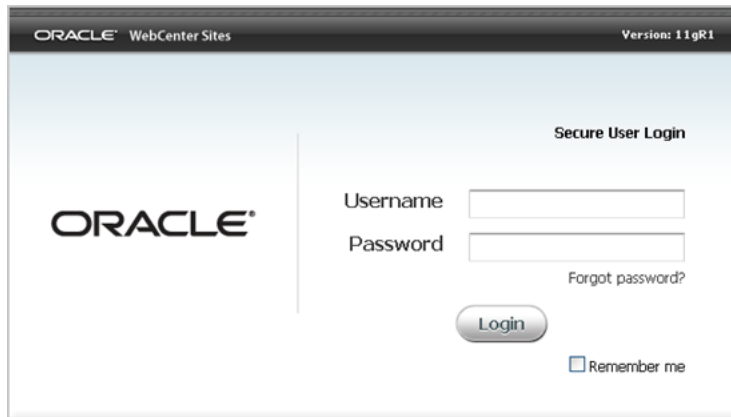
where <server> is the host name or IP address of the server running WebCenter Sites, <port> is the port number of the WebCenter Sites application, and <application context> is the name of the web application that was deployed on the server.

2. Access the Gadgets application by logging in to WebCenter Sites with the credentials of a general or site administrator. **Log in credentials are case-sensitive.**

In this guide we are using the default login credentials of a general administrator:

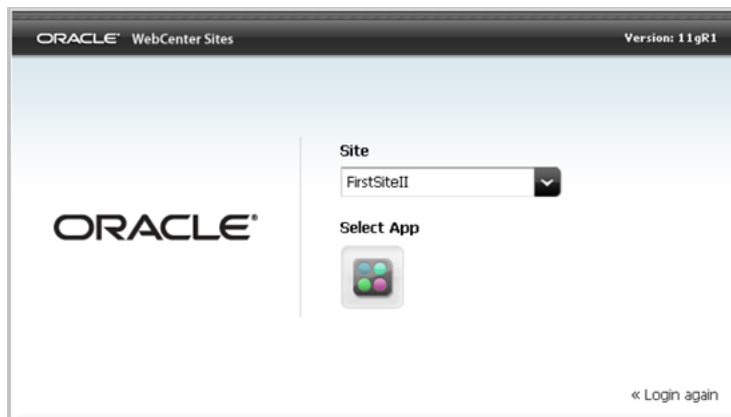
- **Username** – fwadmin
- **Password** – xceladmin

Figure 43–1 WebCenter Sites Login Form



3. Click **Login**.
4. If you are logging in for the first time, or logging in to a site you have never accessed before, the following screen is displayed; otherwise, you are logged in to the application you last visited:

Figure 43–2 WebCenter Sites Select A Site



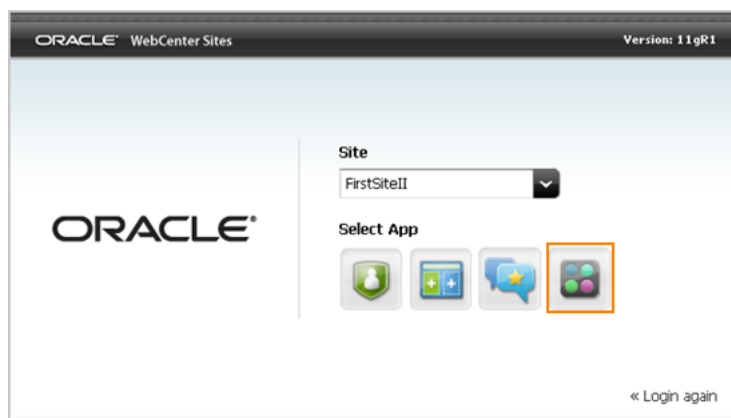
- If you are a general administrator, access the Global Gadget Catalog interface by selecting the site on which the Global Gadget Catalog interface is enabled, and then clicking the **Gadgets** icon.

Figure 43–3 WebCenter Sites Selection, Gadgets Icon



- If you are a general or site administrator, access the Gadgets User interface by selecting the desired site on which the User interface was enabled, and then clicking the **Gadgets** icon.

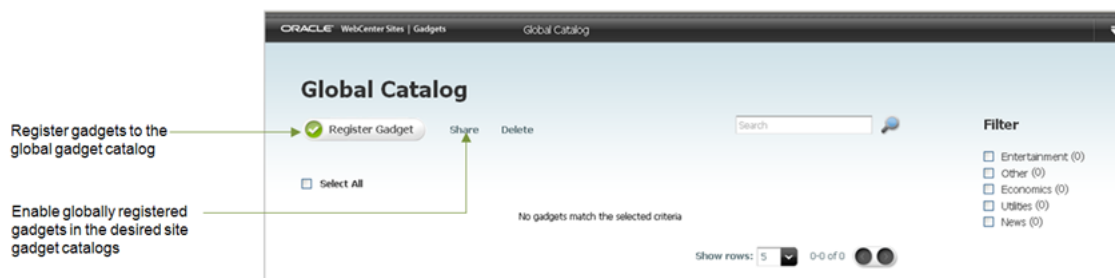
Figure 43–4 WebCenter Sites Selection, Gadgets Icon



5. Your interface is one of the following:

- If you logged in to the site on which the Global Gadget Catalog interface is enabled, the Global Catalog is displayed:

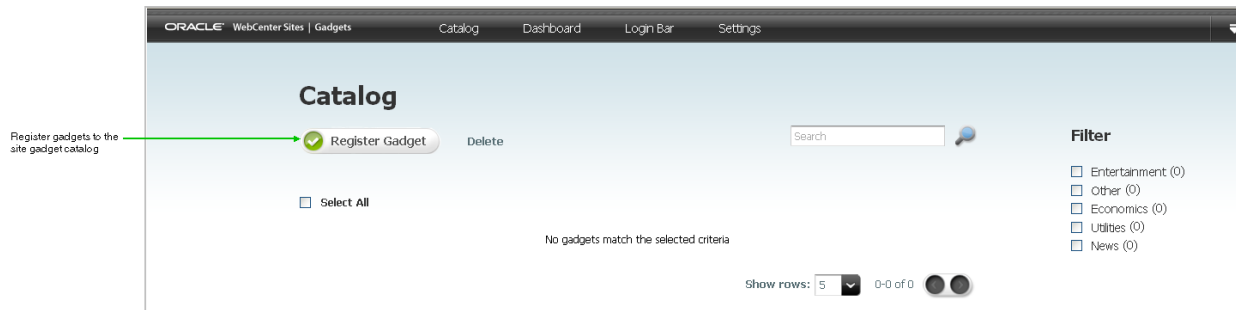
Figure 43–5 Global Gadget Catalog Interface



The Global Gadget Catalog interface contains the global catalog, where general administrators can first register gadgets (of type `FW_CSGadget` or third-party), and then share them to any site gadget catalogs.

- If you logged in to a site on which the Gadgets User interface is enabled, you will see the following screen:

Figure 43–6 Catalog Form, with Register Gadget Button



The Gadgets User interface contains the site gadget catalog, dashboard, Permissions and Settings screens, dashboard deployment screen, single gadget deployment screens, and the Social Login and Login Bar Deployment screens for the site to which you logged in.

Note: For information about the dashboard, and the Permissions and Settings screens, and for instructions on deploying a single gadget and the dashboard, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

43.2 Sample Gadgets

In this part, we assume that the Gadgets Samples Module was installed on the FirstSite II sample site of your WebCenter Sites system. Installing the sample module:

- Installs the `FW_CSGadget` (whose description is CS-Based Gadget) asset type, from which sample and custom gadgets are created.
- Enables the `FW_CSGadget` (whose description is CS-Based Gadget) asset type on FirstSite II.
- Installs sample gadgets on FirstSite II (see [Table 43–1](#)).
- Installs assets that provide content for the sample gadgets.
- Provides templates that render the gadgets. **These templates ([Table 43–2](#)) are required for all gadgets (sample and custom).**

Table 43–1 Sample Gadgets of Type `FW_CSGadget`

Sample Gadgets of Type <code>FW_CSGadget</code>	Description
ListGadget	Displays a list of articles on the FirstSite II sample site.
RSSGadget	Displays a list of entries from an RSS feed.
SlideshowGadget	Displays a list of Product images in a slideshow format.
ThumbListGadget	Displays a list of products with thumbnail images.

Table 43–2 The Templates

Templates	Description
ListSiteGadgets	Lists URLs of the gadget descriptor XML files.
GenerateGadgetXML	Starts the process of rendering a gadget's descriptor XML

More information about the Gadgets Samples Module is available in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*.

43.3 Next Steps

Now that you are familiar with the Gadgets application's interfaces, you can start registering the sample gadgets and third-party gadgets to the global catalog and site gadget catalogs. For information and instructions about registering gadgets, see [Chapter 44, "Registering Gadgets."](#)

Registering Gadgets

This chapter contains instructions on registering gadgets to the Global Catalog and site gadget catalogs. It also contains instructions on removing gadgets from the catalogs.

This chapter contains the following sections:

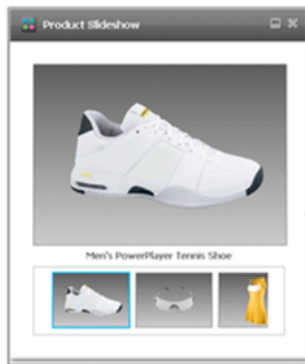
- [Section 44.1, "Overview of Gadget Registration and Deployment"](#)
- [Section 44.2, "Gadget Sources"](#)
- [Section 44.3, "Registering Gadgets to the Global Catalog"](#)
- [Section 44.4, "Registering Gadgets to a Site Gadget Catalog"](#)
- [Section 44.5, "Deleting Gadgets from the Global Catalog or Site Gadget Catalog"](#)

44.1 Overview of Gadget Registration and Deployment

[Figure 44-1](#) – [Figure 44-3](#) provide a closer look at the process of registering a gadget in the global catalog, sharing that gadget to a site gadget catalog, and deploying that gadget either individually, or on the dashboard.

Figure 44–1 Basic Details for Registering and Deploying a Gadget

A gadget is defined by an XML file (gadget descriptor file) responsible for rendering the gadget



Gadget's descriptor XML
(for the gadget you wish to register)

```
http://sitehost.example.com:8080/cs/Satellite?
c=FW_CSGadget&cid=1269873535728&pagename=First
SiteII%2FFW_CSGadget%2FGenerateGadget.XML
```

```
<Module>
- <ModulePrefix title="Product Slideshow" height="350"
  screenshot="http://csoracle.us.oracle.com:8080/cs/FirstSiteII/gadgets
  /Slideshow/screenshot.png"
  thumbnail="http://csoracle.us.oracle.com:8080/cs/FirstSiteII/gadgets
  /Slideshow/thumbnail.png">
  <Require feature="dynamic-height"/>
- </ModulePrefix>
- <Content type="html">
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd"> <style type="text/css"> /*
=== generic styles === */ body { margin: 0; } body #container, body
#container * { padding: 0; margin: 0; font-size: 11px; font-family:
Tahoma, Arial, Helvetica, sans-serif, color: #666; } .error { background-
color: #fcc, color: #c00; } /* === gadget-specific styles === */ body
#container { padding: 15px; } /* styles for image display area */
#container #fullimg { text-align: center; } #container #fullimg {
border: 1px solid #7e7e7e; } #container #imgtitle { width: 100%; color:
#666; font-size: 12px; font-weight: bold; text-align: center; padding:
3px 0; } /* styles for thumbnail strip area */ #container
#stripcontainer { /* container for thumbstrip + buttons */ /> border: 1px
solid #afafaf; height: auto; } #container #thumbstrip { /* horizontal
strip of thumbnail images with buttons, overflow: hidden; white-space: nowrap;
display: inline-block; width: 100%; text-align: center; padding: 5px 0; }
#container #thumbstrip img { border: 1px solid #ccc; width: 50px; height: 50px;
margin: 2px; } #container #thumbstrip a { border: 1px solid #ccc; width: 50px;
height: 20px; margin: 2px; text-align: center; font-size: 8px; color: #666;
background-color: #fff; } #container #thumbstrip a:hover { background-color: #eee;
color: #333; } #container #thumbstrip a:active { background-color: #ddd;
color: #333; } #container #thumbstrip a:disabled { background-color: #ccc;
color: #666; } #container #thumbstrip a[disabled] { pointer-events: none; }
/* styles for buttons */ #container #thumbstrip a { border: 1px solid #ccc;
width: 50px; height: 20px; margin: 2px; text-align: center; font-size: 8px;
color: #666; background-color: #fff; } #container #thumbstrip a:hover {
background-color: #eee; color: #333; } #container #thumbstrip a:active {
background-color: #ddd; color: #333; } #container #thumbstrip a:disabled {
background-color: #ccc; color: #666; } #container #thumbstrip a[disabled] {
pointer-events: none; }
=== end of styles === */
</style>
</Content>
```

- 1 When registering a gadget to the global gadget catalog, the general administrator specifies the URL that points to the gadget's descriptor XML and the gadget's basic metadata.



Gadget descriptor URL

- 2 Saving the gadget completes the registration process. The gadget's thumbnail image and name are displayed in the global gadget catalog.



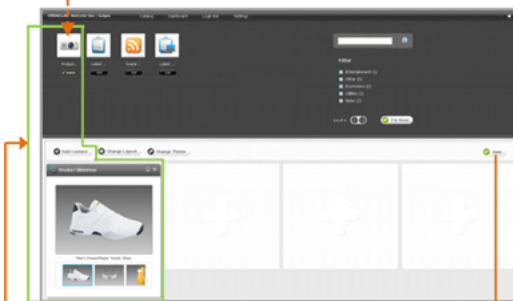
Figure 44–2 Basic Details for Registering and Deploying a Gadget (Continued)

- 3 General administrators can share a globally registered gadget with any site gadget catalog.
Note: General administrators and site administrators can also register gadgets directly to a site gadget catalog.



- 4 The gadget is automatically made available for deployment in two places:

The dashboard's gadget panel, as a thumbnail image



The "Single Gadget Deployment" screen



- 5a An administrator (or a user with the Designer role) can add the gadget to the dashboard panel (by clicking **Add Content**) as a default for website visitors.

- 5b The dashboard widget tag picks up the saved dashboard settings.

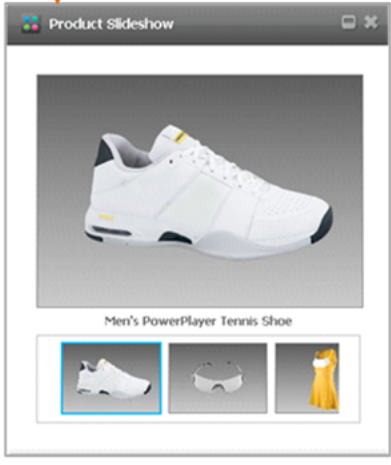
- 5 If an administrator (or a user with the Designer role) clicks the **Deploy** link (located under a gadget), the "Single Gadget Deployment" screen opens and the gadget's metadata is defined in the "Tag" field.

- 6 Copy the gadget defined in the "Tag" field to a page template. Then, publish the template to display the gadget on the web page.

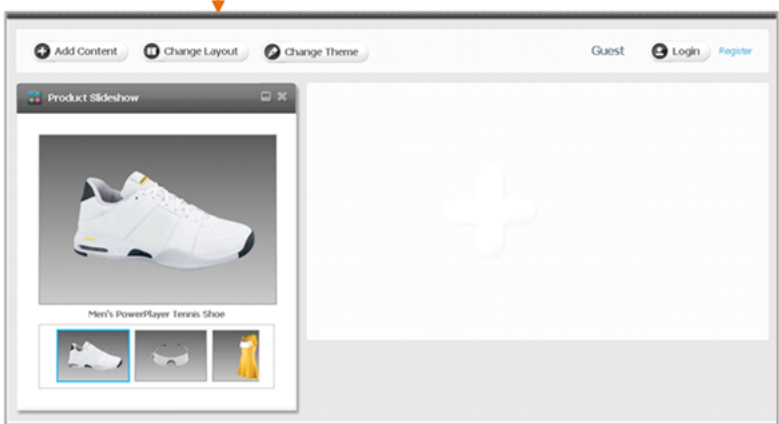
Figure 44–3 Basic Details for Registering and Deploying a Gadget (Continued)

6 Copy the dashboard defined in the “Tag” field to a page template. Then, publish the template to display the dashboard on the web page.

If modifications are made to the gadget's metadata in the site gadget catalog, they are dynamically updated on the deployed gadget.



If modifications are made to the dashboard, they are dynamically updated on the deployed dashboard.



More About Deploying and Updating Gadgets:

To deploy gadgets, you can either insert the dashboard tag or the tag for a single gadget into a template asset and then publish that template. For more information, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Once the dashboard or the individual gadget instance is deployed, any modifications made to the gadget's metadata or the dashboard from the site's Gadgets User interface are dynamically updated to the website. Therefore, you do not need to re-deploy the gadget instance or dashboard, unless you want to display the gadget or dashboard on other pages of the website.

44.2 Gadget Sources

Gadgets are defined by descriptor XML files, which contain the gadgets' logic and data. The source of each XML file is a URL that we call the Gadget Descriptor URL (served by the gadget provider).

Gadgets can be obtained from third-party sources (such as iGoogle), or they can be created in the WebCenter Sites Contributor interface. The Gadgets application's dashboard supports the following content types for gadgets: `text/html`, `text/xml`, and `application/xml`.

Gadgets must be registered to a site gadget catalog directly or shared to a site gadget catalog from the global catalog, if they are to be made available on the Gadgets User interface, and from there deployed to the website. When registering a gadget, you specify the URL of the gadget's descriptor XML:

- For third-party gadgets, you can obtain the URL by viewing the gadget's source code and noting its URL in the address bar.
- For gadgets of type `FW_CSGadget`, you can obtain the URL of the gadget's descriptor XML by previewing the gadget asset in the WebCenter Sites Contributor interface, using the `ListSiteGadgets` template. This template, and the sample gadgets, are included with the Gadgets Samples application.

44.3 Registering Gadgets to the Global Catalog

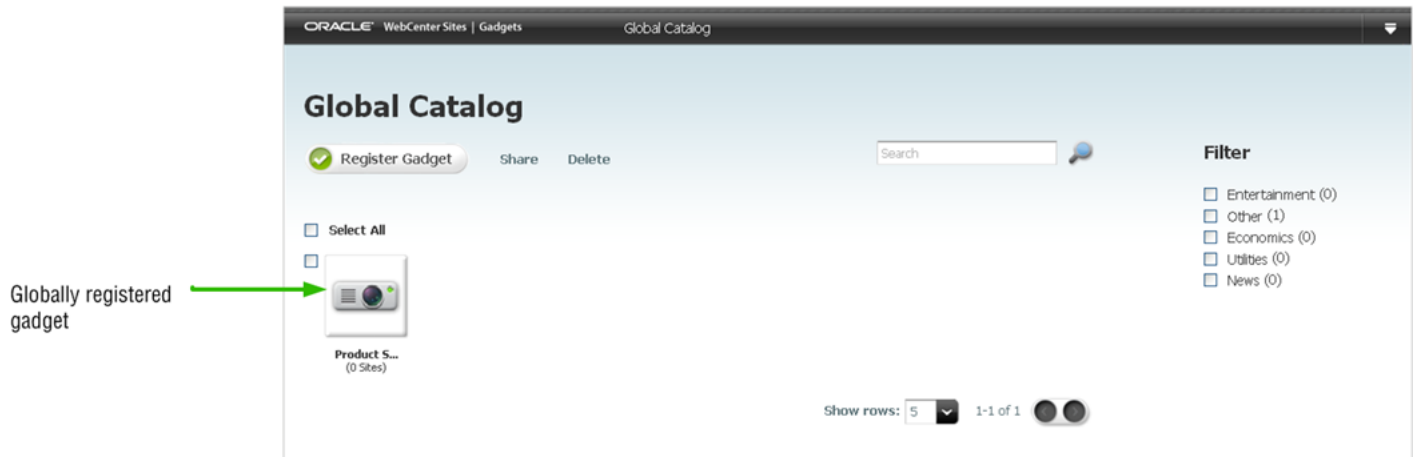
The Global Gadget Catalog interface consists of the global catalog, which is the main repository for your company's gadgets and provides a way for general administrators to distribute gadgets to site gadget catalogs. Once gadgets are registered on or shared to a site gadget catalog, they are made available for single gadget deployment as well as placed on the catalog's associated dashboard, as thumbnails on the gadget selection panel.

To register a gadget with the global catalog

1. Log in to WebCenter Sites as a general administrator, select the site on which the Global Gadget Catalog interface is enabled, and then select the **Gadgets** icon.

The Global Catalog screen opens:

Figure 44-4 Global Gadget Catalog Screen



2. In the Global Catalog screen, click **Register Gadget**.

The Register New Gadget screen opens:

Figure 44–5 Register New Gadget screen

Catalog >

Register New Gadget

STEP 1: GADGET DESCRIPTOR

Gadget Descriptor URL* [What is this?](#)

Step 2: Metadata

Title*

Icon [Browse...](#)

Thumbnail [Browse...](#)

Preview Image

Gadget Name Settings [X](#)

Preview Image Not Available

Available (5) Selected (0)

Filter List

Entertainment
Other
Economics
Utilities
News

Category

Save Cancel

3. In the **Gadget Descriptor URL** field, enter the URL of the gadget's descriptor XML, which contains the gadget's attributes and specifies the source of content.

To locate and specify a gadget's descriptor URL:

- If you are registering a third-party gadget, view the gadget's XML descriptor, copy its URL, and paste the URL into the **Gadget Descriptor URL** field.
- If you are registering a CS-Based Gadget (of type `FW_CSGadget`), follow the steps below to render a list of CS-Based Gadget descriptor URLs:
 - a. Access the WebCenter Sites Contributor interface.
 - b. Select the site where CS-Based Gadgets (of type `FW_CSGadget`) are available (FirstSite II by default).
 - c. In the **Search** field, click the down-arrow to render the **Search Type** drop-down box. Select **Find CS-Based Gadget** and then click the **magnifying glass** button.

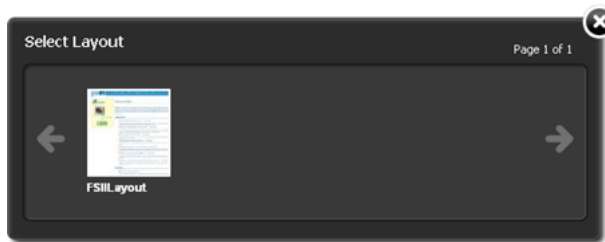
Note: f **Find CS-Based Gadget** is not available from the **Search Type** drop-down box. You must add the `FW_CSGadget` asset type to the search index. For instructions see, [Section 25.2.2, "Adding Asset Types to the Search Index."](#)

The **Search** tab opens displaying a list of all the CS-Based Gadgets on the site.

- d. In the search results list, select any given gadget.
- e. In the gadget's toolbar, click the **Preview** icon.

The Select Layout dialog is displayed:

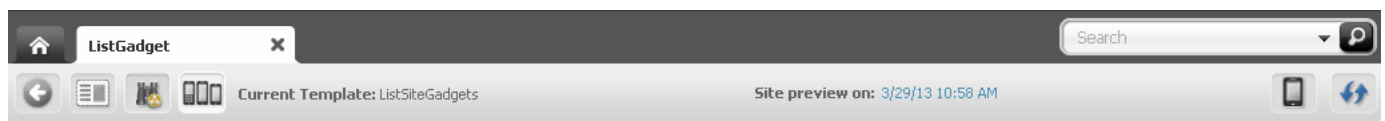
Figure 44–6 Select Layout Dialog



- f. Select a layout and then click **Apply**.
- g. In the asset's toolbar, click the **Preview with Page Layout** icon.
- h. In the **Select preview template** dialog box, select **ListSiteGadgets** and then click **Apply**.

The descriptor URLs for all CS-Based Gadgets (of type `FW_CSGadget`) are listed in the preview screen.

Figure 44–7 CS-Based Gadgets Descriptor URLs



ListGadget

http://webcentersites.ua.oracle.com:8080/cs/Satellite?c=FW_CSGadget&cid=1269873534992&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

RSSGadget

http://webcentersites.ua.oracle.com:8080/cs/Satellite?c=FW_CSGadget&cid=1269873535165&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

ThunbListGadget

http://webcentersites.ua.oracle.com:8080/cs/Satellite?c=FW_CSGadget&cid=12698735353620&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

SlideshowGadget

http://webcentersites.ua.oracle.com:8080/cs/Satellite?c=FW_CSGadget&cid=1269873535728&pagename=FirstSiteII%2FFW_CSGadget%2FGenerateGadgetXML

- i. Copy the descriptor URL of the gadget you wish to register, and paste the URL into the **Gadget Descriptor URL** field.

The Gadgets application processes the gadget descriptor URL to ensure the gadget is valid. Once the gadget is validated, the Gadgets application dynamically

extracts the gadget's default metadata (if available) and populates the fields in the Metadata section with the extracted data.

4. In the Metadata section, do the following:
 - **Title** – Enter a name for the gadget. This name will be displayed under the gadget's thumbnail image in the global catalog (and site gadget catalogs).
 - **Icon** – Click **Browse** to select an icon image. This image will be displayed on the gadget's title bar when the gadget is displayed on a site gadget catalog's associated dashboard panel. If no image is selected, the following default image is displayed:

Figure 44–8 Default Icon



- **Thumbnail** – Click **Browse** to select a thumbnail image. This image will be displayed in the global catalog. If the gadget is registered to or shared with a site gadget catalog, the image is also displayed on that catalog and its associated dashboard's gadget selection panel. If no image is selected, the following default image is displayed:

Figure 44–9 Default Image



- **Preview Image** – Click **Browse** to select the gadget's preview image, which will be displayed when a user clicks the gadget's thumbnail on a dashboard's gadget selection panel. If no preview image is selected, clicking the gadget's thumbnail displays the following image:

Figure 44–10 Gadget Preview Image



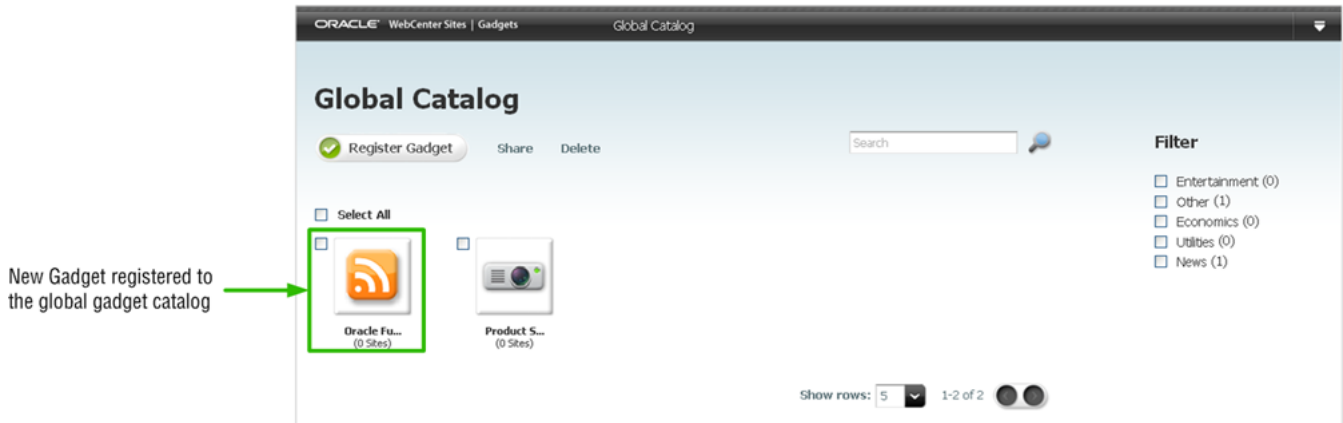
- **Category** – Selecting a category enables users and visitors to filter gadgets by category in the gadget catalogs and the dashboard's gadget selection panel. If

no category is selected, "Other" is assigned by default. The default categories are Entertainment, News, Utilities, Economics, and Other.

5. Click **Save**.

The gadget's name and thumbnail image are now displayed in the Global Catalog.

Figure 44–11 Global Catalog, Showing Gadget Name and Thumbnail



6. Share the gadget with a site gadget catalog. For instructions, see [Section 45.1, "Sharing Gadgets from the Global Catalog."](#)

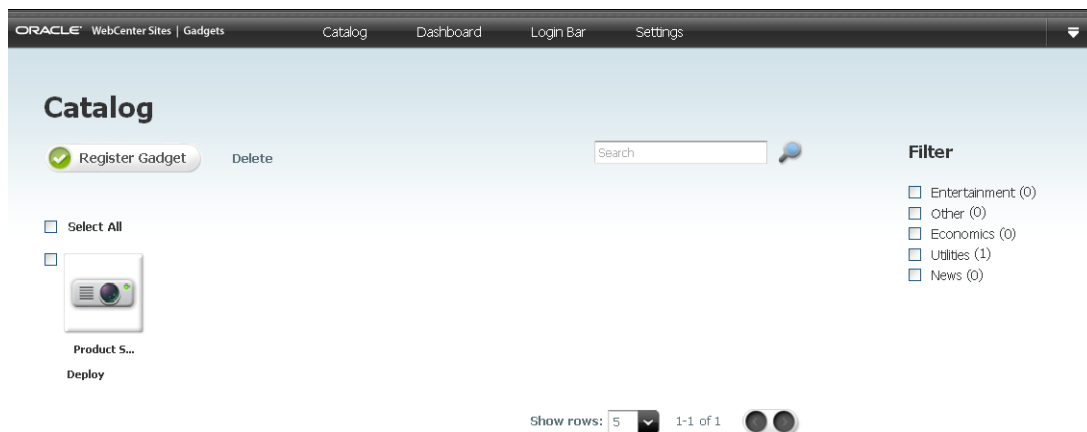
44.4 Registering Gadgets to a Site Gadget Catalog

The Gadgets User interface provides the site on which it is enabled with a site gadget catalog. Gadgets registered to a site gadget catalog are made available for single gadget deployment, and placed on the dashboard (as thumbnail images on the gadget selection panel) for dashboard deployment.

To register gadgets to a site gadget catalog

1. Log in to WebCenter Sites as a general or site administrator, select the desired site on which the Gadgets User interface is enabled, and then click the **Gadgets** icon.

The Catalog screen opens:



2. Register the gadget to the site gadget catalog. For instructions, see steps 2 – 5 in [Section 44.3, "Registering Gadgets to the Global Catalog."](#)

3. Click **Save**.

The gadget is registered to the site gadget catalog. The gadget's name and thumbnail are displayed in the site gadget catalog and placed on the associated dashboard's gadget selection panel.

4. Deploy the gadget either as a single gadget or on the dashboard. For instructions on deploying gadgets, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

44.5 Deleting Gadgets from the Global Catalog or Site Gadget Catalog

Only general administrators can delete gadgets from the global catalog, ensuring that those gadgets cannot be shared with any site gadget catalogs. If the gadgets were *previously* shared with site gadget catalogs, they are not deleted from those catalogs.

General and site administrators can delete gadgets from a site gadget catalog. Because the site gadget catalog, Single Gadget Deployment screens, and dashboard are associated, when gadgets are deleted from a site gadget catalog, those gadgets are no longer available for single deployment and are removed from the dashboard.

To delete gadgets from the global catalog

1. Log in to WebCenter Sites as a general administrator, select the site on which the Global Gadget Catalog interface is enabled, and then click the **Gadgets** icon.
2. In the Global Catalog screen, select the check boxes of the gadgets you wish to delete from the global catalog, and click **Delete**.
3. In the dialog box, click **OK**.

The gadgets are deleted from the global catalog. However, the gadgets are not removed from the site gadget catalogs with which they were previously shared.

To delete gadgets from a site gadget catalog

1. Log to WebCenter Sites as a general or site administrator, select the desired site on which the Gadgets User interface is enabled, and then click the **Gadgets** icon.
2. In the Catalog screen, select the check boxes of the gadgets you wish to delete from the site gadget catalog, and click **Delete**.
3. In the dialog box, click **OK**.

The gadgets are deleted from the site gadget catalog, and are no longer available on the dashboard or for single gadget deployment.

- If the gadgets were registered with the global gadget catalog, they are still available from that catalog.
- If a gadget was deployed individually on a web page, you must remove that gadget's tag from the template. Otherwise, the gadget remains on the web page displaying the message "The gadget is unavailable" to site visitors.

Managing Gadgets

This chapter contains information and instructions about sharing globally registered gadgets with site gadget catalogs. This chapter also contains instructions on modifying a gadget's metadata, such as its name and preview image.

This chapter contains the following sections:

- [Section 45.1, "Sharing Gadgets from the Global Catalog"](#)
- [Section 45.2, "Modifying a Gadget's Metadata"](#)

45.1 Sharing Gadgets from the Global Catalog

Gadgets that are registered in the global catalog can be shared with as many site gadget catalogs as necessary. A site gadget catalog is associated with a gadget's Single Gadget Deployment screen and the dashboard. This means that any gadget listed in the given site gadget catalog is automatically:

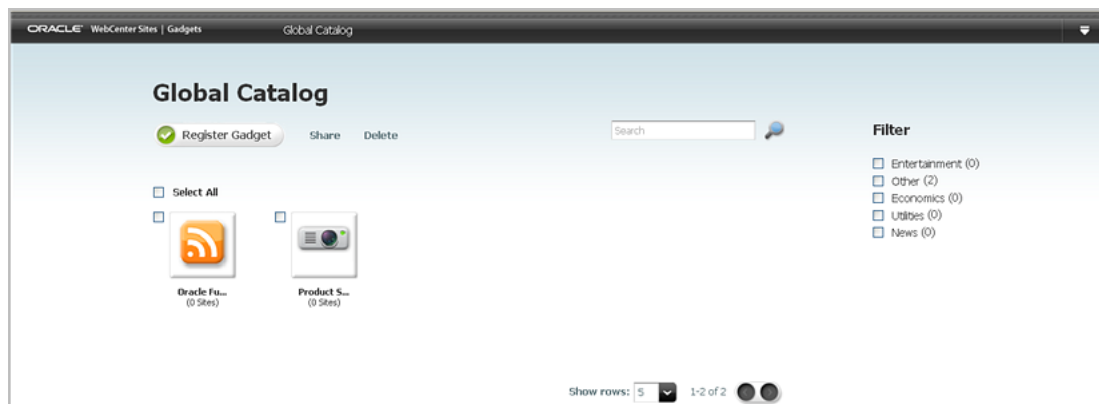
- Available for individual deployment.
- Placed on the dashboard, as a thumbnail on the gadget selection panel.

To share a globally registered gadget

1. Log in to WebCenter Sites as a general administrator, select the site on which the Global Gadget Catalog interface is enabled, and then click the **Gadgets** icon.

The Global Catalog opens:

Figure 45–1 Global Catalog Form

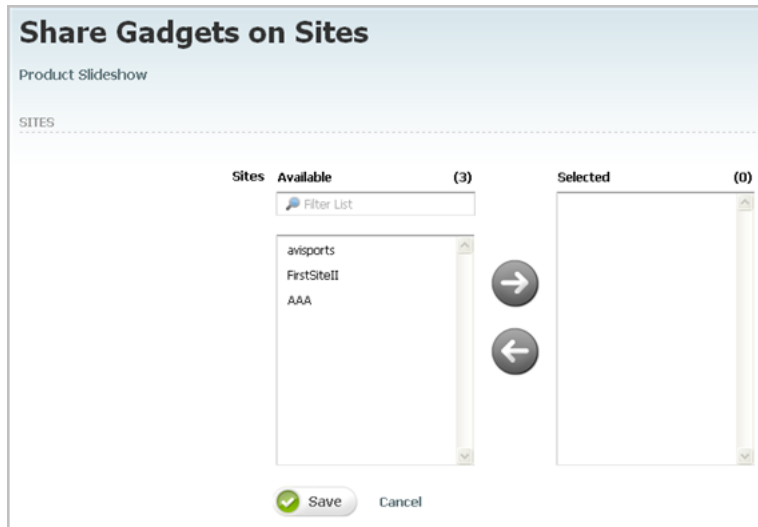


2. Select the check box(es) of the gadget(s) you wish to share with one or more site gadget catalogs, and click **Share**.

Note: If gadgets are not displayed in your interface, register the gadgets. For instructions, see [Section 44.3, "Registering Gadgets to the Global Catalog."](#)

The Share Gadgets on Sites form opens:

Figure 45–2 *Share Gadgets on Sites Form*



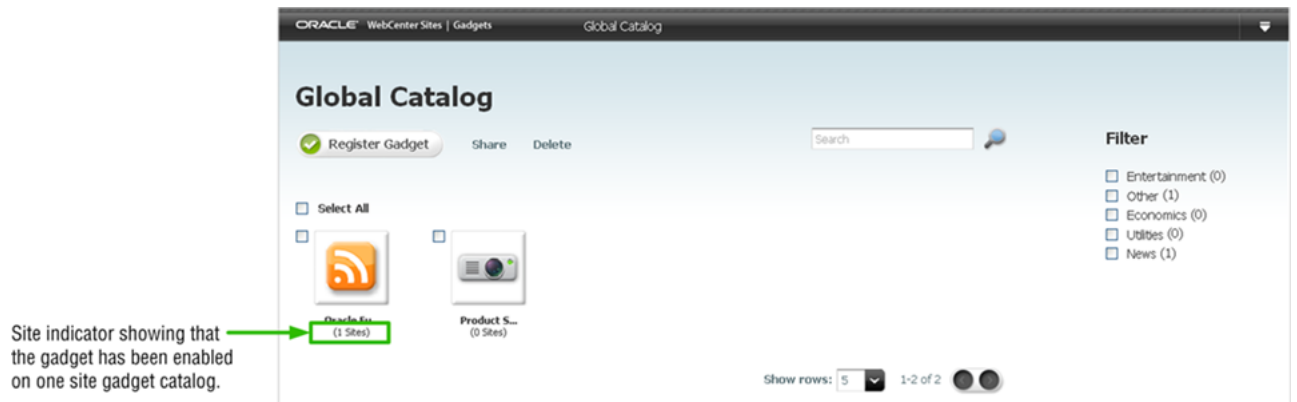
3. From the **Available** list box, select the site(s) to which you wish to share the gadgets, and move the site(s) to the **Selected** list box.

Note: If sites are not displayed in your interface, it means that the Gadgets User interface is not assigned to any sites. For instructions on assigning the interface, see [Section 40.2.2.1, "Enabling the Gadgets User Interface for Existing Site Users."](#)

4. Click **Save**.

The gadgets are made available on the selected site gadget catalog(s), as shown by the site indicator below the gadget's thumbnail:

Figure 45–3 Global Catalog Form, with Callouts



The gadgets are now available for individual deployment in the site gadget catalogs. The gadgets are also placed in the site gadget catalogs' associated dashboards, as thumbnails on the gadget selection panel.

Note: A gadget cannot be shared more than once to a given site gadget catalog.

45.2 Modifying a Gadget's Metadata

A gadget's metadata can be modified from the global catalog or a site gadget catalog. The metadata of a gadget includes its name, descriptor URL, thumbnail, preview, and icon images, and its assigned filter category.

This section contains the following topics:

- [Section 45.2.1, "Modifying a Gadget's Metadata in the Global Catalog"](#)
- [Section 45.2.2, "Modifying a Gadget's Metadata in a Site Gadget Catalog"](#)

45.2.1 Modifying a Gadget's Metadata in the Global Catalog

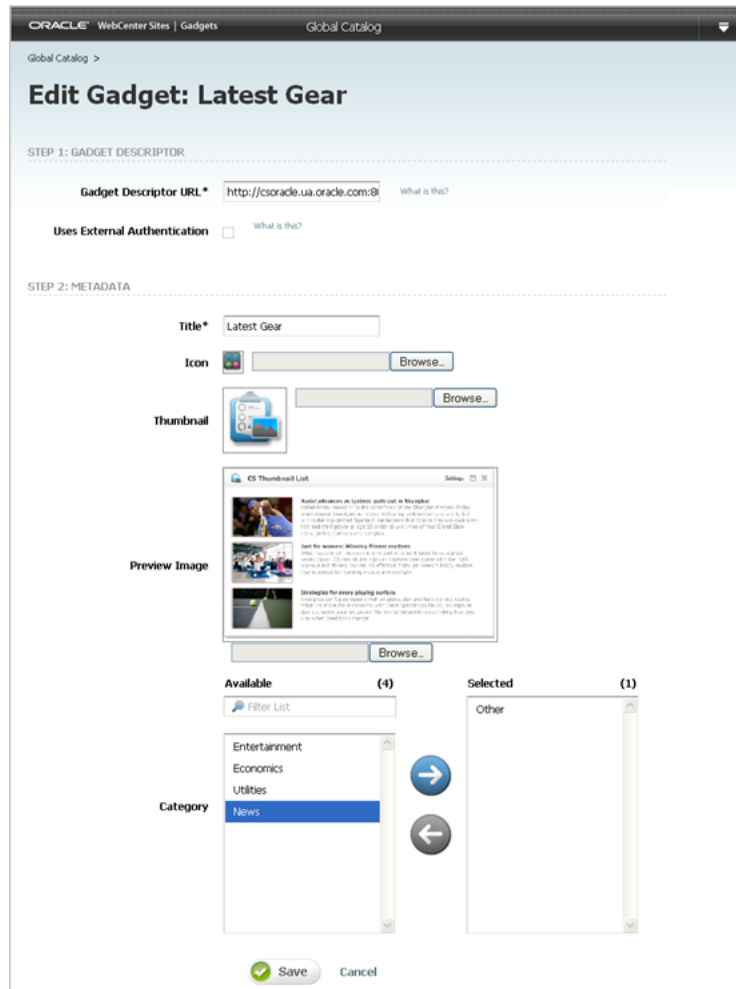
General administrators are the only users who can modify a gadget's metadata in the global catalog.

To modify a gadget's metadata in the global catalog

1. Log in to WebCenter Sites as a general administrator, select the site on which the Global Gadget Catalog interface is enabled, and then click the **Gadgets** icon.
2. In the Global Catalog screen, click the thumbnail of the gadget whose metadata you wish to modify.

The Edit Gadget form opens:

Figure 45–4 Edit Gadget Form



3. In the Edit Gadget form, make the desired changes to the gadget's metadata, and then click **Save**.

The changes you make to the gadget's metadata are only applied to the instance of the gadget in the global catalog. If the gadget is shared with a site gadget catalog, delete the gadget from the site gadget catalog and then share the updated gadget (on the global catalog) with the site gadget catalog.

45.2.2 Modifying a Gadget's Metadata in a Site Gadget Catalog

General and site administrators can modify a gadget's metadata in a site gadget catalog. The modifications affect only the gadget in that catalog. They are propagated from the site gadget catalog to the dashboard, the gadget's Single Gadget Deployment screen, and anywhere the gadget is deployed.

To modify a gadget's metadata on a site gadget catalog

1. Log in to WebCenter Sites as a general or site administrator, select the desired site on which the Gadgets User interface is enabled, and then click the **Gadgets** icon.
2. In the Gadgets screen, click the thumbnail image of the gadget whose metadata you wish to modify.

3. In the Edit Gadget form, make the desired changes to the gadget's metadata, and then click **Save**.

The changes you make to the gadget's metadata take affect in the site gadget catalog, the gadget's Single Gadget Deployment form, the dashboard, and anywhere the gadget is deployed (individually and on the dashboard).

Part V

Administering the Community Blog Module

This part provides administrators with instructions for enabling content providers to use the Community Blogs module to create blogs and modify their author profiles.

This part contains the following chapter:

- [Chapter 46, "Enabling Blog Components"](#)

Enabling Blog Components

This chapter provides instructions for enabling the blog components your content providers will need to access in order to create blog content. It also provides instructions for creating author assets.

This chapter contains the following sections:

- [Section 46.1, "Enabling Blog Asset Types"](#)
- [Section 46.2, "Creating a Blog Tree Tab"](#)
- [Section 46.3, "Sharing Blog Assets"](#)
- [Section 46.4, "Creating Workflows for Blog Assets"](#)
- [Section 46.5, "Working with Blog Authors"](#)
- [Section 46.6, "Publishing Blog Assets"](#)
- [Section 46.7, "Next Steps"](#)

46.1 Enabling Blog Asset Types

If you wish to use the Community Blogs module on different CM sites you must enable the blog asset types on those sites. When you enable the blog asset types you can also enable their start menu items. Start menu items allow content providers to create and search for assets of the types that you enable.

To enable blog asset types

1. Log in to the WebCenter Sites Admin interface as a general administrator.
2. Select a site.
3. In the tree, select the **Admin** tab, expand the **Sites** node, expand the node of the site for which you wish to enable the blog asset types, and expand the **Asset Types** node.
4. Double-click **Enable**.
5. In the Enable Asset Types form, select the check boxes of the blog asset types. For descriptions of the blog asset types, refer to the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
6. Click **Enable Asset Types**.
7. In the start menu selection form, click **Enable Asset Types**.
8. Authorize users to access the start menu items. For instructions, see [Chapter 9, "Managing Access to Content Management Site Components."](#)

46.2 Creating a Blog Tree Tab

A tree tab can be configured to display the hierarchy of blog categories and their assets.

To create a blog tree tab

1. Log in to the WebCenter Sites Admin interface as a general administrator.
2. Select a site.
3. In the **Admin** tab, double click the **Tree** node.
4. Click **Add New Tree Tab**.

The Add New Tree Tab form appears.

5. In the Add New Tree Tab form, fill in the fields:

Figure 46-1 Add New Tree Tab Form

Add New Tree Tab

***Title:**

Tooltip:

***Sites:**

- Any
- avisports
- FirstSiteI
- AdminSite
- Management Site

***Required Roles:**

- Any
- AdvancedUser
- Analytics
- Approver
- ArtworkAuthor

Tab Contents:

<p>Available</p> <div style="border: 1px solid gray; padding: 2px;"> <ul style="list-style-type: none"> Article Image Recommendation Article Category Attribute Editor CGGadget CGPoll CGWidget </div>	<p>Selected</p> <div style="border: 1px solid gray; padding: 2px; height: 100px;"> <!-- Empty selected list --> </div> <p style="text-align: right; margin-top: 5px;">Display Order</p> <div style="text-align: center; margin-top: 5px;"> <input type="button" value="▲"/> <input type="button" value="▼"/> </div>
---	--

- To create a custom section, enter the section name and element name, click "Add New Section", then Save. When the tree tab is reloaded, this element will be called.
- To edit an existing custom section, select the custom section name (in the Selected area above), make the changes, click Edit, then Save.

Section Name:

Element Name:

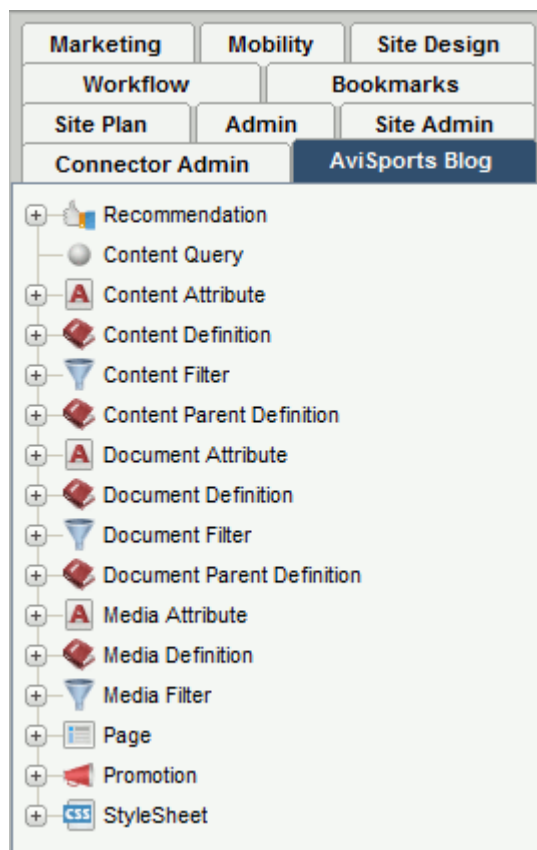
- **Title** - Enter a short, descriptive name, of up to 64 characters.
- **Tooltip** - Enter a short, informative description of the tab, up to 255 characters. This description is displayed when a user mouses over the tab.

- **Site** - Select the sites that will display this tab. The blog asset types must be enabled on these sites. For instructions, see [Section 46.1, "Enabling Blog Asset Types."](#)
 - **Required Roles** - Select the roles a user must be assigned to access this tab.
 - **Tab Contents** - Select the asset types that will be displayed on the tab and then click **Add Selected Items**.
 - (Optional) If you want to add custom functionality to this tab, use the **Section Name** and **Element Name** fields. For more information, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
6. Click **Save**.

The Information page for the newly created blog is displayed.

Additionally, the new blog tab is displayed in the tree (refresh your browser, if necessary).

Figure 46–2 Tree Showing New Blog



46.3 Sharing Blog Assets

You can share blog flex assets with other sites. All types of blog assets must be shared.

Note: The blog asset definition is required to define the attributes that make up the blog asset form.

To share blog assets

1. Log in to the WebCenter Sites Advanced interface as a general administrator.
2. Select a site.
3. Locate the asset you wish to share:
 - a. In the menu bar, click **Search**.
 - b. Click **Find** or **Advanced Search** for the asset type.
 - c. Fill the fields as appropriate, and click **Search**.
4. In the search results list, click the asset you wish to share.
5. At the top of the form in the button bar, click the **Share *element*** button.

The **Share *element*** page displays.

6. In the Share to: field, select the site(s) with which you wish to share the blog asset you selected.
7. Click **Save Changes**.

The blog asset you selected to share is now available on the site(s) with which you shared it.

For detailed information about sharing assets, see [Section 11.2.2.1, "Copying vs. Sharing."](#)

46.4 Creating Workflows for Blog Assets

Workflows can be used if you wish to manage the work on a blog asset. Before a blog asset can be published, it must be approved for publishing. The workflow feature routes blog assets through the steps that you determine as necessary, from creation to approval.

For example, if a blog asset must be reviewed by an editor before it can be approved for publishing, the WebCenter Sites' workflow feature can route that blog asset to the appropriate user at the appropriate time. For more information about workflow processes, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

46.5 Working with Blog Authors

On the sample blog pages, the author's full name and image are displayed next to the author's blog posts. This section provides instructions for creating blog author assets, which contain author information. This section also provides instructions for granting users access to their own blog author assets.

This section contains the following topics:

- [Section 46.5.1, "Creating Author Assets"](#)
- [Section 46.5.2, "Setting Access Permissions to a Blog Author Asset"](#)

46.5.1 Creating Author Assets

To display a blog author's full name and image next to the author's blog posts, you must create an author asset in which you specify that user's full name and optionally, an image.

To create an author asset

1. Log in to the WebCenter Sites Advanced interface as a general administrator.
2. Select the site on which the blog asset types are enabled.
3. In the button bar, click **New**.
4. Click **New Blog Author**.
5. In the Blog Author form, fill in the fields:

Figure 46-3 New Blog Author Form

Blog Author: (BlogAuthorDef)

*** Name:**

Description:

Select a Template: -- choose display style --

Filename:

Path:

Start Date: Format: yyyy-mm-dd hh:mm:ss

End Date: Format: yyyy-mm-dd hh:mm:ss

External Item ID:

Blog Author Definition: BlogAuthorDef

Blog Author Category:

*** BlogAuthorCategory (S):**

*** Full Name:**

Author Profile Image:

Ratings: [no Segments defined]
 no segment ratings apply

Related Items: [show local recommendations]

- **Name** - Enter a name for the blog author asset.

Note: When the author creates a blog asset, she must enter the same name into the blog asset's Author field. Otherwise, the author asset you have created will not be published to its allocated space

- **Description** - Enter a short, descriptive summary about the author asset (for example, the author's full name).
 - **Blog Author Category** - By default, this field is single-valued. The only available value for this field is **BlogAuthors**.
 - **Full Name** - Enter the blog author's full name. This name will be displayed on the blogs that this author creates.
 - **Author Profile Image** - Click **Browse** , and select an image for the content provider.
6. Click **Save**.

46.5.2 Setting Access Permissions to a Blog Author Asset

A blog author asset contains the profile of the content provider who creates blog content for the website. Setting access permissions for a blog author asset entails the following steps:

1. Create a unique role for the blog author.
2. Assign the new role to the blog author.
3. Grant the blog author access to the blog author asset.

The same steps are presented in detail below.

To create a unique role for the blog author

1. Log in to the WebCenter Sites Admin interface as a general administrator, and select a site.
2. In the tree, select the **Admin** tab, expand the **User Access Management** node, then expand the **Roles** node, and double-click **Add New**.

The Add New Role form opens:

Figure 46–4 Add New Role Form

Add New Role

***Name:**

***Description:**

Cancel

Add New Role

- **Name** - Enter a unique name for the role.

- **Description** - Enter a short, descriptive sentence explaining what the role will do.
3. Click **Add New Role**.
Now you must assign the role to the content provider who requires access permissions to the blog author asset.

To assign the role to the blog author

1. Select the **Admin** tab, expand the **Sites** node, and then expand the node of the author's site. Double-click **Users**.
2. In the Username field, enter the name of the user to which you wish to assign the role, and click **Select**.
3. Click the edit (pencil) icon.
4. In the Edit Roles for User form, select the role you created in steps 2 -3.
5. Click **Save**.

To set access permissions to a blog author asset

1. Access the blog author asset to which you want to set permissions:
 - a. In the button bar, click **Search**.
 - b. Click **Find Blog Author**.
 - c. Click **Search**.
2. Select the blog author asset.
3. In the blog author's Inspect form, select the **more...** drop-down menu, and click **Access Permissions**.

Figure 46-5 Access Permissions Form

Blog Author: firstsite

Preview
 Inspect
 Edit
 Delete

 Add to My Active List

Name: firstsite
Description: First Site
Status: Created
Modified: Thursday, May 27, 2010 10:06:12 AM EDT by BrentP

View by role
 View by function

Roles

Roles	Inherited	Functions Granted	Denied
<ul style="list-style-type: none"> AdvancedUser Analyst Analytics Approver ArtworkAuthor ArtworkEditor Author BrentPProfileAccess Checker ContentAuthor ContentEditor DashUser 	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> Grant Deny </div>	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div> <div style="text-align: center; margin-top: 5px;"> Remove </div>	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div> <div style="text-align: center; margin-top: 5px;"> Remove </div>

4. Select the **View by role** radio button.
5. In the Roles list box, select the role you created in steps 2-3 of the section "[To assign the role to the blog author](#)" on page 46-8.
6. In the Inherited list box, select all the functions (**Shift+click** the first and last functions), and click **Grant**.

All the functions in the Inherited list box move to the Granted list box.

7. Click **Save**.

The only user who can work with this blog author asset, besides the administrator, is the blog author who was assigned the selected role.

46.6 Publishing Blog Assets

If you are planning to publish a blog asset, you must pre-approve its author asset. This is to ensure that the author's profile is published with the blog asset.

46.7 Next Steps

This chapter provides administrators with instructions for enabling content providers to create and publish blog assets. For more information on testing and further

developing the blog data model, to create blog pages for publication, and to work with the Community application's functionality on blog pages, see the *Oracle Fusion Middleware WebCenter Sites User's Guide*.

Part VI

Implementing Oracle WebCenter Sites: Site Capture

This part describes Oracle WebCenter Sites: Site Capture, a web application used to download dynamically published websites. This part begins with an overview of the Site Capture application. It shows you how to navigate the Site Capture application and its file system to manage crawl sessions and downloaded websites. The concluding chapter presents configuration code used to control the crawler's site capture process. Examples illustrate the use of Java methods and interfaces for implementing link extraction logic, rewriting URLs, enabling email notification at the end of a crawl session, and otherwise customizing the capture process.

This part contains the following chapters:

- [Chapter 47, "Overview of the Site Capture Application"](#)
- [Chapter 48, "Managing Downloaded Sites"](#)
- [Chapter 49, "Site Capture File System"](#)

Overview of the Site Capture Application

This chapter introduces the Oracle WebCenter Sites: Site Capture application and shows you how to navigate its interface.

This chapter contains the following topics:

- [Section 47.1, "Site Capture Model"](#)
- [Section 47.2, "Logging in to the Site Capture Application"](#)
- [Section 47.3, "Using the Default Crawlers"](#)
- [Section 47.4, "Setting Up a Site Capture Operation"](#)
- [Section 47.5, "Enabling Publishing-Triggered Site Capture"](#)

47.1 Site Capture Model

Crawls can be initiated manually from the Site Capture interface, or they can be triggered by the completion of a WebCenter Sites RealTime publishing session. In each scenario, the crawler downloads the website to disk in one of the following modes: static or archive, depending on how you choose to run the crawler.

This section contains the following topics:

- [Section 47.1.1, "Capture Modes"](#)
- [Section 47.1.2, "Crawlers"](#)

47.1.1 Capture Modes

When a site is downloaded in either static or archive mode, the same files (html, css, and so on) are stored to disk, but with several differences. For example, statically downloaded sites are available only in the file system, whereas archived sites are available in both the file system and the Site Capture interface. Capture mode, then, determines how crawlers download sites and how you manage the results.

Static Mode	Archive Mode
Static mode supports rapid deployment, high availability scenarios.	Archive mode is used to maintain copies of websites on a regular basis for compliance purposes or similar reasons.
In static mode, a crawled site is stored as files ready to be served. Only the latest capture is kept (the previously stored files are overwritten).	In archive mode, all crawled sites are kept and stored as zip files (archives) in time-stamped folders. Pointers to the zip files are created in the Site Capture database.

Static Mode	Archive Mode
Static crawl sessions can be initiated manually from the application interface or at the end of a publishing session. However, the downloaded sites can be accessed and managed only from the Site Capture file system.	Archive crawl sessions, like static sessions, can be initiated manually from the Site Capture interface or at the end of a publishing session. However, because the zip files are referenced by pointers in the Site Capture database, they can be managed from the Site Capture interface. Here, you can download the files, preview the archived sites, and set capture schedules.

For any capture mode, logs are generated at the end of the crawl session to provide such information as crawled URLs, HTTP status, and network conditions. In static capture, the logs must be obtained from the file system. In archive capture, they can be downloaded from the Site Capture interface. For any capture mode, you have the option of configuring crawlers to email reports as soon as they are generated.

47.1.2 Crawlers

Starting any type of site capture process requires a crawler to be defined in the Site Capture interface. To help you get started quickly, Site Capture comes with two sample crawlers, Sample and FirstSiteII. We assume the crawlers were installed during the Site Capture installation process. This guide uses mainly the Sample crawler.

Creating your own crawler involves naming the crawler (typically, after the target site), and uploading a text file named `CrawlerConfigurator.groovy`, which controls the crawler's site capture process. The `groovy` file must be coded with methods in the `BaseConfigurator` class that specify at least the starting URI(s) and link extraction logic for the crawler. Although the `groovy` file controls the crawler's site capture process, the crawler's capture mode is set outside the file.

Using a crawler for publishing-triggered site capture requires taking an additional step – you will name the crawler and specify its capture mode on the publishing destination definition on the WebCenter Sites source system that is integrated with Site Capture, as described in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*. (Bear in mind that on every publishing destination definition, you can specify one or more crawlers, but only a single capture mode.) Information about the success of crawler invocation is stored in the Site Capture file system and in the log files (`futuretense.txt`, by default) of the WebCenter Sites source and target systems.

The exercises in this chapter cover both types of crawler invocation scenarios: manual and publishing-triggered.

47.2 Logging in to the Site Capture Application

The Site Capture application runs on WebCenter Sites. You will access the Site Capture application by logging in to WebCenter Sites.

To log in to the Site Capture application

1. Access WebCenter Sites at the following URL:

```
http://<server>:<port>/<context>/login
```

where `<server>` is the host name or IP address of the server running WebCenter Sites, `<port>` is the number of the WebCenter Sites application, and `<context>` is the name of the WebCenter Sites web application that was deployed on the server.

2. Log in as a general administrator. **Login credentials are case sensitive.** In this guide, we use the default credentials:

Username: fwadmin
Password: xceladmin

Figure 47-1 Login Dialog



3. Click **Login**.
4. If you are logging in for the first time, the following dialog opens:

Figure 47-2 Login Dialog



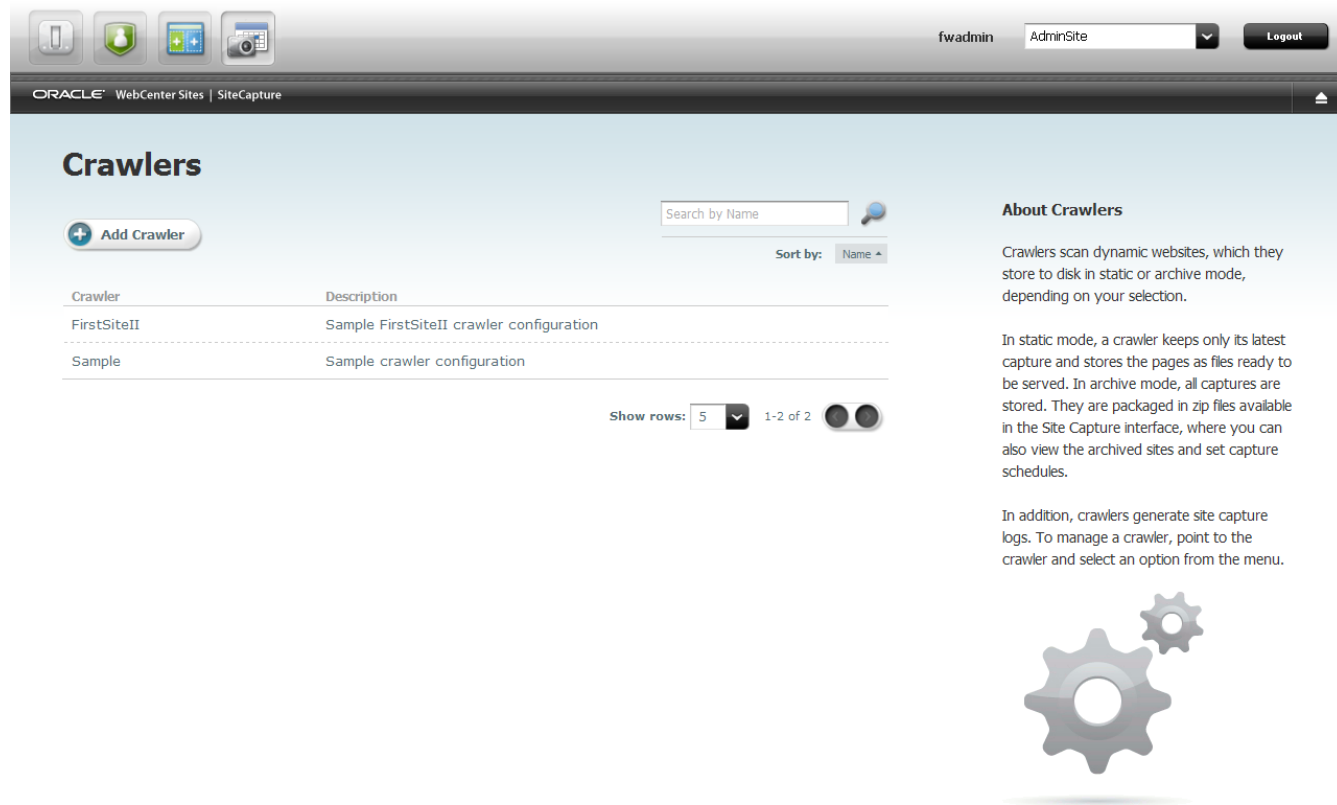
Select the AdminSite (to which the Site Capture application is assigned by default) and select the Site Capture icon.

Figure 47-3 Login Dialog



5. The first page you see is named Crawlers. If the default crawlers were installed with Site Capture, you will see them listed under the names **Sample** and **FirstSiteII**.

Figure 47–4 Crawlers Page



6. Your next step can be any one of the following, depending on your requirements:
 - If you wish to learn more about the default crawlers, continue to [Section 47.3, "Using the Default Crawlers."](#)
 - To set up your own site capture operation and, in the process, learn to navigate the Site Capture interface, skip to [Section 47.4, "Setting Up a Site Capture Operation."](#)
 - To learn about crawler configuration code, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

47.3 Using the Default Crawlers

In this guide, we assume that the default crawlers Sample and FirstSiteII were installed with the Site Capture application, and they are displayed in its interface (as shown in step 1 in [Section 47.2, "Logging in to the Site Capture Application"](#)). If you wish to define your own crawlers, see [Section 47.4, "Setting Up a Site Capture Operation."](#)

This section contains the following topics:

- [Section 47.3.1, "Sample Crawler"](#)

- [Section 47.3.2, "FirstSiteII Crawler"](#)
- [Section 47.3.3, "Running a Default Crawler"](#)

47.3.1 Sample Crawler

The Sample crawler can be used to download any site. The purpose of the Sample crawler is to help you quickly download the site and to provide you with required configuration code, which you will reuse when creating your own crawlers. The Sample crawler is minimally configured with the required methods and an optional method that limits the duration of the crawl by limiting the number of links to crawl.

- The required methods are `getStartURI` and `createLinkExtractor` (which defines the logic for extracting links from crawled pages).
- The optional method is `getMaxLinks`, which specifies the number of links to crawl.

More information about these methods as well as crawler customization methods and interfaces is available in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

47.3.2 FirstSiteII Crawler

The FirstSiteII crawler is used to download WebCenter Sites' dynamic FirstSiteII sample website as a static site. The purpose of the crawler is to provide you with advanced configuration code that shows how to create a custom link extractor and resource rewriter, using the `LinkExtractor` and `ResourceRewriter` interfaces. More information about the interfaces is available in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

47.3.3 Running a Default Crawler

In this section, you will run either the Sample crawler or the FirstSiteII crawler. Using the FirstSiteII crawler requires WebCenter Sites' FirstSiteII sample site to be published.

To run a default crawler

1. On the Crawlers page, point to one of the default crawlers – **Sample** or **FirstSiteII** – and select **Edit Configuration**.

Note: If the default crawlers are not listed, skip to [Section 47.4, "Setting Up a Site Capture Operation"](#) to define your own crawler.

2. Set the crawler's starting URI by editing the crawler's configuration file. For instructions, skip to 1 in [Section 47.4.2, "Defining a Crawler,"](#) and continue with the rest of the steps to run the crawler and manage its captured data.

47.4 Setting Up a Site Capture Operation

In this section, you will step through the process of creating and running your own crawler to understand how the Site Capture interface and file system are organized.

This section contains the following topics:

- [Section 47.4.1, "Creating a Starter Crawler Configuration File"](#)
- [Section 47.4.2, "Defining a Crawler"](#)
- [Section 47.4.3, "Editing the Crawler Configuration File"](#)

- [Section 47.4.4, "Starting a Crawl"](#)

47.4.1 Creating a Starter Crawler Configuration File

Before you can create a crawler, you must have a configuration file that controls the crawler's site capture process. The fastest way to create a useful file is to copy sample code and recompile, as necessary.

To create a starter crawler configuration file

1. Copy the **Sample** crawler's configuration file to your local machine in one of the following ways:
 - Log in to the Site Capture application. If the Crawlers page lists the Sample crawler, do the following (otherwise, skip to the item directly below):
 - a. Point to **Sample** and select **Edit Configuration**.
 - b. Go to the **Configuration File** field, copy its code to a text file on your local machine, and save the file as `CrawlerConfigurator.groovy`.
 - Go to the Site Capture host machine and copy the `CrawlerConfigurator.groovy` file from `<SC_INSTALL_DIR>/fw-site-capture/crawler/Sample/app/` to your local machine.

Note:

Every crawler is controlled by its own `CrawlerConfigurator.groovy` file. The file is stored in a custom folder structure. For example:

When you define a crawler, Site Capture creates a folder bearing the name of the crawler (`<crawlerName>`, or `Sample` in our scenario) and places that folder in the following path: `<SC_INSTALL_DIR>/fw-site-capture/crawler/`. Within the `<crawlerName>` folder, Site Capture creates an `/app` subfolder to which it uploads the `groovy` file from your local machine.

When the crawler is used for the first time in a given mode, Site Capture creates additional subfolders (in `/<crawlerName>/`) to store sites captured in that mode. For more information about the Site Capture file system, see [Section 48.1, "Managing Statically Captured Sites."](#)

2. Your sample `groovy` file specifies a sample starting URI, which you will reset for the crawler you will be creating in the next step. (In addition to the starting URI, you can set crawl depth and similar parameters, invoke post-crawl commands, and implement interfaces to define logic specific to your target sites.)

At this point, you have the option to either customize the downloaded `groovy` file now, or first create the crawler and then customize its `groovy` file (which is editable in the Site Capture interface).

- To follow this exercise, continue to the next step [Section 47.4.2, "Defining a Crawler."](#)
- For information about crawler configuration methods and interfaces, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

47.4.2 Defining a Crawler

1. Go to the Crawlers page and click **Add Crawler**.

Figure 47–5 Crawlers Page

The screenshot shows the Oracle WebCenter Sites SiteCapture interface. At the top, there are navigation icons and a user profile for 'fwadmin' with a dropdown menu showing 'AdminSite' and a 'Logout' button. The main heading is 'Crawlers'. Below the heading is an 'Add Crawler' button. To the right is a search box labeled 'Search by Name' and a 'Sort by: Name' dropdown. A table lists the following crawlers:

Crawler	Description
FirstSiteII	Sample FirstSiteII crawler configuration
Sample	Sample crawler configuration

Below the table, there are pagination controls: 'Show rows: 5' (with a dropdown), '1-2 of 2', and two radio buttons. On the right side, there is an 'About Crawlers' section with text explaining that crawlers scan dynamic websites and store captures in static or archive mode. It also mentions that crawlers generate site capture logs. At the bottom right, there is a graphic of two interlocking gears.

2. On the Add Crawlers page:

Figure 47–6 Add Crawler Page

The screenshot shows the 'Add Crawler' page in the Oracle WebCenter Sites Site Capture interface. The page has a header with 'ORACLE WebCenter Sites | SiteCapture' and a user menu for 'fwadmin' with 'AdminSite' selected and a 'Logout' button. The main content area is titled 'Add Crawler' and includes a 'Crawler Configuration' section with the following fields:

- Name:** MySite
- Description:** This crawler is used for scheduled archive capture.
- Configuration File:** CrawlerConfigurator.groovy (with a 'Browse' button)

At the bottom of the form are 'Save' and 'Cancel' buttons. To the right, the 'Adding a Crawler' section contains the following text:

Be sure of the name you are assigning to the crawler. Once saved, a crawler cannot be renamed.

The configuration file controls the crawler's site capture process. This text file, named `CrawlerConfigurator.groovy`, can be edited in the Site Capture interface once the crawler is saved. For information about coding the file, see the Site Capture documentation.

Below the text is an icon of two interlocking gears.

- a. Name the crawler after the site to be crawled.

Note:

- Once saved, a crawler cannot be renamed.
 - Throughout this guide, we assume that every custom crawler is named after the target site and will not be used to capture any other site.
-
-

- b. Enter a description (optional). For example: "This crawler is reserved for publishing-triggered site capture" or "This crawler is reserved for scheduled captures."
- c. In the **Configuration File** field, browse to the `groovy` file that you created in [Section 47.4.1, "Creating a Starter Crawler Configuration File."](#)
- d. Save the new crawler.

Your `CrawlerConfigurator.groovy` file is uploaded to the `<SC_INSTALL_DIR>/fw-site-capture/crawler/<crawlerName>/app` folder on the Site Capture host machine. The file can be edited directly in the Site Capture interface.

3. Continue to [Section 47.4.3, "Editing the Crawler Configuration File."](#)

47.4.3 Editing the Crawler Configuration File

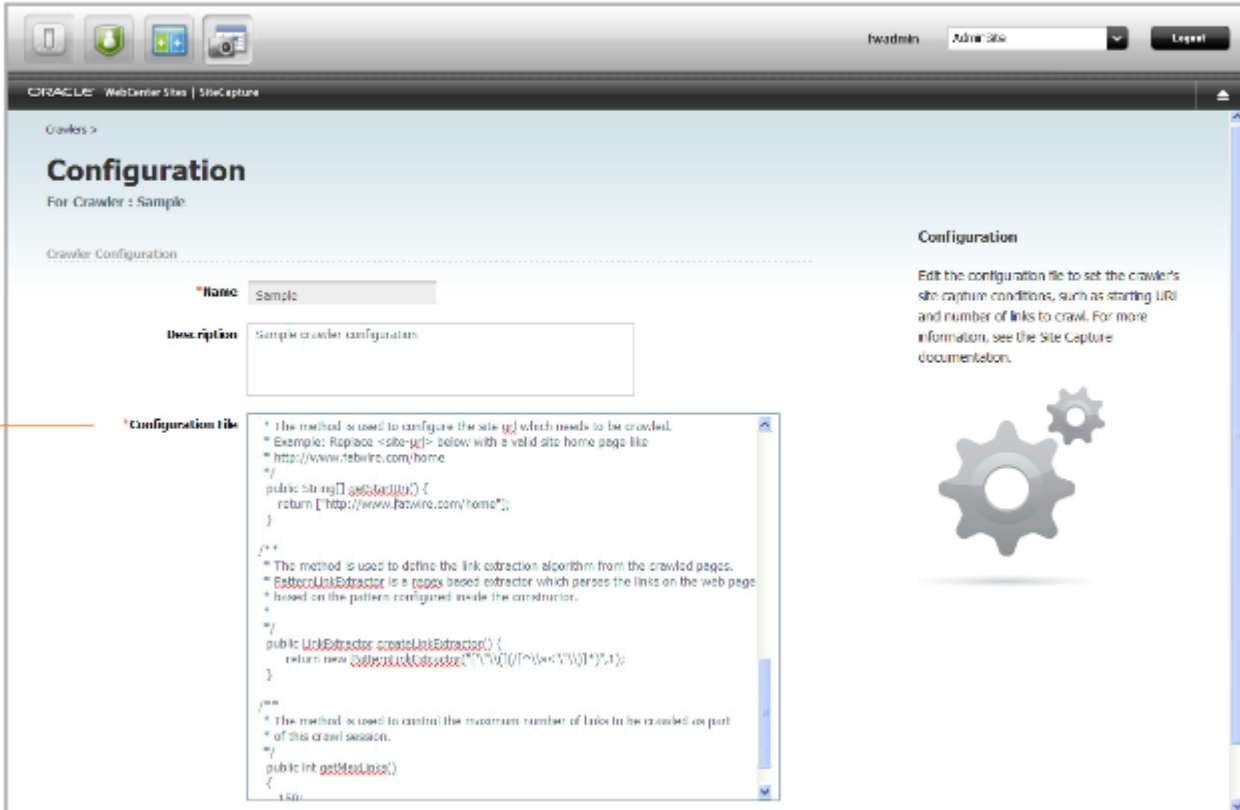
From the Site Capture interface, you can recode the entire crawler configuration file. In this example, we simply set the crawler's starting URI.

To edit the crawler configuration file

1. On the Crawlers page, point to the crawler you just defined and select **Edit Configuration**.

Figure 47-7 Configuration Page

This field displays the crawler's `CrawlerConfigurator.groovy` file, located in `<SC_INSTALL_DIR>/fw-site-capture/crawler/<crawlerName>/app`.

The screenshot shows the Oracle WebCenter Sites Site Capture interface. The main content area is titled 'Configuration' and is for the crawler 'Sample'. It includes fields for 'Name' (Sample) and 'Description' (Sample crawler configuration). The 'Configuration File' field is expanded, showing a Groovy script. The script includes comments and methods for configuring the site URI, defining link extraction algorithms, and controlling the maximum number of links to be crawled. A callout box with an orange border points to the 'Configuration File' field, containing the text: 'This field displays the crawler's CrawlerConfigurator.groovy file, located in <SC_INSTALL_DIR>/fw-site-capture/crawler/<crawlerName>/app'. On the right side of the configuration page, there is a 'Configuration' section with a gear icon and text: 'Edit the configuration file to set the crawler's site capture conditions, such as starting URI and number of links to crawl. For more information, see the Site Capture documentation.'

2. Set the crawler's starting URI in the following method:

```
public String[] getStartUri() {
    return ["http://www.mycompany.com/home"]
}
```

Note: Take care of the following:

You can set multiple starting URIs. They must belong to the same site. Enter a comma-separated array, as shown in the example below:

```
public String[] getStartUri() { return
["http://www.fatwire.com/product", "http://www.fatwire.com/support"]
; }
```

Your configuration file includes the `createLinkExtractor` method, which calls the logic for extracting the links to be crawled. The links will be extracted from the markup that is downloaded during the crawl session. For additional information about this method and the extraction logic, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

Your configuration file also includes the `getMaxLinks()` method, which specifies the number of links to crawl. Its default value is set to 150 to ensure a quick run. If for some reason you have to stop a static capture, you will have to stop the application server. Archive captures can be stopped from the Site Capture interface.

For information about crawler configuration methods and interfaces, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

3. Click **Save**.
4. Continue to [Section 47.4.4, "Starting a Crawl."](#)

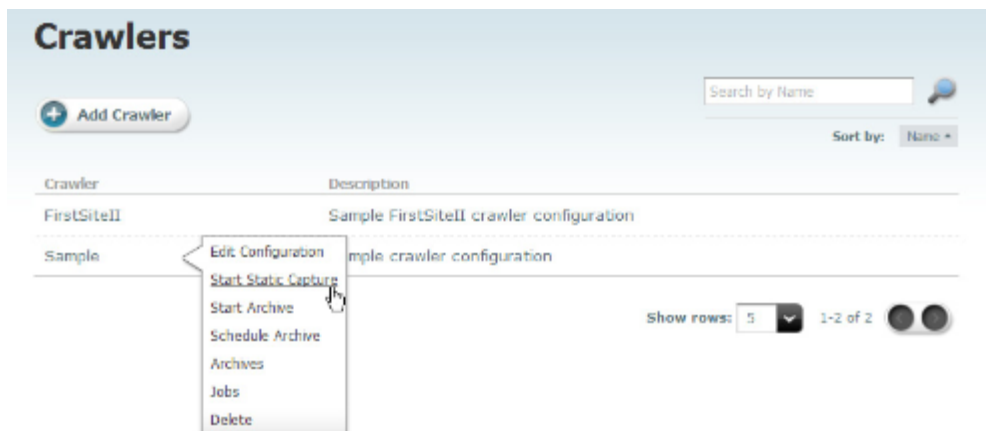
47.4.4 Starting a Crawl

You can start a crawl in several ways:

- [Section 47.4.4.1, "Run the Crawler Manually in Static Mode"](#)
- [Section 47.4.4.2, "Run the Crawler Manually in Archive Mode"](#)
- [Section 47.4.4.3, "Schedule the Crawler for Archive Capture"](#)
- [Section 47.4.4.4, "Publish a Site in RealTime Mode"](#)

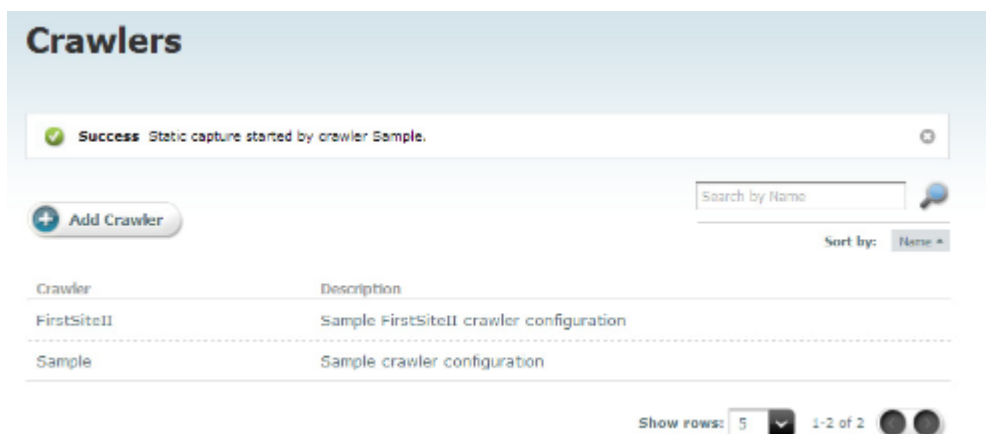
47.4.4.1 Run the Crawler Manually in Static Mode

1. On the Crawlers page, point to the crawler that you created and select **Start Static Capture** from the drop-down menu.

Figure 47–8 Crawlers Page Detail with Drop-Down Menu Expanded

The Crawlers page displays the following message when capture begins:

"**Success.** Static capture started by crawler <crawlerName>."

Figure 47–9 Crawlers Page Detail Showing Status Message

2. At this point, the Site Capture interface does not display any other information about the crawler or its process, nor will it make the downloaded site available to you. Instead, you will use the Site Capture file system to access the downloaded files and various logs:
 - To monitor the static capture process, look for the following files:
 - The lock file in <SC_INSTALL_DIR>/fw-site-capture/<crawlerName>/logs. The lock file is transient. It is created at the start of the static capture process to prevent the crawler from being invoked for an additional static capture. The lock file is deleted when the crawl session ends.
 - The crawler.log file in <SC_INSTALL_DIR>/fw-site-capture/logs/. (This file uses the term "VirtualHost" to mean "crawler.")
 - The inventory.db file in <SC_INSTALL_DIR>/fw-site-capture/<crawlerName>. This file lists the crawled URLs. The inventory.db file is used by the Site Capture system and must not be deleted or modified.

- The `audit.log`, `links.txt` file, and `report.txt` files are available in `/fw-site-capture/crawler/<crawlerName>/logs/yyyy/mm/dd`.
- To access the downloaded files, go to `<SC_INSTALL_DIR>/fw-site-capture/crawler/<crawlerName>/www`.

For more information about the Site Capture file system, see [Section 48.1, "Managing Statically Captured Sites."](#)

47.4.4.2 Run the Crawler Manually in Archive Mode

If a crawler was used in one mode, it can be rerun in a different mode.

To run the crawler in archive mode

1. On the Crawlers page, point to the crawler that you created and select **Start Archive**.

A dialog box opens.

2. In the dialog, add a comment about the crawler's upcoming job:

Figure 47–10 Comment Dialog



Note: A comment cannot be added once a crawler starts running.

If you choose to add a comment in the dialog above, it will be displayed in the following places:

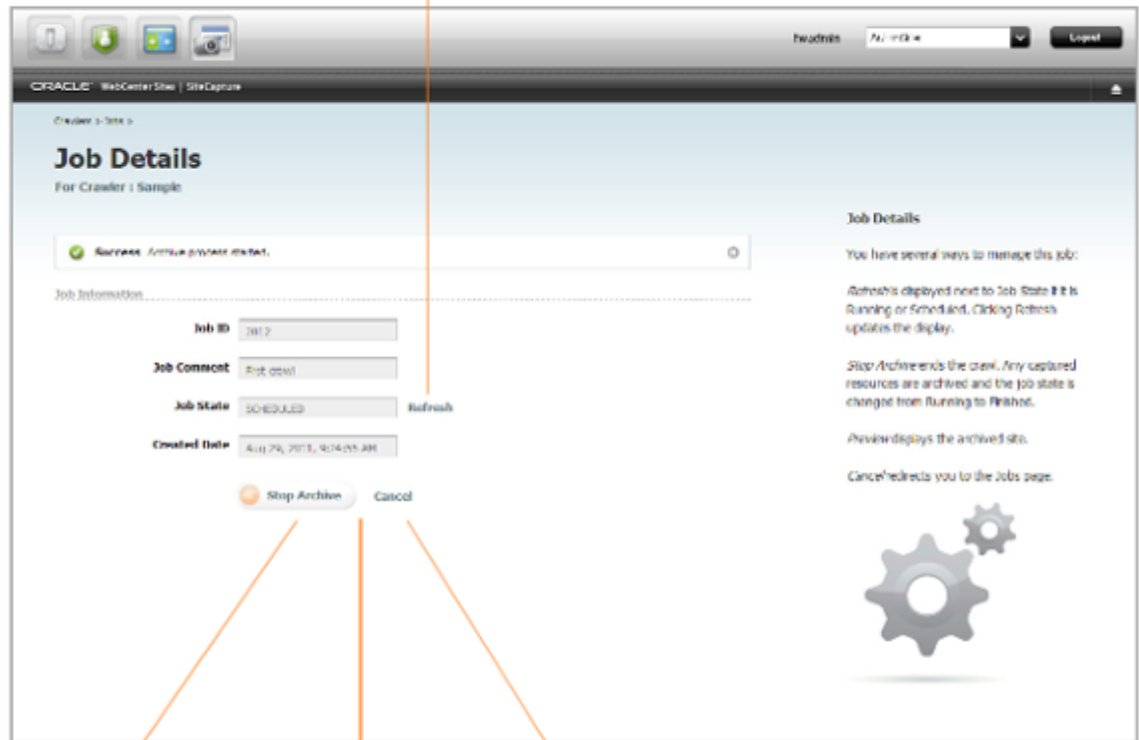
- Job Comment field on the Job Details page (shown in the next step).
 - Job Comment field on the Jobs page.
 - Comment field on the Archives page.
-

3. Click **Start Archive**.

The Job Details page is displayed, where you can manage the archive process in several ways, as shown in [Figure 47–11](#). To follow this exercise, click **Refresh** (next to **Job State**) until Finished is displayed, then continue to the next step.

Figure 47–11 Job Details Page

Refresh is shown as long as the job state is Scheduled or Running. Clicking **Refresh** updates the displayed job state. Possible job states are Scheduled, Running, Finished, Stopped, and Failed.



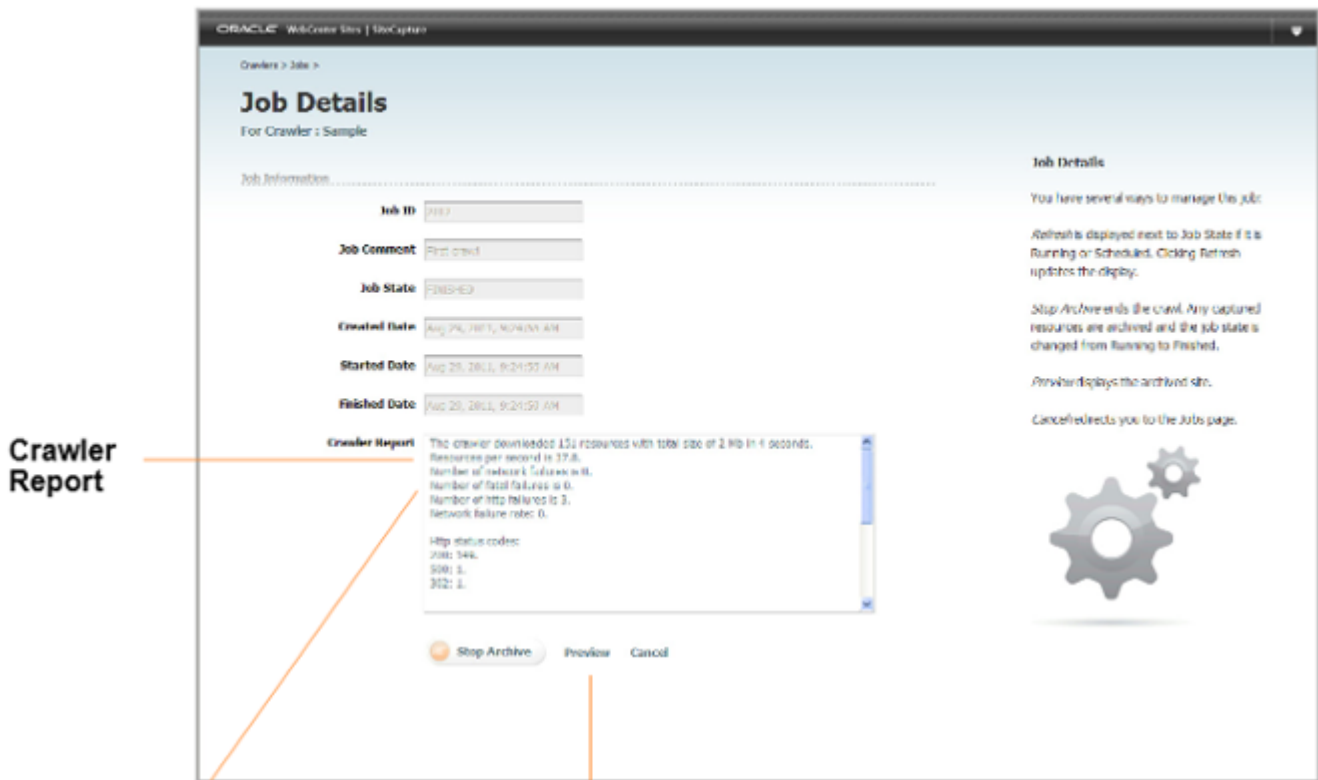
Stop Archive ends the crawler's session. Any captured resources are archived and the job state is changed from "Running" to "Finished" (click **Refresh** to see the change).

Cancel redirects you to the "Jobs" screen. The crawler, if running, continues to run.

Preview will be displayed here when the job state is "Finished." Clicking **Preview** displays the archived site.

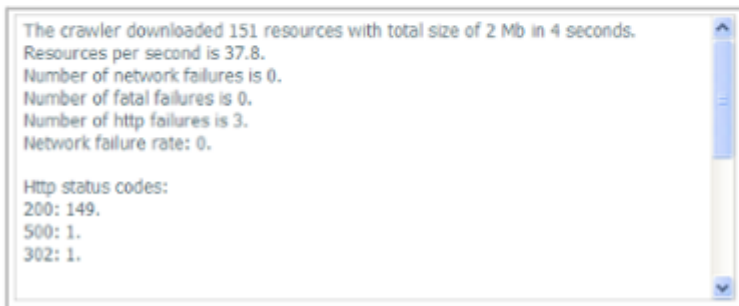
4. When the archive crawl ends, results are made available in the Site Capture interface. For example:
 - The crawler report is displayed on the Job Details page. The report lists the number of downloaded resources, their total size and download time, network conditions, HTTP status codes, and additional notes as necessary.

Figure 47-12 Job Details Page, Showing Crawler Report



Crawler Report

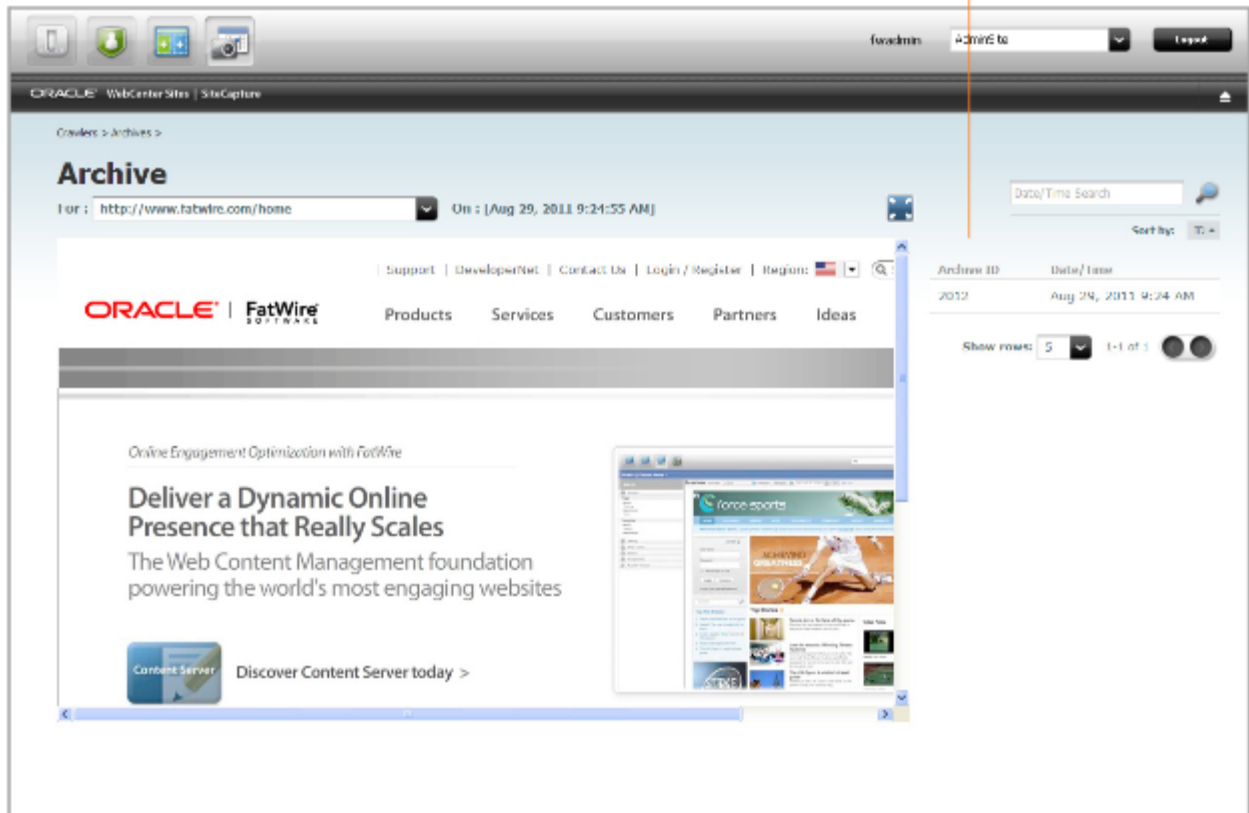
Clicking **Preview** renders the archived site



- Clicking **Preview** on the Job Details page renders the archived site (as shown in Figure 47-13). Next to the site is the Archive ID table with archive management options, which are displayed when you point to an archive.

Figure 47-13 Archived Site

Point to the archive to open its management menu with options to Preview, Download, Archive, and View URLs



Note: If your archived site contains links to external domains, its preview is likely to include those links, especially when the crawl depth and number of links to crawl are set to large values (in the `CrawlerConfigurator.groovy` file). Although the external domains can be browsed, they are not archived.

- For a summary of pathways to various data, see [Section 48.2, "Managing Archived Sites."](#)

47.4.4.3 Schedule the Crawler for Archive Capture

Only archive captures can be scheduled. For a given crawler, you can create multiple schedules – for example, one for capturing periodically, and another for capturing at a particular *and* unique time.

Note: If you set multiple schedules, ensure they do not overlap.

To schedule a crawler for archive capture

1. Go the Crawlers page, point to the crawler that you created and select **Schedule Archive**.
2. Click **Add Schedule** and make selections on all calendars: Days, Dates, Months, Hours, and Minutes.

Figure 47-14 Add Schedule Page Detail

Add Schedule
For Crawler : Sample

Comment

Enabled Date and Time

Days: Any
Dates: 1 of each selected month
Months: January, April, July, October
Hours: 12AM
Minutes: at :05 of each selected hour

Recurrence Pattern

Days Dates Months

Select: All | None

January	February	March	April	May	June
July	August	September	October	November	December

Information:
Event will run only on the selected days of the selected months.

Times of Recurrence

Hours	Minutes
12 1 2 3 4 5 6 7 8 9 10 11 AM	0 5 10
12 1 2 3 4 5 6 7 8 9 10 11 PM	15 20 25
	30 35 40
	45 50 55

Information:
Event will run only on the selected minutes during the selected hours.

Save Cancel

3. Click **Save** and add another schedule if necessary.

47.4.4.4 Publish a Site in RealTime Mode

If you configure your WebCenter Sites publishing systems to communicate with the Site Capture application, you can set up a RealTime publishing process to invoke one or more crawlers to capture the newly published site. For instructions, see [Section 47.5, "Enabling Publishing-Triggered Site Capture."](#)

47.4.5 Managing Captured Data

Information about accessing various data associated with static and archive captures can be found in [Chapter 48, "Managing Downloaded Sites."](#) [Section 48.3, "Summary"](#) presents a collection of notes and tips to bear in mind when managing crawlers and captured data. Topics include the following:

- [Section 48.3.1, "Creating and Editing Crawlers"](#)

- [Section 48.3.2, "Deleting a Crawler"](#)
- [Section 48.3.3, "Scheduling a Crawler"](#)
- [Section 48.3.4, "Monitoring a Static Crawl"](#)
- [Section 48.3.5, "Stopping a Crawl"](#)
- [Section 48.3.6, "Downloading Archives"](#)
- [Section 48.3.7, "Previewing Sites"](#)
- [Section 48.3.8, "Configuring Publishing Destination Definitions"](#)
- [Section 48.3.9, "Accessing Log Files"](#)

47.5 Enabling Publishing-Triggered Site Capture

Your main steps for enabling publishing-triggered site capture are the following:

- [Section 47.5.1, "Integrating the Site Capture Application with Oracle WebCenter Sites"](#)
- [Section 47.5.2, "Configuring a RealTime Publishing Destination Definition for Site Capture"](#)
- [Section 47.5.3, "Matching Crawlers"](#)
- [Section 47.5.4, "Next Steps"](#)

An administrative user can configure as many publishing destination definitions for Site Capture as necessary and invoke as many crawlers as necessary.

47.5.1 Integrating the Site Capture Application with Oracle WebCenter Sites

You can enable site capture at the end of a RealTime publishing session only if the Site Capture application is first integrated with the WebCenter Sites source and target systems used in the publishing process. If Site Capture is not integrated, see the *Oracle Fusion Middleware WebCenter Sites Installation Guide* for integration instructions, then continue with the steps below.

47.5.2 Configuring a RealTime Publishing Destination Definition for Site Capture

When configuring a publishing destination definition, you will name the crawler(s) that will be invoked at the end of the publishing session. You will also specify the capture mode.

To configure the publishing destination definition

1. Go to the WebCenter Sites source system that is integrated with the Site Capture application (as shown in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*).
 - a. Create a RealTime publishing destination definition pointing to the WebCenter Sites target system that is integrated with Site Capture. For instructions on creating a RealTime publishing destination, see [Chapter 20, "Configuring the RealTime Publishing Process."](#)
 - b. In the More Arguments section of the publishing destination definition, name the crawler(s) to be invoked at the end of the publishing session, and set the capture mode by using the following parameters to control crawler invocation:

- `CRAWLERCONFIG`: Specify the name of each crawler. If you are using multiple crawlers, separate their names with a semicolon (;).

Examples:

For a single crawler:`CRAWLERCONFIG=crawler1`

For multiple crawlers:`CRAWLERCONFIG=crawler1; crawler2; crawler3`

Note: The crawler(s) that you specify here must also be configured and identically named in the Site Capture interface. Crawler names are case sensitive.

- `CRAWLERMODE`: To run an archive capture, set this parameter to `dynamic`. By default, static capture is enabled.

Example: `CRAWLERMODE=dynamic`

Note:

- If `CRAWLERMODE` mode is omitted or set to a value other than `dynamic`, static capture will start when the publishing session ends.
 - Both crawler parameters can be set in a single statement as follows: `CRAWLERCONFIG=crawler1; crawler2&CRAWLERMODE=dynamic`
 - While you can specify multiple crawlers, you can set only one mode. All crawlers will run in that mode. To run some of the crawlers in a different mode, configure another publishing destination definition.
-
-

2. Continue to the next section.

47.5.3 Matching Crawlers

Crawlers named in the publishing destination definition must exist in the Site Capture interface. Do the following:

- Verify that crawler names in the destination definition (step b in [Section 47.5.2, "Configuring a RealTime Publishing Destination Definition for Site Capture"](#)) and Site Capture interface are identical. The names are case sensitive.
- Ensure that a valid starting URI for the target site is set in each crawler's configuration file. For information about navigating to the crawler's configuration file, see [Section 47.4.3, "Editing the Crawler Configuration File."](#) For more information about writing configuration code, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

47.5.4 Next Steps

1. Having enabled publishing-triggered site capture, you are ready to publish the target site. When publishing ends, site capture begins. The invoked crawlers capture pages in either static or archive mode, depending on how you set the `CRAWLERMODE` parameter in the publishing destination definition (step b in [Section 47.5.2, "Configuring a RealTime Publishing Destination Definition for Site Capture"](#)).

2. To monitor the site capture process.

- For static capture, the Site Capture interface does not display any information about the crawl, nor will it make the captured site available to you.
 - To determine that the crawlers were invoked, open the `futuretense.txt` file on the source or target WebCenter Sites system.

Note: The `futuretense.txt` file on the WebCenter Sites source and target systems contains crawler invocation status for any type of crawl: static and archive.

- To monitor the capture process, go to the Site Capture file system and review the files that are listed in step 2 in [Section 47.4.4.1, "Run the Crawler Manually in Static Mode."](#)
- For dynamic capture, you can view the status of the crawl from the Site Capture interface.
 - a. Go to the Crawlers page, point to the crawler, and select **Jobs** from the pop-up menu.
 - b. On the Job Details page, click **Refresh** next to the Job State until you see "Finished." (Possible value for Job State are Scheduled, Running, Finished, Stopped, or Failed.) For more information about the Job Details page, see steps 3 and 4 in [Section 47.4.4.2, "Run the Crawler Manually in Archive Mode."](#)

3. Managing captured data.

When the crawl session ends, you can manage the captured site and associated data as follows:

- For a statically captured site, go to the Site Capture file system. For more information, see [Section 48.1, "Managing Statically Captured Sites."](#)
- For an archived site, use the Site Capture interface to preview the site and download the zip file and logs. For more information, see [Section 48.2, "Managing Archived Sites."](#)

Managing Downloaded Sites

Downloaded sites are managed either from the Site Capture file system or the interface, depending on whether they are statically captured or archived.

This chapter contains the following topics:

- [Section 48.1, "Managing Statically Captured Sites"](#)
- [Section 48.2, "Managing Archived Sites"](#)
- [Section 48.3, "Summary"](#)

48.1 Managing Statically Captured Sites

For every crawler that a user creates in the Site Capture interface, Site Capture creates an identically named folder in its file system. This custom folder, `<crawlerName>`, is used to organize the crawler's configuration file, captures, and logs as shown in [Figure 48-1, "Site Capture's Custom Folders: <crawlerName>"](#), while describes the `<crawlerName>` folder and its contents.

Note: To access static captures and logs, you will have to use the file system. Archive captures and logs are managed from the Site Capture interface (their location in the file system is included in this section).

Figure 48-1 Site Capture's Custom Folders: `<crawlerName>`

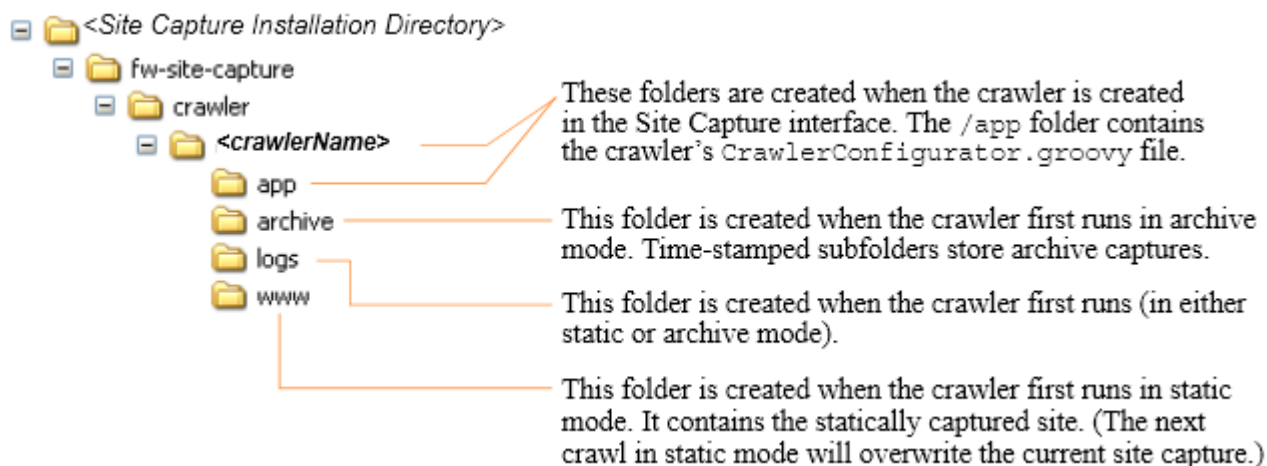
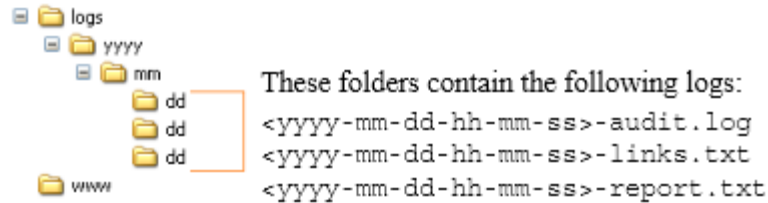


Table 48-1 *<crawlerName> Folder and Its Contents*

Folder	Description
/fw-site-capture/crawler/<crawlerName>	<p>Represents a crawler. For every crawler that a user defines in the Site Capture interface, Site Capture creates a /<crawlerName> folder. For example, if you installed the sample crawlers FirstSiteII and Sample, you will see both crawlers listed in the Site Capture interface, and you will find identically named folders in the Site Capture file system.</p> <p>Note: In addition to the subfolders (described below), the <crawlerName> folder contains an <code>inventory.db</code> file, which lists statically crawled URLs. The file is created when the crawler takes its first static capture. Do not delete or modify <code>inventory.db</code>. It is used by the Site Capture system.</p>
/fw-site-capture/crawler/<crawlerName>/app	<p>Contains the crawler's <code>CrawlerConfiguration.groovy</code> file. Its code controls the crawl process. The /app folder is created when the crawler is created and saved.</p>
/fw-site-capture/crawler/<crawlerName>/archive	<p>The /archive folder is used strictly for archive capture. This folder contains a hierarchy of <code>yyyy/mm/dd</code> subfolders. The /dd subfolder stores all of the crawler's archive captures in time-stamped zip files.</p> <p>The /archive folder is created when the crawler first runs in archive mode. The zip files (located in /dd) are referenced in the database and therefore made available in the Site Capture interface for you to download and display as websites.</p> <p>Note: Archive captures are accessible from the Site Capture interface. Each zip file contains a URL log named <code>__inventory.db</code>. Do not delete or modify <code>__inventory.db</code>. It is used by the Site Capture system.</p>
/fw-site-capture/crawler/<crawlerName>/www	<p>Contains only the latest <i>statically</i> captured site (when the same crawler is rerun in static mode, it overwrites the previous capture). The site is stored as <code>html</code>, <code>css</code>, and other files that can be readily served.</p> <p>The /www folder is created when the crawler first runs in static mode.</p> <p>Note: Static captures are accessible only from the Site Capture file system.</p>

Table 48–1 (Cont.) <crawlerName> Folder and Its Contents

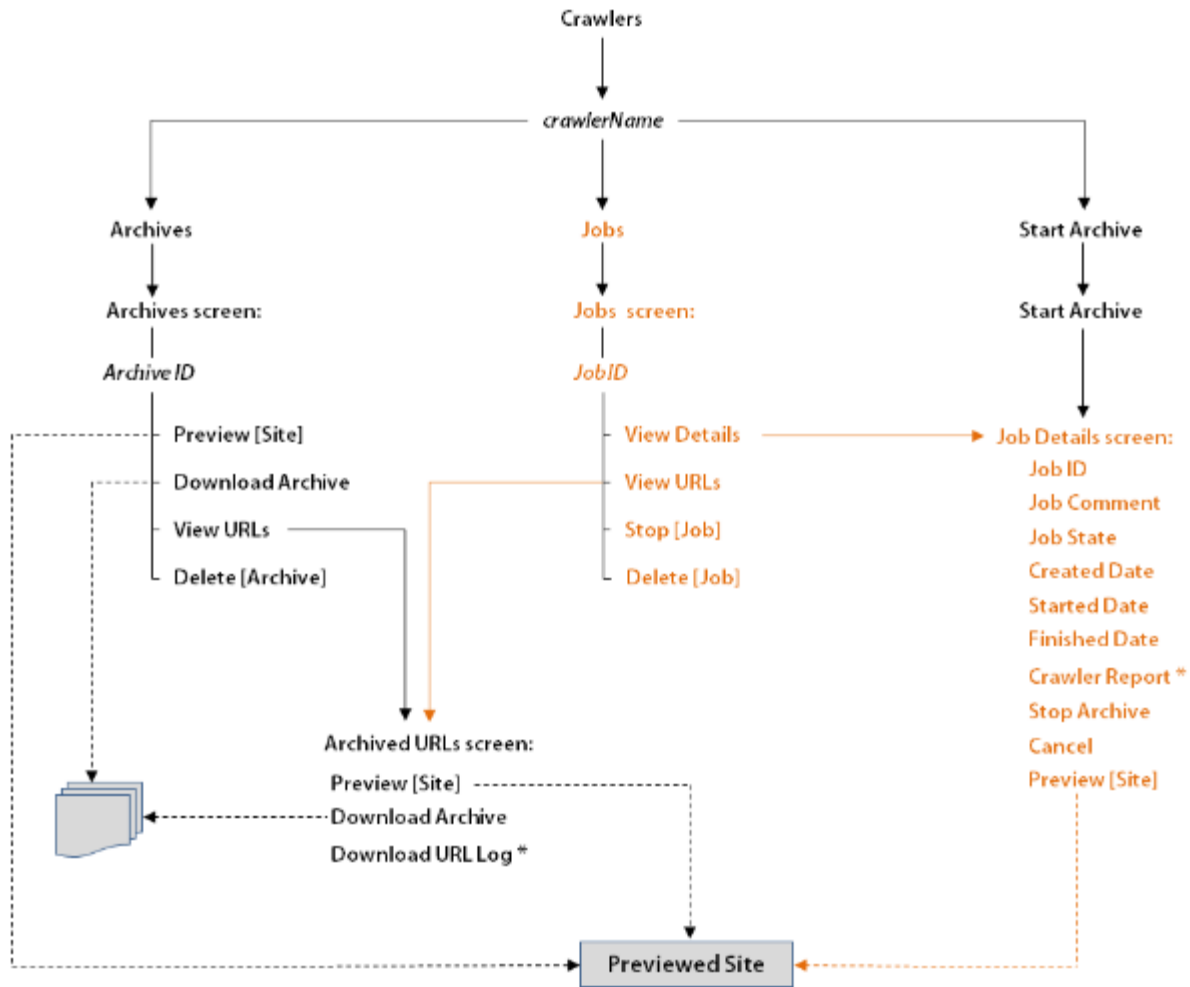
Folder	Description
/fw-site-capture/crawler/<crawler Name>/logs/yyyy/mm/dd	<p>Contains log files with information about crawled URLs. Log files are stored in the /dd subfolders and named as shown in Figure 48–2.</p> <ul style="list-style-type: none"> ▪ The <code>audit.log</code> file lists the crawled URLs with data such as timestamps, crawl depth, HTTP status, and download time. ▪ The <code>links.txt</code> file lists the crawled URLs. ▪ The <code>report.txt</code> file lists the overall crawl statistics such as number of downloaded resources, total size, download size and time, and network conditions. For archive capture, this report is available in the Site Capture interface as the crawler report (on the Job Details form. Paths to the Job Details form are shown in Figure 48–3, "Paths to Archive Information"). <p>Note: If the crawler captured in both static mode and archive mode, the /dd subfolders contain logs for static captures and archive captures.</p> <p>The /logs folder is also used to store a transient file named <code>lock</code>. The file is created at the start of the static capture process to prevent the crawler from being invoked for additional static captures. The <code>lock</code> file is deleted when the crawl session ends.</p>

Figure 48–2 Log Files Location

48.2 Managing Archived Sites

Archived sites can be managed from different forms of the Site Capture interface. [Figure 48–3](#), "Paths to Archive Information" shows some of the pathways to various information: archives, jobs, site preview, crawler report, and URL log:

Figure 48-3 Paths to Archive Information



- For example, to preview a site, start at the Crawlers form, point to a crawler (*crawlerName*), select **Archives** from the pop-up menu (which opens the Archives form), point to an **Archive ID**, and select **Preview** from the pop-up menu.
- Dashed lines represent multiple paths to the same option. For example, to preview a site, you can follow the crawler's **Archives** path, **Jobs** path, or **Start Archive** path. To download an archive, you can follow the **Archives** path or the **Jobs** path.
- The crawler report and URL log are marked by an asterisk (*).

48.3 Summary

This section summarizes notes and tips for managing crawlers and captured data.

This section contains the following topics:

- [Section 48.3.1, "Creating and Editing Crawlers"](#)
- [Section 48.3.2, "Deleting a Crawler"](#)
- [Section 48.3.3, "Scheduling a Crawler"](#)
- [Section 48.3.4, "Monitoring a Static Crawl"](#)

- [Section 48.3.5, "Stopping a Crawl"](#)
- [Section 48.3.6, "Downloading Archives"](#)
- [Section 48.3.7, "Previewing Sites"](#)
- [Section 48.3.8, "Configuring Publishing Destination Definitions"](#)
- [Section 48.3.9, "Accessing Log Files"](#)

48.3.1 Creating and Editing Crawlers

When creating crawlers and editing their configuration code, consider the following information:

- Crawler names are case sensitive.
- Every crawler's configuration file is named `CrawlerConfigurator.groovy`. The file is used to inject dependency. Hence, its name must not be changed.
- A crawler can be configured to start at one or more seed URIs on a given site and to crawl one or more paths. Additional Java methods enable you to set parameters such as crawl depth, invoke post-crawl commands, specify session timeout, and more. Interfaces can be implemented to define logic for extracting links, rewriting URLs, and sending email at the end of a crawl session. For more information, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.
- When a crawler is created and saved, its `CrawlerConfigurator.groovy` file is uploaded to the Site Capture file system and made editable in the Site Capture interface.
- While a crawler is running a static site capture process, it cannot be invoked to run a second static capture process.
- While a crawler is running an archive capture process, it can be invoked to run a second archive capture process. The second process will be marked as Scheduled and will start after the first process ends.

48.3.2 Deleting a Crawler

If you need to delete a crawler (which includes all of its captured information), do so from the Site Capture interface, not the file system. Deleting from the interface prevents broken links. For example, if a crawler ran in archive mode, deleting it from the interface removes two sets of information - the crawler's archives and logs, and database references to those archives and logs. Deleting the crawler from the file system retains database references to archives and logs that no longer exist, thus creating broken links in the Site Capture interface.

48.3.3 Scheduling a Crawler

Only archive crawls can be scheduled.

- When setting a crawler's schedule, consider the site's publishing schedule and avoid overlapping the two.
- You can create multiple schedules for a single crawler – for example, one schedule to invoke the crawler periodically, and another schedule to invoke the crawler at one particular and unique time.
- When creating multiple schedules, ensure they do not overlap.

48.3.4 Monitoring a Static Crawl

To determine whether a static crawl is in progress or completed, look for the crawler's lock file in the `<SC_INSTALL_DIR>/fw-site-capture/<crawlerName>/logs` folder. The lock file is transient. It is created at the start of the static capture process to prevent the crawler from being invoked for an additional static capture. The lock file is deleted when the crawl session ends.

48.3.5 Stopping a Crawl

Before running a crawler, consider the number of links to be crawled and the crawl depth, both of which determine the duration of the crawler's session.

- If you need to terminate an archive crawl, use the Site Capture interface (Select **Stop Archive** on the Job Details form).
- If you need to terminate a static crawl, you will have to stop the application server.

48.3.6 Downloading Archives

Avoid downloading large archive files (exceeding 250MB) from the Site Capture interface. Instead, use the `getPostExecutionCommand` to copy the files from the Site Capture file system to your preferred location.

Archive size can be obtained from the crawler report, on the Job Details form. Paths to the Job Details form are shown in [Figure 48–3, "Paths to Archive Information"](#). For information about the `getPostExecutionCommand` method, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.

48.3.7 Previewing Sites

If your archived site contains links to external domains, its preview is likely to include those links, especially when the crawl depth and number of links to crawl are set to large values (in the crawler's `groovy` file). Although the external domains can be browsed, they are not archived.

48.3.8 Configuring Publishing Destination Definitions

- If you are running publishing-triggered site capture, you can set crawler parameters in a single statement on the publishing destination definition:

```
CRAWLERCONFIG=crawler1;crawler2&CRAWLERMODE=dynamic
```
- While you can specify multiple crawlers on a publishing destination definition, you can set only one capture mode. All crawlers will run in that mode. To run some of the crawlers in a different mode, configure another publishing destination definition.

48.3.9 Accessing Log Files

- For statically captured sites, log files are available only in the Site Capture file system:
 - The `inventory.db` file, which lists statically crawled URLs, is located in the `/fw-site-capture/crawler/<crawlerName>` folder.

Note: The `inventory.db` file is used by the Site Capture system. It must not be deleted or modified.

- The `crawler.log` file is located in the `<SC_INSTALL_DIR>/fw-site-capture/logs/` folder. (The `crawler.log` file uses the term "VirtualHost" to mean "crawler.")
- For statically captured and archived sites, a common set of log files exists in the site Capture file system:
 - `audit.log`, which lists the crawled URLs, timestamps, crawl depth, HTTP status, and download time.
 - `links.txt`, which lists the crawled URLs
 - `report.txt`, which is the crawler report

The files named above are located in the following folder:

```
/fw-site-capture/crawler/<crawlerName>/logs/yyyy/mm/dd
```

Note: For archived sites, `report.txt` is also available in the Site Capture interface, on the Job Details form, where it is called the Crawler Report. (Paths to the Job Details form are shown in [Figure 48-3, "Paths to Archive Information"](#)).

- The archive process also generates a URL log for every crawl. The log is available in two places:
 - In the Site Capture file system, where it is called `__inventory.db`. This file is located within the zip file in the following folder:

```
/fw-site-capture/crawler/<crawlerName>/archive/yyyy/mm/dd
```

Note: The `__inventory.db` file is used by the Site Capture system. It must not be deleted or modified.

- In the Site Capture interface, in the Archived URLs form (whose path is shown in [Figure 48-3, "Paths to Archive Information"](#)).

Site Capture File System

The Site Capture file system is created during the Site Capture installation process to store installation-related files, property files, sample crawlers, and sample code used by the FirstSiteII crawler to control its site capture process. The file system also provides the framework in which Site Capture organizes custom crawlers and their captures.

This chapter contains the following topics:

- [Section 49.1, "General Directory Structure"](#)
- [Section 49.2, "Custom Folders"](#)

49.1 General Directory Structure

[Figure 49-1](#) shows Site Capture's most frequently accessed folders to help administrators find commonly used Site Capture information. All folders, except for <crawlerName>, are created during the Site Capture installation process. For information about <crawlerName> folders, see [Table 49-1, "Site Capture's Frequently Accessed Folders"](#) and [Section 49.2, "Custom Folders."](#)

Figure 49–1 Site Capture File System

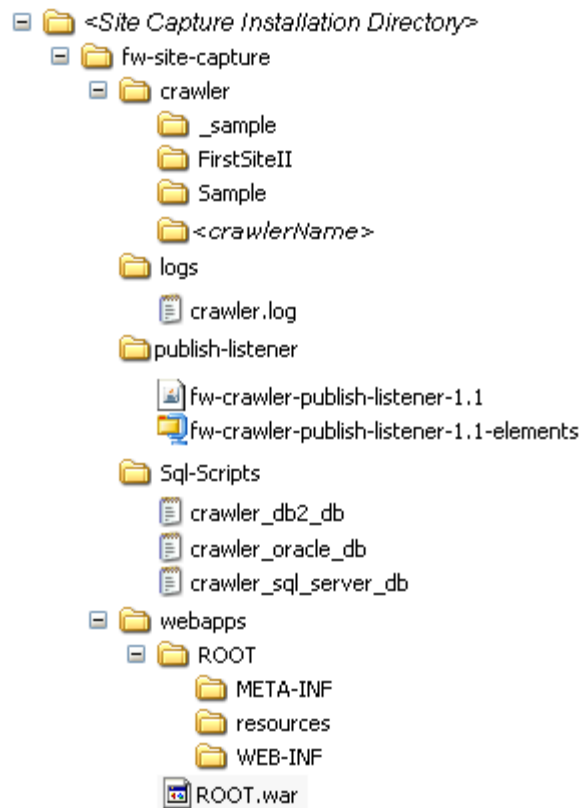


Table 49–1 Site Capture's Frequently Accessed Folders

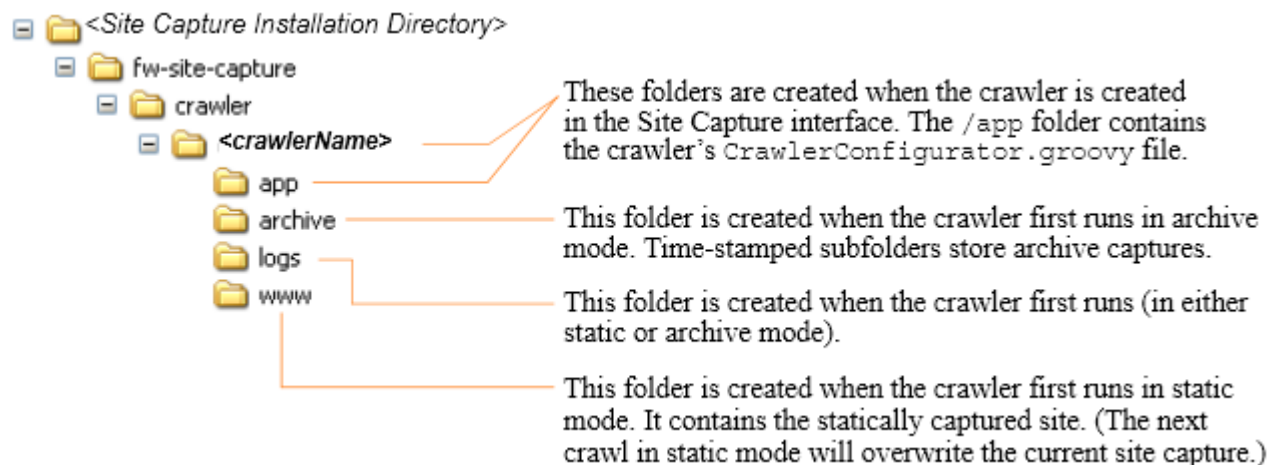
Folder	Description
/fw-site-capture	The parent folder.
/fw-site-capture/crawler	Contains all Site Capture crawlers, each stored in its own crawler-specific folder.
/fw/site-capture/crawler/_sample	Contains the source code for the FirstSiteII sample crawler. Note: Folder names beginning with the underscore character ("_") are not treated as crawlers. They are not displayed in the Site Capture interface.
/fw-site-capture/crawler/Sample	Represents a crawler named "Sample." This folder is created only if the "Sample" crawler was installed during the Site Capture installation process. The Sample folder contains an /app folder, which stores the CrawlerConfiguration.groovy file specific to the "Sample" crawler. The file contains basic configuration code for capturing any dynamic site. The code demonstrates the use of required methods (such as getStartUri) in the BaseConfigurator class. When the Sample crawler is invoked in static or archive mode, subfolders are created within the /Sample folder.
/fw-site-capture/logs	Contains the crawler.log file, a system log for Site Capture.
/fw-site-capture/publish-listener	Contains the following files needed for installing Site Capture for publishing-triggered crawls: <ul style="list-style-type: none"> ■ fw-crawler-publish-listener-1.1-elements.zip ■ fw-crawler-publish-listener-1.1.jar

Table 49–1 (Cont.) Site Capture's Frequently Accessed Folders

Folder	Description
/fw-site-capture/Sql-Scripts	Contains the following scripts, which create database tables that are needed by Site Capture to store its data: <ul style="list-style-type: none"> crawler_db2_db.sql crawler_oracle_db.sql crawler_sql_server_db.sql
/fw-site-capture/webapps	Contains the ROOT/WEB-INF/ folder.
/fw-site-capture/webapps/ROOT/WEB-INF	Contains the log4j.xml file, used to customize the path to the crawler.log file.
/fw-site-capture/webapps/ROOT/WEB-INF/classes	Contains the following files: <ul style="list-style-type: none"> sitecapture.properties file, where you can specify information for the WebCenter Sites application on which Site Capture is running. The information includes WebCenter Sites' host machine name (or IP address) and port number. root-context.xml file, where you can configure the Site Capture database.

49.2 Custom Folders

A custom folder is created for every crawler that a user creates in the Site Capture interface. The custom folder, `<crawlerName>`, is used to organize the crawler's configuration file, captures, and logs, as summarized in [Figure 49–2](#).

Figure 49–2 Site Capture's Custom Folders: `<crawlerName>`

Part VII

Oracle WebCenter Content Integration

This part contains procedures for enabling and configuring Oracle WebCenter Content Integration (WCC Integration) to support the synchronization of content from Oracle WebCenter Content to Oracle WebCenter Sites.

This part contains the following chapters:

- [Chapter 50, "About WebCenter Content Integration"](#)
- [Chapter 51, "Enabling the WebCenter Content Integration Feature"](#)
- [Chapter 52, "Setting Up Content Synchronization"](#)
- [Chapter 53, "Resynchronization Options"](#)
- [Chapter 54, "WCC Connector Configuration Files"](#)

About WebCenter Content Integration

The Oracle WebCenter Content Integration feature, also called WCC Integration, is a GUI-controlled connector. Its function is to synchronize content items from WebCenter Content to WebCenter Sites, where they are managed as native assets. The assets can be edited, deleted, previewed, revision tracked, published, and otherwise treated like any asset that was created directly in WebCenter Sites.

This chapter first outlines your prerequisites for working with WCC Integration. It then discusses basic concepts to help you plan, set up, and manage content synchronization.

This chapter contains the following sections:

- [Section 50.1, "Administrator's Prerequisites"](#)
- [Section 50.2, "Basics of WebCenter Content Integration"](#)
- [Section 50.3, "WebCenter Content Data Model"](#)
- [Section 50.4, "Asset Type Mapping and Connector Rules"](#)
- [Section 50.5, "WebCenter Content Integration and Its Parts"](#)
- [Section 50.6, "First Steps"](#)

50.1 Administrator's Prerequisites

WCC Integration is for administrators with a working knowledge of WebCenter Content and WebCenter Sites, especially their data models. Requirements include a clear understanding of content items, renditions, conversions, and metadata defined in WebCenter Content. Also required is a clear picture of WebCenter Sites asset types, the flex family in particular. A critical portion of your work involves mapping content item metadata to asset type attributes. If the need arises to create the asset types, you would collaborate with developers.

Detailed information about WebCenter Content is available in *Oracle Fusion Middleware Using Oracle WebCenter Content*.

50.2 Basics of WebCenter Content Integration

WCC Integration leverages WebCenter Content and the conversion capabilities of its content transformation engines. Specifically, WCC Integration is designed to synchronize two types of content from WebCenter Content to WebCenter Sites – documents and digital assets, including images and videos – as native files, renditions, and HTML conversions. Renditions can be created by the core transformation engine or Digital Asset Manager, depending on the type of native file and which renditions

are required. HTML conversions are generated by Dynamic Converter. File formats are unlimited. More information about content transformation is available in ... *Oracle Fusion Middleware Using Oracle WebCenter Content*.

Before showing you how to enable and set up WCC Integration, we start with concepts that are most important to your success with WCC Integration. These concepts are used throughout the WCC Integration feature and the chapters in this guide. They include the WebCenter Content data model, its mapping to asset types in WebCenter Sites, and the rules you will be configuring to guide the connector when it polls WebCenter Content. The concepts are discussed in [Section 50.3, "WebCenter Content Data Model"](#) and [Section 50.4, "Asset Type Mapping and Connector Rules."](#)

50.3 WebCenter Content Data Model

The WebCenter Content data model is based on an object called *content item*, which holds several types of information: a native file, its renditions, conversions, and the associated metadata.

- Native files are created by users directly in WebCenter Content and kept in their original format. For WCC Integration, valid native files are documents (such as docx files) and digital assets (such as image files and videos).
- Renditions are converted content items. Renditions are different formats of native content. They are files generated by content transformation engines and associated with the content item that stores the native (original) file.

For example:

- Native documents can be converted by the core transformation engine (available by default in WebCenter Content) into web formats for easier viewing on the web. Typically, this would mean converting a Microsoft Word document into a PDF file. The native document is known as the primary rendition, and its web-viewable copy is known as the web rendition. The native file and its rendition are contained within the same content item.
- Native image files and videos can be converted by Digital Asset Manager into multiple renditions of different sizes and formats. The original image file is the primary rendition, its small thumbnail version could be called small thumbnail rendition, its large thumbnail version could be called large thumbnail rendition, and so on.
- HTML conversions are specific renditions of native documents (such as Microsoft Word). The renditions are created by Dynamic Converter, which takes a single native document containing both text and graphics and converts it into an HTML rendition. The rendition consists of (1) an HTML file containing all the text of the native document, and (2) a collection of image files representing the graphics.
- Metadata associated with content items is analogous to attributes associated with asset types.

When it runs for the first time, the WCC connector imports the latest released versions of native files, renditions, conversions, metadata, or combinations of them into WebCenter Sites. Which content it imports depends on the connector rules and the mappings they specify.

If the content items are then modified on WebCenter Content, the connector synchronizes them to WebCenter Sites in its next session. Updated content items are re-imported into WebCenter Sites, and deleted content items have their counterpart assets deleted from WebCenter Sites (unless dependencies on those assets were created since the last synchronization session).

Import and synchronization can be triggered manually, or they can be scheduled to run on a timed basis.

Synchronization is always unidirectional, from WebCenter Content to WebCenter Sites. WCC Integration supports a many-to-one client-server model. Any number of WebCenter Sites clients can be connected to a single WebCenter Content instance.

For more information about the WebCenter Content data model, see *Oracle Fusion Middleware Understanding Oracle WebCenter Content* and the *Oracle WebCenter Content Administrator's Guide for Dynamic Converter*.

50.4 Asset Type Mapping and Connector Rules

In WebCenter Content, managed items are referred to as content items, while in WebCenter Sites the managed items are referred to as assets.

- Content items in WebCenter Content have metadata associated with them.
- Assets in WebCenter Sites are instances of asset types, which have attributes associated with them.

The goal of using WCC Integration is to import content items as assets into WebCenter Sites and keep them synchronized as the content items are updated. Your success depends on many factors, but rests largely on the target asset type (or types), especially on the mapping of content item metadata to asset type attributes, how you configure the mapping in rules the connector uses throughout its polling sessions, and how you prioritize the rules.

This section contains the following topics:

- [Section 50.4.1, "Asset Type Mapping"](#)
- [Section 50.4.2, "Connector Rules"](#)

50.4.1 Asset Type Mapping

Target asset types can be created or re-used. How many and which asset types to use depends on your content requirements. There is, however, a minimum set of requirements:

- Target asset types must belong to the flex family.
- A target asset type must have at least the following types of attributes:
 - A content identifier attribute that maps to the content identifier metadata field used in WebCenter Content. For more information, see [Section 50.4.1.1, "Content Identifier Attribute."](#)
 - Content storage attributes to hold the imported files (native files, renditions, conversions, or any combination of them). For more information, see [Section 50.4.1.2, "Content Storage Attributes."](#)
 - A template storage attribute to enable asset previewing, assuming asset previewing is a requirement (which is typically the case).

If a template attribute is created, special attention must be given to template code for previewing dynamically converted HTML files and their referenced images, as described in [Section 50.4.1.3, "Template Attributes."](#)

- The rest of the requirements are specific to your needs and outlined in [Section 50.4.1.4, "Administrator-Defined Specifications."](#)

50.4.1.1 Content Identifier Attribute

In WebCenter Content, each item has a unique ID, which is stored in a database field named `dDocName` (and displayed in the item's **Content ID** field). By default, `dDocName` is mapped to the `name` attribute. Whenever an asset is created in WebCenter Sites to store a content item, the value of `dDocName` for that content item is recorded in the asset's `name` attribute.

The `dDocName` (**Content ID**) is used in several ways. It enables the WCC connector (and you) to trace assets to their counterpart content items. It enables the connector to determine if a content item has already been synchronized to WebCenter Sites and, therefore, to either create a new asset, update the existing asset, or delete the existing asset, depending on whether the content item is created, updated, or deleted.

The default mapping of `dDocName` to `name` can be changed, as described in [Section 54.2, "User Configurable Parameters."](#) This should be done only once, on a new system, before connector rules are created and the first synchronization session is started.

50.4.1.2 Content Storage Attributes

Given that content items are imported into WebCenter Sites as files, the target asset type requires attributes of type `blob` (either single- or multi-valued) to accept the files.

Dynamically converted documents require some special handling, because the process of converting a single native file into HTML generates multiple artifacts. For example, a single Microsoft Word document containing both text and graphics gets converted to (1) an HTML file containing all the text, and (2) a collection of image files representing all the graphics.

The HTML and image files are mapped to separate attributes. The HTML file is mapped to a single-valued attribute of type `blob`. The image files are mapped to a multi-valued attribute of type `blob`. A template attribute and template for previewing the asset must also be configured, as described in [Section 50.4.1.3, "Template Attributes."](#)

50.4.1.3 Template Attributes

Typically, asset types in WebCenter Sites have template attributes to support asset previewing.

Templates Requiring Special Attention

Templates for dynamically converted HTML files require special attention for the following reason:

A dynamically converted HTML file contains image links in WebCenter Content format. For example:

```

```

Because WebCenter Sites does not recognize a format such as `{wcmUrl('dresource', '20673.jpg')}`, the links must be converted to `blob` links. Conversion of link format is illustrated in a sample template named `WccConversionPreview.jsp`, located in the `/misc/Samples/WCC` directory of the exploded WebCenter Sites distribution file. Developers can use the sample template as is, or include parts of its logic in their own templates. The sample template is hard-coded with the names of the file-storage attributes shown in [Table 50-1](#):

Table 50–1 Attributes in WccConversionPreview.jsp

Attribute Name	Data Type
html	blob
images	blob (M)

If the attributes in your target asset type are named other than `html` and `images`, developers will have to update the sample `WccConversionPreview.jsp` template with the attribute names used in the target asset type.

50.4.1.4 Administrator-Defined Specifications

In addition to the required attributes described in sections [Section 50.4.1.1, "Content Identifier Attribute,"](#) [Section 50.4.1.2, "Content Storage Attributes,"](#) and [Section 50.4.1.3, "Template Attributes,"](#) you always have the option to add attributes for any metadata that you need to synchronize to a target asset type.

Target asset types can be enabled on as many content management sites as necessary. During a synchronization session, content items will be synchronized to the target asset types on all sites that you specify in the connector rules. For information about connector rules, see [Section 50.4.2, "Connector Rules."](#)

50.4.2 Connector Rules

The WCC connector requires at least one active (enabled) rule. Without an active rule, the connector cannot poll WebCenter Content.

An *active* rule is an enabled set of instructions for the connector to execute when it runs. A rule is configured and enabled on WebCenter Sites. It contains several instructions: Which metadata to poll on WebCenter Content, which target asset type to invoke on which content management site(s) where the target asset type is enabled, and how to map content item metadata to asset type attributes. The connector uses the map to determine in which attributes to store the information it retrieves.

When planning and configuring connector rules, consider the information presented in the rest of this section to help ensure that content items are processed to your expectations:

- [Rules and Target Asset Types](#)
- [Prioritizing Active Rules](#)
- [Resynchronizing Content Items to WebCenter Sites](#)
- [Disabling and Deleting Active Rules](#)
- [Additional Considerations](#)

Rules and Target Asset Types

A connector rule allows you to choose only a single target asset type, which means that you are likely to create multiple rules if you plan to synchronize a variety of content items to WebCenter Sites. For example:

- You need to import a dynamically converted HTML file and its referenced images into a single asset type. You would then create a single rule that names the target asset type and maps the HTML file with its images to an `html` attribute and an `images` attribute.

- You need to import Microsoft Word documents and unrelated pdf files. You have two dedicated asset types: one asset type for docx files, and the other asset type for pdf files. You would then create two rules: one rule for synchronizing the docx files to the first asset type, and another rule for synchronizing pdf files to the second asset type.

Prioritizing Active Rules

If multiple active rules map the same content items to different asset types, only one of the rules will be executed by the connector – the first rule that matches the content items. The order in which rules are executed determines which content items are synchronized to which asset types.

For example, you have new items named `surfing.jpg`, `surfing.png`, and `skiing.png`. You also have two active rules:

- Rule 1 maps `surfing.any_extension` items to the `Images_avi` asset type on the `avisports` site.
- Rule 2 maps `png` items to the `Images_FSII` asset type on `FirstSite II`.

Table 50–2 shows that the items are distributed differently between the two asset types when rule 1 and rule 2 are executed in different orders. Therefore, you must prioritize the rules in a way that produces the results you require.

Table 50–2 How the Order of Active Connector Rules Determines Content Distribution

Active Rule	Images_avi Target Asset Type	Images_FSII Asset Type
If rule 1 is executed first	<code>surfing.jpg</code> and <code>surfing.png</code> are imported into this asset type	<code>skiing.png</code> is imported into this asset type
If rule 2 is executed first	<code>surfing.jpg</code> is imported into this asset type	<code>surfing.png</code> and <code>skiing.png</code> are imported into this asset type

Note: Here is more information about how the connector uses rules to synchronize content items to WebCenter Sites.

1. When the connector runs, it examines new and updated content items, but only if they are (or once were) enabled for synchronization. The connector examines the content items one at a time, as follows:

The connector compares the metadata of a content item to the metadata in the rule. If the connector fails to find matching metadata, it uses the next active rule to re-test the same content item.

If it finds matching metadata, the connector continues to test the content item by looking for renditions that match renditions specified in the rule. Once its search is complete, the connector turns to WebCenter Sites (regardless of whether it finds matching renditions).

2. On WebCenter Sites, the connector operates on the database:

The connector invokes the target asset type named in the rule, creates an asset, uses the rule to determine which metadata maps to which attributes, and imports the metadata values and renditions (if any) into the asset's attributes.

Once the imported content item is synchronized, the connector skips testing all other rules on that content item and moves on to test the next content item, starting with step 1.

Resynchronizing Content Items to WebCenter Sites

Once synchronized to WebCenter Sites, a content item will be resynchronized only when it is modified (updated or deleted) on WebCenter Content.

Note: If a content item is synchronized to WebCenter Sites and then deleted from WebCenter Content, its counterpart asset is deleted from WebCenter Sites, unless dependencies were created on that asset since the last synchronization session.

Disabling and Deleting Active Rules

If an active rule is disabled or deleted, content items imported into WebCenter Sites by that rule remain in WebCenter Sites. Disabled rules are ignored by the connector.

Additional Considerations

Creating and managing rules is only one part of setting up content synchronization. Your complete set of steps includes first enabling several WCC Integration parts (a one-time operation) and then configuring the parts to support content synchronization. The parts are described in [Section 50.5, "WebCenter Content Integration and Its Parts."](#)

50.5 WebCenter Content Integration and Its Parts

The WCC Integration feature consists of several parts, which you will first enable and then configure to set up and manage content synchronization. The parts are the SitesIntegration component, the WCC connector, and the Connector Admin tab, which is used to control the connector.

- **SitesIntegration:** This component is installed by default on WebCenter Content during the installation process and must be enabled to support the WCC connector.

Enabling the SitesIntegration component adds the following features to WebCenter Content:

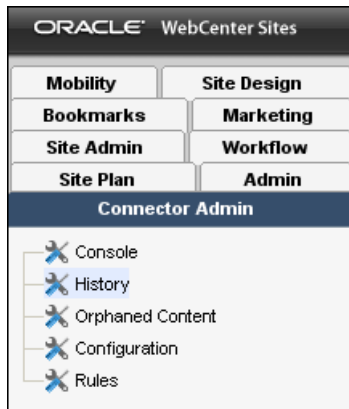
- **Sync to Sites** option. This option is displayed on content checkin, update, and search forms. Setting the option to `True` makes the content item available to the connector for retrieval. Whether the content is actually retrieved and stored depends on whether that content matches the criteria in the active rule(s) you create.
- **WCS Queue Support Information.** This graphical interface lists token history for each of the WebCenter Sites clients. Token information can be used to roll back to previous updates.

For information about enabling the SitesIntegration component, see [Section 51.1, "Enabling the SitesIntegration Component."](#)

- **WCC Connector:** This connector is installed on WebCenter Sites to communicate with WebCenter Content. It can be selected for installation during the WebCenter Sites installation or upgrade process, or it can be installed at a later time (the former method is preferred). The connector polls WebCenter Content once it is configured to do so from the **Connector Admin** tab in the WebCenter Sites Admin interface.
- **Connector Admin Tab:** This is a graphical interface used to control the connector. When WCC Integration is installed, the **Connector Admin** tab is added to the

navigation tree of the WebCenter Sites Admin interface, as shown in [Figure 50–1](#).

Figure 50–1 WCC Connector Admin Tab



The **Connector Admin** tab is used to set up, run, and manage the connector. For example, you can create and enable connector rules, establish a connection to WebCenter Content, manually run the connector, monitor connector activity, view connector history, and obtain token information.

For information about enabling the Connector Admin tab, see [Section 51.2, "Enabling Access to the Connector Admin Tab."](#)

50.6 First Steps

Before any content can be imported into WebCenter Sites, two WCC Integration parts must be enabled to support content synchronization, as described in [Chapter 51, "Enabling the WebCenter Content Integration Feature."](#) Your first (and one-time) step is to complete (or ensure the completion of) all procedures in [Chapter 51, "Enabling the WebCenter Content Integration Feature."](#) They are:

1. Enable the SitesIntegration component on WebCenter Content, as described in [Section 51.1, "Enabling the SitesIntegration Component."](#)
2. Enable the Connector Admin tab on WebCenter Sites for yourself and other users, as described in [Section 51.2, "Enabling Access to the Connector Admin Tab."](#)

Once the above procedures are completed, you are ready to set up your system for content synchronization sessions, as described in [Chapter 52, "Setting Up Content Synchronization."](#)

Enabling the WebCenter Content Integration Feature

Before you can start configuring your system for content synchronization, you need to enable two WCC Integration parts: the Sites Integration component and the Connector Admin tab. Each of your steps is described in its own section.

This chapter contains the following sections:

- [Section 51.1, "Enabling the SitesIntegration Component"](#)
- [Section 51.2, "Enabling Access to the Connector Admin Tab"](#)
- [Section 51.3, "Next Steps"](#)

51.1 Enabling the SitesIntegration Component

The SitesIntegration component is installed during the full Oracle WebCenter Content installation or upgrade process. Once the SitesIntegration component is enabled, the following features are added to WebCenter Content:

- The `SiSyncToSites` metadata field is added to WebCenter Content schema and displayed as the **Sync To Sites** option in content checkin, update, and search forms.
- The WCS Queue Support Information page is added to provide a listing of WebCenter Sites clients and their token histories. Once synchronization begins, the same page logs the synchronized content items and their update histories.

To enable the SitesIntegration component

1. On WebCenter Content, open the Component Manager:
 - a. Click **Administration** on the portal navigation bar for your Content instance.
 - b. Click **Admin Server**.
2. On the Component Manager page:
 - a. Scroll down to the Integration section and select the **SitesIntegration** check box.
 - b. Click **Update**.

Note: If you need more information about enabling components, refer to *Oracle Fusion Middleware Installing and Configuring Oracle WebCenter Content* or the online Help.

3. Restart WebCenter Content.
4. Open any content checkin, update, or search form to verify that the **Sync To Sites** check box is displayed in the form, as shown in [Figure 51-1](#).

Figure 51-1 Sync To Sites Option on a Content Item Update Form

The screenshot shows the Oracle Content Server interface for updating a content item. The page title is 'Info Update Form for 'DGT''. The form contains the following fields:

- Content ID: DGT
- * Type: Document - Any generic document (dropdown)
- * Title: DGT (text input)
- * Author: weblogic (text input) | weblogic (dropdown)
- * Security Group: Public (dropdown)
- * Revision: 1 (text input)
- Comments: updated comment (text area)
- Profile: (dropdown)
- Template Type: (dropdown)
- Sync To Sites: True (dropdown)

5. Verify the display of the WCS Queue Support Information page:
 - a. Log in to WebCenter Content as an administrator.
 - b. Select **Administration** and then **WCS Administration** to open the WCS Queue Support Information page ([Figure 51-2](#)).

Figure 51-2 WCS Queue Support Information Page on WebCenter Content Server

The screenshot shows the 'WCS Queue Support Information' page. It contains two tables:

WCS Queue

Token	QueueType	QueueId	Time	Sync	Message
* Indicates previously marked Sync To Sites					

WCS Recent Queue Queries

ClientId	AccessCount	LastAccessTime	RequestToken	SentToken	Message
----------	-------------	----------------	--------------	-----------	---------

6. Continue to [Section 51.2, "Enabling Access to the Connector Admin Tab."](#)

51.2 Enabling Access to the Connector Admin Tab

Anyone who needs to use WCC Integration must be assigned the ConnectorAdmin role. The only exception is noted below.

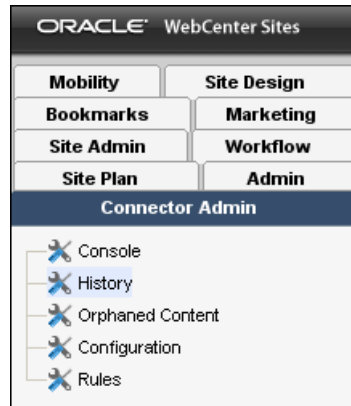
Note: The ConnectorAdmin role is automatically assigned to only one user, that is the general administrator whose credentials were used during the WebCenter Sites installation process. All other users, including general administrators, must have the role assigned to them manually.

The ConnectorAdmin role grants access to the **Connector Admin** tab, enabling its user (typically an administrator) to control the connector by performing such operations as creating and managing its rules, establishing communication with WebCenter Content, running the connector, monitoring its activity, and viewing its history.

To enable access to the Connector Admin tab

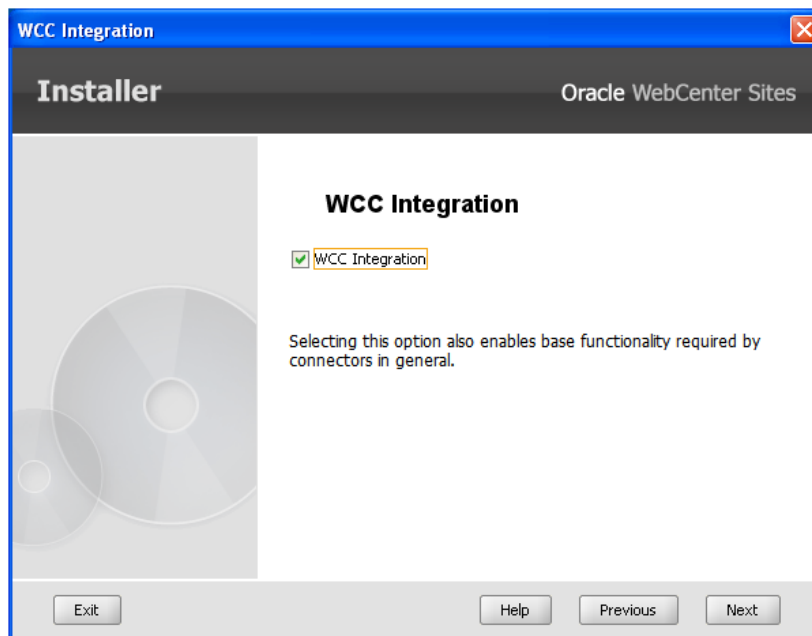
1. Open the WebCenter Sites Admin interface on the content management site you intend to work with, and look for the **Connector Admin** tab (shown in [Figure 51-3](#)).

Figure 51-3 Connector Admin Tab



2. If the **Connector Admin** tab is not displayed, assign yourself (and other intended users) the ConnectorAdmin role. For procedures on assigning roles, see [Chapter 4, "Working with ACLs and Roles."](#)
3. Log out of the Admin interface, and log back in to the same content management site as in step 1 of this procedure.
4. If the **Connector Admin** tab is still not displayed, WCC Integration was not installed during the WebCenter Sites installation process. To install WCC Integration, do the following:
 - a. Back up all customizations for your current installation and re-run the WebCenter Sites installer with the **WCC Integration** option enabled on the WCC Integration dialog (shown in [Figure 51-4](#)). Instructions for running the installer are available in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*.

Figure 51–4 WebCenter Sites Installer: WCC Integration Step



- b. When the installer completes its process, restore your customizations, and repeat the steps in this section to enable your access to the **Connector Admin** tab.

51.3 Next Steps

Having successfully completed the steps in this chapter, you are now ready to set up your system for content synchronization. Continue to [Chapter 52, "Setting Up Content Synchronization."](#)

Setting Up Content Synchronization

This chapter contains procedures for configuring your system to import content items from WebCenter Content into WebCenter Sites, and to keep the two systems synchronized as the content items are updated. Once the required items are imported as assets, you can work with content contributors to help them identify and work with the assets. Each of your steps is described in its own section.

Note: Before starting the steps in this chapter:

- Read [Chapter 50, "About WebCenter Content Integration"](#) for information about the WebCenter Content data model, as well as the asset types and connector rules you will be configuring.
 - Ensure that all procedures in [Chapter 51, "Enabling the WebCenter Content Integration Feature"](#) have been completed.
 - Read this chapter to gain an understanding of the steps you will be following. Advance knowledge will help you plan.
 - If possible, set up a test WebCenter Sites system on which to configure and test content synchronization. Verify the results, make adjustments as necessary, and reproduce your final configurations on your actual system.
-
-

This chapter contains the following sections:

- [Section 52.1, "Step 1: Enabling Content Items for Synchronization"](#)
- [Section 52.2, "Step 2: Determining Which Renditions to Synchronize"](#)
- [Section 52.3, "Step 3: Selecting or Creating Target Asset Types"](#)
- [Section 52.4, "Step 4: Connecting to WebCenter Content"](#)
- [Section 52.5, "Step 5: Creating an Active Rule"](#)
- [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules"](#)
- [Section 52.7, "Step 7: Manually Running the Connector"](#)
- [Section 52.8, "Step 8: Monitoring the Connector"](#)
- [Section 52.9, "Step 9: Viewing Connector History"](#)
- [Section 52.10, "Step 10: Verifying Imported Content"](#)
- [Section 52.11, "Step 11: If You Need to Modify Your Setup"](#)
- [Section 52.12, "Step 12: Scheduling the Connector"](#)
- [Section 52.13, "Step 13: Managing Synchronized Assets"](#)

52.1 Step 1: Enabling Content Items for Synchronization

Content items are typically enabled for synchronization by WebCenter Content users. They set each item's **Sync to Sites** option to `True`, as shown in [Figure 52–1](#). Valid content items are documents and digital assets.

Figure 52–1 Sync To Sites Option on the WebCenter Content Server Update Form

The screenshot shows the Oracle Content Server interface for updating a content item. The form is titled "Info Update Form for 'DGT'". It contains several fields:

- Content ID:** DGT
- * Type:** Document - Any generic document (dropdown)
- * Title:** DGT (text input)
- * Author:** weblogic (text input) and weblogic (dropdown)
- * Security Group:** Public (dropdown)
- * Revision:** 1 (text input)
- Comments:** updated comment (text area)
- Profile:** (dropdown)
- Template Type:** (dropdown)
- Sync To Sites:** True (dropdown)

Synchronization-enabled content items are available to the WCC connector for processing. However, they can be imported into WebCenter Sites only when the WCC connector is enabled to import them into the correct asset types. You will enable the connector in the next few steps. Continue to [Section 52.2, "Step 2: Determining Which Renditions to Synchronize."](#)

52.2 Step 2: Determining Which Renditions to Synchronize

Content items enabled for synchronization in [Section 52.1, "Step 1: Enabling Content Items for Synchronization"](#) have renditions. Default rendition types are `primary` and `web`:

- A `primary` rendition is the native (original) file that was checked in to WebCenter Content (for example, a Microsoft Word document, before any renditions were created).
- The `web` rendition of a content item is a web-viewable version of a native file. For example, a PDF version of a Microsoft Word document is a web rendition of the Word document.

Custom renditions may also be available. To determine which renditions to import into WebCenter Sites, consult with site designers who work with WebCenter Sites. You will specify the renditions for the connector in a later step (in [Section 52.4, "Step 4: Connecting to WebCenter Content"](#)).

Continue to [Section 52.3, "Step 3: Selecting or Creating Target Asset Types."](#)

52.3 Step 3: Selecting or Creating Target Asset Types

Complete the following steps:

- Ensure that your target asset types meet synchronization requirements:
 - Your asset types meet the minimum requirements outlined in [Section 50.4.1, "Asset Type Mapping."](#) For example, they must belong to the flex family.
 - Your asset types are configured to accept the content items that were enabled in [Section 52.1, "Step 1: Enabling Content Items for Synchronization."](#)
- If you need to change the default mapping of the `dDocName` metadata field (to an attribute other than `name`), do so now by resetting the `wcc.keyfield` parameter in the `wcc-integration.ini` file. Once the parameter is reset and synchronization has started, its value should not be edited again.

For information about `dDocName`, see [Section 50.4.1.1, "Content Identifier Attribute."](#) For information about the `wcc.keyfield` parameter, see [Section 54.2, "User Configurable Parameters."](#)

- If your asset types have a template attribute, copy the display names of the templates exactly as they appear in the WebCenter Sites Admin interface and paste them into a text file. You will provide the template names when configuring connector rules, in [Section 52.4, "Step 4: Connecting to WebCenter Content."](#)

52.4 Step 4: Connecting to WebCenter Content

In this section, you will connect WebCenter Sites to WebCenter Content and specify which renditions (determined in [Section 52.2, "Step 2: Determining Which Renditions to Synchronize"](#)) must be available for use in the attribute mapping part of the connector rules.

To connect to WebCenter Content

1. From the **Connector Admin** tab, select **Configuration** to open the Connector Configuration form ([Figure 52–2](#)).

Figure 52–2 Connector Configuration Form with Example Values

The screenshot shows a web form titled "Connector Configuration". It contains the following fields and controls:

- * Server URL:** A text input field containing "www.example.com:16200/cs/ldcplg".
- * Server User:** A text input field containing "weblogic".
- Server Password:** A text input field containing "*****".
- * Rendition Names:** A section with a "+ Add New Rendition" link and two text input fields. The first field contains "primary" and the second contains "web". Each field has a small "X" icon to its right.

At the bottom of the form are two buttons: "Test" and "Save".

2. In the Connector Configuration form, fill in the following fields:

- a. In the **Server URL** field, provide the address of the WebCenter Content instance to connect to. The HTTP, HTTPS, or IDC protocols can be used. For more information on determining the URL path of a WebCenter Content instance, see *Oracle Fusion Middleware Administering Oracle WebCenter Content*.
- b. In the **Server User** field, provide the name of a user with administration privileges for the WebCenter Content instance.

Note: If you specified the IDC protocol in the **Server URL** field, the user name must be defined as that of a local user with administrative privileges (by default, **sysadmin**).

- c. In the **Server Password** field, provide a password for the administrative user only if you specified an HTTP or HTTPS connection in the **Server URL** field.

Note: For IDC connections, a server password is not required. The IDC protocol uses a trusted port configured on the WebCenter Content server.

- d. In the **Rendition Names** field, add the names of all renditions you plan to import (as determined in [Section 52.2, "Step 2: Determining Which Renditions to Synchronize"](#)).

To add a rendition, click **Add New Rendition** and enter its name. If you are adding a video rendition, use a colon-separated format such as `media:rendition-name` (see [Figure 52–3](#)). For example, `media:flash8`.

Figure 52–3 Example Video Rendition

* Rendition Names: + Add New Rendition

media:rendition-name	X
primary	X
web	X

The rendition types `primary` and `web` are provided by default. They can be kept or deleted, depending on your requirements.

3. Click the **Test** button to test your entries. If your entries are valid, a confirmation message is displayed.
4. Click the **Save** button to save the entries. A confirmation message is displayed when the configuration has been saved successfully.
5. Continue to [Section 52.5, "Step 5: Creating an Active Rule."](#)

52.5 Step 5: Creating an Active Rule

An active rule is a set of instructions enabled for the connector to follow when it runs. The connector will use active rules to do the following:

1. Sample the content items that were enabled in [Section 52.1, "Step 1: Enabling Content Items for Synchronization,"](#) and retrieve items that match your active rules.
2. Create receiving assets in the target asset types that you selected or created in [Section 52.3, "Step 3: Selecting or Creating Target Asset Types"](#) and named in your active rules.
3. Synchronize the renditions that you specified in [Section 52.4, "Step 4: Connecting to WebCenter Content"](#) and in your active rules.

Note: Before creating rules, read [Section 50.4.2, "Connector Rules"](#) for planning and configuration guidelines.

You can enable rules when you create them (in this step), or in a later step (in [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules"](#)).

To Create a Rule

1. From the **Connector Admin** tab, select **Rules**.
2. In the Connector Rules form, click **Add New Rule** ([Figure 52-4](#)).

Figure 52-4 Connector Rules Form

The screenshot shows a rectangular box titled "Connector Rules". Below the title is a dashed horizontal line. At the bottom left of the box is a dark button with the text "Add New Rule" in white.

3. The rules wizard opens ([Figure 52-5](#)). Perform each step in the rules wizard, as described below.

Rules Wizard: Step 1 Name

Use this step to name your rule, add a description for this rule, and enable your rule.

Figure 52-5 Step 1: Name Page with Example Values

The screenshot shows a wizard interface with a progress bar at the top. The progress bar has four steps: "STEP", "1 Name", "2 Rule", "3 Target", and "4 Attributes". The "1 Name" step is highlighted. Below the progress bar, the text "Name : image-rule" is displayed. A dashed horizontal line follows. Below that, there are three form fields:

- * Name: A text input field containing "image-rule".
- * Description: A text input field containing "Move all images to FSII".
- Enabled: A checkbox that is currently unchecked.

1. Enter a name for this rule (required). Only alphanumeric characters and hyphens are allowed.
2. Enter a description for this rule (required).

3. Select the **Enabled** option to enable this rule.

Note: If the **Enabled** option is not selected, the rule is not activated and the connector will not run the rule. The rule can be enabled or disabled at any time from the current wizard or the Connector Rules form, described in [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules."](#)

4. Continue to **Step 2: Rule** of the rules wizard.

Rules Wizard: Step 2 Rule

Use this step to set up a rule statement specifying which metadata the connector must sample when polling the content items you enabled in [Section 52.1, "Step 1: Enabling Content Items for Synchronization."](#)

In this step of the rules wizard, you will select a metadata name, choose an operator, and then specify the metadata value. You can specify many metadata names and values by adding AND/OR statements.

Figure 52–6 Step 2 Rule Page with Example Values

The screenshot shows the 'Step 2 Rule' page of the Rules Wizard. The page has a progress bar at the top with four steps: 1 Name, 2 Rule (active), 3 Target, and 4 Attributes. Below the progress bar, the rule name is 'image-rule'. The rule statement is defined by three conditions:

- Condition 1: **dDocName** (selected from a dropdown) **contains** (selected from a dropdown) **wcc** (text input).
- Condition 2: **dDocType** (selected from a dropdown) **matches** (selected from a dropdown) **image** (text input).
- Condition 3: **dExtension** (selected from a dropdown) **ends with** (selected from a dropdown) **jpg** (text input).

The conditions are connected by **AND** and **OR** operators. There are buttons to add 'OR' and 'AND' statements.

To specify which metadata the connector must sample

1. In the left-hand drop-down list, select a metadata name.

The list displays all metadata names defined in WebCenter Content. For descriptions of the metadata, refer to the *Oracle Fusion Middleware Developing with Oracle WebCenter Content*.

Common metadata names include, for example:

- **dDocName:** The Content ID for the content item on WebCenter Content.

- **dDocType:** The content item type on WebCenter Content.
 - **dDocTitle:** The content item title on WebCenter Content.
 - **dExtension:** The file extension (such as `txt`, `doc`, `pdf`, and `jpg`) on WebCenter Content.
 - **xComments:** The comment field on WebCenter Content.
2. In the middle drop-down list, select an operator:
 - **matches:** The connector will search for metadata values that match the entire string you provide in the right-hand field. For example, if your statement is **dDocType matches image**, the connector will recognize `image`, but not `images`.
 - **contains:** The connector will perform a substring search on the metadata value you provide in the right-hand field. For example, if your statement is **dDocName contains abc**, the connector will recognize `abc`, `abcde`, and any other `dDocName` that contains `abc`.
 - **starts with:** The connector will search for metadata values that begin with the string you provide in the right-hand field.
 - **ends with:** The connector will search for metadata values that end with the string you provide in the right-hand field.
 - **is greater than:** Applies to integers only (not valid for strings).
 - **is less than:** Applies to integers only (not valid for strings).
 3. In the right-hand field, enter a value for your chosen metadata field.
 4. Add as many OR statements as desired.
 5. Add as many AND statements as desired.
 6. Continue to **Step 3 Target** of the rules wizard.

Rules Wizard: Step 3 Target

Use this step to select the target asset type, content management site(s) where the asset type is enabled, and any other information you may be prompted for.

Note: The content of this step depends on the asset type.

Figure 52–7 Step 3 Target Page with Example Values

STEP 1 Name 2 Rule 3 Target 4 Attributes

Name : image-rule

* Asset Type: Media_C

* Asset Subtype: FSII_Image

Parent(s): * FSII_ImageCategory
FSII Article Images

* Site(s): Site Name
 FirstSiteII

To specify the target asset type and the content management site where it is enabled

1. Select the target asset type from the drop-down list.
Depending on the asset type you selected, different options are displayed on the form.
2. Select the asset subtype.
3. Enable the desired parent(s) from the list.
4. The Site(s) drop-down list names the content management site(s) on which your selected asset type is enabled. Select one or more sites. The connector will import content items and metadata into the target asset type on all of your selected sites.
5. Provide any other information you may be prompted for.
6. Continue to **Step 4 Attributes** of the rules wizard.

Rules Wizard: Step 4 Attributes

Use this step to map the metadata of the synchronization-enabled content items (from [Section 52.1, "Step 1: Enabling Content Items for Synchronization"](#)) to the attributes of your target asset type.

The **Step 4. Attributes** page of the rules wizard ([Figure 52–8](#)) lists the attribute names and their data types in the asset type you selected in **Step 3. Target** of the rules wizard ([Figure 52–7](#)). Attributes marked with a red asterisk are required. Attributes whose data type contain (M) are multi-valued attributes. Such attributes can contain a single value or multiple values separated by a semicolon (;). For example: value1;value2;

Figure 52–8 Step 4 Attributes Page with Example Values

STEP **1** Name **2** Rule **3** Target **4** Attributes

Name : image-rule

Asset Type: Media_C
 Asset Subtype: FSII_Image
 Key Field: name

Attribute	Type	Take from	Value
description	string	Literal	WCC image for Ski Banner
fwtags	string (M)	Select	
template	string	Select	
*FSII_ImageFile	blob	Rendition	web
FSII_AltText	string	Literal	300
startdate	date	Literal	Apr 3, 2013 11:25:57 AM
enddate	date	Metadata	dReleaseDate

To map content item metadata to asset type attributes

For each attribute, go to the **Take From** column and select an option from the drop-down list. Options depend on the attribute type. The options are:

- **Literal or Metadata:** These options apply only to attributes of type other than blob.
- **Rendition or Conversion:** These options apply only to attributes of type blob.
- **Literal or Metadata:** These options apply only to attributes of type other than blob.
 - If you select **Literal**, enter a standard text string in the **Value** column. Your selected attribute will display the same text for all content items that you import with this rule.

Also note the following:

- * If your asset type has a template attribute, enter the template name in the **Value** column, exactly as it appears in the WebCenter Sites Admin interface. We recommend copying the name and pasting it into the **Value** column (you can use the name that you pasted into the text file in [Section 52.3, "Step 3: Selecting or Creating Target Asset Types"](#)).
- * For attributes of type date (for example, attributes named `startdate` or `enddate`), selecting **Literal** allows you to pick a date using the date-picker (see [Figure 52–9](#)).

Figure 52–9 Date-picker for Attribute of Type Date with a Literal Value



- If you select **Metadata** from the drop-down list in the **Take From** column, select the metadata name from the **Value** column.

Note: When you select **Metadata**, the drop-down list in the **Value** column lists metadata fields whose data type matches the data type of the attribute you are targeting. For example, if your attribute is of type date, the **Value** column lists only metadata fields of type date. For detailed descriptions of metadata types, refer to the *Oracle WebCenter Content Services Reference Guide*.

- **Rendition or Conversion:** These options apply only to attributes of type blob.
 - If you select **Rendition** from the drop-down list in the **Take From** column, select the desired rendition type from the **Value** column (for example, *web*, *primary*, or other rendition type).

Note: When you select **Rendition**, the drop-down list in the **Value** column names only the rendition types that you specified in the Connector Configuration form, in [Section 52.4, "Step 4: Connecting to WebCenter Content."](#)

- If you select **Conversion** from the drop-down list in the **Take From** column, select the desired conversion type from the **Value** column. Your only options are **html** and **images** (the **Conversion** option is strictly for HTML conversions).

For example, to import an HTML conversion, you can create a map such as the following (assuming the target attributes are named *HTML* and *Images*):

Table 52–1 Attribute Map Entries

Attribute	Type	Take from	Value
HTML	blob	Conversion	html
Images	blob (M)	Conversion	images

Rules Wizard: Final Steps

1. Click the **Save** icon in the toolbar to commit your rule to the WebCenter Sites database.

Note: If you did not select the **Enabled** option in "Rules Wizard: Step 1 Name," the rule is saved but not enabled. It can be enabled or disabled at any time from the Connector Rules form. See [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules."](#)

2. Create additional rules, if necessary.
3. Continue to [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules."](#)

52.6 Step 6: Prioritizing Active Rules and Managing Rules

One of the most important aspects of managing rules is to enable them for the connector and prioritize them in order of execution. The order can determine how the connector distributes content items among the target asset types in a synchronization session. For more information about prioritization, see [Section 50.4.2, "Connector Rules."](#)

Other operations, such as editing, deleting, and adding rules can also be completed in the same form for enabling and prioritizing rules.

To prioritize and otherwise manage rules

1. From the **Connector Admin** tab, select **Rules** to open the Connector Rules form.

Figure 52–10 Connector Rules Form with Example Rules

Connector Rules			
	Enabled	Name	Description
	<input checked="" type="checkbox"/>	FSII Manufacturer Logos	Description from WCC
	<input checked="" type="checkbox"/>	FSIIImageFiles	FirstSiteII image files
	<input type="checkbox"/>	image-rule	Move all images to avisports

2. If your immediate need is to prioritize rules for the connector, continue with this step (otherwise, skip to step 3 for descriptions of all the rule management options, such as editing and deleting, in the Connector Rules form):
 - a. Ensure that the rules you need to prioritize have their **Enabled** check box selected.
 - b. Click the **Move Up** or **Move Down** arrows to change the rule order. Enabled rules are executed by the connector in the arranged order. If a content item matches one active rule, the content item will be synchronized according to that one rule. All other rules for synchronizing that content item will be ignored.

- c. Click **Save**.
3. If you need to perform any of the possible operations listed on the Connector Rule form, see the list below for descriptions of your options:
 - Click the **Edit** icon (pencil) of a rule to edit that rule. The rules wizard is displayed. For information about configuring rules, see [Section 52.5, "Step 5: Creating an Active Rule."](#)
 - Click the **Delete** icon (trash can) of a rule to delete that rule.
 - Click the **Move Up** or **Move Down** arrows to change the rule order. Enabled rules are executed during connector polling in the arranged order. If a content item matches one active rule, it will be synchronized according to that one rule. All other rules for processing that content item are ignored.
 - Select or de-select the **Enabled** option of a rule to enable or disable that rule. Click **Save** after making any changes.

Note:

- If a rule is disabled, that rule and its matching content items will be ignored by the connector when it runs.
 - If an active rule is executed and then disabled (or deleted), content items imported according to that rule remain in WebCenter Sites.
-
-
- In the **Name** column, click the rule name to view or edit that rule. The rules wizard is displayed.
 - The **Description** column provides the rule description. This description is defined when you create a rule.
 - Click the **Add New Rule** button to add a new rule. The rules wizard is displayed.
 - Click the **Save** button to save any changes, such as enabling or disabling a rule.
 4. Once you have optimized, enabled, and prioritized your active rules, continue to [Section 52.7, "Step 7: Manually Running the Connector."](#)

52.7 Step 7: Manually Running the Connector

In this step, you will manually run the connector to test the rules you configured and prioritized in [Section 52.5, "Step 5: Creating an Active Rule"](#) and [Section 52.6, "Step 6: Prioritizing Active Rules and Managing Rules."](#)

To run the connector

1. From the **Connector Admin** tab, select **Console** to open the Connector Console.

Figure 52–11 Connector Console with Example Server Information

Connector Console

Server URL: www.example.com:16200/cs/ldcplg
 Server User: weblogic

Recent Activity

Timestamp	Launched By	State	Batch size	Updated	No Rule	Error
No record found						

2. Verify that the Connector Console displays the information you specified in [Section 52.4, "Step 4: Connecting to WebCenter Content"](#):
 - The **Server URL** field lists the address of the WebCenter Content instance.
 - The **Server User** field lists the user with administration privileges for the WebCenter Content instance.
3. Click **Run Now With Token**.

The connector starts running and the token field displays the current token value. For more information about tokens, see [Section 53.1, "Rolling Back the Synchronization Point."](#)

Note: The connector can be scheduled to run on a timed basis, as described in [Section 52.12, "Step 12: Scheduling the Connector."](#)

4. While the connector runs, you can monitor its activity from the same form you are now using. For more information, continue to [Section 52.8, "Step 8: Monitoring the Connector."](#)

52.8 Step 8: Monitoring the Connector

When the connector runs, the Connector Console (populated in the previous step) displays the Recent Activity table, shown in [Figure 52–12](#).

Figure 52–12 Connector Console Form with Filled Recent Activity Table








Connector Console

Server URL: www.example.com:16200/cs/fdcplg
 Server User: weblogic

Run Now With Token

Recent Activity

Refresh **Cancel**

Timestamp	Launched By	State	Batch size	Updated	No Rule	Error
 4/2/13 4:35:23 PM	fwadmin	Finished	1	1		
 4/2/13 4:22:28 PM	fwadmin	Finished	1	1		
4/2/13 4:21:42 PM	fwadmin	Finished	No Updates			
 4/2/13 3:39:33 PM	fwadmin	Finished	1	1		
4/2/13 3:27:25 PM	fwadmin	Finished	No Updates			
 4/2/13 3:26:05 PM	fwadmin	Finished	22	17	3	
 4/2/13 3:25:36 PM	fwadmin	Finished	30	27	3	
 4/2/13 3:25:13 PM	fwadmin	Finished	30	23	7	
 4/2/13 3:24:40 PM	fwadmin	Finished	30	24	6	

The Recent Activity table lists ten rows at a time, within the last 24 hours. The table must be manually refreshed to provide up-to-date information. Each row represents a batch. By default, the connector processes content items in batches, up to 50 items per batch. Batch size can be changed, as described in [Section 54.2, "User Configurable Parameters."](#)

- To determine whether the connector is importing any content items into WebCenter Sites, refresh the table and read the Updated column.

All options and statistics in the Recent Activity table are described below:

- Clicking **Refresh** updates the table.
- Clicking **Cancel** cancels the currently running connector. Note that processing of the next batch (if any) will be cancelled. The current batch will be processed to completion.
- Click the **Inspect this item** icon (magnifying glass) to view connector history for the selected batch. See [Section 52.9.2, "Viewing Connector History for a Single Batch"](#) for more information.
- The **Timestamp** column shows the date and time of batch processing.
- The **Launched By** column lists whether batch processing was launched manually (from the Connector Console) by the named user, or launched automatically (AUTO) by the connector schedule in the `SystemEvents` table.

For information about connector scheduling, see [Section 52.12, "Step 12: Scheduling the Connector."](#)

- The **State** column lists batch processing status. Possible values are Submitted, Running, Cancelled, Error, Reset, or Finished. Click **Refresh** to view the updated state. This table does not auto-refresh.
 - The **Batch size** column lists the number of items in the batch. By default, the maximum is 50 items per batch. For information about changing maximum batch size, see [Chapter 54, "WCC Connector Configuration Files."](#)
 - The **Updated** column lists the number of items that were updated in WebCenter Sites during batch processing. The number includes assets that were created, updated, or deleted.
 - The **No Rule** column lists the number of content items that did not match any active rule.
 - The **Error** column lists the number of processing errors for this batch.
2. You can gather additional information about connector activity by viewing the connector's history (for all or selected batches). Continue to [Section 52.9, "Step 9: Viewing Connector History."](#)

If you prefer to spot-check the synchronized assets to ensure they are stored in the intended asset types, skip to [Section 52.10, "Step 10: Verifying Imported Content."](#)

52.9 Step 9: Viewing Connector History

All connector sessions are archived. You can view the entire archive, or selected batches (one by one). You can also drill down to individual content items for detailed information about their processing. Archived information can be deleted.

This section contains the following topics:

- [Section 52.9.1, "Viewing the Entire Connector History"](#)
- [Section 52.9.2, "Viewing Connector History for a Single Batch"](#)

52.9.1 Viewing the Entire Connector History









To view connector history for all batches

1. From the **Connector Admin** tab, select **History** to open the Connector History table.

Figure 52–13 Connector History Table with Example Activity

Connector History

Page 1 of 8 | Next 10 ► | Last ►►

Timestamp	Launched By	State	Batch size	Updated	No Rule	Error	All
 4/2/13 4:35:23 PM	fwadmin	Finished	1	1			<input type="checkbox"/>
 4/2/13 4:22:28 PM	fwadmin	Finished	1	1			<input type="checkbox"/>
4/2/13 4:21:42 PM	fwadmin	Finished	No Updates				<input type="checkbox"/>
 4/2/13 3:39:33 PM	fwadmin	Finished	1	1			<input type="checkbox"/>
4/2/13 3:27:25 PM	fwadmin	Finished	No Updates				<input type="checkbox"/>
 4/2/13 3:26:05 PM	fwadmin	Finished	22	17	3		<input type="checkbox"/>
 4/2/13 3:25:36 PM	fwadmin	Finished	30	27	3		<input type="checkbox"/>
 4/2/13 3:25:13 PM	fwadmin	Finished	30	23	7		<input type="checkbox"/>
 4/2/13 3:24:40 PM	fwadmin	Finished	30	24	6		<input type="checkbox"/>
 4/2/13 3:19:11 PM	fwadmin	Finished	21	16	3		<input type="checkbox"/>

Show 10 20 30 50 100 200 300 items per page

Delete

- The Connector History table displays the following information and options:
 - Each row of the Connector History table represents a batch. By default, the connector processes content items in batches, up to 50 items per batch.
 - Click the **Inspect this item** icon (magnifying glass) to view connector history for the selected batch. For more information, see [Section 52.9.2, "Viewing Connector History for a Single Batch."](#)
 - The **Timestamp** column shows the date and time of batch processing.
 - The **Launched By** column lists whether batch processing was launched manually (from the Connector Console) by the named user, or launched automatically (AUTO) by the connector schedule in the `SystemEvents` table. For information about connector scheduling, see [Section 52.12, "Step 12: Scheduling the Connector."](#)
 - The **State** column lists the batch processing state. Possible values are Submitted, Running, Cancelled, Error, Reset, or Finished. Click **Refresh** to view the updated state.
 - The **Batch size** column lists the number of items in the batch. By default, the maximum is 50 items per batch. For information about changing maximum batch size, see in [Chapter 54, "WCC Connector Configuration Files."](#)
 - The **Updated** column lists the number of items that were updated in WebCenter Sites during batch processing. The number includes assets that were created, updated, or deleted.
 - The **No Rule** column lists the number of items that did not match any rule.


- The **Error** column lists the number of processing errors for this batch.
 - Select the **All** option and then click the **Delete** button to delete all activity listings.
 - Select a batch (a row) and click the **Delete** button to delete that batch.
3. To view connector history for a single batch, select the batch. The Connector History table lists all items that were processed in the batch. For an example, see [Figure 52–14](#), in [Section 52.9.2, "Viewing Connector History for a Single Batch."](#)

52.9.2 Viewing Connector History for a Single Batch

1. From the **Connector Admin** tab, select **History** to open the Connector History table.
2. Locate the desired batch (a row), and click its **Inspect** icon (magnifying glass).
The Connector History table lists all items that were processed in the selected batch (see [Figure 52–14](#) for an example).

Figure 52–14 Connector History Details for a Single Batch

Connector History - June 6, 2013 4:05:14 PM PDT


Content ID	Asset ID	Status	Remark 
TV_SAMSUNG_60	1369931666407	Added	Asset created: Media_C:1369931666407 to sites: FirstSiteII
TV_VIZIO_70	1369931666377	Added	Asset created: Media_C:1369931666377 to sites: FirstSiteII

Back

3. The Connector History table for a single batch displays the following information and options:
 - Each row represents a single item in the batch.
 - The **Content ID** column lists Content ID for each item in the batch.
 - The **Asset ID** column lists Asset ID for each item in the batch.
 - The **Status** column lists the synchronization status for each item in the batch. Common status flags are *Added*, *Error*, *Updated*, *Not Matched* (no rule matched), *Not Present* (asset not found), and *Deleted* (asset was deleted).
 - The **Remark** column provides additional information for each item in the batch. Toggle its **Show Details** icon (plus sign) or **Hide Details** icon (minus sign). For an example and descriptions of commonly returned messages, skip to step 4.
 - Clicking the **Back** button takes you back to the previous table (which displays the history for all batches).
4. [Figure 52–15](#) displays an example of possible messages in the **Remark** column. A more comprehensive list is available below the figure.

Figure 52–15 Connector History of a Single Batch with Details in the Remark Column

Connector History - June 6, 2013 4:05:14 PM PDT

Content ID	Asset ID	Status	Remark 
TV_SAMSUNG_60	1369931666407	Added	Checkin QueueToken 19418 Rule matched {TVs}, Wcc request: Rendition(s) {primary} Asset created: Media_C:1369931666407 to sites: FirstSiteII
TV_VIZIO_70	1369931666377	Added	Checkin QueueToken 19419 Rule matched {TVs}, Wcc request: Rendition(s) {primary} Asset created: Media_C:1369931666377 to sites: FirstSiteII

Back

Some of the common messages are listed here:

- No valid rule found—Indicates that either no rules are defined, or the defined rules are invalid.
- No record found—Indicates that no record could be found in the database table, meaning that there are no entries for the particular table being viewed.
- Delete QueueToken {0}—Value {0} is the token ID. This message indicates that the item from WebCenter Content is marked for removal from WebCenter Sites.
- Asset {0} deleted from site {1}—Value {0} is the *assetType_assetID*. Value {1} is the site name. This message confirms deletion of the listed asset from the specified content management site.
- Asset deleted—Confirms asset deletion.
- Asset not found—Typically, this message means that an attempt was made to locate an item to delete, but the item has already been deleted.
- Asset created: {0:1} to sites: {2}—Value {0:1} is the asset type and asset ID. Value {2} is the site name or a comma-separated list of site names. This message indicates that the listed asset was created in the listed site or sites.
- Asset updated: {0:1} to sites: {2}—Value {0:1} is the asset type and asset ID. Value {2} is the site name or a comma-separated list of site names. This message indicates that the listed asset was updated in the listed site or sites.
- Asset {0} delete failed from site {1} because: {2}—Value {0} is the asset ID. Value {1} is the site name. Value {2} is a message as to why the deletion failed. Typically, a deletion fails because the asset was modified since the last synchronization session and now has references to it. If deletion fails, the item is listed in the Orphaned Content form (see [Section 52.13, "Step 13: Managing Synchronized Assets"](#) for more information).
- Checkin QueueToken {0}—Value {0} is the token ID. This message indicates a queue token checkin request, that is, a new or updated item is ready to be imported into WebCenter Sites from WebCenter Content.

- Rule matched {0}, Wcc request: {1}—Value {0} is the rule name. Value {1} is the WebCenter Content rendition or conversion that was requested in the attribute mapping section of the rule.
 - No rule matched—Indicates that no rule matched the item (compare with No valid rule found).
 - Rule matched {0}, but mappings not found—Value {0} is the rule name. This message indicates that a rule matched the item being imported from WebCenter Content, but no mappings of metadata to attributes were found.
 - Rendition {0} was not returned—Value {0} is the rendition name. This message indicates that WebCenter Content did not return the requested rendition (often due to the rendition not being available). Compare with Available Renditions and Requested Renditions not available.
 - Available Renditions: {0}—Value {0} is a list of the item's available renditions on WebCenter Content.
 - Requested Renditions not available—Indicates that the requested rendition is not available. Refer to Available Renditions for a list of available renditions.
 - Error on field {0}, fieldDef: {1}:{2}—Values {0} and {1} are the attribute name (the values are identical). Value {2} is the error message for the exception. This message is returned when an unknown or unexpected exception occurs while the connector is mapping a content item metadata field to an attribute.
5. Before automating synchronization to run routinely, verify the imported content to ensure it is stored in the intended asset types and with the expected level of detail. Continue to [Section 52.10, "Step 10: Verifying Imported Content."](#)

52.10 Step 10: Verifying Imported Content

At this point, you are likely to have enough information about how well content synchronization meets your requirements. To confirm, spot check the imported content items to ensure they are stored in the intended asset types, that all metadata values are recorded in the correct attributes, and all other conditions that were specified in the attribute mapping part of the rules are met. To locate the content items and the assets, refer to the Connector History table for Content IDs and their counterpart Asset IDs.

Once verification is complete, do one of the following:

- If adjustments are necessary, continue to [Section 52.11, "Step 11: If You Need to Modify Your Setup."](#)
- If the results meet your requirements, continue to the following sections:
 - [Section 52.12, "Step 12: Scheduling the Connector"](#)
 - [Section 52.13, "Step 13: Managing Synchronized Assets"](#)

52.11 Step 11: If You Need to Modify Your Setup

If the results of your synchronization sessions do not meet your requirements, you have a number of ways to adjust system configuration. Before taking action, it is best to review your options. They are as follows:

- You can return to some of the steps in this chapter; for example, enable more content items for synchronization, disable content items, modify rules, re-prioritize active rules, and so on.

Note: If you choose to modify or re-prioritize active rules, first delete the synchronized assets from WebCenter Sites. Clearing asset types helps to simplify interpretation of synchronization results after the connector runs again. It also prepares your system for rollback of its synchronization point to the original starting point or to a particular update, should you decide to take that step.

- You can use the `wcc-integration.ini` file to reconfigure various parameters, such as maximum batch size. For more information, see [Section 54, "WCC Connector Configuration Files."](#)
- You can roll back the synchronization point to the original starting point, or to a particular update by resetting the token.

Caution: If you choose to reset the token, all content items at the new synchronization point will be re-examined by the connector and resynchronized to WebCenter Sites.

Before rolling back, prepare your system by deleting the synchronized assets from WebCenter Sites. This step is especially important if you modified connector rules in any of the following ways: changed the target asset types, changed their content management sites, or re-prioritized the rules. Once the assets are deleted, you can reset the token. For more information about tokens, see [Section 53.1, "Rolling Back the Synchronization Point."](#)

Once you have successfully modified your setup, continue to the following sections:

- [Section 52.12, "Step 12: Scheduling the Connector"](#)
- [Section 52.13, "Step 13: Managing Synchronized Assets"](#)

52.12 Step 12: Scheduling the Connector

Once you have configured and verified content synchronization, you may want to schedule the connector to run on a timed basis. Use the Oracle WebCenter Sites: Explorer utility to add a timed entry to the `SystemEvents` table.

To schedule the connector

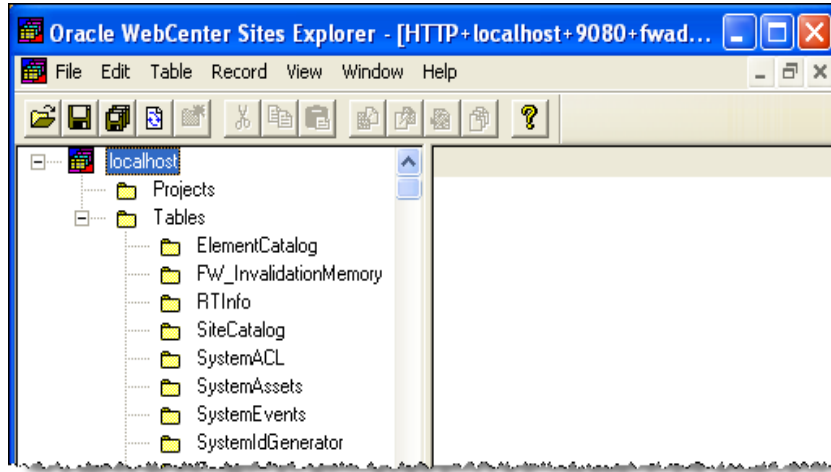
1. Launch the Sites Explorer executable file (`ContentServerExplorer.exe`) located in the `SitesExplorer` directory of the distribution file.

Note: Sites Explorer is a Windows-only utility.

2. Log in to Sites Explorer. (For information about login credentials, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*.)
3. In Sites Explorer, do the following:

- a. Expand the **Tables** node.

Figure 52–16 Oracle WebCenter Sites: Explorer with Expanded Tables Node



- b. Select the **SystemEvents** table (see [Figure 52–16](#)).
- c. From the main menu, select **File**, then **New**, and then **Record**.
- d. Define the scheduling event for the WCC connector by entering the following information:
 - * **eventname:** WCCConnector
 - * **type:** 1
 - * **enabled:** 1
 - * **times:** Leave this field blank for now. You will set it in step e.
 - * **target:** ContentServer
 - * **params:** pagename=WCC/Pull
- e. Define the schedule for running the connector by entering a time pattern into the **times** field, in the following format:

h:m:s W/D/M

Fields of the time pattern and their legal values are defined in [Table 52–2](#). Below the table is an example of a time pattern.

Table 52–2 Fields of a Time Pattern in the SystemEvents Table

Field	Legal Values	All Legal Values	Single Value Format	Inclusive Range Format	List Format
<i>h</i> (hours)	0–23	*	<i>h</i>	<i>h-h</i>	<i>h, h</i>
<i>m</i> (minutes)	0–59	*	<i>m</i>	<i>m-m</i>	<i>m, m</i>
<i>s</i> (seconds)	0–59	*	<i>s</i>	<i>s-s</i>	<i>s, s</i>
<i>W</i> (day of the week)	0–6, where 0=Sunday	*	<i>W</i>	<i>W-W</i>	<i>W, W</i>
<i>D</i> (day of the month)	0–31	*	<i>D</i>	<i>D-D</i>	<i>D, D</i>
<i>M</i> (month)	1–12	*	<i>M</i>	<i>M-M</i>	<i>M, M</i>

Example Time Pattern:

```
2,5-7:0:0 5/*\/*
```

The first field, `2,5-7:0:0` is a comma-separated list of times that includes a range. It means the event is triggered at 2 AM, 5 AM, 6 AM, and 7 AM at 0 minutes and 0 seconds for every specified hour. The next field (`5/`) specifies that the event is triggered every Friday. The next field (`*\`) ensures that the event is not triggered every day of the month. The last field (`/*`) ensures that the event will be triggered in all months.

The specification of days can be made by two fields: day of the week (WD) and day of the month (DD). If both are specified, both take effect. For example, `1:0:0 1/15/*` means the event is triggered at 1 AM, 0 minutes and 0 seconds, every Monday, and on the fifteenth day of each month. To specify days by only one field, set the other field to `*`.

4. Click **Save**.
5. Continue to [Section 52.13, "Step 13: Managing Synchronized Assets"](#) for information about helping content contributors identify and manage synchronized assets.

52.13 Step 13: Managing Synchronized Assets

Once the required content items are imported into WebCenter Sites, you can work with content contributors to help them identify and manage the synchronized assets, especially if synchronization sessions will be running on a regular basis. Administrators who publish also need to know how the assets are affected by synchronization sessions.

This section contains the following topics:

- [Section 52.13.1, "Mapped Attributes"](#)
- [Section 52.13.2, "Orphaned Content"](#)

52.13.1 Mapped Attributes

If content contributors modify assets by changing the values of attributes that were explicitly mapped in the attributes mapping part of the connector rules, those values will be updated when the content items are updated and resynchronized. All other attributes will have their values preserved.

Note: If the value of a mapped attribute is changed, it will be overwritten when its content item is updated and resynchronized. Keeping changes to mapped attributes requires copying the assets.

52.13.2 Orphaned Content

One task that synchronization cannot accomplish is deleting synchronized assets that have dependents. This results in orphaned content.

Orphaned content is an asset that cannot be deleted when its counterpart content item is either deleted, expired, or disabled from synchronization (by having its **Sync To Sites** option set to `False`). The orphaned asset cannot be deleted because dependencies were created on that asset since the last synchronization session.

Note: If a synchronized asset is shared across content management sites, it will be deleted, unless it has dependents.

To determine orphaned assets

1. From the WebCenter Sites Admin interface, open the **Connector Admin** tab.
2. Select **Orphaned Content** to open the Connector Orphaned Content table (Figure 52–17).

Figure 52–17 Connector Orphaned Content Table with Example Information

Connector Orphaned Content				
Content ID	Connector Token	Asset ID	Referenced By	Remark
DGT	11816	Media_C:1359061879217	FSII AudioCo iAC-008 Portable Audio Player FSII AudioCo iAC-008 Portable MP3 Player	Exception: com.fatwire.assetapi.common.AssetAccessException "Error voiding asset Media_C:1359061879217 with erno number is: -12074"

Show 10 20 30 50 100 200 300 items per page

The Connector Orphaned Content table displays the following columns, which identify the deleted source content items, the orphaned assets, and their dependent assets:

- The **Content ID** column lists Content IDs for the content items that were deleted or disabled from synchronization in WebCenter Content.
 - The **Connector Token** column lists the token's `update_number`, from which you can derive the date and time when the content items were processed by the connector. Date and time can be obtained from the WCS Queue Support Information table, described in [Chapter 53, "Resynchronization Options."](#)
 - The **Asset ID** column lists the orphaned assets by their WebCenter Sites asset IDs.
 - The **Referenced By** column lists the dependent assets (which reference the orphaned assets).
 - The **Remark** column lists additional information (if any) for orphaned assets.
3. Use the Connector Orphaned Content table to locate the orphaned assets in WebCenter Content. More information about deleted content items is available in WebCenter Content.

Resynchronization Options

When setting up your WCC Integration system for content synchronization, you may have to roll back the synchronization point. This chapter describes rollback procedures and provides information about the synchronization token.

This chapter contains the following sections:

- [Section 53.1, "Rolling Back the Synchronization Point"](#)
- [Section 53.2, "About the Generated Token Value"](#)

53.1 Rolling Back the Synchronization Point

WebCenter Sites and WebCenter Content are synchronized through a generated token, which rarely needs to be reset once the content synchronization process is established.

Note: It is not recommended to reset the token for an established content synchronization process.

During system setup, however, you may have re-configured rules or asset types and therefore need to roll back the synchronization point to either the original starting point or a particular update. Rolling back involves resetting the token.

Before discussing token reset, this section briefly describes a token and how it is updated when exchanged between WebCenter Content and WebCenter Sites:

1. When WebCenter Sites contacts WebCenter Content for the first time, it receives a token from WebCenter Content. The token value has the following format:

Token Value Format:

```
ClientID: Internally_used_database_identifier: update_number
```

Example:

```
c10013:1344632434389:10012
```

2. When checking for updates, WebCenter Sites passes the token to WebCenter Content, which uses the `update_number` to determine if any changes were made since the last call. Updates will include content items whose **Sync to Sites** option is (or was once) enabled. If updates were made, WebCenter Content generates a new `update_number` and returns the updated token to WebCenter Sites.

If the need arises, a token value can be rolled back to its original value (the original synchronization point), or to a particular update, as described next:

Caution: Tokens should be reset only during system setup, when synchronization-related objects (such as rules) are being configured, tested, and re-configured. Once your system is configured for actual, uninterrupted operation, token values should not be modified.

- Replacing a token value with a blank value rolls back the point of synchronization to the original starting point. The next time the connector runs, it will re-examine and resynchronize every content item whose **Sync to Sites** option is enabled or was once enabled.

Note: The connector re-examines previously enabled content items (which are no longer marked as **Sync to Sites**) in case their counterpart assets need to be deleted from WebCenter Sites.

- Replacing the token's `update_number` with a previously generated `update_number` rolls back the point of synchronization to the previous update. The next time the connector runs, it will start polling content item versions at the new synchronization point. The connector will re-examine and resynchronize every item whose **Sync to Sites** option is enabled or was once enabled. The connector re-examines previously enabled content items (which are no longer marked as **Sync to Sites**) in case their counterpart assets need to be deleted from WebCenter Sites.

A history of token values and `update_numbers` is available in WebCenter Content, on the WCS Queue Support Information page. For information about resetting the synchronization point, see the following sections:

- [Section 53.1.1, "Rolling Back to the Original Starting Point"](#)
- [Section 53.1.2, "Rolling Back to a Particular Update"](#)

For more information about token value, see [Section 53.2, "About the Generated Token Value."](#)

53.1.1 Rolling Back to the Original Starting Point

Resetting the token to a blank value rolls back the current point of synchronization to the original starting point.

To resynchronize to the original starting point

1. From the WebCenter Sites Admin interface, open the **Connector Admin** tab.
2. Select **Console**.
3. In the Connector Console form ([Figure 53–1](#)), delete the token value (next to the **Run with Token** button).

Figure 53–1 Connector Console Form

Connector Console

Server URL: www.example.com:16200/cs/idcplg
 Server User: weblogic

Run Now With Token

53.1.2 Rolling Back to a Particular Update

Resetting the token's `update_number` to a previously generated value rolls back the current point of synchronization to that particular update.

To roll back to a particular update

1. Locate the token value to roll back to:
 - a. Log in to WebCenter Content as an administrator.
 - b. Select **Administration** and then **WCS Administration**.

The WCS Queue Support Information page opens (Figure 53–2).

Figure 53–2 WCS Queue Support Information in Table WebCenter Content Server

ORACLE Content Server						
Search New Check-In						
v My Content Server v Web Sites v Browse Content v Content Management v Administration						
WCS Queue Support Information						
WCS Queue						
Token	QueueType	QueueId	Time	Sync	Message	
11526	G	IMAGE_DMA_LOGO_2_5_7	1/11/13 2:54 PM	True	(+Y)	
11527	G	ICON_THREE_COL_2_5_6	1/11/13 2:55 PM	True	(+Y)	
11528	G	IMAGE_DMA_LOGO_5_7	1/11/13 2:56 PM	True	(+Y)	
11529	G	IMAGE_DMA_LOGO_3_4_5_6_7	1/11/13 2:56 PM	True	(+Y)	
11530	G	IMAGE_DMA_LOGO_1_5	1/11/13 2:57 PM	False	(+Y)	
* Indicates previously marked Sync To Sites						
WCS Recent Queue Queries						
ClientId	AccessCount	LastAccessTime	RequestToken	SentToken	Message	
c1	20	1/18/13 2:19 PM	c1:1357943779870:11616	c1:1357943779870:11618	10.255.250.250	
c11489	5	1/22/13 2:07 PM	c11489:1357943779870:1	c11489:1357943779870:11624	10.255.251.251	
c11610	4	1/18/13 11:51 AM	c11610:1357943779870:11542	c11610:1357943779870:11613	10.255.252.252	
c11613	3	1/18/13 12:30 PM	c11613:1357943779870:11542	c11613:1357943779870:11613	10.255.253.253	

- c. At the top of the page, is the WCS Queue Support Information table. Look in the **QueueId** column (equivalent to **Content ID**) to locate the item that was updated. Look in the **Token** column to obtain the item's `update_number`.
2. Reset the token value in WebCenter Sites:
 - a. From the WebCenter Sites Admin interface, open the **Connector Admin** tab.
 - b. Select **Console** to open the Connector Console form and replace the token's `update_number` (last set of digits), with the new `update_number`.

53.2 About the Generated Token Value

A generated token value has three parts. Each part is separated by a colon (:).

`ClientID:Internally_used_database_identifier:update_number`

This section describes the parts by using the following value as an example:

`c10013:1344632434389:10012`

- The first part of a token value is the ClientID (c10013, in our example).
This value is generated by WebCenter Content when a WebCenter Sites client connects for the first time. The value is returned to the client and used in all subsequent calls by that client. If a client re-connects with a blank token, it will get a new ClientID.
- The middle part of a token value (1344632434389, in our example) is the unique identifier for the database table in WebCenter Content.

Note: The middle part of a token value must not be edited. It is used internally for client-server resynchronization.

- The last part of a token value (10012, in our example) is the `update_number`.
WebCenter Content generates `update_numbers` for content items as follows:
 - When a new content item is checked in and its **Sync to Sites** option is enabled, it receives an `update_number`. The item's Content ID and `update_number` are logged in the WCS Queue table. That information is kept in the table, even if the item is deleted.
 - When a content item has its **Sync to Sites** option enabled, it receives a new `update_number`, which is logged in the WCS Queue table. The logged information will always remain in the table, even if the content item is deleted.
 - When a content item is updated or its **Sync to Sites** option is reset, it receives a new `update_numbers`, which is logged in the WCS Queue table.
 - When a content item is deleted and its **Sync to Sites** option was enabled, it receives a new `update_number`. Although the item has been deleted, its ContentId and new `update_number` remain in the WCS Queue table. This allows notifications of deleted content to be provided.

WCC Connector Configuration Files

This chapter briefly describes configuration files used by the WCC Integration feature. It also provides detailed information about `wcc-integration.ini`, which contains user configurable parameters.

This chapter contains the following sections:

- [Section 54.1, "About WCC Connector Configuration Files"](#)
- [Section 54.2, "User Configurable Parameters"](#)

54.1 About WCC Connector Configuration Files

WCC connector configuration files consist of several INI files located in the `/Shared/clustersynch/ucm/ini` directory of the WebCenter Sites installation. The only file that may need editing is the `wcc-integration.ini` file, which is used to optimize conditions for the WCC connector and to troubleshoot, as described in [Section 54.2, "User Configurable Parameters."](#)

Note: The following configuration files must not be edited. They are used internally for data storage:

- `wcc-matchers.ini`: Contains the rule statements that were created via the rules wizard in the WebCenter Sites Connector Admin tab. Each rule is listed by name and identified as either enabled or disabled. Also included are a description of the rule and its statement logic. This file is paired with the `wcc-mappers.ini` file.
 - `wcc-mappers.ini`: Contains the rule details defined in the rules wizard. This file is paired with the `wcc-matchers.ini` file.
 - `wcc-token.ini`: Contains a token received from WebCenter Content indicating the last time the server synchronized with WebCenter Sites.
-

54.2 User Configurable Parameters

The `wcc-integration.ini` configuration file contains several system parameters that are user configurable. These parameters are not configurable in the WebCenter Sites Admin interface.

The `wcc-integration.ini` configuration file is located in the `/Shared/clustersynch/ucm/ini` directory of the WebCenter Sites installation, and it uses the Java property file format, as shown in the following example:

```
#wcc-integration.ini
wcc.batch.size=50
wcc.batch.autorepeat=true
wcc.download.dir=C:\Oracle\WebCenter\Sites\tomcat\temp
wcc.download.autoclean=true
wcc.keyfield=name
wcc.rule.case-sensitive=true
```

Parameters listed in the example above are user configurable and defined as follows:

- `wcc.batch.size=[50]`

When a lot of content needs to be processed, processing is done in batches. This parameter specifies the maximum size of the batch.

When the `wcc-integration.ini` configuration file is created, the default batch size is set to 50. This value can be changed, but the parameter cannot be left blank and must contain a valid value.
- `wcc.batch.autorepeat=[true | false]`

If the connector runs and receives a response that the total number of items exceeds the number defined in the `wcc.batch.size` parameter, then the connector can be set to either automatically run again once the batch is processed (`true`), or the connector can wait for the next manual or next timed run (`false`).

When the `wcc-integration.ini` configuration file is created, this parameter is set to a default value of `true`. The value can be changed, but the parameter cannot be left blank and must contain a valid value. Valid values are `true` or `false`.
- `wcc.download.dir=[temporary_zip_file_location]`

This parameter defines the temporary download location of the ZIP files that are transferred from WebCenter Content.
- `wcc.download.autoclean=[true | false]`

This parameter defines whether to delete the ZIP file when processing is complete. This is a debugging parameter that will either force the temporary files to be cleaned up (`true`), or leave files on disk for analysis (`false`).

When the `wcc-integration.ini` configuration file is created, this parameter is set to a default value of `true`. The value can be changed, but the parameter cannot be left blank and must contain a valid value. Valid values are `true` or `false`.
- `wcc.keyfield=[name]`

This parameter specifies the name of the asset type attribute that will receive the value of the `dDocName` metadata field in WebCenter Content. By default, the value of `wcc.keyfield` is `name`, because `name` is a default attribute available in every asset type. It is unlikely that this value would need to be edited.

This value may be changed once, on a new system, before rules are created, and before content synchronization begins. Once synchronization starts, this value should not be edited. Otherwise, assets may be duplicated. An example situation where the default value might be changed is when the `name` field is already in use. For more information about `dDocName`, see [Section 50.4.1.1, "Content Identifier Attribute."](#)
- `wcc.rule.case-sensitive=[true | false]`

This parameter is used to compare string metadata values specified in connector rules with the actual metadata values of content items. If the parameter is set to `true`, case-sensitivity is enabled.

When the `wcc-integration.ini` configuration file is created, this parameter is set to a default value of `true`. Valid values are `true` or `false`.

Part VIII

Implementing Oracle WebCenter Sites Content Integration Platform for File Systems and Microsoft SharePoint

This part describes using the Content Integration Platform (CIP) to publish to the Oracle WebCenter Sites web application from file systems and Microsoft SharePoint.

This part contains the following chapters:

- [Section 55, "Integrating with Oracle WebCenter Sites"](#)
- [Chapter 56, "Publishing"](#)
- [Chapter 57, "Configuring Event Notification"](#)
- [Chapter 58, "Remapping"](#)
- [Chapter 59, "Adding Assets to a Flex Family"](#)
- [Chapter 60, "File Systems: Default Mapping Specifications"](#)
- [Chapter 61, "Microsoft SharePoint Systems: Default Mapping Specifications"](#)

Integrating with Oracle WebCenter Sites

This chapter begins with an overview of the Oracle WebCenter Sites: Content Integration Platform (CIP) for File Systems and Microsoft SharePoint. The middle section provides information about the default mapping framework, which supports default publishing scenarios. The last section outlines your option to set up workflows that inform administrators of events that occur on the target system when monitored content is modified on the source system.

This chapter contains the following sections:

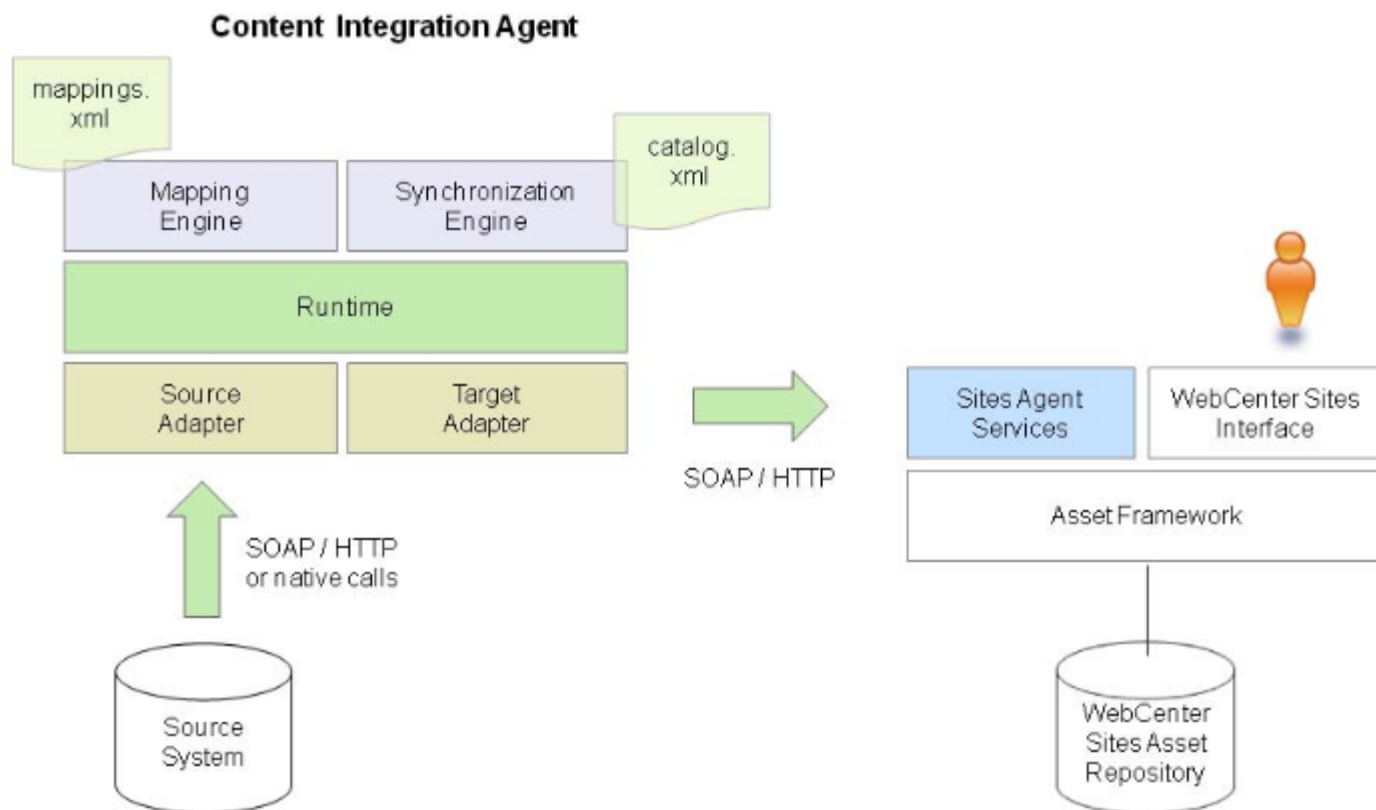
- [Section 55.1, "Overview"](#)
- [Section 55.2, "System Defaults"](#)
- [Section 55.3, "Event Notification Option"](#)

Note: The following feature is deprecated in WebCenter Sites 11g Release 1 (11.1.1.8.0): SOAP-based web services. This feature is replaced by REST services.

55.1 Overview

Content Integration Platform (CIP) for File Systems and Microsoft SharePoint enables you to publish files and SharePoint objects to WebCenter Sites. The publishing process uses the CIP components Content Integration Agent and Sites Agent Services, both shown in [Figure 55-1](#).

Figure 55–1 System Architecture



- **Content Integration Agent** is a process, either daemon or standalone, responsible for synchronizing the metadata of objects selected for publication. Its main components are:
 - The `mappings.xml` file, which defines the metadata of objects selected for publication.
 - `CIPCommander`, used to initiate publishing sessions from the command line, where you name the folder to be published. Publishable content includes subfolders and documents (and pictures in SharePoint systems).

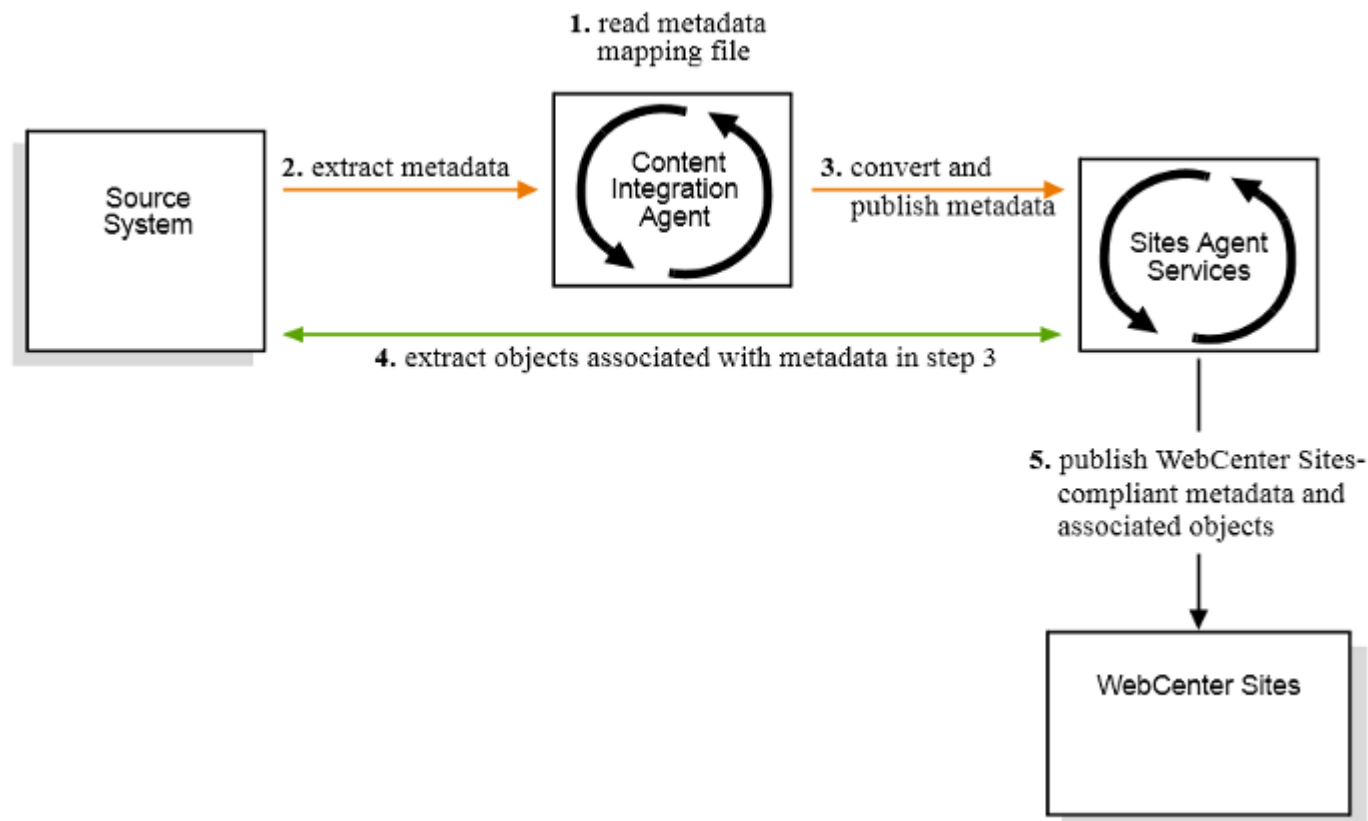
During a publishing session, the Content Integration Agent process 1) reads the `mappings.xml` file, 2) extracts the mapped metadata from the source system, 3) converts the metadata to a format recognized by WebCenter Sites and invokes the synchronization engine to publish the WebCenter Sites-compliant metadata to Agent Services. Having completed the publishing process, the synchronization engine starts monitoring the source system's published folder. Then, every time a new object is created, deleted, or modified in the monitored folder, the synchronization engine updates Sites Agent Services with the new metadata. (The synchronization interval is configurable.)

- **Sites Agent Services** is a web application responsible for receiving WebCenter Sites-compliant metadata from Content Integration Agent and storing it in the WebCenter Sites database. Agent Services also runs a background process that extracts and stores the primary binary contents associated with the metadata. Agent Services exposes the Web Services interface needed by Content Integration Agent to perform the synchronization process.

- `catalog.xml` (also within Content Integration Agent), stores information about published objects. When the objects are "unpublished," their information is deleted from `catalog.xml`.

The publishing process, illustrated in [Figure 55–2](#), also applies to the synchronization event that takes place when published objects are modified or deleted, or new objects are added to the monitored repository.

Figure 55–2 Publishing to WebCenter Sites



55.2 System Defaults

Publishing an object to WebCenter Sites requires:

- A flex family to store the source system's published metadata (object types and attributes) and the associated objects.
- A mappings file (`mappings.xml`), which maps the object's metadata to asset types and assets in the flex family.

55.2.1 Default Mapping Framework

The default mapping framework in Content Integration Platform defines publishable objects to be of the type and attributes shown in [Table 55–1](#).

Table 55–1 Source-System Metadata Supported by Default

Source System	Default Object Types and Attributes	See Also...
File System	<p>File Type</p> <p>Folder type named <code>Folder</code></p> <p>Document type named <code>Document</code></p> <p>Attributes</p> <p><code>contentAttr</code></p> <p><code>contentURL</code></p> <p><code>FileSize</code></p>	Chapter 60, "File Systems: Default Mapping Specifications"
SharePoint	<p>Content Type:</p> <p>Folder type named <code>Folder</code></p> <p>Document type named <code>Document</code></p> <p>Picture type named <code>Picture</code></p> <p>Attributes:</p> <p>Documents:</p> <p><code>contentAttr</code></p> <p><code>contentURL</code></p> <p><code>File Size</code></p> <p>Pictures:</p> <p><code>Date Picture Taken</code></p> <p><code>Description</code></p> <p><code>Keywords</code></p> <p><code>Picture Height</code></p> <p><code>Picture Width</code></p>	Chapter 61, "Microsoft SharePoint Systems: Default Mapping Specifications"

Attribute names are display names. The following system-defined attributes – `DateCreated`, `DateModified`, and `MimeType` – are less commonly used and therefore omitted from the default `mappings.xml` file and flex family.

To give you a quick start, the default mapping framework provides the following components:

- A flex family named after its source system: `FileSystem` or `SharePoint`. The purpose of each flex family is to store the source system's default metadata (in [Table 55–1](#)) on WebCenter Sites and thus provide asset type tables to store the associated objects as assets.
- A pre-configured `mappings.xml` file, which maps the default object types and attributes in [Table 55–1](#) to WebCenter Sites assets in the `FileSystem` and/or `SharePoint` flex families. (The `mappings.xml` file is located on the server that hosts Content Integration Agent.)

For complete specifications on the default mapping framework, see the appendices at the end of this guide

55.2.2 Implications for Publishing and Synchronization

The success of publishing and synchronization processes depends on the `mappings.xml` file and the source system's flex family.

55.2.2.1 Default mappings.xml and Flex Families

If your source objects are of the types specified in the default `mappings.xml` file (see also [Table 55-1](#)), you can publish those objects to WebCenter Sites without having to modify either `mappings.xml` or the default flex family. During publishing, the objects are automatically re-created in their respective flex family as either `flexparent` assets (if they are folders) and `flex child` assets (if they are documents or pictures).

Following the publication process, changes to monitored folders and their contents (on the source side) are propagated to the flex family by the synchronization engine. For example, if you modify or delete published objects on the source side, or you create new objects (of the mapped types) in the monitored folders, your changes will be automatically propagated to the flex family by the synchronization engine.

55.2.2.2 Custom mappings.xml and Flex Families

If the schema of the source system is modified, the source system's flex family must be updated in order for publishing and synchronization to produce the expected results. The `mappings.xml` file, however, may or may not require updates, depending on the nature of the schema changes. For example:

- If new document types are created for documents that will be published, both the flex family and `mappings.xml` must be updated with the new document types.
- If a newly added attribute will be propagated to WebCenter Sites, that attribute must be added to the relevant flex family and assigned to the relevant asset type definition.

Mapping an attribute, however, is conditional:

An attribute must be mapped (in `mappings.xml`) if it will be propagated to WebCenter Sites and it is named differently on the source and WebCenter Sites systems. (If attributes are named identically, they can be omitted from `mappings.xml`)

Incorrect mapping of attributes does not stop the publication process, but it does produce a warning message and an entry in the log file. The publication process continues by skipping to the next publishable object.

55.3 Event Notification Option

When events such as asset creation occur in published folders on the source system, Content Integration Platform responds by synchronizing the target system to the source system. Content Integration Platform can be configured for event notification. Events would then trigger notices to CIP administrators, informing them of the events and whether synchronization took place. Notices are delivered to CIP administrators in a simple workflow process. Content Integration Platform ships with several sample workflows. You have the option to enable any or all of the workflows during the CIP installation procedure.

This chapter contains information about publishing to WebCenter Sites from file systems and Microsoft SharePoint systems.

This chapter contains the following sections:

- [Section 56.1, "Overview"](#)
- [Section 56.2, "Publishing to WebCenter Sites"](#)
- [Section 56.3, "Examples"](#)
- [Section 56.4, "When an Object is Published"](#)
- [Section 56.5, "Synchronization"](#)
- [Section 56.6, "Unpublish Command"](#)

56.1 Overview

Once the source system schema is mapped to WebCenter Sites, any object that is based on the schema can be published to WebCenter Sites.

This section contains the following topics:

- [Section 56.1.1, "Synchronization"](#)
- [Section 56.1.2, "Unpublishing"](#)

56.1.1 Synchronization

Manual publishing is unnecessary after the initial session, because the synchronization engine takes over. The engine monitors published folders (on the source system) and mirrors the changes it detects:

- Modification to published objects
- Deletion of published objects
- Addition of objects to the monitored folder(s), as long as their object types are mapped
- Attributes require special treatment, as explained in [Chapter 58, "Remapping,"](#)

The synchronization interval can be configured in the `publish` command (see [Section 56.2, "Publishing to WebCenter Sites"](#)).

Event notification for the synchronization process can be enabled by installing CIP-related default workflows, or specially creating custom workflows. For more information, see [Chapter 57, "Configuring Event Notification."](#)

56.1.2 Unpublishing

The unpublish command is used to clear `catalog.xml` of all entries that are associated with published objects. Including the `-delete` parameter removes the same entries from the WebCenter Sites database. For more information, see [Section 56.6, "Unpublish Command."](#)

56.2 Publishing to WebCenter Sites

If the source system schema is mapped to WebCenter Sites, you can successfully publish objects that are based on the schema. (Otherwise, you will need to remap the schema. Instructions are available in [Chapter 58, "Remapping"](#)).

To publish to WebCenter Sites

1. Make sure Content Integration Agent is running.

Note: If you changed the port in the section "Restart the Content Integration Agent Executable" in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*, make sure that the new port is set in `facilities.xml`, and add `-p <port>` to the command in the following step (which starts CIPCommander).

2. Run the CIPCommander executable (located in the `bin` folder of the system where Content Integration Agent is installed):

```

cipcommander
publish <source_providerid> <target_providerid>
-source_repname <source_repname>
-source_path <source_path>
-target_repname <target_repname>
-mapping <mapping_id>
-replic_mode <full | ingestion>
-bulk_resynch_interval <seconds>

```

where:

- `<source_providerid>` is the provider ID for the source system:

File System:

2023b849-688e-4009-af1f-903fe62d85b7

Microsoft SharePoint:

7137dd5d-9ed7-4327-b4fd-8caeebd5889a

- `<target_providerid>` is the WebCenter Sites provider ID:

70b1e307-26a1-499c-9295-cf0b6bd01342

Parameters and values for the `publish` command are defined in [Table 56-1, "Publishing Parameters"](#). Examples on the usage of the `publish` command are available in [Section 56.3, "Examples."](#) Information about an object's publication data is available in [Section 56.4, "When an Object is Published."](#)

Table 56–1 Publishing Parameters

Publishing Parameter	Value
-source_repname	<p><source_repname> : Name of the source repository from which content will be published. Enter the name exactly as it appears in the URL.</p> <p>Legal values:</p> <p>File System: Empty string ("").</p> <p>SharePoint: Name of the document library or picture library that contains the folder to be published.</p> <p><i>Examples:</i> See Section 56.3, "Examples."</p>
-source_path	<p><source_path>: Path to the object you want to publish.</p> <p>Legal values:</p> <p>File System: /<folder>/<folder>/ .../<folder>/ (to publish the last folder in the path)</p> <p>SharePoint: / (to publish a library [document or picture] and its contents)</p> <p>/<folder>/<folder> .../<folder>/ (to publish the last folder in the path)</p>
-target_repname	<p><target_repname>: Name of the content management site (in WebCenter Sites) in which the flex family is enabled for this source system.</p>
-mapping	<p><mapping_id> : Value of the mapping id in mappings.xml . If you are using the default mappings.xml , enter one of the following values:</p> <ul style="list-style-type: none"> ■ sharepoint2cs ■ filesystem2cs
-replic_mode	<p>full ingestion</p> <ul style="list-style-type: none"> ■ full means that a full replication will be performed (by default). ■ ingestion means that only item creation events will be propagated. Modifications and deletions on the source side will not be reflected on the target.
-bulk_resynch_interval	<p><seconds>: Number of seconds between two successive synchronization events.</p> <p>An optional publishing parameter. For more information, see Section 56.5, "Synchronization."</p>

56.3 Examples

This section contains the following topics:

- [Section 56.3.1, "Publishing from a File System"](#)
- [Section 56.3.2, "Publishing from Microsoft SharePoint"](#)

56.3.1 Publishing from a File System

To publish the C:\publish folder to the CIPDemo content management site, using filesystem2cs default mapping:

```

cipcommander publish 2023b849-688e-4009-af1f-903fe62d85b7
70b1e307-26a1-499c-9295-cf0b6bd01342
-source_repname ""

```

```
-source_path c:\publish-mapping filesystem2cs-target_rename CIPDemo
```

56.3.2 Publishing from Microsoft SharePoint

- To publish the Images picture library to the CIPDemo content management site using, sharepoint2cs default mapping:

```
cipcommander publish
7137dd5d-9ed7-4327-b4fd-8caeebd5889a70b1e307-26a1-499c-9295-cf0b6bd01342-source
_rename Images-source_path /-mapping sharepoint2cs-target_rename CIPDemo
```

- To publish the Cool/Bright folder in the Images picture library to the CIPDemo content management site, using sharepoint2cs default mapping:

```
cipcommander publish
7137dd5d-9ed7-4327-b4fd-8caeebd5889a70b1e307-26a1-499c-9295-cf0b6bd01342-source
_rename Images-source_path /Cool/Bright-mapping sharepoint2cs-target_rename
CIPDemo
```

56.4 When an Object is Published

When an object is published, `catalog.xml` (located in `integration_agent/conf/`) is updated with an entry that identifies both the source system and the WebCenter Sites system (within the `<workspace>` tags), and specifies replication data for the published object (within the `<replication>` tag). The code below is an example of a publication entry for an object that was published from a file system:

```
<workspace id="8a55488d-97c9-4290-92e1-d7bb9e476dc7">
  <provider-ref refid="2023b849-688e-4009-af1f-903fe62d85b7" />
  <init-params>
    <param name="rename" />
    <param name="path">c:/temp/CIP_xml/test1/test3</param><param name="repid" />
  </init-params>
</workspace>
<workspace id="f917ddcb-1cbb-46c5-8840-fb194b693629">
  <provider-ref refid="70b1e307-26a1-499c-9295-cf0b6bd01342" />
  <init-params>
    <param name="rename">CIPTest</param>
    <param name="repid">39e1e988-f1a6-4913-a16d-45a1c5ad9976</param>
  </init-params>
</workspace>
<replication>
  <link id="04e4f0f2-8535-492b-8590-691a510884d4">
    <source-ref refid="8a55488d-97c9-4290-92e1-d7bb9e476dc7" />
    <target-ref refid="f917ddcb-1cbb-46c5-8840-fb194b693629" />
    <mapping-ref refid="filesystem2cs" />
    <init-params>
      <param name="BulkResynchInterval">600</param>
      <param name="ReplicMode">full</param>
    </init-params></link>
</replication>
```

56.5 Synchronization

Once objects are published, the synchronization engine monitors the status of the folder(s) from which the objects were published and mirrors the changes it detects. For optimal performance, set the synchronization interval to a value that agrees with the

frequency of updates to the source system. To set the synchronization interval, include the `bulk_resynch_interval` parameter in the publish command (Section 56.2, "Publishing to WebCenter Sites").

56.6 Unpublish Command

You can unpublish objects from `catalog.xml` alone (and additionally, from WebCenter Sites) by executing the `cipcommander unpublish` command with parameters that suit your requirements.

Note: The unpublish command clears `catalog.xml` of all entries that are associated with published objects (for a sample publication entry, see the code in Section 56.4, "When an Object is Published"). To remove the same entries from the WebCenter Sites database, you must include the `-delete` parameter.

The unpublish command takes the following form and parameters:

```
cipcommander unpublish <parameters>
```

Table 56–2 Unpublish Parameters

Unpublish Parameter	Description
-all	Use this parameter to clear <code>catalog.xml</code> of all publication entries.
-linkid	Use this parameter to clear <code>catalog.xml</code> of selected publication entries. linkid specifies the published object's link to the WebCenter Sites system. Use the value in the published object's <code><link ></code> tag, which is nested within the object's <code><replication></code> tag (for sample code, see Section 56.4, "When an Object is Published.") For example: To unpublish a single object from <code>catalog.xml</code> , obtain its <code><linkid></code> and issue the following command: <pre>cipcommander unpublish linkid 04e4f0f2-8535-492b-8590-691a510884d4</pre> To unpublish multiple objects, add their linkid's to the unpublish command.
-delete	Use this parameter to remove, from the WebCenter Sites database, the same objects that you are unpublishing from <code>catalog.xml</code> . Legal values: <code><true false></code> Default value: <code>true</code>

Configuring Event Notification

This chapter contains the following sections:

- [Section 57.1, "Overview"](#)
- [Section 57.2, "Sample Workflows"](#)

57.1 Overview

Event notification keeps Content Integration Platform administrators informed about the synchronicity of source and target systems. When changes are made to monitored folders (published folders on the source system), CIP administrators receive confirmation that the same changes were either successfully propagated to WebCenter Sites, or propagation failed to occur. Event-driven notices are delivered to administrators in a simple workflow process.

A default workflow is available for each of the following events:

Event in WebCenter Sites	Workflow
Asset creation	CIPAssetCreated. Invoked when an object is created in a monitored folder and the counterpart asset is created in WebCenter Sites.
Asset deletion	CIPAssetDeleted. Invoked when an object is deleted from a monitored folder and the counterpart asset is deleted from WebCenter Sites.
Asset deletion failure	CIPAssetDeletionFailed. Invoked when: An object that was deleted from the monitored folder is checked out on the WebCenter Sites system. An object that was deleted from the monitored folder has dependencies that would become unresolved on the WebCenter Sites system if the counterpart asset were to be deleted.
Asset modification	CIPAssetModified, invoked when an object in the monitored folder is modified and the counterpart asset is created in WebCenter Sites.
Asset modification failure	CIPAssetModificationFailed, invoked when an object in the monitored folder is modified, but its counterpart asset is checked out in WebCenter Sites.

Custom Workflows

Although CIP-related workflows can be created, in most cases it is more convenient to use the sample workflows provided with Content Integration Platform. If you wish to install sample workflows, continue reading this section. If you wish to create custom workflows, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

57.2 Sample Workflows

Sample workflows are available to use as the basis for modifying to use for your own particular needs. It can be an easier way to provide for workflows in your organization rather than constructing basic workflows from scratch..

Note: If you have already published to WebCenter Sites, install and enable only the following workflows for the published content: CIPAssetCreated, CIPAssetModified, and CIPAssetModificationFailed. The remaining workflows (CIPAssetDeleted and CIPAssetDeletionFailed) must be installed and enabled before an asset is published.

This section contains the following topics:

- [Section 57.2.1, "Installing Sample Workflows"](#)
- [Section 57.2.2, "Verifying Sample Workflows"](#)
- [Section 57.2.3, "Enabling Sample Workflows"](#)

57.2.1 Installing Sample Workflows

To install sample workflows

1. Run `catalogmover.bat` (or `catalogmover.sh` on Linux) from the WebCenter Sites installation directory.
2. Go to **Catalog, and then Auto Import Catalog(s)**.
 - Select `workflows.zip` (in the same directory or level as all `cs*_schema.zip` files).
 - In the import dialog, fill in the fields as shown below:
 Catalog Data Directory: Leave the default value
 Catalog ACL List: `Browser, SiteGod, xceleeditor, xceladmin`
3. Create the sample workflows by invoking the following URL:

```
http://<host>:<port>/<context_path>/ContentServer?pagename=OpenMarket/Xcelerate/Installation/CIPCreateWorkflows&username=<username>&<password>=<password>
```

where:

- `<host>` is the address of the WebCenter Sites installation
- `<port>` is the port of the WebCenter Sites installation
- `<context_path>` is the context path where the WebCenter Sites web application is deployed
- `<username>` is the WebCenter Sites administrator's user name
- `<password>` is the WebCenter Sites administrator's password

For example, the URL of the default configuration is:

```
http://localhost:8080/cs/ContentServer?pagename=OpenMarket/Xcelerate/Installation/CIPCreateWorkflows&username=fwadmin&password=xceladmin
```

When the workflows are installed, the following message will be displayed:

```
"Workflows for Content Integration Platform were created successfully"
```

57.2.2 Verifying Sample Workflows

When the sample workflows are created, associated objects are also created in WebCenter Sites.

To verify the sample workflows and associated objects

1. Log in to the WebCenter Sites Advanced interface as an administrator.
2. Verify that the following objects have been created:
 - CIPAdmin role, which will be used as the management role for all CIP workflows. All users with the CIPAdmin role will be notified about all CIP events in the sample workflows.
 - Workflow processes:
CIP Asset Created, CIP Asset Deleted, CIP Asset Deletion Failed, CIP Asset Modified, and CIP Asset ModificationFailed
 - Workflow states:
CIP Asset Created, CIP Asset Deleted, CIP Asset Deletion Failed, CIP Asset Modified, and CIP Asset Modification Failed
 - Workflow step action:
CIP Asset Deleted, which results in an email notice to the CIP administrators.
 - Email object:
CIP Asset Event

57.2.3 Enabling Sample Workflows

Sample workflows are pre-configured in the default `mappings.xml` file. Each asset type that is listed in the default `mappings.xml` file contains a commented workflow configuration section.

To enable a CIP workflow

1. Uncomment the following section for each asset type that you wish to enable for event notification:

```
<descriptor-mapping sourceid="assetCreatedProcess"
  targetid="CIPAssetCreated" type="static" />
<descriptor-mapping sourceid="assetModifiedProcess"
  targetid="CIPAssetModified" type="static" />
<descriptor-mapping sourceid="assetDeletedProcess"
  targetid="CIPAssetDeleted" type="static" />
<descriptor-mapping sourceid="assetDeletionFailedProcess"
  targetid="CIPAssetDeletionFailed" type="static" />
```

2. Assign the CIPAdmin role to CIP administrators. Ensure that CIP administrators are able to receive email. For instructions, see [Section 5.5.1, "Creating and Editing a User Profile."](#)
3. If the number of events occurring on the source system is relatively large, it is best to use workflow groups, as they will allow you to resolve tasks in bulk. Workflow groups are not packaged by default. They must be created manually. For

instructions on creating workflow groups, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

Note: If a workflow group has the name of the invoked workflow process, the workflow process will be automatically added to the group.

Sample workflows behave in the following way:

- For creation, deletion failure, modification, and modification failure events, a task is assigned to all WebCenter Sites users with the CIPAdmin role. The task is simply a way of notifying the users of events and their propagation to the target system. The task can be removed; there is no obligation to take a step.
- When an event occurs, only the first step of the corresponding workflow is taken. If the option **Assign from list of participants** for the first step is chosen, all members of the selected roles will be assigned the next task. (For simplicity, notification workflows comprise one state and two steps.)
- For deletion events, all members with the CIPAdmin role receive an email notice. Because the asset no longer exists once the deletion event occurs, no tasks are displayed.

If you plan to publish from a source system whose schema differs from the default schema defined by the Content Integration Platform, you will have to update the flex family corresponding to the source system and possibly `mappings.xml`, depending on how the schema differ.

This chapter contains the following sections:

- [Section 58.1, "When to Remap"](#)
- [Section 58.2, "Remapping Procedures"](#)

58.1 When to Remap

A commonly made schema change is adding a new attribute or attributes. For remapping procedures for adding a new attribute, see [Section 58.2.1, "File Systems"](#) or [Section 58.2.2, "Microsoft SharePoint Systems"](#) as necessary.

Note: Procedures in the rest of this chapter call for updates to the default `mappings.xml` file and the default flex families, both supplied as part of the Content Integration Platform. If you wish to create your own flex family, see the *Oracle Fusion Middleware WebCenter Sites Developer's Guide* for instructions. When creating your own flex family, keep in mind the source system's default flex family as a model. Basic procedures for remapping to a custom flex family remain the same (except for the names you choose for the family and its members). If you create flex filters (for either a default flex family or a custom flex family), makes sure to add the corresponding `jar` files to both the WebCenter Sites and the Sites Agent Services applications.

58.2 Remapping Procedures

This section contains the following topics:

- [Section 58.2.1, "File Systems"](#)
- [Section 58.2.2, "Microsoft SharePoint Systems"](#)

58.2.1 File Systems

If you create new attributes for documents of type `Document` and wish to propagate the attributes to WebCenter Sites, you must update the `FileSystem` flex family with the new attributes. You must also update `mappings.xml` if the new attributes will be named differently on the WebCenter Sites side. (A list of default attributes and

document types is available in [Chapter 60, "File Systems: Default Mapping Specifications."](#)

To update the flex family and mappings.xml

1. Stop the Content Integration Agent.
2. Follow the remapping procedures in [Table 58–1, " Remapping File System Schema"](#). For descriptions of tags and parameters, see [Table 58–2, " Mapping File System Data to Oracle WebCenter Sites"](#).
3. Test your changes by running a publishing session. For instructions, see [Chapter 56, "Publishing."](#)

Table 58–1 Remapping File System Schema

File System	WebCenter Sites	Content Integration Agent: mappings.xml
If you add a new attribute.	<p>Create a FileSystem Attribute asset.</p> <p>Create the attribute in the FileSystem flex family and assign the attribute to the relevant asset type definition. For instructions, see Section 59.1, "Adding a New Attribute" and take note of the Value fields in step 5.</p>	<p>Map the new attribute in mappings.xml <i>only</i> if the attribute is named differently on the WebCenter Sites side. Map the attribute within the <descriptor-mapping> tag, and nest the tag in the relevant object type mapping (example below).</p>
For example, CreationDate	Example fs_CreationDate	<p>In this example, the CreationDate attribute belongs to the Document file type. Nest the attribute in the Document type mapping.</p> <pre data-bbox="902 993 1360 1192"><assettype-mappingsourceid="Document" targetid="FileSystem_Document;fs_ document" id="fs_ document"><descriptor-mappingsourceid ="CreationDate"targetid="fs_ CreationDate" type="static" /> </assettype-mapping></pre>

Tags and Parameters for Mapping File System Data to Oracle WebCenter Sites

```
<assettype-mappingsourceid="FileType" targetid="ChildAssetType|ParentAssetType;
ChildDefinitionInstance|ParentDefinitionInstance" id="assettypeMappingID">
  <descriptor-mappingsourceid="FileSystemAttribute" targetid="CSAttribute"/>
</assettype-mapping>
```

Table 58–2 Mapping File System Data to Oracle WebCenter Sites

Tag	Parameter	Parameter Description
asstype-mapping	sourceid	Specifies the type of document that will be published. Legal values: Document Folder
asstype-mapping	targetid	WebCenter Sites' counterpart to sourceid. For document objects, targetid takes the following value: ChildAssetType;ChildDefinitionInstance <ul style="list-style-type: none"> ChildAssetType : Stores file system documents. Default value: FileSystem_Document <ul style="list-style-type: none"> ChildDefinitionInstance : <ul style="list-style-type: none"> Asset of type FileSystem Child Definition. Defines the type of document that will be stored in ChildAssetType. Default value: fs_document Sample value: pdfDoc Example: If sourceid="pdfDoc" then targetid could be targetid="FileSystem_Document;fs_pdfDoc" that is, the (child) asset type named FileSystem_Document stores documents of type fs_pdfDoc. Note: If you were to map a folder type, targetid would take the following value: ParentAssetType;ParentDefinitionInstance <ul style="list-style-type: none"> ParentAssetType : <ul style="list-style-type: none"> Stores file system folders. Default value: FileSystem_Folder <ul style="list-style-type: none"> ParentDefinitionInstance : <ul style="list-style-type: none"> Asset of type FileSystem Parent Definition. Defines the type of folder that will be stored in ParentAssetType. Default value: fs_folder For more information about the FileSystem flex family, see Chapter 60, "File Systems: Default Mapping Specifications."
asstype-mapping	id	Unique value that identifies the asset type mapping.
descriptor-mapping	sourceid	Attribute's name in the file system.
descriptor-mapping	targetid	Attribute's name in WebCenter Sites.

58.2.2 Microsoft SharePoint Systems

If you add new attributes and/or document types for the documents you will publish, you must update the SharePoint flex family accordingly. You may also have to update mappings.xml, as explained in this section. (A list of default attributes and document types is available in [Chapter 60, "File Systems: Default Mapping Specifications."](#))

To modify the flex family and mappings.xml

1. Stop the Content Integration Agent.
2. Follow the remapping procedures in [Table 58–3, "Remapping Microsoft SharePoint Schema"](#). For descriptions of tags and parameters, see [Table 58–4, "Mapping Microsoft SharePoint Data to Oracle WebCenter Sites"](#)

- Test your changes by running a publishing session. For instructions, see [Chapter 56, "Publishing."](#)

Table 58–3 Remapping Microsoft SharePoint Schema

MS SharePoint	WebCenter Sites	Content Integration Agent: mappings.xml
If you add a new attribute.	Create a SharePoint Attribute asset. Create the attribute in the SharePoint flex family and assign the attribute to the relevant asset type definition. For instructions, see Section 59.1, "Adding a New Attribute" and take note of the Value fields in step 5.	Map the new attribute in mappings.xml only if the attribute is named differently on the WebCenter Sites side. Map the attribute within the <descriptor-mapping> tag, and nest the tag in the relevant object type mapping (example below).
For example, CreationDate	sp_CreationDate	In this example, the CreationDate attribute belongs to the Document content type. Nest the attribute in the Document type definition. <pre><assettype-mapping sourceid="Document" targetid="SharePoint_Document;sp_ document" id="sp_PDF" extends="sp_ item"><descriptor-mappingsourceid="Creati onDate"targetid="sp_ CreationDate" /></assettype-mapping></pre>
If you add a new document type.	Create a SharePoint Child Definition asset. For instructions on creating child definition assets, see Chapter 59, "Adding Assets to a Flex Family."	Map the new document type in mappings.xml, using the <assettype-mapping> tag.
For example, PDF	sp_PDF	<pre><assettype-mapping sourceid="PDF" targetid="SharePoint_Document;sp_ PDF" id="sp_PDF" extends="sp_ item"></assettype-mapping></pre>

Table 58–3 (Cont.) Remapping Microsoft SharePoint Schema

MS SharePoint	WebCenter Sites	Content Integration Agent: mappings.xml
If you add a new attribute and new document type.	<ol style="list-style-type: none"> 1. Create a SharePoint Attribute asset. Create the attribute in the SharePoint flex family. For instructions, see Chapter 59, "Adding Assets to a Flex Family" and take note of the Value fields in step 5 of Section 59.1, "Adding a New Attribute." 2. Create a SharePoint Child Definition asset. When creating the child definition asset, assign the new attribute to the asset. For instructions, see Chapter 59, "Adding Assets to a Flex Family." 	<ol style="list-style-type: none"> 1. Map the new document type in mappings.xml, using the <assettype-mapping> tag (example below). 2. Map the new attribute in mappings.xml only if the attribute is named differently on the WebCenter Sites side. Map the attribute within the <descriptor-mapping> tag, and nest the tag in the relevant object type mapping (example below).
For example, Attribute: CreationDate New Document type: PDF	Attribute: sp_CreationDate New Document type: sp_PDF	<pre><assettype-mapping sourceid="PDF" targetid="SharePoint_ Document;sp_PDF" id="sp_PDF" extends="sp_item"> <descriptor-mapping sourceid="CreationDate" targetid="sp_CreationDate" /> </assettype-mapping></pre>
If you add a new picture type.	Mapping a new picture type is analogous to mapping a document type. The targetid parameter takes the value SharePoint_Document; <ChildDefinitionInstance>	

58.2.2.1 Tags and Parameters for Mapping Microsoft SharePoint Data to Oracle WebCenter Sites

```
<assettype-mappingsourceid="SharePointContentType"
targetid="ChildAssetType|ParentAssetType;
ChildDefinitionInstance|ParentDefinitionInstance"
id="assettypeMappingID" extends="sp_item">
<descriptor-mappingsourceid="SharePointAttribute" targetid="CSAttribute"/>
</assettype-mapping>
```

Table 58–4 Mapping Microsoft SharePoint Data to Oracle WebCenter Sites

Tag	Parameter	Parameter Description
assettype-mapping	sourceid	Specifies the type of content that will be published. Default value: Document Picture Folder
assettype-mapping	targetid	WebCenter Sites' counterpart to sourceid. For document and picture objects, targetid takes the following value: ChildAssetType;ChildDefinitionInstance ChildAssetType: Stores SharePoint documents and pictures. Default value: SharePoint_Document ChildDefinitionInstance Asset of type SharePoint Child Definition. Defines the type of document (or picture) that will be stored in ChildAssetType. Default value: sp_document sp_picture Sample values: pdfDoc Example: If sourceid="pdfDoc", then targetid could be targetid="SharePoint_Document;sp_pdfDoc" that is, the (child) asset type named SharePoint_Document stores documents of type sp_pdfDoc. Note: If you were to map a folder type, targetid would take the following value: ParentAssetType;ParentDefinitionInstance ParentAssetType: Stores SharePoint folders Default value: SharePoint_Folder ParentDefinitionInstance: Asset of type SharePoint Parent Definition. Defines the type of folder that will be stored in ParentAssetType. Default value: sp_folder For more information about the SharePoint flex family, see Chapter 60, "File Systems: Default Mapping Specifications."
assettype-mapping	id	Unique value that identifies the asset type mapping. By default, assettypeMappingID takes the following values: sp_document sp_picture sp_folder The extends parameter is a pointer to the mapping id for the next higher-level object in the schema hierarchy. Because the Microsoft SharePoint content model requires that content types always have item defined as one of their top-level objects, the extends parameter will always be used, although it is not technically a required field. Including the extends parameter allows attributes defined in parent types to be inherited by child types, starting at the item level. By default: extends=sp_item
descriptor-mapping	sourceid	Attribute's name in the Microsoft SharePoint system.
descriptor-mapping	targetid	Attribute's name in WebCenter Sites.

Adding Assets to a Flex Family

This chapter contains procedures that support the remapping steps in [Chapter 58, "Remapping."](#)

This chapter contains the following sections:

- [Section 59.1, "Adding a New Attribute"](#)
- [Section 59.2, "Adding a New Child Definition Asset"](#)

59.1 Adding a New Attribute

If you define new attributes on the source system and associate them with an object that is or will be published, you must create the corresponding attributes in WebCenter Sites. Otherwise, objects associated with the attributes cannot be properly synchronized (or published).

To add a new attribute to WebCenter Sites

1. Log in to the WebCenter Sites Admin interface as an administrator.
2. Select the site in which the flex family for your source system is enabled.
3. Select **New** from the top navigation bar.
4. Select **New *SourceSystem* Attribute** from the list of options.
5. When filling in fields in the *SourceSystem* Attribute form, fill in the following fields as explained below:
 - **Value Type:** Select a value type that is as close as possible to the corresponding attribute's data type on the source system. Although you can map any data type to a string, you gain flexibility by mapping to a similar data type. (For example, a date for a given locale cannot be modified for any other locale unless it is mapped to the date type.) If you are using Microsoft SharePoint, use [Table 59-1, "Suggested Conversions for SharePoint Attributes"](#) for suggested conversions.
 - **Number of Values:** If you are using Microsoft SharePoint and re-creating a multivalued attribute, select the multivalued option. For file systems, all attributes are single-valued.
6. Save the attribute.
7. Assign the attribute to the relevant flex parent definition or flex child definition.

Figure 59–1 SharePoint Attributes

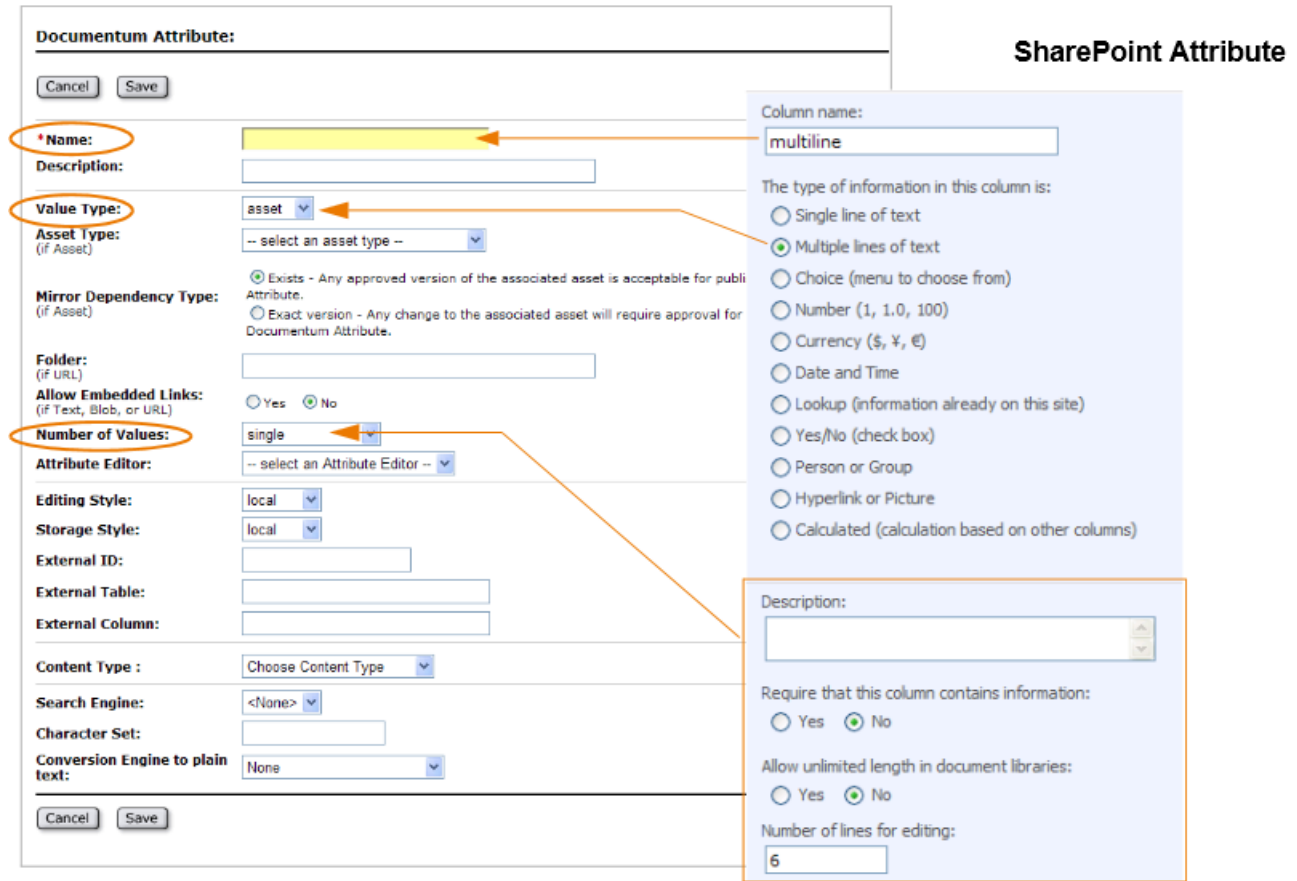


Table 59–1 Suggested Conversions for SharePoint Attributes

SharePoint Attribute Type	WebCenter Sites Attribute Type	Multi-valued
Single line of text	string / text	N/A
Multiple lines of text	string / text	N/A
Choice (menu to choose from)	string	single- or multi-valued, depending on selection
Number (1, 1.0, 100)	float	N/A
Currency (\$, ¥, i)	money	N/A
Date and Time	date	N/A
Lookup (information already on this site)	string	single- or multi-valued, depending on selection
Yes/No (check box)	string	N/A
Person or Group	string	single- or multi-valued, depending on selection
Hyperlink or Picture	string	N/A
Calculated (calculation based on other columns)	string	N/A

59.2 Adding a New Child Definition Asset

Note: If you are creating a child definition asset with new attributes, create the attributes first (as shown below). You will assign them to the child definition asset during the procedure below.

To add a new child definition asset to WebCenter Sites

1. Log in to the WebCenter Sites Admin interface as an administrator.
2. Select the site in which the flex family is enabled.
3. Select **New** from the top navigation bar.
4. Select **New *SourceSystem* Child Definition** from the list of options and fill in the fields.
5. Save the asset.

File Systems: Default Mapping Specifications

The default mapping framework in the Content Integration Platform enables publishing from file systems, as long as the publishable content is based on the following default constructs of the file system: `Folder` and `Document` file types, and attributes `contentAttr`, `contentURL`, and `FileSize`.

This chapter contains the following sections:

- [Section 60.1, "Overview"](#)
- [Section 60.2, "Default mappings.xml"](#)
- [Section 60.3, "FileSystem Flex Family Specifications"](#)

60.1 Overview

The mapping framework supplies the following default components:

- The `FileSystem` flex family, pre-configured to match the file types and attributes listed above.
- A `mappings.xml` file, in which file types and attributes (listed above) are mapped to assets in the `FileSystem` flex family:
 - The `Folder` file type is mapped to a flex parent definition asset named `fs_folder`.
 - The `Document` file type is mapped to a flex definition asset named `fs_document`.
 - Attributes are mapped to flex assets of type `FileSystem Attribute`.

The mapping is coded in `mappings.xml`, and summarized in [Table 60–1, "FileSystem Default Data and Flex Family Analogs"](#) (for more information, see [Section 60.2, "Default mappings.xml"](#)).

Once the mapping is established, folders of type `Folder` (and their contents) can be published:

- Folders are published as flex parent assets to the `FileSystem Folder` asset type.
- Documents are published as flex assets to the `FileSystem Document` asset type.

During publishing, the Content Integration Platform refers to the `mappings.xml` file to determine the types of objects to publish (`Folder` and `Document`). The folder that is named in the `publish` command is the starting point of the publication process. The

folder is published as a flex parent asset of the `FileSystem Folder`, along with all the subfolders and documents it contains.

To reproduce the folder's structure, (subfolders and documents), the Content Integration Platform refers to path information. If subfolders exist, the Content Integration Platform chains their counterpart `FileSystem Folder` assets to reproduce the hierarchy. Documents, treated as flex assets of file type `FileSystem Document`, are placed under their respective `FileSystem Folder` parent assets.

Figure 60–1 Mapping the Document Type to the FileSystem Flex Family

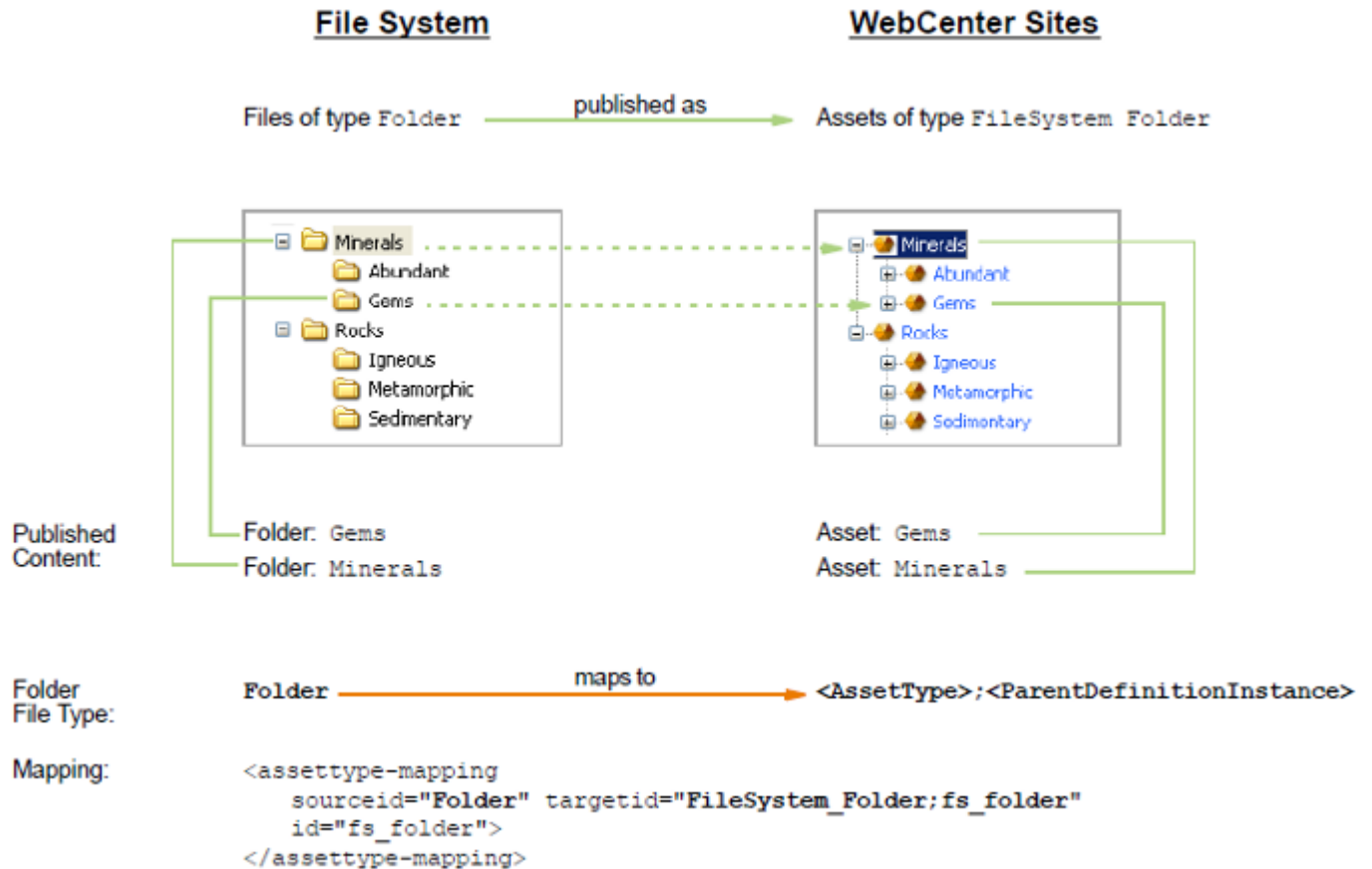
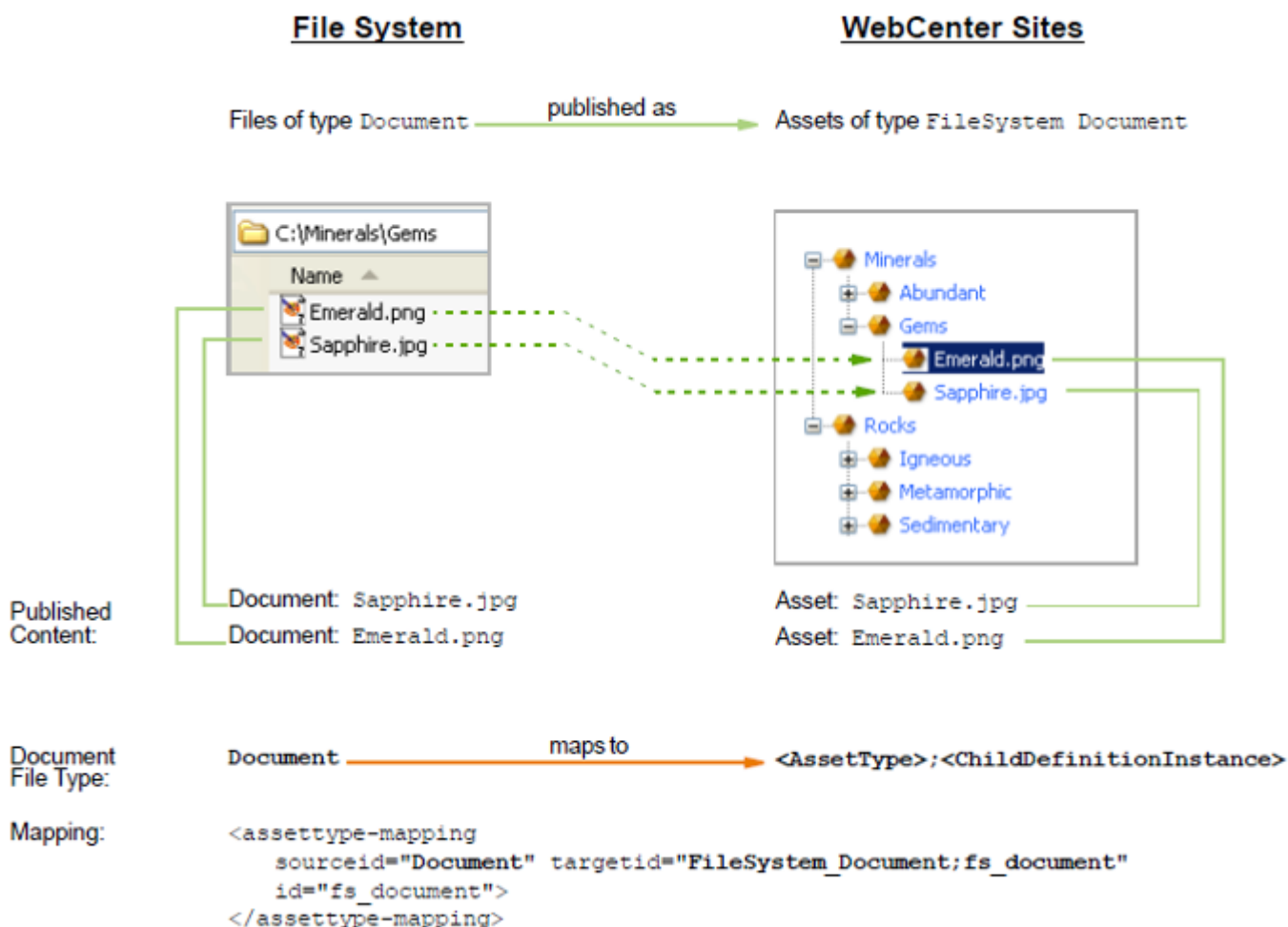


Figure 60–2 Mapping the Document Type to the FileSystem Flex Family



60.2 Default mappings.xml

The default mappings.xml file, located in the Content Integration Agent, maps the file system's default schema to WebCenter Sites' FileSystem flex family. The mapping is uniquely identified by the id attribute, specifically in this example as <mapping id="filesystem2cs">. Parameters for the <assettype-mapping> and <descriptor-mapping> tags are defined in "Tags and Parameters for Mapping File System Data to Oracle WebCenter Sites" on page 58-2.

```
<mappings>
<!-- File System to CS -->
<mapping id="filesystem2cs">
  <assettype-mapping
    sourceid="Document"
    targetid="FileSystem_Document;fs_document"
    id="fs_document">
  <descriptor-mapping
    sourceid="contentAttr"
    targetid="file" type="static" />
  <descriptor-mapping
    sourceid="contentURL"
    targetid="contentURL" type="dynamic" />
</mapping>
</mappings>
```

```

    <descriptor-mapping
      sourceid="FileSize"
      targetid="file_size" />
  </assettype-mapping>
  <assettype-mapping
    sourceid="Folder"
    targetid="FileSystem_Folder;fs_folder"
    id="fs_folder" />
</mapping>
</mappings>

```

60.3 FileSystem Flex Family Specifications

Table 60–1, "FileSystem Default Data and Flex Family Analogs" summarizes the default mapping of file system schema to WebCenter Sites' FileSystem flex family. For customized implementations, you can either re-use the flex family or create your own.

Table 60–1 *FileSystem Default Data and Flex Family Analogs*

Type of Data	File System Default Data	Maps To FlexAssetType	Maps to Flex Family Member	Description	Assets of This Type Are Created By
Schema	Attribute Attributes (for documents) ¹ : contentURL contentAttr FileSize	FileSystem Attribute Stores attribute instances: ² content URL file file_size	Flex Attribute	This flex asset type stores attributes for documents.	WebCenter Sites Administrator
Schema	Folder File Type Folder	FileSystem Parent Definition Stores parent definition instance: ³ fs_folder	Flex Parent Definition	This flex asset type stores parent definition instances.	WebCenter Sites Administrator
Schema	Document File Type Document	FileSystem Child Definition Stores child definition instance: fs_document	Flex (Child) Definition	This flex asset type stores child definition instances.	WebCenter Sites Administrator
Content	Published Folders Folders of file type Folder	FileSystem Folder Stores flex parent assets (of type fs_folder and any other type)	Flex Parent	This flex asset type stores folder assets.	Content Integration Platform

Table 60–1 (Cont.) FileSystem Default Data and Flex Family Analogs

Type of Data	File System Default Data	Maps To FlexAssetType	Maps to Flex Family Member	Description	Assets of This Type Are Created By
Content	Published Documents Documents of file type Document	FileSystem Document Stores flex (child) assets (of type fs_ document and any other type)	Flex (Child) Asset	This flex asset type stores document assets. For an example, see Figure 60–1 .	Content Integration Platform

¹ Attribute names are display names. The less commonly used system-defined attributes have been omitted from the default mappings.xml file and flex family. The attributes are: DateCreated, DateModified, MimeType.

² "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

³ "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

Microsoft SharePoint Systems: Default Mapping Specifications

The default mapping framework in the Content Integration Platform enables publishing from SharePoint libraries on selected sites.

This chapter contains the following sections:

- [Section 61.1, "Overview"](#)
- [Section 61.2, "Default mappings.xml"](#)
- [Section 61.3, "SharePoint Flex Family Specifications"](#)

61.1 Overview

The default mapping framework in the Content Integration Platform enables publishing from SharePoint libraries on selected sites, as long as the publishable content is based on the following default SharePoint constructs: Content types Document, Folder, and Picture; and attributes File Size, contentAttr, contentURL, Date Picture Taken, Description, Keywords, Picture Height, and Picture Width.

The mapping framework supplies the following default components:

- The SharePoint flex family, pre-configured to match the content types and attributes listed above.
- A mappings.xml file, in which content types and attributes (listed above) are mapped to assets in the SharePoint flex family:
 - The Folder type is mapped to a flex parent definition asset named sp_folder.
 - The Document type is mapped to a flex definition asset named sp_document.
 - The Picture type is mapped to a flex definition asset named sp_picture.
 - Attributes are mapped to flex assets of type SharePoint Attribute.

The mappings are coded in mappings.xml in [Section 61.2, "Default mappings.xml"](#) and summarized in [Table 61–1, "SharePoint Default Data and Flex Family Analogs"](#).

Once the mapping is established, folders of type Folder (and their contents) can be published:

- Folders are published as flex parent assets to the SharePoint Folder asset type.
- Documents and pictures are published as flex assets to the SharePoint Document asset type.

During publishing, the Content Integration Platform refers to the mappings.xml file to determine the types of content to publish (sp folder, sp document, and sp picture). The folder that is named in the publish command is the starting point of the publication process. The folder is published as a flex parent asset of type SharePoint Folder, along with all the subfolders, documents, and pictures it contains.

To reproduce the folder's structure (subfolders, documents, and pictures), the Content Integration Platform refers to path information:

- If subfolders exist, the Content Integration Platform chains their corresponding SharePoint Folder assets to reproduce the hierarchy.
- Documents, treated as SharePoint Document assets, are placed under their respective SharePoint Folder parent assets.
- Pictures, treated as SharePoint Picture assets, are placed under their respective SharePoint Folder parent assets.

Figure 61-1 Mapping the Document Type to WebCenter Sites' SharePoint Flex Family

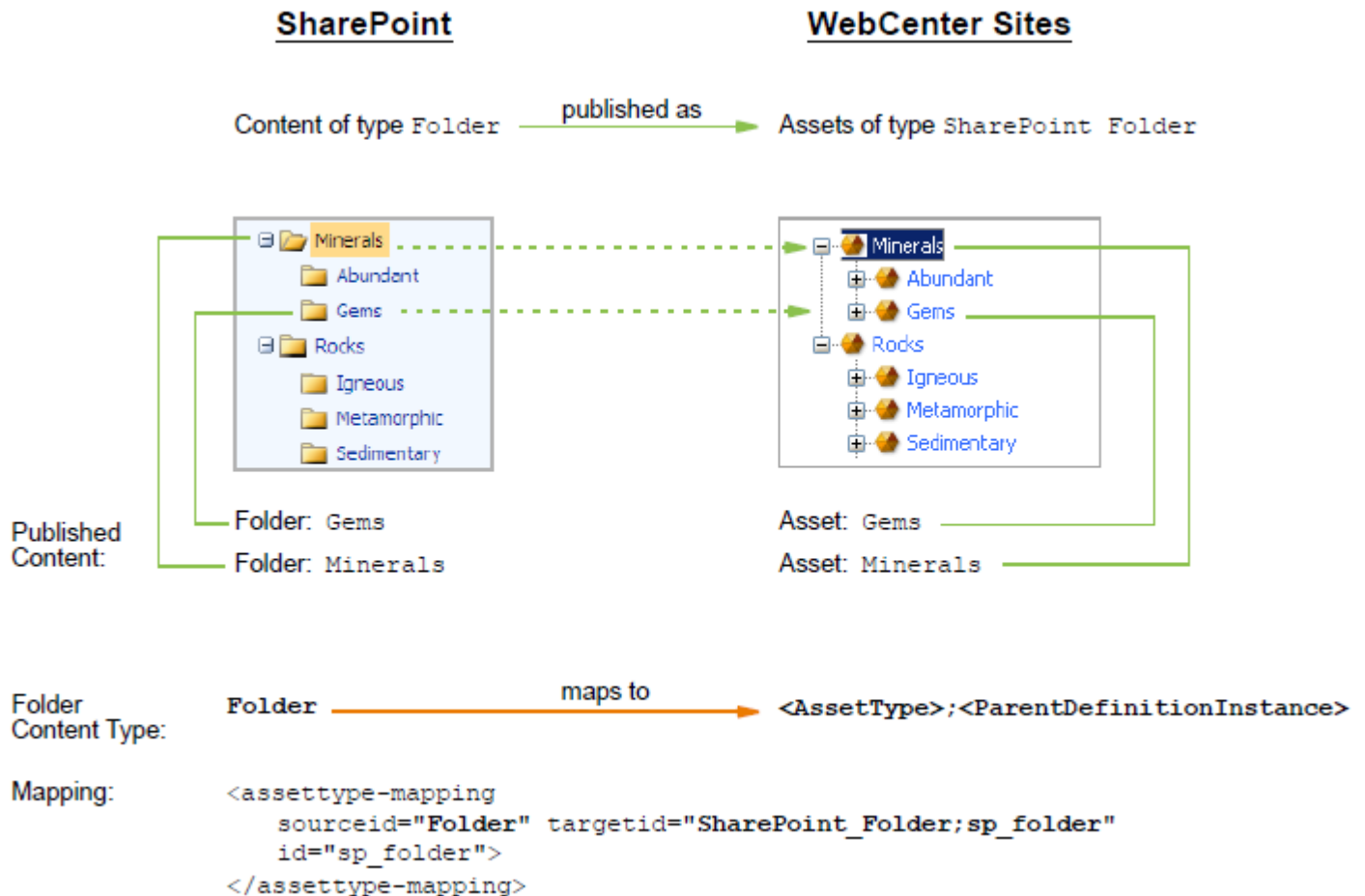
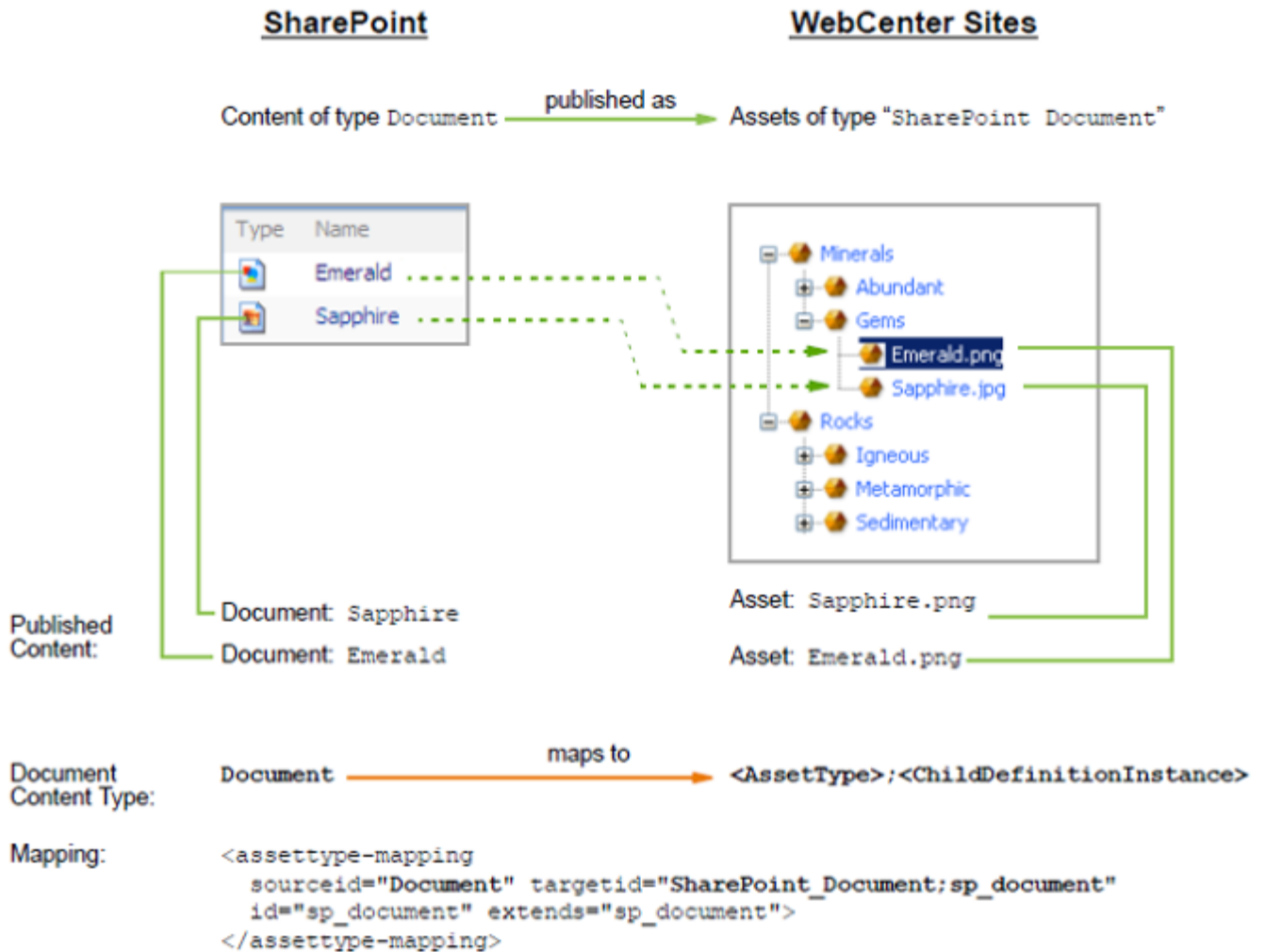


Figure 61–2 Mapping the Document Type to WebCenter Sites' SharePoint Flex Family



Note: The Picture content type is mapped similarly to the Document content type:

sourceid takes the value Picture

targetid takes the value SharePoint_Document;sp_picture

For more information, see [Section 61.2, "Default mappings.xml."](#)

61.2 Default mappings.xml

The default mappings.xml file, located in the Content Integration Agent, maps SharePoint schema to WebCenter Sites' SharePoint flex family. The mapping is uniquely identified by the ID as shown in the code below. Parameters for the <assettype-mapping> and <descriptor-mapping> tags are defined in [Section 58.2.2.1, "Tags and Parameters for Mapping Microsoft SharePoint Data to Oracle WebCenter Sites."](#)

```
<mappings>
  <!-- MS SharePoint to CS -->
```

```

<mapping id="sharepoint2cs">
  <assettype-mapping sourceid="Item" targetid="" id="sp_item">
    <descriptor-mapping sourceid="Title" targetid="title" />
  </assettype-mapping>
  <assettype-mapping sourceid="Document"
    targetid="SharePoint_Document;sp_document" id="sp_document"
    extends="sp_item">
    <descriptor-mapping sourceid="File Size" targetid="file_size" />
    <descriptor-mapping sourceid="contentAttr" targetid="file" type="static" />
    <descriptor-mapping sourceid="contentURL" targetid="contentURL"
      type="dynamic" />
  </assettype-mapping>
  <assettype-mapping sourceid="Folder" targetid="SharePoint_Folder;sp_folder"
    id="sp_folder" extends="sp_item" />
  <assettype-mapping sourceid="Picture"
    targetid="SharePoint_Document;sp_picture" id="sp_picture"
    extends="sp_document">
    <descriptor-mapping sourceid="Date Picture Taken" targetid="date_taken" />
    <descriptor-mapping sourceid="Description" targetid="description" />
    <descriptor-mapping sourceid="Keywords" targetid="keywords" />
    <descriptor-mapping sourceid="Picture Height" targetid="picture_height" />
    <descriptor-mapping sourceid="Picture Width" targetid="picture_width" />
  </assettype-mapping>
</mapping>
</mappings>

```

61.3 SharePoint Flex Family Specifications

Table 61–1, "SharePoint Default Data and Flex Family Analogs" summarizes the default mapping of SharePoint schema to WebCenter Sites' SharePoint flex family. For customized implementations, you can either re-use the flex family or create your own.

Table 61–1 *SharePoint Default Data and Flex Family Analogs*

Type of Data	SharePoint Default Data	Maps To: Flex Asset Type	Maps To: Flex Family Member	Description	Assets of This Type Are Created By
Schema	Attribute	SharePoint Attribute	Flex Attribute	This flex asset type stores attributes for documents and pictures.	WebCenter Sites Administrator
	Attributes:	Stores attribute instances:			
	For documents:	File Size			
	contentAttr	file_size			
	contentURL	file			
	For pictures:	contentURL			
	Date Picture Taken	date_taken			
	Description	description			
	Keywords	keywords			
	Picture Height	picture_height			
	Picture Width	picture_width			

Table 61–1 (Cont.) SharePoint Default Data and Flex Family Analogs

Type of Data	SharePoint Default Data	Maps To: Flex Asset Type	Maps To: Flex Family Member	Description	Assets of This Type Are Created By
Schema	Folder Content Type Folder	SharePoint Parent Definition Stores parent definition instance: ² sp_folder	Flex Parent Definition	This flex asset type stores parent definition instances.	WebCenter Sites Administrator
Schema	Document Content Type Document	SharePoint Child Definition Stores child definition instance: ³ sp_document	Flex (Child) Definition	This flex asset type stores child definition instances.	WebCenter Sites Administrator
Schema	Picture Content Type Picture	SharePoint Child Definition Stores child definition instance: ⁴ sp_picture			
Content	Published Folders Folders of content type Folder	SharePoint Folder Stores flex parent assets (of type sp_folder and any other type)	Flex Parent	This flex asset type stores folder assets.	Content Integration Platform
Content	Published Documents Documents of content type Document Published Pictures Pictures of content type Picture	SharePoint Document Stores flex (child) assets of type sp_document and sp_picture, and any other type.	Flex (Child) Asset	This flex asset type stores document assets and picture assets. For an example, see Figure 61–1 .	Content Integration Platform

¹ "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

² "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

³ "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

⁴ "Instance" means "asset." "Instance" is used only to help differentiate metadata (instances) from published content (assets).

Part IX

Implementing Oracle WebCenter Sites Content Integration Platform for EMC Documentum

This part of the documentations contains procedures for configuring and using Oracle WebCenter Sites: Content Integration Platform (CIP) for EMC Documentum to transfer content between Oracle WebCenter Sites and EMC Documentum. Supported modes include publishing Documentum objects to WebCenter Sites and archiving the assets of WebCenter Sites to EMC Documentum.

This part contains the following chapters:

- [Chapter 62, "Publishing to Oracle WebCenter Sites"](#)
- [Chapter 63, "Archiving to EMC Documentum"](#)
- [Chapter 64, "Configuring Event Notification"](#)
- [Chapter 65, "Default Mapping Specifications for Publishing"](#)

Publishing to Oracle WebCenter Sites

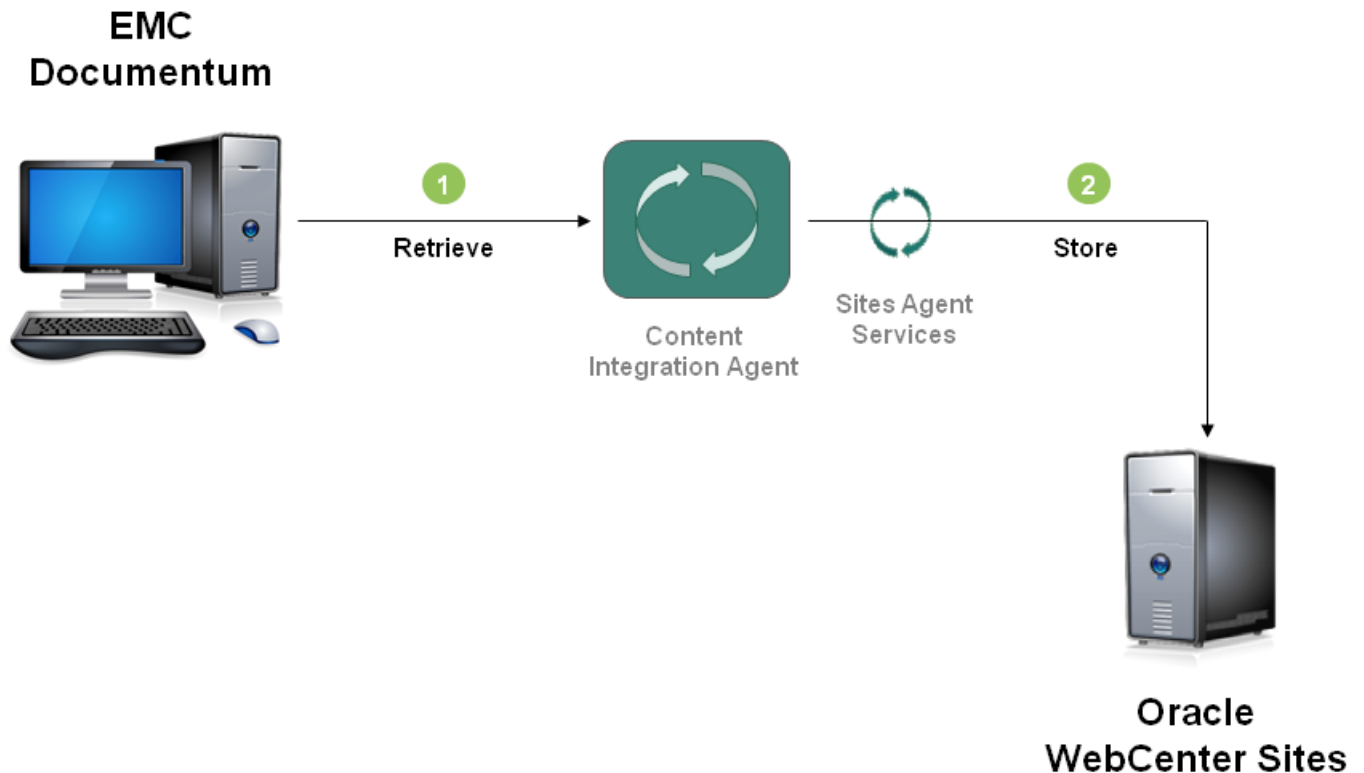
This chapter provides instructions on publishing from EMC Documentum to Oracle WebCenter Sites.

This chapter contains the following sections:

- [Section 62.1, "Overview of the Publishing Process"](#)
- [Section 62.2, "Publishing Procedures"](#)
- [Section 62.3, "Maintaining the Integrated Systems"](#)

62.1 Overview of the Publishing Process

Publishing from EMC Documentum to Oracle WebCenter Sites requires the CIP components Content Integration Agent and Sites Agent Services, shown in [Figure 62-1, "System Architecture for Publishing to Oracle WebCenter Sites"](#).

Figure 62–1 System Architecture for Publishing to Oracle WebCenter Sites

This section contains the following topics:

- [Section 62.1.1, "System Architecture and Process Flow"](#)
- [Section 62.1.2, "Mapping Framework"](#)

62.1.1 System Architecture and Process Flow

Content Integration Agent is used to synchronize the source and target workspaces via the `cipcommander publish` command, the synchronization engine, and the `mappings.xml` file, which provides the metadata map. Sites Agent Services exposes the web interface used by Content Integration Agent to perform the synchronization process. Following a publishing session, the synchronization process runs automatically. Details of the implementation are described below.

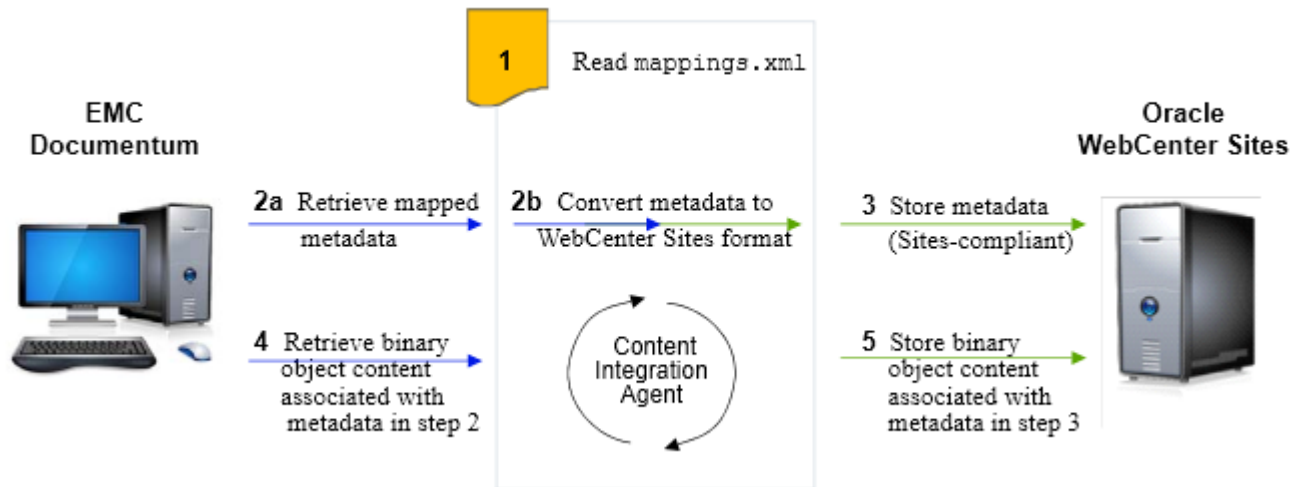
Initial Synchronization

When the `cipcommander publish` command is issued, the synchronization engine initializes the source and target by replicating metadata and associated content in this manner:

1. The synchronization engine reads the `mappings.xml` file.
2. The synchronization engine refers to the Documentum workspace (that is, cabinet or folder) named in the `cipcommander publish` command.
 - it reads the workspace's objects to retrieve their mapped metadata
 - it converts the metadata to WebCenter Sites-compliant format, using the `mappings.xml` file

3. The synchronization engine stores the WebCenter Sites-compliant metadata to WebCenter Sites (via Sites Agent Services).
4. The synchronization engine retrieves from Documentum the content of binary objects associated with the metadata.
5. The synchronization engine stores the objects (via Sites Agent Services) to the target flex family.

Figure 62–2 Publishing Process



Monitoring the Data Source

Following the initial synchronization, the synchronization engine starts monitoring the published cabinet or folder and automatically replicates changes to WebCenter Sites (via Sites Agent Services). For example, when an object based on the published metadata is created, modified, or deleted in the monitored cabinet or folder, the synchronization engine replicates the new or modified object, or the object's deletion to WebCenter Sites. If metadata schema is modified, the workspace must be republished.

Tuning the Integration

Content Integration Agent contains a configuration file named `catalog.xml`, which stores information about the publishing session. Various parameters in the file can be tuned once the synchronization process is initialized. When objects are unpublished, their information is deleted from `catalog.xml`. For more information about tuning, see [Section 62.3.1, "Tuning the Synchronization Process."](#)

62.1.2 Mapping Framework

The CIP mapping framework determines the success of the publishing and synchronization processes. Documentum objects can be published to WebCenter Sites as long as their metadata is mapped. The basic mapping framework for publishing involves two configurable components: the Documentum flex family and the `mappings.xml` file.

Documentum Flex Family

The Documentum flex family, provided with CIP, stores published objects and their object type definitions as WebCenter Sites assets. [Chapter 65, "Default Mapping Specifications for Publishing"](#) provides flex family specifications.

For simplicity, we recommend using the default flex family. Should you need to create your own flex family, refer to instructions in the *Oracle Fusion Middleware WebCenter Sites Developer's Guide*. Use the information in [Chapter 65, "Default Mapping Specifications for Publishing"](#) of this guide as a model of the flex family.

mappings.xml

The default mappings.xml file contains a documentum2cs section, which specifies the mappings listed below:

- The dm_folder type maps to Documentum_Folder; dm_folder in the Documentum flex family, where
 - Documentum_Folder is a flex parent asset type that stores folder assets
 - dm_folder is a parent definition (of type Documentum Parent Definition) that defines the folder type
- The dm_document type maps to Documentum_Document; dm_document in the Documentum flex family, where
 - Documentum_Document is a flex asset type that stores document assets
 - dm_document is a child definition (of type Documentum Child Definition) that defines the document type
- Attributes are mapped in the <descriptor-mapping .../> tags. Attributes are named as listed under the **Types** node of the Documentum WebTop interface:
 - title
 - subject
 - keywords
 - r_version_label
 - r_full_content_size

Mapping, Publishing, and Synchronization

When publishing, bear in mind the following mapping specifications:

- Documentum objects based on default metadata can be published to WebCenter Sites without you having to modify either the default mappings.xml file or the Documentum flex family. Running the cipcommander publish command replicates the specified workspace and its subfolders to the Documentum flex family as flex parents; documents are replicated as flex child assets. The published workspace is then monitored by the synchronization engine.
- Publishing objects based on custom metadata (such as a new document type) requires you to first update at least the flex family with the new metadata. The mappings.xml file may or may not require updates, depending on the nature of the metadata. Details of custom mappings can be found in [Section 62.2.2, "Publishing via Customized Mappings."](#)

62.2 Publishing Procedures

This section contains the following topics:

- [Section 62.2.1, "Publishing via the Default Mapping"](#)
- [Section 62.2.2, "Publishing via Customized Mappings"](#)

62.2.1 Publishing via the Default Mapping

In this section, you will publish cabinets and folders to WebCenter Sites using the default mappings.xml file and Documentum flex family.

To publish via the Default Mapping

1. Start Content Integration Agent.

Note: If you changed the port in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*, make sure that the new port is set in facilities.xml, and add `-p <port>` to the cipcommander publish command in 3, below (which starts CIPCommander).

2. Run the CIPCommander executable (located in the bin folder of the system where Content Integration Agent is installed).
3. Publish objects of the types that are specified in the default mappings.xml file (for definitions of publishing parameters, see the table [Table 62-1, "Publishing Parameters"](#)):

```
cipcommanderpublish <source_providerid> <target_providerid>-source_rename
<cabinet_name>-source_path <path_in_cabinet>-target_rename <CS_content_
management_site>-mapping <mapping_id>-replic_mode <full | new | updated>-bulk_
resynch_interval <seconds>
```

Examples:

- To publish the Images cabinet to the CIPDemo content management site:

```
cipcommander publish d7a96a63-e78c-407c-8d7f-e84988806e49
70b1e307-26a1-499c-9295-cf0b6bd01342-source_rename Images-source_path /
-target_rename CIPDemo-mapping documentum2cs
```

- To publish the /Sample/Trees folder in the Images cabinet to the CIPDemo content management site:

```
cipcommander publish d7a96a63-e78c-407c-8d7f-e84988806e49
70b1e307-26a1-499c-9295-cf0b6bd01342-source_rename Images-source_path
/Sample/Trees-target_rename CIPDemo-mapping documentum2cs
```

4. When the publishing session ends, the synchronization engine starts monitoring the published object.

Verify that modifications and deletions are replicated to WebCenter Sites (for example, modify the replicated objects, add folders and documents to the monitored object, and delete documents).

Objects can also be unpublished. For information, see [Section 62.3.2, "Unpublishing."](#)

To optimize the synchronization process, see [Section 62.3.1, "Tuning the Synchronization Process."](#)

Table 62–1 Publishing Parameters

Publishing Parameter	Required	Value
<source_providerid>	R	Provider ID for Documentum: d7a96a63-e78c-407c-8d7f-e84988806e49
<target_providerid>	R	Provider ID for WebCenter Sites: 70b1e307-26a1-499c-9295-cf0b6bd01342
-source_repname	R	<cabinet_name>: Name of the cabinet containing the objects to be published. Enter the name exactly as it appears in the URL.
-source_path		<path_in_cabinet>: Path to the object you want to publish. <ul style="list-style-type: none"> ▪ / (to publish the cabinet specified by <source_repname>) ▪ /<folder>/<folder>/... /<folder> (to publish the last folder in the path)
-target_repname	R	Name of the content management site (on WebCenter Sites) on which the target flex family is enabled. Enter the site's display name, exactly as it appears on the Admin tab in the WebCenter Sites Advanced interface.
-mapping	R	<mapping_id>: Value of the mapping id in mappings.xml . The value is documentum2cs .
-replic_mode		full new updated <ul style="list-style-type: none"> ▪ full means that a full replication will be performed (by default), i.e., newly created items, updated items, and deletions. ▪ new means that only newly created items will be replicated (updates and deletions will not be replicated). ▪ updated means that only new and updated items will be replicated (deletions will not be replicated).
-bulk_resynch_interval		<seconds>: Number of seconds between two successive synchronization events. Default value:600 For optimal performance, set the synchronization interval to a value that agrees with the frequency of updates to the monitored folders. For more information, see Section 62.3.1, "Tuning the Synchronization Process."

62.2.2 Publishing via Customized Mappings

If the objects you plan to publish are based on unmapped metadata, you must first map the object types.

To publish via customized mappings

1. Depending on which type of metadata you have created, update the relevant mapping components as shown below. (We suggest reusing the Documentum flex family. The mappings.xml file is located on the Content Integration Agent host).

New Metadata	Update	Guidelines
New folder type	Documentum flex family	Create a parent definition for the new folder type. (The default parent definition is <code>dm_folder</code> .)
New folder type	<code>mappings.xml</code>	Map the new folder type (<code>source id</code>). The <code>target id</code> takes the value <code>Documentum_Parent</code> ; <code><parent definition></code> (where <code>Documentum_Parent</code> defines the storage table for folder assets).
New document type	Documentum flex family	Create a child definition for the new document type. (The default child definition is <code>dm_document</code> .)
New document type	<code>mappings.xml</code>	Map the new folder type (<code>source id</code>). The <code>target id</code> takes the value <code>Documentum_Child</code> ; <code><child definition></code> (where <code>Documentum_Parent</code> defines the storage table for folder assets).
New document attribute	Documentum flex family	Add the new attribute to the Documentum flex family.
New document attribute	<code>mappings.xml</code>	Mapping the new attribute is required only if its name differs on the source and target. Note: Incorrect mapping of attributes does not stop the publication process, but it does produce a warning message and an entry in the log file.

- If you create flex filters, add the corresponding `jar` files to both the WebCenter Sites and the Sites Agent Services applications.
- Publish the objects. For instructions, see [Section 62.2, "Publishing Procedures."](#)

62.3 Maintaining the Integrated Systems

This section contains the following topics:

- [Section 62.3.1, "Tuning the Synchronization Process"](#)
- [Section 62.3.2, "Unpublishing"](#)

62.3.1 Tuning the Synchronization Process

When a cabinet or folder is published, `catalog.xml` is updated with data points from the `cipcommander publish` command. The data points identify the Documentum and WebCenter Sites systems (in the `<workspace>` tags) and specify replication settings (in the `<replication>` tag).

Following a publishing session, the synchronization engine monitors the published cabinet (folder), using `catalog.xml`. The `BulkResynchInterval` and `ReplicMode` parameters listed in the sample file below can be reset as shown in [Table 62–1, "Publishing Parameters"](#) (The `catalog.xml` file is located in the `conf` folder on the Content Integration Agent server).

Sample catalog.xml

```
<workspace id="41ce3f11-0411-46cb-b974-429162249462">
  <provider-ref refid="d7a96a63-e78c-407c-8d7f-e84988806e49" />
  <init-params>
```

```

        <param name="repname">Documents</param>
        <param name="path">/Images</param>
        <param name="repid">0c000001800045fe</param>
        <param name="itemid">0b0000018002c58e</param>
    </init-params>
</workspace>
<workspace id="6af59904-c6a3-4588-af7b-76f1422d6c10">
    <provider-ref refid="70b1e307-26a1-499c-9295-cf0b6bd01342" />
    <init-params>
        <param name="repname">FirstSiteII</param>
        <param name="repid">68ef906a-6c59-406a-84c2-b73b098cdb93</param>
    </init-params>
</workspace>
<replication>
    <link id="332e73c8-e977-4ba4-85c2-d7636281e192">
        <source-ref refid="41ce3f11-0411-46cb-b974-429162249462" />
        <target-ref refid="6af59904-c6a3-4588-af7b-76f1422d6c10" />
        <mapping-ref refid="documentum2cs" />
        <init-params>
            <param name="BulkResynchInterval">600</param>
            <param name="ReplicMode">full</param>
            <param name="IncrementalSyncDelay">10</param>
        </init-params>
    </link>
</replication>

```

62.3.2 Unpublishing

You can unpublish objects from `catalog.xml` and WebCenter Sites by executing the `cipcommander unpublish` command. The command clears `catalog.xml` of all entries that are associated with published objects (for a sample publication entry, see the sample file in "[Sample catalog.xml](#)" on page 62-7). The `-delete` parameter removes the same entries from WebCenter Sites' database.

The unpublish command takes the following form and parameters:

```
cipcommander unpublish <parameters>
```

Table 62–2 Unpublish Parameters

Unpublish Parameter	Description
-all	Clears <code>catalog.xml</code> of all publication entries.
-linkid	Clears <code>catalog.xml</code> of selected publication entries. linkid specifies the published object's link to the WebCenter Sites system. Use the value in the published object's <code><link></code> tag, which is nested within the object's <code><replication></code> tag (for sample code, see " Sample catalog.xml " on page 62-7). For example, to unpublish an object with linkid <code>332e73c8-e977-4ba4-85c2-d7636281e192</code> from <code>catalog.xml</code> , run the following command: <pre>cipcommander unpublish linkid 332e73c8-e977-4ba4-85c2-d7636281e192</pre>
-delete	Removes from WebCenter Sites' database the same objects that you are unpublishing from <code>catalog.xml</code> . Legal values: <code><true false></code> Default value: <code>true</code>

Archiving to EMC Documentum

This chapter provides instructions on archiving WebCenter Sites assets to EMC Documentum.

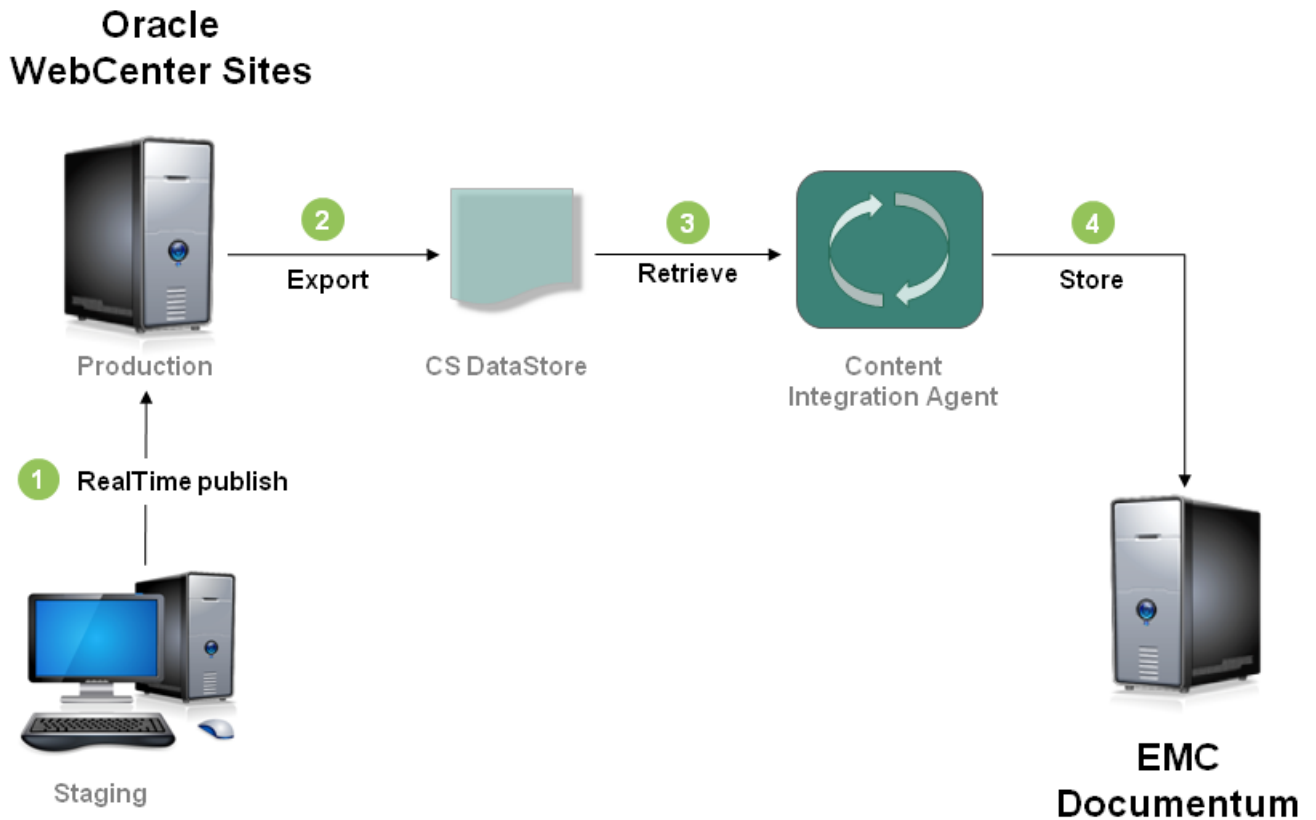
This chapter contains the following sections:

- [Section 63.1, "Overview of the Archival Process"](#)
- [Section 63.2, "Steps for Archiving WebCenter Sites Assets to Documentum"](#)
- [Section 63.3, "Testing Synchronization"](#)
- [Section 63.4, "Tuning the Synchronization Process"](#)

63.1 Overview of the Archival Process

Archiving WebCenter Sites assets to EMC Documentum requires the components shown in [Figure 63-1, "System Architecture for Archiving"](#). Users who are familiar with the process of publishing Documentum objects to WebCenter Sites will recognize archiving to be a similar process.

Figure 63–1 System Architecture for Archiving



The main differences between publishing and archiving are the following:

- In the archival process, Content Integration Agent treats the Content Server (CS) DataStore, rather than WebCenter Sites, as its source of data. The CS DataStore contains assets in folders and files that map to Documentum folders of type `dm_folder` and files of type `fw_document`. The CS DataStore is created when assets in the WebCenter Sites system are published to a RealTime destination while archiving is enabled.
- Content Integration Agent archives data to Documentum directly. It does not require the Sites Agent Services component to store data on the target system.

This section contains the following topics:

- [Section 63.1.1, "System Architecture and Process Flow"](#)
- [Section 63.1.2, "CS DataStore"](#)
- [Section 63.1.3, "Mapping Framework"](#)

63.1.1 System Architecture and Process Flow

Content Integration Agent synchronizes the CS DataStore and target Documentum folder via the `publish` command, the synchronization engine, and the `mappings.xml` file, which provides the metadata map. Following the initial archival session, the synchronization process runs automatically. Details of the implementation are described below.

Initial Synchronization

Issuing the `publish` command invokes the synchronization engine to archive the CS DataStore to Documentum and thereby initialize the synchronization process. The complete set of steps is outlined below (and shown in [Figure 63-1](#)):

1. Assets in WebCenter Sites are RealTime published with archiving enabled.
2. The CS DataStore is created.
3. Issuing the `publish` command invokes the synchronization engine, which then:
 - refers to the CS DataStore (which is specified in the `publish` command)
 - reads the files in the CS DataStore, retrieves their metadata, and
 - converts the metadata to a Documentum-compliant format, using `mappings.xml`
4. The synchronization engine then stores the CS DataStore files to the target Documentum folder.

Monitoring the CS DataStore

Following the initialization process, the synchronization engine monitors the archived CS DataStore and automatically replicates changes to the Documentum target folder. When an asset based on the archived metadata is created or modified in the monitored CS DataStore (during a RealTime publishing process), the synchronization engine replicates the new or modified asset to the target folder on Documentum. If metadata is modified, `mappings.xml` must be reconfigured and the CS DataStore must be republished.

Tuning the Integration

Content Integration Agent contains a configuration file named `catalog.xml`, which stores information about the archival session and allows tuning of the synchronization interval. For more information about tuning, see [Section 63.4, "Tuning the Synchronization Process."](#)

63.1.2 CS DataStore

The CS DataStore contains WebCenter Sites assets in folders and files that map to Documentum folders of type `dm_folder` and files of type `fw_document`. A CS DataStore is created when assets are published to a RealTime destination with archiving enabled. The export path is:

```
<CS DataStore = cs.pgexportfolder>/CIP_DataStore/<CS DataStore for Publishing Destination>
```

Figure 63–2 CS DataStore Archival Example

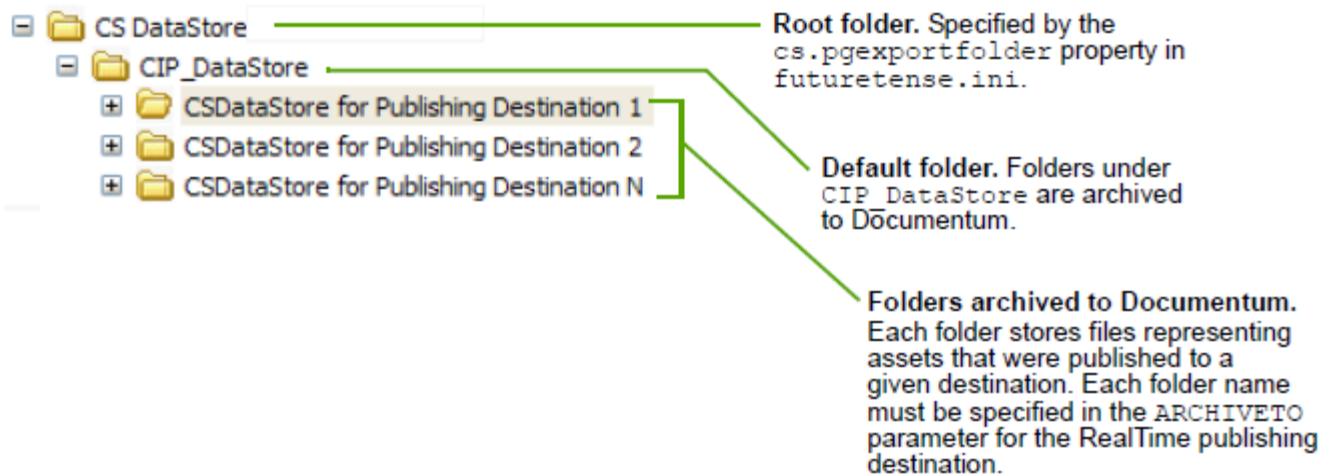


Figure 63–3 illustrates the archival of a sample CS DataStore.

Figure 63–3 Sample CS DataStore Archived to Documentum

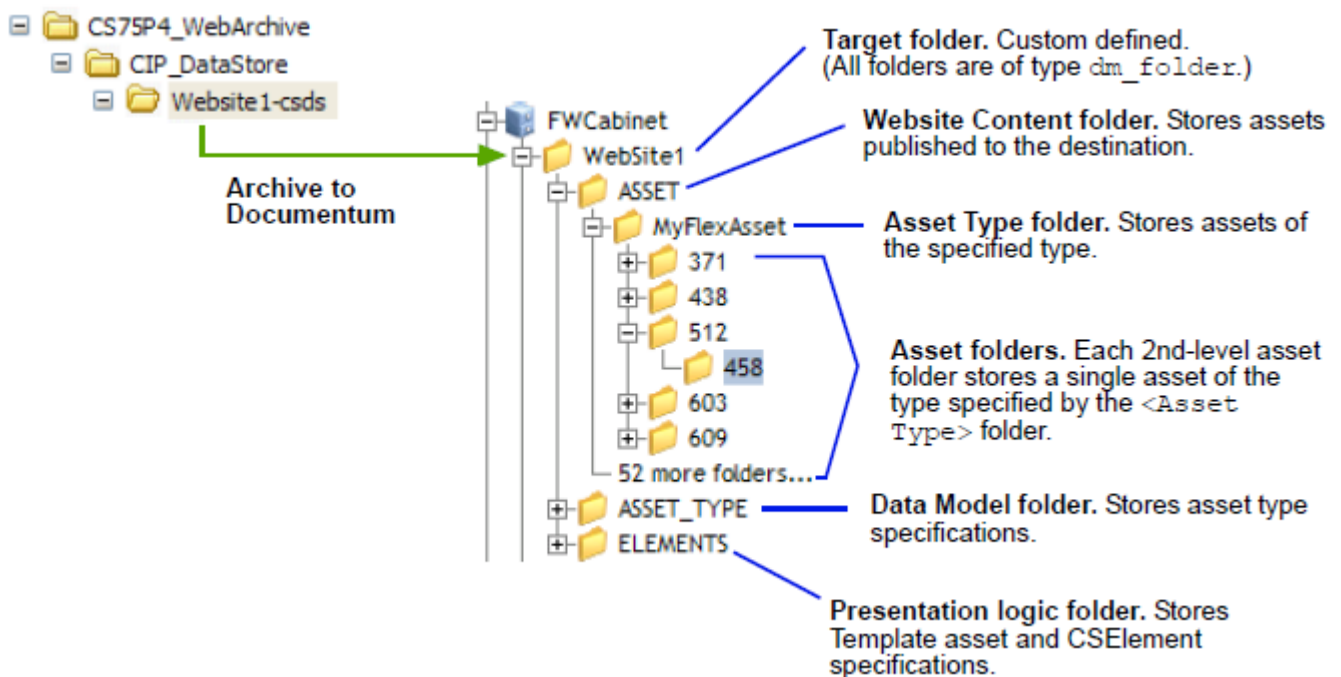
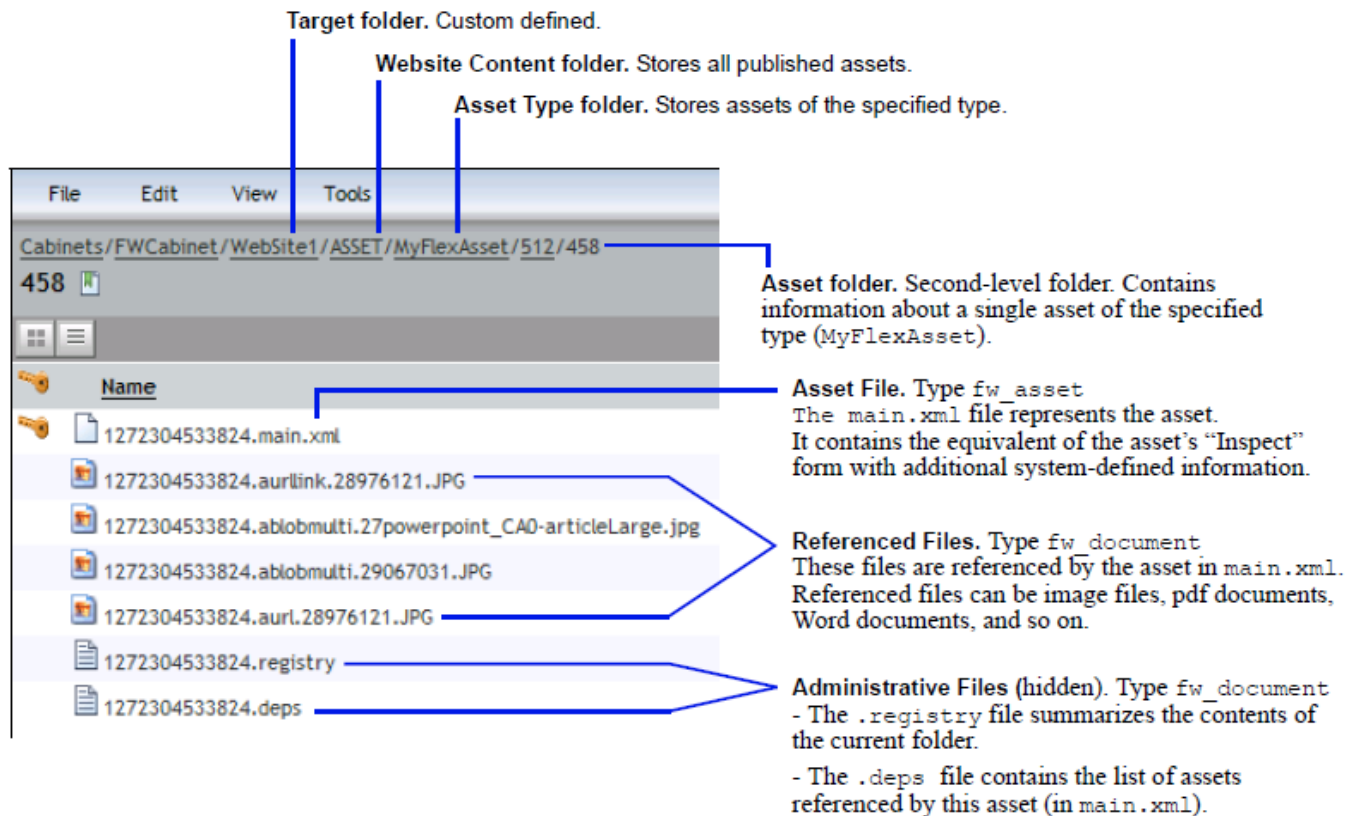


Figure 63–4, "Sample Asset Folder" shows the contents of a typical asset folder (the same folder, named 458, is also shown in Figure 63–3, "Sample CS DataStore Archived to Documentum"). Note that the administrative files shown in Figure 63–4 are hidden files. They are shown here only for illustration purposes.

Figure 63–4 Sample Asset Folder



63.1.3 Mapping Framework

The CIP mapping framework determines the success of the archival and synchronization processes. A CS DataStore can be archived to Documentum as long as its metadata is mapped. The basic mapping framework involves two configurable components: object types in the Documentum workspace and the `mappings.xml` file.

This section contains the following topics:

- [Section 63.1.3.1, "Object Types in the Documentum Workspace"](#)
- [Section 63.1.3.2, "mappings.xml"](#)
- [Section 63.1.3.3, "Mappings, Publishing, and Synchronization"](#)

63.1.3.1 Object Types in the Documentum Workspace

Default Documentum target types are `fw_asset` and `fw_documentum`, which could be modified or replaced with custom types.

63.1.3.2 mappings.xml

The default `mappings.xml` file contains a `documentum2cs` section, which specifies the following mappings:

- `csds_folder` is the data type for all folders in the CS DataStore. Folders of type `csds_folder` map to Documentum folders of type `dm_folder`:

```
<assettype-mappingsourceid="csds_folder" targetid="dm_folder" id="Folder"
extends="Item" />
```

- `csds_Document` is the data type for all files (except `main.xml`) in the CS DataStore. Files of type `csds_Document` map to Documentum documents of type `fw_document`.

```
<assettype-mappingsourceid="csds_Document" targetid="fw_document" id="Document"
extends="Item" />
```

- `csds_Asset` is the data type of the `main.xml` file, which defines the asset (see [Figure 63-4, "Sample Asset Folder"](#)). Files of type `csds_Asset` map to Documentum documents of type `fw_asset`.

```
<assettype-mappingsourceid="csds_Asset" targetid="fw_asset" id="Asset"
extends="Document">
```

- Attributes of the `csds_Asset` document type map to Documentum attributes as shown. All WebCenter Sites attributes are system-defined.

Table 63-1 Document Type to Documentum Attribute Mapping

source id	target id	source id	target id
id	fw_id	publist	fw_publist
name	fw_name	status	fw_status
createdby	fw_createdby	subtype	fw_subtype
createddate	fw_createddate	updatedby	fw_updatedby
description	fw_description	updateddate	fw_updateddate

- The `datemodified` attribute is a special attribute that stores the last publication date. The date is taken from the last modification time of the corresponding asset file in the CS DataStore. The `datemodified` attribute maps to the Documentum attribute `fw_publisheddate`.

63.1.3.3 Mappings, Publishing, and Synchronization

When archiving, bear in mind the following mapping specifications:

- **Default mapping:** Any CS DataStore can be archived to Documentum without your having to modify the default `mappings.xml` file. Running the `publish` command archives the CS DataStore to the target Documentum folder. The source CS DataStore is then monitored by the synchronization engine. When a RealTime publishing session updates the CS DataStore, the new assets and modified assets are automatically replicated to the target Documentum folder by the synchronization engine.
- **Custom mapping:** You can map selected asset types and definitions to your own Documentum object type. Instructions are available in [Section 63.2.3](#).

63.2 Steps for Archiving WebCenter Sites Assets to Documentum

[Section 63.2.1, "Prepare the Documentum System to Store WebCenter Sites Assets"](#)

[Section 63.2.2, "Configure the Path to the CS DataStore"](#)

[Section 63.2.3, "Add Metadata to mappings.xml"](#)

[Section 63.2.4, "RealTime Publish the Site"](#)

[Section 63.2.5, "Archive the CS DataStore on Documentum"](#)

Section 63.2.6, "Archive Visitor-Generated Content"

63.2.1 Prepare the Documentum System to Store WebCenter Sites Assets

Follow these steps to prepare the Documentum system to store WebCenter Sites assets.

1. Create a cabinet for WebCenter Sites assets (`FWCabinet`, for example).
2. Create a folder for the given assets (`WebSite1`, for example).
3. If you configured event notification, store the associated workflow processes anywhere on the Documentum system.

63.2.2 Configure the Path to the CS DataStore

In this step, you will enable the WebCenter Sites RealTime publishing system to export your selected site in a CS DataStore along the following path (see also [Section 63.1.2, "CS DataStore"](#)):

```
<CS DataStore=cs.pgexportfolder>/CIP_DataStore/<CSDataStore for Publishing Destination>
```

where:

- `<cs.pgexportfolder>` defines the root directory of the CS DataStore. (The `cs.pgexportfolder` property is located in the `futuretense.ini` file on the WebCenter Sites delivery system.)
- `CIP_DataStore` is a default subdirectory of `<cs.pgexportfolder>`. Its subfolders will be archived on Documentum.
- `<CSDataStore for Publishing Destination>` is a subfolder that holds the assets of the published site.

To configure the path to the CS DataStore

1. Configure the root directory of the CS DataStore by setting the `cs.pgexportfolder` property in the `futuretense.ini` file of the delivery system.
2. Configure the RealTime publishing process to support archiving.
 - a. Configure publishing as shown in [Chapter 20, "Configuring the RealTime Publishing Process."](#) If RealTime publishing is already configured, start with [Chapter 20.2.4, "Create a RealTime Destination Definition on the Source System."](#)
 - b. on the Add New Destination form, set the **More Arguments** field as follows:

```
ARCHIVETO=<CSDataStore for Publishing Destination>
```

Note: If you are publishing a small number of assets (less than thousands) and wish to store their files to the same folder in the CS DataStore, specify `USEHASHDIRS =false` to disable hash folders.

- c. Complete the remaining steps up to and including mirroring site configuration data to the destination database.

63.2.3 Add Metadata to mappings.xml

To add custom mappings to `mappings.xml`, include the following information, depending on whether you are mapping a flex or basic asset type:

For flex asset types, sourceid takes the form <asset type>;<Definition>. For basic asset types, sourceid takes the form <asset type>. The targetid specifies the corresponding Documentum object type.

For example:

- To map all flex assets of type Content_C with asset definition FSII Article to the fw_content object type, add the following line to mappings.xml :

```
<assettype-mapping sourceid="Content_C;FSII Article" targetid="fw_content"
id="Content"></assettype-mapping>
```

- To map all basic assets of type FW_Article to the fw_article object type, add the following line to mappings.xml :

```
<assettype-mapping sourceid="FW_Article" targetid="fw_article"
id="Article"></assettype-mapping>
```

63.2.4 RealTime Publish the Site

Publish the content management site and verify that the CS DataStore was created in the specified path (in [Section 63.2.2, "Configure the Path to the CS DataStore"](#)).

63.2.5 Archive the CS DataStore on Documentum

In this step, you will run the CIP publish command to archive the <CS DataStore for Publishing Destination> and initialize the synchronization process.

Note: If you changed the port in the *Oracle Fusion Middleware WebCenter Sites Installation Guide*, make sure that the new port is set in facilities.xml, and add -p <port> to the command in 2, below (which starts CIPCommander).

1. Start Content Integration Agent.
2. Run the CIPCommander executable (located in the bin folder of the system where Content Integration Agent is installed):

```
cipcommander publish <source_providerid> <target_providerid> -source_repid
<path to data store>-target_repname <cabinet name>-target_path <path within the
cabinet>-mapping <mapping_id>-bulk_resynch_interval <seconds>-handlerset
csdatastore-create false-replic_mode updated
```

For example:

```
cipcommanderpublish 7833d862-4f8b-4285-84f2-731d5af81865
d7a96a63-e78c-407c-8d7f-e84988806e49-source_repid c:\temp\csdatastore-target_
repname Archive-target_path /fatwire-mapping csds2documentum-bulk_resynch_
interval 60-handlerset csdatastore-create false-replic_mode updated
```

Table 63–2 Publishing Parameters

Publishing Parameter	Value
<source_providerid>	provider ID for the WebCenter Sites system: 7833d862-4f8b-4285-84f2-731d5af81865
<target_providerid>	Provider ID for the Documentum system: d7a96a63-e78c-407c-8d7f-e84988806e49

Table 63–2 (Cont.) Publishing Parameters

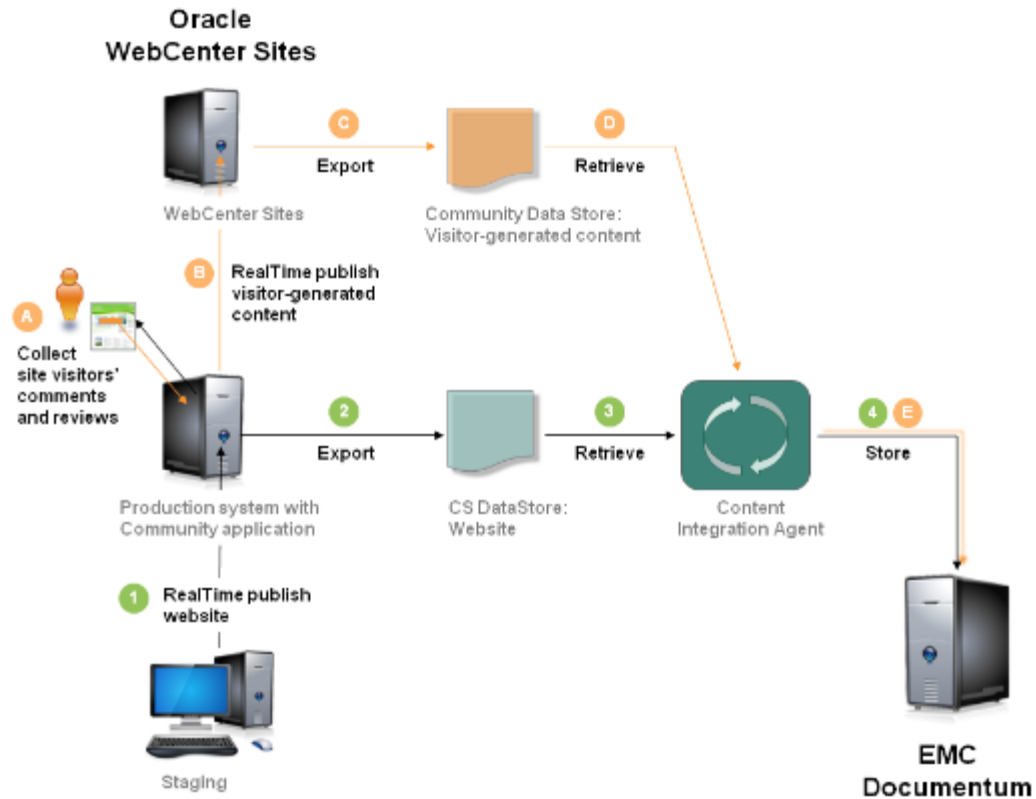
Publishing Parameter	Value
-source_repid	<path to data store>: Path to the <CS DataStore=cs.pgexportfolder> folder on the file system.
-target_repname	<cabinet name>: Name of the Documentum cabinet to which <CSDataStore for Publishing Destination> will be archived.
-target_path	<path within the cabinet>: Path to the Documentum folder in the cabinet specified by target_repname . To publish to the cabinet itself, skip this parameter.
-mapping	<mapping_id> : mapping identifier from the mappings.xml file. Default value : csds2documentum
-bulk_resynch_ interval	<seconds> : Number of seconds between two successive synchronization events. For optimal performance, set this value to a number that correlates with the frequency of RealTime publishing sessions. Default value: 600
-handlerset	References the handlerset element from handlers.xml . Allowed value: csdatastore
-create	Specifies whether to create target repository. Allowed value: false
-replic_mode	Specifies which types of changes will be replicated. Allowed value: updated (only new and updated items will be replicated; deletions will not be replicated).

3. Verify on Documentum the directory structure of the archived assets. For background information, see [Section 63.1.2, "CS DataStore."](#)

63.2.6 Archive Visitor-Generated Content

If your WebCenter Sites delivery system runs the WebCenter Sites: Community application and you wish to archive its visitor-generated content (comments and reviews), publish the content from the delivery system to a RealTime destination on a separate WebCenter Sites system ([Figure 63–5](#)). Procedures for archiving visitor-generated content are identical to those for archiving assets.

Figure 63–5 Archiving Visitor-Generated Content



63.3 Testing Synchronization

After the publish command executes, the archive session ends and the synchronization engine starts monitoring the source CS DataStore. Verify that modifications are replicated to Documentum.

To verify that modifications are replicated to Documentum

1. Create and modify assets on the published content management site.
2. Republish the site to export the same <CS DataStore for Publishing Destination> folder.
3. Look for updates in the target Documentum folder.

63.4 Tuning the Synchronization Process

When the publish command executes, the CS DataStore is archived to Documentum and catalog.xml is updated with data points from the publish command. The data points identify the CS DataStore and Documentum system (in the <workspace> tags) and specify replication settings for the CS DataStore (in the <replication> tag).

Following the archival session, the synchronization engine starts monitoring the published CS DataStore. The synchronization interval can be reset in catalog.xml. See BulkResynchInterval in Table 63–2, "Publishing Parameters". (The catalog.xml file is located in the conf folder.)

```
<workspace id="776a0536-1af8-4e10-9b55-ecb9cfd715b8">
  <provider-ref refid="7833d862-4f8b-4285-84f2-731d5af81865" />
```

```
<init-params>
  <param name="repid">C:\cs\export\CIP_datastore\DCTM</param>
  <param name="repname"></param>
</init-params>
</workspace>
<workspace id="ae679aa2-572c-492a-b553-b7a866b60a2f">
  <provider-ref refid="d7a96a63-e78c-407c-8d7f-e84988806e49" />
  <init-params>
    <param name="repname">Archive</param>
    <param name="path">/FirstSiteII</param>
    <param name="repid">0c0000018002bd87</param>
    <param name="itemid">0b0000018002bd91</param>
  </init-params>
</workspace>
<replication>
  <link id="b5be4bc4-a8dc-4928-b3df-e0ac2851f4e4">
    <source-ref refid="776a0536-1af8-4e10-9b55-ecb9cfd715b8" />
    <target-ref refid="ae679aa2-572c-492a-b553-b7a866b60a2f" />
    <mapping-ref refid="csds2documentum" />
    <handlerset-ref refid="csdatastore" />
    <init-params>
      <param name="BulkResynchInterval">3</param>
      <param name="ReplicMode">updated</param>
      <param name="IncrementalSyncDelay">10</param>
    </init-params>
  </link>
</replication>
```

Configuring Event Notification

This chapter provides instructions on configuring event notifications. Event notifications keep Content Integration Platform administrators informed about the synchronicity of source and target systems as changes to content are made on either system.

This chapter contains the following sections:

- [Section 64.1, "Overview of Event Notification"](#)
- [Section 64.2, "Configuring Event Notification Workflows for WebCenter Sites"](#)
- [Section 64.3, "Configuring Event Notification Workflows for EMC Documentum"](#)

64.1 Overview of Event Notification

Event-driven notices are delivered to administrators in a simple workflow process which can be enabled globally or selectively for specific events (such as item creation) and for specific types of items.

Note: If you choose to configure CIP for event notification, you must do so before publishing objects to WebCenter Sites or archiving assets to Documentum. For instructions, see [Section 64.2, "Configuring Event Notification Workflows for WebCenter Sites"](#) and [Section 64.3, "Configuring Event Notification Workflows for EMC Documentum."](#)

64.2 Configuring Event Notification Workflows for WebCenter Sites

In the CIP publishing model, event notification is the process of informing CIP administrators of actions that occur on WebCenter Sites when changes are made to monitored workspaces - cabinets or folders - on Documentum.

Monitoring of a workspace begins once its metadata and associated objects are first published to WebCenter Sites via the `cipcommander publish` command. The synchronization engine starts listening to the workspace for changes to content and automatically replicates them to WebCenter Sites, where they are treated as events. If event notification is configured, the events invoke WebCenter Sites workflows to send notices confirming the events. Failed events also trigger notices if the associated workflows are configured. [Table 64-1, "Default workflows for WebCenter Sites"](#) lists the supported events and default workflows.

Table 64–1 Default workflows for WebCenter Sites

Event in WebCenter Sites	Workflow Process
Asset creation	CIPAssetCreated . Invoked when an object is created in a monitored cabinet or folder and the counterpart asset is created in WebCenter Sites.
Asset deletion	CIPAssetDeleted . Invoked when an object is deleted from a monitored cabinet or folder and the counterpart asset is deleted from WebCenter Sites.
Asset deletion failure	CIPAssetDeletionFailed . Invoked when: <ul style="list-style-type: none"> ■ An object that was deleted from the monitored cabinet or folder is checked out on the WebCenter Sites system. ■ An object that was deleted from the monitored cabinet or folder has dependencies that would become unresolved on the WebCenter Sites system if the counterpart asset were to be deleted.
Asset modification	CIPAssetModified . Invoked when an object in the monitored cabinet or folder is modified and the counterpart asset is created in WebCenter Sites.
Asset modification failure	CIPAssetModificationFailed . Invoked when an object in the monitored cabinet or folder is modified, but its counterpart asset is checked out in WebCenter Sites.

Default workflows have the following properties:

- Default workflows consist of one state and two steps. When an event occurs, only the first step of the corresponding workflow is taken. If the option **Assign from list of participants** is selected, all members with the selected roles are assigned the first task.
- All supported events trigger notices to users with role CIPAdmin . Tasks may or may not be assigned, depending on the event:
 - A task is assigned for the following events: creation, deletion failure, modification, and modification failure. The task is simply a way of notifying the user whether an event occurred or failed on WebCenter Sites. The task can be removed; there is no obligation to take a step.
 - For deletion events, no task is assigned (the asset no longer exists).

Note: Although workflows related to the publishing processes can be created, it is typically more convenient to use workflows that are provided with CIP. If you wish to create your own workflows, refer to [Chapter 10, "Creating and Managing Workflow Processes."](#)

This section contains the following topics:

- [Section 64.2.1, "Installing Default Workflows on WebCenter Sites"](#)
- [Section 64.2.2, "Verifying the Installation of Default Workflows"](#)
- [Section 64.2.3, "Enabling Default Workflows"](#)

64.2.1 Installing Default Workflows on WebCenter Sites

To install default workflows

1. Run `catalogmover.bat` (or `catalogmover.sh` on Linux) from the WebCenter Sites installation directory.
2. Go to **Catalog**, then **Auto Import Catalog(s)** .
 - a. Select `workflows.zip` (in the same directory or level as all `cs_*_schema.zip` files).
 - b. In the import dialog, fill in the fields as shown below:
 Catalog Data Directory: Leave the default value
 Catalog ACL List: Browser, SiteGod, xceleditor, xceladmin
3. Create the default workflows by invoking the following URL:

```
http://<host>:<port>/<context_
path>/ContentServer?pagename=OpenMarket/Xcelerate/Installation/CIPCreat
eWorkflows&username=<username>&password=<password>
```

where:

- `<host>` is the address of the WebCenter Sites installation
- `<port>` is the port of the WebCenter Sites installation
- `<context_path>` is the context path where the WebCenter Sites web application is deployed
- `<username>` is the WebCenter Sites administrator's user name
- `<password>` is the WebCenter Sites administrator's password

For example, the URL of the default configuration is:

```
http://localhost:8080/cs/ContentServer?pagename=OpenMarket/Xcelerate/In
stallation/CIPCreateWorkflows& username=fwadmin&password=xceladmin
```

When the workflows are created, the following message is displayed:

```
"Workflows for Content Integration Platform were created successfully"
```

64.2.2 Verifying the Installation of Default Workflows

When the default workflows are created, associated items (such as roles and email objects) are also created in WebCenter Sites.

To verify default workflows and their associated items

1. Log in to the WebCenter Sites Advanced interface as a general administrator (default credentials: `fwadmin / xceladmin`).
2. Verify that the following items have been created:
 - CIPAdmin role, which will be used as the management role in all CIP workflows. All users with the CIPAdmin role will be notified of all CIP events in the default workflows.
 - Workflow processes:
 CIP Asset Created, CIP Asset Deleted, CIP Asset Deletion Failed, CIP Asset Modified, and CIP Asset ModificationFailed

- Workflow states:
CIP Asset Created, CIP Asset Deleted, CIP Asset Deletion Failed, CIP Asset Modified, and CIP Asset Modification Failed
- Workflow step action:
CIP Asset Deleted, which results in an email notice to the CIP administrators.
- Email object: CIP Asset Event

64.2.3 Enabling Default Workflows

Default workflows are pre-configured in the default `mappings.xml` file. Each listed asset type contains a commented workflow configuration section.

To enable a CIP workflow

1. Open `mappings.xml` (on the Content Integration Agent host), go to the section `<mapping id= "documentum2cs">` and uncomment the required workflows (below) for each asset type that must be enabled for event notification:


```
<param name="assetCreatedProcess">CIPAssetCreated</param>
<param name="assetModifiedProcess">CIPAssetModified</param>
<param name="assetDeletedProcess">CIPAssetDeleted</param>
<param name="assetDeletionFailedProcess">CIPAssetDeletionFailed</param>
<param name="assetModificationFailedProcess">CIPAssetModification
Failed</param>
```
2. Assign the CIPAdmin role to CIP administrators. Verify that CIP administrators are able to receive email. For instructions, see [Section 5.5.1, "Creating and Editing a User Profile."](#)
3. If a relatively large number of events occur on the source system, it is best to use workflow groups, as they allow you to resolve tasks in bulk. Workflow groups are not packaged by default. They must be created manually. For instructions, see [Chapter 10, "Creating and Managing Workflow Processes."](#)

Note: If a workflow group has the name of the invoked workflow process, the workflow process will be automatically added to the group.

64.3 Configuring Event Notification Workflows for EMC Documentum

In the archival model, event notification is the process of informing CIP administrators of actions that occur in Documentum folders when changes are made to monitored CS DataStores.

Monitoring of a CS DataStore begins when its metadata and associated files are first archived to Documentum via the `cipcommander publish` command. The synchronization engine starts listening for changes to the content of the CS DataStore and automatically replicates them to the Documentum target folders, where they are treated as events. If event notification is configured, the events invoke Documentum workflows to send notices confirming the events. [Table 64–2, "Event Notification Workflows for the Archival Process"](#) lists the supported events.

Table 64–2 Event Notification Workflows for the Archival Process

Event in Documentum	Workflow
Object creation	Invoked when a new asset is created in the CS DataStore and its counterpart object is created in Documentum.
Object modification	Invoked when an asset is modified in the CS DataStore and its counterpart object is created in Documentum.

Note: Event notification related to the archival process requires you to create or reuse Documentum workflows.

To enable notification workflows for archival events

1. Create your own workflow processes in Documentum or re-use workflow processes. Define the packages as necessary.
2. Open `mappings.xml` and go to `<mapping id= "csds2documentum">`.
 - a. Uncomment the required workflow processes and packages (below) for each asset type that must be enabled for event notification:


```
<param name="objectCreatedProcess">ProcessName</param>
<param name="objectModifiedProcess">ProcessName</param>
<param name="objectCreatedPackage">PackageName</param>
<param name="objectModifiedPackage">PackageName</param>
```
 - b. Replace `ProcessName` with the name of the workflow process that should be automatically started.
 - c. `PackageName` is an optional parameter. If `PackageName` is not specified, an asset will be placed into the first available package.

Default Mapping Specifications for Publishing

This chapter contains the following sections:

- [Section 65.1, "Mapping Framework"](#)
- [Section 65.2, "Documentum Flex Family Specifications"](#)

65.1 Mapping Framework

The default mapping framework for the publishing model supplies the following components (also listed in [Table 65–1, " Documentum Default Data and Flex Family Analogs"](#) and [Table 65–2, " Default mappings.xml: <assettype mapping>"](#)):

- The Documentum flex family, pre-configured to match the object types and attributes listed.
- A `mappings.xml` file, in which the object types and attributes listed above are mapped to assets in the Documentum flex family:
 - The `dm_folder` type is mapped to a flex parent definition asset named `dm_folder`.
 - The `dm_document` type is mapped to a flex definition asset named `dm_document`.
 - Attributes are mapped to flex assets of type `Documentum Attribute`.

When the `publish` command is issued:

- Folders are published as flex parent assets to the `Documentum Folder` asset type.
- Documents are published as flex assets to the `Documentum Document` asset type.

During publishing, Content Integration Agent refers to the `mappings.xml` file to determine the types of objects to publish. The folder that is named in the `publish` command is the starting point of the publication process. The folder is published as a flex parent asset of type `Documentum Folder`, along with all the subfolders and documents it contains.

To reproduce the folder's structure (subfolders and documents), Content Integration Agent refers to path information. If subfolders exist, Content Integration Agent chains the counterpart `Documentum Folder` assets to reproduce the hierarchy.

Documents are treated as flex assets of type `Documentum Document`. They are placed under their respective `Documentum Folder` parents.

65.2 Documentum Flex Family Specifications

Table 65–1, "Documentum Default Data and Flex Family Analogs" and Table 65–2, "Default mappings.xml: <assettype mapping>" summarize the mapping of default object types in Documentum to asset types and assets in WebCenter Sites' Documentum flex family. In customized implementations, you can either re-use the flex family or create your own.

Table 65–1 Documentum Default Data and Flex Family Analogs

Documentum Object Type	Flex Asset Type	Assets
Document Attribute ¹ <ul style="list-style-type: none"> ■ title ■ subject ■ keywords ■ r_version_label ■ r_full_content_size 	Documentum Attribute Stores document attributes.	<ul style="list-style-type: none"> ■ title ■ subject ■ keywords ■ version label ■ file size
Folder dm_folder (default)	Documentum Parent Definition Stores folder type definitions.	Parent definitions dm_folder (default)
Folder	Documentum_Folder Flex Parent asset type. Stores folder assets.	Documentum folders
Document dm_document (default)	Documentum Child Definition Stores document type definitions.	Document definitions dm_document (default)
Document	Documentum_Document Flex (Child) asset type. Stores document assets.	Documentum documents

¹ Attribute names are those listed in the **Types** node of the Documentum WebTop interface.

Table 65–2 Default mappings.xml: <assettype mapping>

sourceid	assetid
dm_folder	Documentum_Folder;<dm_folder>
dm_document	Documentum_Document;<dm_document>