Oracle® Fusion Middleware

Administering Oracle WebCenter Sites

12c (12.2.1.3.0)
E96359-03
October 2020

Oracle Fusion Middleware Administering Oracle WebCenter Sites, 12c (12.2.1.3.0)

E96359-03

Primary Author: Puneeta Bharani

Contributing Authors: Ankita Chowdhury, Promila Chitkara

Contributors: Anil Yamarti, Aswini Kalyanam, Brian Cheyne, Chandra Busireddy, Eric Gandt, Eugene Shevchenko, Hareesh Kadlabalu, Kannan Appachi, Kuldeep Tiwari, Naveen Chintala, Revanth Potnuru, Ravi Khanuja, Saikat Chaudhuri, Sailaxmi Rajanala, Viswadas Leher

Contents

Preface

Audience xxvii

Documentation Accessibility xxvii

Product Accessibility xxvii

Related Documents xxvii

Conventions xxviii

Part I Understanding Oracle WebCenter Sites Administration

1 Introduction to Oracle WebCenter Sites

About Oracle WebCenter Sites 1-1

Who Are the WebCenter Sites Administrators? 1-1

Prerequisites for Administrators 1-2

What Tasks Do Administrators Perform to Manage Websites? 1-3

- Collaborating with Your Content Management Team 1-3
- Managing Users 1-4
- Managing Access Control Lists (ACLs) and Roles 1-4
- Managing Asset Types and Assets 1-4
- Managing User Access to Applications 1-5
- Managing Workflow 1-5
- Managing the Publishing System 1-5
- Administering the Performance of WebCenter Sites and Its Components 1-6
- Integrating WebCenter Content with WebCenter Sites 1-6

Sample Sites Used In This Guide 1-6

2 Exploring Content Management Concepts

Introduction to Oracle WebCenter Sites Concepts 2-1

Websites 2-1

Content Management Sites 2-2

Configuration Components 2-3

Content Management Site Definition 2-4
Data Model 2-4
Users and Their Table-Level Permissions 2-5
Users and Roles 2-6
Start Menus 2-8
Workflow Processes 2-9
Publishing System 2-9
User Interface Options 2-9
Content Management Models 2-9
1:1 Model 2-10
1:n Model 2-10
x:n Model 2-11
Oracle WebCenter Sites Environment 2-12

3 Logging In and Getting Started

Logging In to Oracle WebCenter Sites 3-1
If You Selected the Admin Icon 3-2
If You Selected the WEM Admin Icon 3-4
How to Reset a WebCenter Sites Password 3-5
How to Edit Your Oracle WebCenter Sites Profile 3-5
Logging Out of WebCenter Sites 3-6

4 Roadmap for Administering WebCenter Sites

Making Pre-Configuration Decisions 4-1
Understanding Your Job On Each WebCenter Sites System 4-3
Configuring the Development System 4-4
Configuring the Management System 4-5
Configuring the Delivery System 4-9
Configuring the Testing System 4-9
Moving from a Test to a Production Environment 4-9
Common Tasks for WebCenter Sites Administrators 4-9

Part II Managing the Site Foundation

5 Assembling Content Management Sites

Working with Sites 5-1
Creating a Site From the Admin Interface 5-1
Obtaining Site Configuration Information 5-3

Editing a Site in the Admin Interface 5-4
Deleting a Site in the Admin Interface 5-4
Assigning and Managing Site Users 5-5
Granting Users Access to a Site (Assigning Roles to Users) 5-5
Viewing Site Users and Reassigning Roles 5-7
Deleting Users from a Site 5-8
Enabling and Managing Asset Types 5-8
Enabling Asset Types for a Site 5-8
Removing Asset Types from a Site 5-10

6 Working with ACLs and Roles

Overview of ACLs and Roles 6-1
Access Control Lists (ACLs) 6-1
ACLs for User Accounts 6-2
ACLs for Database Tables 6-3
ACLs for Page Entries in the SiteCatalog Table 6-3
System ACLs 6-4
Custom ACLs 6-4
Working with ACLs 6-4
Creating an ACL 6-5
Editing a Custom ACL 6-6
Deleting a Custom ACL 6-6
Assigning ACLs to Custom Tables 6-7
Assigning ACLs to WebCenter Sites Pages 6-8
Roles 6-8
Working with Roles 6-9
Creating a Role From the Admin Interface 6-9
Editing a Role From the Admin Interface 6-10
Deleting a Role in the Admin Interface 6-10

Part III Managing User Interfaces and Assets

7 Configuring the User Interfaces

Setting the Locale for the WebCenter Sites Interfaces 7-1
Using the System Default Locale for WebCenter Sites Users 7-2
System Locale Properties 7-3
Single-Language Restrictions 7-4
Locale and Asset Types 7-5
Locale and the Article Asset Type 7-6

User Interface Fields that are Exceptions to Localization 7-6
Using Web Mode 7-6
Maintaining Separate Browser Sessions for Preview 7-7
Reconfiguring Keyboard Shortcuts for the Contributor Interface 7-8
Default Keyboard Shortcuts 7-8
Customizing Keyboard Shortcuts for the Contributor Interface 7-9
Setting the User Interface to Right-to-Left (RTL) Orientation 7-10

8 Creating and Authorizing Users

Creating Users 8-1
Authorizing Users to Work with Applications 8-4
Authorizing a Predefined User 8-8
Authorizing Developers to Register Applications 8-8
Asking Your Developers About Available Applications 8-9

9 Managing Access to Content Management Site Components

Overview 9-1
Managing Access to Asset Types Using Start Menus 9-1
Working with Start Menu Items 9-3
Start Menu Items for Different Kinds of Users 9-4
Start menu Items for Flex Assets 9-4
Managing Start Menu Items 9-5
About Start Menu Items 9-5
Creating a Start Menu Item 9-6
Creating Search Start Menu Items 9-10
Editing Start Menu Items 9-12
Showing a List of All Start Menu Items 9-12
How to Set Access Permissions to Assets 9-12
Understanding the Permission Structure 9-13
Working with Authorization Types 9-14
Setting Access Permissions 9-15
The Asset Toolbar 9-16
Setting Access Permissions from the Admin Interface 9-16
Setting Access Permissions from Property Files 9-18
Other Options for Asset Types 9-19
Options for Managing Access to the Tree (Admin Interface Only) 9-19
Showing and Hiding the Tree 9-20
About Creating Tree Tabs 9-21
Creating a Tree Tab 9-22

Creating the Workflow Groups Tab 9-25
Editing Trees and Nodes 9-25
Deleting Tree Tabs 9-26
Changing Tree Tab Order 9-26
Configuring the Number of Items Shown Under Nodes on Tabs 9-27

10 Working with the Admin Interface

About the Admin Window 10-1
Administrator-Specific Trees and Nodes 10-2
Administrative Nodes in the General Admin Tree 10-3
About the Site Admin Node in the Site Tree 10-6
Developer-Related Trees and Nodes 10-6
About the Developer Nodes in the General Admin Tree 10-7
About the Developer Nodes in the Content Tree 10-8
Miscellaneous Trees and Nodes 10-8
About System Defaults 10-9

11 Using the Web Experience Management Framework

About the WEM Admin Window 11-1
Overview of the Web Experience Management Framework 11-2
Administrative Roles and Permissions 11-4
General Administrator 11-4
Site Administrator 11-5
WebCenter Sites Applications 11-5
Managing WebCenter Sites CM Sites in the WEM Framework 11-6
Enabling Tree Tabs 11-6
Managing the WEM Framework - Quick Tips and Roadmaps 11-7
Common WEM Tasks for WebCenter Sites Administrators 11-8
Managing Sites in the WEM Admin Interface 11-8
Creating a Site in the WEM Admin interface 11-8
Editing a Site in the WEM Admin interface 11-9
Deleting a Site in the WEM Admin Interface 11-9
Adding an Application to a Site From the Sites List in the WEM Admin Interface 11-9
Assigning Users to a Site From the Sites List in the WEM Admin Interface 11-9
Reassigning Roles to a User from the Sites List in the WEM Admin Interface 11-10
Reassigning Roles to an Application from the Sites List in the WEM Admin Interface 11-10
Removing a User From a Site in the WEM Admin Interface Sites List 11-11
Removing an Application From a Site in the WEM Admin Interface Sites List 11-11

Managing Applications 11-11
Modifying an Application 11-11
Assigning an Application to a Site in the WEM Admin Interface 11-12
Reassigning Roles to an Application from the Apps List in the WEM Admin Interface 11-12
Removing an Application From a Site in the WEM Admin Interface Apps List 11-12
Managing Users in the WEM Admin Interface 11-13
Creating a User in the WEM Admin Interface From the Users List 11-13
Enabling a User From the Users List in the WEM Admin Interface 11-13
Assigning a User to a Site From the Users List in the WEM Admin Interface 11-14
Editing a User From the Users List in the WEM Admin Interface 11-14
Reassigning Roles to a User from the Users List in the WEM Admin Interface 11-14
Assigning a User to a Group from the WEM Admin Interface 11-15
Removing a User From a Site in the WEM Admin Interface Users List 11-15
Deleting a WebCenter Sites User from the Users List in the WEM Admin Interface 11-15
Managing Roles From the WEM Admin Interface 11-15
Adding a Role to the WebCenter Sites System From the WEM Admin Interface 11-16
Editing a Role From the Roles List in the WEM Admin Interface 11-16
Deleting a Role from the Roles List in the WEM Admin Interface 11-16
Managing Profiles From the WEM Admin Interface 11-16
Modifying a User Profile 11-16
Modify Your Own Profile 11-17

Part IV Managing Users and Security

12 Configuring Users, Profiles, and Attributes

About Creating WebCenter Sites Users 12-1
Understanding User Management Options 12-1
Understanding Native WebCenter Sites User Manager 12-2
Understanding the LDAP Plug-In 12-2
Configuring Users in WebCenter Sites 12-2
Working with User Accounts 12-3
Creating a User in the Admin Interface 12-4
Creating Common User Types 12-6
Making the User a General Administrator 12-6
Making the User a Site Administrator 12-7
Making the User a Workflow Administrator 12-8
Making the User a Content Contributor 12-9

Editing a User From the Admin Interface 12-10
Deleting a User from the System Using the Admin Interface 12-10
Working with User Profiles and User Attributes 12-11
Creating and Editing a User Profile 12-12
Deleting a User Profile in the Admin Interface 12-13
Modifying, Adding, and Deleting User Attributes 12-13

13 Setting Up External Security

Setting Up External Security 13-1
Introduction to WebCenter Sites Security 13-1
Understanding ACLs and Security 13-2
Understanding DefaultReader, secure.CatalogManager, and secure.TreeManager 13-3
Understanding BlobServer and Security 13-3
Implementing Security Goals 13-4
Implementing Security 13-5
Using Properties to Configure Security Settings 13-5
Working with Users and Passwords 13-6
Working with SSL and Digital Certification 13-6
Working with URLs and the Web Server (Delivery System Only) 13-7
Working with WebCenter Sites Forms and Pages (Delivery System Only) 13-7
Testing Security 13-7
Implementing Cryptography and Encryption Key Security 13-8
WebCenter Sites Security Overview 13-8
About Security Configurations in WebCenter Sites 13-9
About the SitesSecurityContext.xml Configuration File 13-9
Working with WebCenter Sites Security Utilities 13-11
Working with ESAPI Security Validation 13-13
Using REST Security 13-13
REST Authorization 13-13
Security Model 13-13
Privilege Resolution Algorithm 13-15
Authorizing Users to Access Application Resources 13-15
Viewing REST Security Configurations 13-16
Creating a Group 13-18
Adding Users to a Group 13-18
Configuring Security for REST Resources 13-20
REST Security Configuration Reference 13-22
Configuring REST Security for ACL Resources 13-22
Configuring REST Security for Application Resources 13-23
Configuring REST Security for Asset Resources 13-24

Configuring REST Security for Asset Type Resources 13-24
Configuring REST Security for Engage Resources 13-26
Configuring REST Security for Group Resources 13-26
Configuring REST Security for Indexed Asset Type Resources 13-27
Configuring REST Security for Role Resources 13-28
Configuring REST Security for Site Resources 13-28
Configuring REST Security for User Resources 13-29
Configuring REST Security for UserDef Resources 13-29
Configuring REST Security for UserLocale Resources 13-30
Configuring REST Security for Visitor Resources 13-30

Part V Managing Publishing

14 Understanding Publishing

Prerequisites for Publishing Assets 14-1
Publishing Destinations 14-2
Authorizing Users to Publish Assets from the Contributor Interface 14-3
Understanding the Approval System 14-4
About Scheduled Publishing Sessions 14-4
Understanding the Publishing Session 14-5
Obtaining Information About a Publishing Session 14-6

15 Using RealTime Publishing

RealTime Publishing Overview 15-1
Source System Configuration for Publishing 15-3
Destination System Configuration for Publishing 15-4
Working with RealTime Publishing 15-5
Configuring Your System for RealTime Publishing 15-7
Creating the Batch User Account (If One Does Not Exist) 15-8
Setting Up the Destination System 15-10
Identifying a Proxy Server to the Source System 15-11
Identifying the Local Proxy Server to the Source System for All Destinations 15-11
Identifying the Local Proxy Server to the Source System Per Destination 15-11
Creating a RealTime Destination Definition on the Source System 15-12
Initializing the Destination Database 15-15
Mirroring the Site Configuration Data to the Destination Database 15-16
Testing the Publishing Process 15-18
Approving Assets 15-19
Publishing the Assets 15-19

Testing the Published Site 15-19
Setting Up the Schedule 15-20
Turning Off Asset Invalidation on the Delivery System 15-21
Adding a New RealTime Destination Definition 15-21
About Synchronizing Site Configuration Data 15-21
How to Differentiate Complete and Delayed Publishing 15-22
About the Publish Console 15-22
Working with Active Publishing Sessions 15-23
Monitoring a Publishing Session 15-24
Resuming a Delayed Publishing Session 15-25
Canceling a Publishing Session 15-26
Canceling a Publishing Session from the Publish Console 15-27
Canceling a Publishing Session from the Publishing Status Form 15-27
Working with Scheduled Publishing Sessions 15-29
Working with Prior Publishing Sessions 15-29
Viewing Past Publishing Sessions 15-30
Redoing a Publishing Session 15-31
Working with Publishing Logs 15-32
Viewing and Searching Logs 15-33
Downloading Logs 15-35
Deleting Publish History 15-35

16 Scheduling Publish Events

Scheduling a Publish Event 16-1
Reading the Schedule Abbreviations 16-3
Editing Publish Events 16-4
Overriding the Schedule 16-5
Assigning Approval or Preview Templates 16-5
Monitoring Sessions in the Publishing Console 16-6
Verifying Publishing Readiness 16-7
About Publish History 16-8
About Session History 16-8
About Publishing All Approved Assets 16-9
Troubleshooting 16-10
Understanding Publishing System Error Messages 16-10
Understanding Numeric Messages 16-11
Indicating Other System or Configuration Issues 16-13

Part VI Managing Approval and Workflow

17 Understanding the Approval System

About the Approval System 17-1
Understanding Dependency Analysis 17-2
Understanding Publishing Terms and Definitions 17-5
Understanding Approval: Intent to Publish vs. Permission to Publish 17-5
Understanding Approval Dependencies 17-6
Examples of Approval Dependencies 17-8
Notes About Approval Dependencies 17-11
Working with Approval States 17-11
Using Rules for Dependency Analysis 17-17
Understanding Approval Dependencies and Parent-Child Relationships 17-17
Working with Export to Disk Publishing 17-19
Evaluating Approval Dependencies 17-19
Understanding Test Rendering 17-20
Working with Mirror to Server Publishing and Export to XML 17-20
Ensuring Version-Matched Content 17-21
Using Exists-Exact Dependencies in Mirror to Server and Export to XML Publishing 17-22
Using Exists-Exact Dependencies in Export to Disk Publishing 17-23
Working with Exists/Exact Dependency Rules 17-23
Understanding Dependency Rules for Basic Asset Types 17-24
Understanding Dependency Rules for CSElement and SiteEntry Assets 17-24
Understanding Dependency Rules for Flex Families 17-24
Understanding Dependency Rules for Engage Visitor Data Assets 17-25
Evaluating Published Content 17-26
Approval System Example 17-26
Using Approval States: Reference 17-30

18 Creating and Managing Workflow Processes

About the Workflow Feature 18-1
Understanding Workflow Participants 18-2
Understanding Workflow States 18-2
Moving Assets from State to State 18-3
Working with Multiple Paths for the Asset 18-3
Managing Deadlocks 18-4
Understanding Workflow Groups 18-5
Delegating and Clearing Assignments 18-6
Placing an Asset in Workflow 18-6

Restricting Access to Assets While They Are in Workflow 18-6
Working with Deadlines 18-7
Scheduling a Deadline Calculation 18-8
Ending a Workflow Process 18-8
Planning Your Workflow Processes 18-9
Starting with a Sketch 18-9
Determining Roles and Participants 18-9
Determining the Email Objects, Actions, and Conditions 18-11
Understanding Email Objects 18-12
Understanding Timed Actions 18-13
Understanding Delegate Actions 18-14
Understanding Step Actions 18-15
Understanding Step Conditions 18-16
Understanding Deadlock Actions 18-16
Understanding Group Deadlock Actions 18-16
Determining States 18-17
Determining Steps 18-18
Determining Function Privileges 18-20
Implementing Simplified Access Control 18-23
Determining Additional Workflow Process Details 18-24
Configuring Your Workflow Processes 18-24
Setting Up Email Objects 18-26
Creating Email Objects 18-27
Editing Email Objects 18-28
Deleting Email Objects 18-28
Setting Up the Actions and Conditions 18-28
Creating Actions and Conditions 18-29
Editing Actions and Conditions 18-30
Deleting Actions and Conditions 18-30
Configuring Approve for Publish Step Actions 18-31
Setting Up the Timed Action Event 18-31
Setting Up the States 18-32
Creating Workflow States 18-33
Editing Workflow States 18-34
Deleting Workflow States 18-35
Setting Up the Workflow Processes 18-35
Editing Workflow Processes 18-35
Editing Workflow Steps 18-36
Editing Function Privileges 18-36
Copying Workflow Processes 18-36
Deleting Workflow Processes 18-37

Deleting Workflow Steps 18-37
Deleting Function Privileges 18-38
Creating Workflow Processes 18-38
Name and Set Global Settings for the Workflow Process 18-38
Create the Start Step for the Workflow Process 18-40
Create the Subsequent Workflow Steps 18-41
(Optional) Create the End Step for the Workflow Process 18-43
(Optional) Configure Function Privileges for the Workflow Process 18-44
Testing Your Workflow Process 18-46
Moving Your Work 18-46
Clearing Workflow Assignments 18-46
Creating and Viewing Workflow Reports 18-48
Creating Workflow Groups 18-48

Part VII Enabling Blog Components

19 Enabling Blog Asset Types

20 Creating a Blog Tree Tab

21 Sharing Blog Assets

22 Working with Blog Authors

Creating Author Assets 22-1
How to Set Access Permissions to a Blog Author Asset 22-3
Creating a Unique Role for the Blog Author 22-3
Assigning the Role to the Blog Author 22-4
Setting Access Permissions to a Blog Author Asset 22-4

23 About Creating Workflows for and Publishing Blog Assets

Creating Workflows for Blog Assets 23-1
About Publishing Blog Assets 23-1

Part VIII Enabling A/B Testing

Part IX Enabling Eloqua Cloud Marketing Service

24 Administering Eloqua Cloud Marketing Service in WebCenter Sites

Eloqua Integration Overview 24-1
Accessing WebCenter Sites Content in Eloqua 24-1
  Configuring Eloqua for Access to the WebCenter Sites Content Service 24-2
  Configuring OAuth Authentication 24-4
  Making WebCenter Sites Assets Available within Eloqua 24-4
Registering the Eloqua Firehose Service 24-4
Creating the Rendering Template for the EloquaForm Assets 24-5

Part X Enabling Bluekai Integration

25 Administering Bluekai Integration

Working with Bluekai Partner Site 25-1
Configuring Bluekai Integration Properties 25-1

Part XI Enabling Content and Experience Cloud Integration

26 Administering Content and Experience Cloud Integration

Getting Content and Experience Cloud Access Token 26-1
Configuring Content and Experience Cloud with WebCenter Sites 26-2
Configuring Content and Experience Cloud Properties 26-3
Publishing Content and Experience Cloud Assets 26-3

27 Publishing Content from WebCenter Sites to Oracle Content and Experience

Configuring Assets Types 27-1
Creating OCE Destination 27-2
Approving Assets for Publishing 27-3
Publishing Assets 27-3
Downloading Published Assets 27-4
Importing Published Assets to Oracle Content and Experience 27-4

Limitations 27-7

Part XII Managing and Tuning Caching

28 Using The inCache Framework

Overview 28-1
- Overview of the Caching Framework 28-2
- Overview of Page Caching 28-2
- Overview of Asset Caching 28-3

Understanding How inCache Works 28-3
Restarting a Node 28-5
Regenerating Pages During RealTime Publishing 28-5
Double-Buffered Caching 28-6
Configuring Your System for inCache Page Caching 28-6
Tuning Options 28-11
- Striping the Disk Cache 28-11
- Configuring for Page Regeneration During RealTime Publishing 28-12
- Setting Up Page Propagation 28-14
  - Enabling Page Propagation 28-15
  - Setting Up Page Propagation on Restart 28-16
- Configuring for Pagelet Regeneration in the Background 28-16

Working with the AssetCache Container 28-17
Working with Asset Caching Operations 28-18
- Working with Asset Loading 28-18
- Comparing Asset Caching with inCache Page Caching 28-19
- Flushing AssetCache 28-20

Types of Asset Caching 28-21
Customizing Asset Caching 28-22
Disabling Asset Caching 28-23
Using the Cache Management Tool 28-23
Using inCache Features for Remote Satellite Server 28-23

Part XIII Monitoring and Maintaining WebCenter Sites

29 Troubleshooting Using System Tools

Using the Configure Log ODL Tool 29-1
- Viewing Loggers 29-1
- Changing Logger Levels 29-3

- Adding New Loggers 29-4

Using the System Information Tool 29-4

- Viewing WebCenter Sites Information 29-5
- Viewing Database Information 29-6
- Viewing Thread Information 29-8

About Using the Cache Management Tool 29-9

Working with Cache Management - CAS Cache 29-9

- Overview of the CAS Cache Summary Form 29-9
- Working with Cluster Information 29-11

Working with Cache Management - Resultset Cache 29-12

- Enabling and Configuring Resultset Caching over inCache 29-12
- Summary View 29-13
- Viewing Cluster Information 29-15

Working with Cache Management - Sites Cache 29-16

- Overview of the Sites Cache Summary Form 29-16
- Working with Page Cache 29-18
- Working with the Dependency Cache 29-19
- Working with the Asset Cache 29-21
- Working with the URL Cache 29-22
- Viewing Cluster Information 29-23

Working with Cache Management - Co-Res Cache 29-24

- Overview of the Co-Res SS Cache Summary Form 29-24
- Working with the Page Cache 29-26
- Working with the Dependency Cache 29-27
- Working with the URL Cache 29-29
- Viewing Cluster Information 29-30

Using the Log Viewer 29-30

- Viewing the Log 29-30
- Tailing the Log 29-31
- Downloading the Log 29-32
- Searching the Log 29-32

Performance Test 29-33

- File System Test 29-33
- Lucene Indexing Test 29-36

Working with Slots Assets 29-38

Working with the URL Node 29-39

Working with the Purge Voided Assets Node 29-39

30 Replicating Content Management Sites

Site Launcher Overview 30-1
Preparing for Replication 30-2
Replication Requirements for the Source Site 30-2
Planning the New Site 30-2
Copying vs. Sharing 30-2
Prefixes for Asset Names 30-4
Planning Users 30-4
Replication Schedule 30-4
Enabling Site Replication 30-5
Configuring the Source Site for Site Launcher 30-5
Replicating the Source Site 30-6
Completing Post-Replication Tasks 30-8

31 Managing Revision Tracking

Revision Tracking Overview 31-1
Tracker Tables and Storage Directories 31-2
RTInfo Table 31-2
Revision Tracking and the Two Asset Models 31-3
Implicit vs. Explicit Checkin and Explicit Checkout 31-3
Revision Tracking and Non-Asset Tables 31-3
How Many Versions? 31-4
Enabling Revision Tracking 31-4
Enabling Revision Tracking for Assets 31-5
Enabling Revision Tracking for Non-Asset Tables 31-5
Editing Revision Tracking Settings 31-5
Changing the Root Storage Directory 31-6
Changing the Number of Revisions 31-6
Disabling Revision Tracking 31-7
Unlocking Revisions 31-7

32 Configuring Vanity URLs

Generating Vanity URLs 32-1
Configuring Auto-Generated URLs 32-2
Working with WebRoots 32-6
Different Types of WebRoots Required For a URL: Examples 32-7
Creating a New WebRoot 32-7
Viewing a WebRoot 32-9
Editing an Existing WebRoot 32-10

Approving and Publishing a WebRoot 32-11
Working with URL Patterns 32-12
Using Functions 32-13
Using Patterns 32-14
Deleting Patterns 32-15
About Publishing URL Patterns 32-15
Disabling Redirection to Vanity URLs 32-15
Using Vanity URL with System Tools 32-15
Viewing All Vanity URLs 32-16
Resolving Vanity URL Conflicts 32-16
Resolving Vanity URLs Using a Rewriter Filter 32-17
Using Vanity URLs for Mobile Sites 32-18
Using the Web Server with Vanity URLs 32-18
Using mod_rewrite with Vanity URLs 32-19
Setting Up mod_rewrite 32-20
Setup for Provided Examples 32-20
About Converting with mod_rewrite 32-22
Using the WebCenter Sites Filter with mod_rewrite 32-23
Using the Rewriter Filter with mod_rewrite 32-24
Using Static Content with mod_rewrite 32-25
Customizing Vanity URLs 32-25
Creating a Customized Vanity URL 32-26

33 Configuring the Lucene Search Engine

Overview of WebCenter Sites Search Functions 33-1
Indexing for Search Functions 33-1
Using WebCenter Sites Search Functions 33-2
Setting Up Search Indices 33-4
Enabling the Lucene Search Engine 33-5
Adding Asset Types to the Search Index 33-5
Configuring Attributes for Asset Type Index 33-7
Indexing Binary Files 33-8
Enabling Indexing of Binary Files 33-8
Disabling Indexing of Binary Files 33-8
Disabling the Lucene Search Engine 33-9
Maintaining Search Indexes 33-10
Pausing and Resuming Indexing 33-10
Pausing Global and Asset Type Indexing 33-10
Resuming Global and Asset Type Indexing 33-11
Re-indexing 33-11

Deleting Index Data 33-12
Writing Code that Queries the Search Index 33-13

34 Using the Site Capture Application

Site Capture Model 34-1
Capture Modes 34-1
Crawlers 34-2
Logging in to the Site Capture Application 34-3
Using the Default Crawlers 34-4
Sample Crawler 34-4
FirstSiteII Crawler 34-4
Running a Default Crawler 34-4
Setting Up a Site Capture Operation 34-5
Creating a Starter Crawler Configuration File 34-5
Defining a Crawler 34-6
Editing the Crawler Configuration File 34-7
Starting a Crawl 34-8
Run the Crawler Manually in Static Mode 34-8
Run the Crawler Manually in Archive Mode 34-9
Schedule the Crawler for Archive Capture 34-11
About Publishing a Site in RealTime Mode 34-11
About Managing Captured Data 34-11
Enabling Publishing-Triggered Site Capture 34-11
About Integrating the Site Capture Application with Oracle WebCenter Sites 34-11
Configuring a RealTime Publishing Destination Definition for Site Capture 34-12
Matching Crawlers 34-13
Managing Site Capture 34-13
Managing Statically Captured Sites 34-14
About Managing Archived Sites 34-16
Notes and Tips for Managing Crawlers and Captured Data 34-17
Tips for Creating and Editing Crawlers 34-18
Notes for Deleting a Crawler 34-18
Notes for Scheduling a Crawler 34-18
About Monitoring a Static Crawl 34-19
About Stopping a Crawl 34-19
About Downloading Archives 34-19
Notes About Previewing Sites 34-19
Tips for Configuring Publishing Destination Definitions 34-19
About Accessing Log Files 34-19
General Directory Structure 34-20

Custom Folders 34-22

35 Administering WebCenter Sites and Its Components in Enterprise Manager

Starting WebCenter Sites and Its Components in the WebLogic Server Console 35-1
Starting WebCenter Sites and Its Components in Enterprise Manager 35-2
Viewing the Home Pages of WebCenter Sites and Its Components 35-3
Managing Log Files for WebCenter Sites and Its Components 35-4
Configuring Log File Settings for WebCenter Sites and Its Components 35-4
Viewing Log Files for WebCenter Sites and Its Components 35-4
Monitoring the Performance Metrics of WebCenter Sites and Its Components 35-5
Viewing the Status of WebCenter Sites and Its Components 35-5
Viewing the Performance Metrics of WebCenter Sites and Its Components 35-6
WebCenter Sites Performance Metrics 35-6
Visitor Services Performance Metrics 35-10

Part XIV Integrating with Oracle WebCenter Content

36 Understanding WebCenter Content Integration

Prerequisites for Administering WCC Integration 36-1
Basics of WebCenter Content Integration 36-1
WebCenter Content Data Model 36-2
Asset Type Mapping and Connector Rules 36-3
Asset Type Mapping 36-3
Content Identifier Attribute 36-4
Content Storage Attributes 36-4
Template Attributes 36-4
Administrator-Defined Specifications 36-5
Connector Rules 36-5
WebCenter Content Integration and Its Parts 36-7
Enabling the SitesIntegration Component 36-9
Enabling Access to the Connector Admin Node 36-10

37 Setting Up Content Synchronization

Enabling Content Items for Synchronization 37-2
Determining Which Renditions to Synchronize 37-2
Selecting or Creating Target Asset Types 37-3
Connecting to WebCenter Content 37-3

Creating an Active Rule 37-5
Prioritizing Active Rules and Managing Rules 37-10
Manually Running the Connector 37-11
Monitoring the Connector 37-12
Viewing Connector History 37-14
Viewing the Entire Connector History 37-14
Viewing Connector History for a Single Batch 37-16
Verifying Imported Content 37-18
Modifying Your Setup 37-19
Scheduling the Connector 37-19
Managing Synchronized Assets 37-21
Mapped Attributes 37-22
Orphaned Content 37-22
Rolling Back the Synchronization Point 37-23
Rolling Back to the Original Starting Point 37-24
Rolling Back to a Particular Update 37-25
About the Generated Token Value 37-25

38 Working With WCC Connector Configuration Files

About WCC Connector Configuration Files 38-1
User Configurable Parameters 38-1

Part XV Reference

39 System Defaults

ACLs 39-1
Permissions 39-1
Accessing ACLs 39-2
System ACLs and Their Permissions 39-2
ACLs of Default Users 39-5
Required ACLs for Custom Users 39-6
System Roles 39-6
System Asset Types 39-7
Default Tree Nodes 39-8

40 WebCenter Sites Database Tables

Cache Management Tables 40-1
Approval System Tables 40-1

Publishing System Tables 40-2
Workflow Tables 40-2
Basic Asset Tables 40-3
Flex Asset Tables 40-3
Visitor Tables (Engage) 40-4
Managing the Attribute Tables 40-5
Managing the Session Objects (Scratch) Table 40-6
Deleting Unnecessary .class Files 40-6

41 Users, Sites, and Roles in LDAP-Integrated Sites Systems

LDAP-Integrated WebCenter Sites Systems 41-1
LDAP Schema 41-1
LDAP Connectivity for Site and Role Management 41-3
LDAP Users and Their Permissions to LDAP Servers 41-3
LDAP-Integrated Operations 41-4
User Management Operations 41-4
LDAP Server Response for Creating an ACL 41-7
LDAP Server Response for Editing an ACL 41-7
LDAP Server Response for Deleting an ACL 41-7
LDAP Server Response for Assigning ACLs to Tables 41-8
LDAP Server Response for Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries) 41-8
LDAP Server Response for Creating a User (Granting ACLs for Access Privileges) 41-8
LDAP Server Response for Editing a User 41-9
LDAP Server Response for Deleting a User 41-9
LDAP Server Response for Creating/Editing a User Profile 41-10
LDAP Server Response for Deleting a User Profile 41-10
LDAP Server Response for Creating User Attributes 41-10
LDAP Server Response for Editing/Deleting User Attributes 41-11
Site and Role Management Operations 41-11
Operations with Flat-Schema LDAP Servers 41-11
Operations with Hierarchical-Schema LDAP Servers 41-13
LDAP Server Response for Creating a Site 41-15
LDAP Server Response for Editing a Site 41-15
LDAP Server Response for Deleting a Site 41-16
LDAP Server Response for Granting Users Access to Sites 41-16
LDAP Server Response for Removing Users' Access to Sites 41-17
LDAP Server Response for Creating a Role 41-17
LDAP Server Response for Editing a Role 41-18

LDAP Server Response for Deleting a Role 41-19

42 Alternate Publishing Methods

About Alternate Publishing Methods 42-2
Using On-Demand Publishing 42-3
Unapproving Selected Assets from a Publishing Session 42-6
Learning Export to Disk Publishing Terminology 42-9
Page 42-9
Publish Key 42-9
Publish Queue 42-10
Primary Asset 42-10
Using Approvals and Export to Disk Publishing 42-10
Using the Approval Template 42-11
Using the Approval Queue 42-11
Using Approval Dependencies 42-11
Working with Export to Disk Publishing 42-12
Working with the Publish-Time Template 42-13
Working with Starting Points 42-13
Working with Compositional Dependencies 42-13
Frequently Asked Questions About the Export to Disk Publishing Process 42-13
Investigating Publishing Behavior and Functionality 42-14
Investigating Template Dependencies 42-14
Investigating Reference Dependencies 42-15
Understanding the Approval and Publish-Time Templates: What Happens When They Differ? 42-16
Differing Templates: Example 1 42-16
Differing Templates: Example 2 42-17
Differing Templates: Example 3 42-17
Working with Sample Templates 42-18
Rendering Export to Disk Pages 42-18
Understanding Export to Disk Publishing 42-19
Working with the Path Naming Convention 42-21
Working with Export Path Variables and SIMPLEDIR 42-21
Working with Export Path Construction 42-23
Using a Forced Export Path 42-23
Using an Export Path for Page Assets 42-23
Using an Export Path for Assets Other than Page 42-24
Using Sample Export Paths 42-24
Working with Paths for Links Within Exported Files 42-25
Working with File Naming Conventions 42-26
Working with Export Starting Points 42-28

Configuring Your System for Export to Disk Publishing 42-28

Creating the Batch User Account For Export Publishing (If One Does Not Exist) 42-29

Specifying the Root Export Directory 42-29

Configuring an Export Destination 42-30

Mapping a URL Prefix for Your Web Server 42-34

Creating the Export Starting Points 42-34

Approving Your Assets for Export to Disk Publishing 42-37

Publishing and Testing the Results 42-37

Setting Up the Schedule 42-38

Understanding Mirror to Server Publishing 42-38

Pre-Configuring Mirror to Server Publishing 42-41

Working with Users and Mirror to Server Publishing 42-41

Working with CacheManager 42-41

Configuring a Mirror Destination 42-41

Determining When to Use Mirror Destination Configuration 42-42

Configuring Your System for Mirror to Server Publishing 42-42

Creating the Batch User Account for Mirror Publishing (If One Does Not Exist) 42-43

Setting Up the Destination System 42-43

Identifying the Mirror User to the Source System 42-44

Identifying the Local Proxy Server to the Source System (If One Exists) 42-45

Creating a Mirror Destination 42-46

Initializing the Destination 42-50

Configuring the Mirror Destination 42-51

Approving Your Assets 42-53

Publishing the Assets 42-54

Testing the Results 42-54

Setting Up the Schedule 42-55

Turning Off Asset Invalidation on the Delivery System 42-56

Retrieving Logs From Delivery WebCenter Sites Systems 42-56

Exporting Assets to XML Publishing Method 42-57

Understanding the XML Output 42-57

Understanding File Naming Conventions for Export Assets to XML 42-59

Configuring Your System for Export Assets to XML 42-59

Creating the Batch User Account for XML Publishing (If One Does Not Exist) 42-59

Specifying the Root Export Directory 42-59

Configuring an Export Assets to XML Destination 42-60

Approving Your Assets for XML Publishing 42-60

Publishing and Testing the Results 42-61

Setting Up the Schedule 42-61

Migrating a Site from One System to Another 42-61

Moving a Site from a Development System to a Management System 42-62

Moving a Site to a Delivery System 42-62

Approving Multiple Assets 42-63

Creating Destinations 42-65

Editing Destinations 42-65

Deleting Destinations 42-65

Creating Export Starting Points 42-66

Preface

This guide explains how to administer WebCenter Sites including managing user interfaces, assests, users, security, and publishing.

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.

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.

Product Accessibility

The Oracle WebCenter Sites: Contributor interface uses high contrast mode when this option is selected in supported operating systems.

Related Documents

For more information, see the following documents in the Oracle WebCenter Sites 12c (12.2.1.3.0) documentation set:

- *Oracle WebCenter Sites Release Notes*
- For user information, see *Using Oracle WebCenter Sites*.
- For developer information, see *Developing with Oracle WebCenter Sites*.
- For properties information, see *Property Files Reference for Oracle WebCenter Sites*.

Conventions

The following text conventions are used in this document:

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

Part I

Understanding Oracle WebCenter Sites Administration

You can administer all tasks that are involved in creating and maintaining a website through Oracle WebCenter Sites, from creation to delivery. You will be involved in performing content managements tasks, creating and updating user profiles, configuring different environments, and so on.

Topics

- Introduction to Oracle WebCenter Sites
- Exploring Content Management Concepts
- Logging In and Getting Started
- Roadmap for Administering WebCenter Sites

1

Introduction to Oracle WebCenter Sites

With Oracle WebCenter Sites, you can create scalable and robust websites that are easy to manage and administer.
This section describes how to get started with Oracle WebCenter Sites for general, site, and workflow administrators.

- About Oracle WebCenter Sites
- Who Are the WebCenter Sites Administrators?
- Prerequisites for Administrators
- What Tasks Do Administrators Perform to Manage Websites?
- Sample Sites Used In This Guide

About Oracle WebCenter Sites

Oracle WebCenter Sites is a Web Experience Management (WEM) system that enables its users to build and manage desktop and mobile websites. With Oracle WebCenter Sites, you can create scalable and robust websites that are easy to manage and administer.

As an administrator, you are responsible for creating content management sites, assigning users to the sites, enabling the Oracle WebCenter Sites: Contributor interface on the sites, and managing user access to content management functions within the interface. You are also responsible for managing workflow, revision tracking, publishing, and marketing.

For administrators, Oracle WebCenter Sites provides the Admin interface and the WEM Admin interface, which together provide the tools needed to manage all aspects of a content management site. In addition, you can enable features such as Oracle WebCenter Sites: A/B Testing, and Conversions, which help your content contributors and marketers measure website successfulness so they can enhance it accordingly.

Who Are the WebCenter Sites Administrators?

There are three different types of administrators for WebCenter Sites: general, site, and workflow. A general administrator has full access while site and workflow administrators access are restricted to their specific functions.

- **General administrator**, anyone who 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 Exploring Content Management Concepts.
- **Site administrator**, anyone who manages only certain sites on the management system.
- **Workflow administrator**, anyone who creates workflow processes on the management system.

Each administrator is given permissions to relevant parts of the WebCenter Sites Admin and WEM Admin interfaces.

While some concepts of this guide are specific to a certain type of administrators, the general administrator should know all of the concepts.

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

Prerequisites for Administrators

As a WebCenter Sites administrator, you must know about the various interfaces, system defaults, content management, your collaborators, online sites specifications, and sites environments.

The following list summarizes the information you must know or be familiar with:

- Oracle WebCenter Sites Admin interface and system defaults

  The WebCenter Sites Admin interface provides you with site configuration tools and access to system defaults such as ACLs, roles, and asset types. See Working with the Admin Interface, System Defaults, and WebCenter Sites Database Tables.

- Oracle WebCenter Sites WEM Admin interface

  The WebCenter Sites WEM Admin interface provides you with site configuration tools which enable you to create and manage sites, users, roles, and applications. See Using the Web Experience Management Framework.

- 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

  See Collaborating with Your Content Management Team and Migrating Your Work to Your Content Management System in *Developing with Oracle WebCenter Sites*.

- 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. See Oracle WebCenter Sites Environment, Roadmap for Administering WebCenter Sites, and The WebCenter Sites Development Process in *Developing with Oracle WebCenter Sites*.

What Tasks Do Administrators Perform to Manage Websites?

As an administrator, you’re responsible for creating the foundation of the website by defining the users and basic elements (assets and asset types). As the website is maintained, you continue to maintain users and roles, as well as set up and maintain workflow, blogs, and the regular publishing of website updates.

You maintain users and roles, set up and maintain workflow, blogs, and the regular publishing of website updates.

In WebCenter Sites, an administrator's duties vary from installation to installation. In general, the administrator is responsible for performing the following tasks:

- Collaborating with Your Content Management Team
- Managing Users
- Managing Access Control Lists (ACLs) and Roles
- Managing Asset Types and Assets
- Managing User Access to Applications
- Managing Workflow
- Managing the Publishing System
- Administering the Performance of WebCenter Sites and Its Components
- Integrating WebCenter Content with WebCenter Sites

Collaborating with Your Content Management Team

Before development begins, you typically collaborate with a team comprised of many different specialists, for example:

- Site designers
- XML and JSP developers
- Java application developers
- Database administrators
- System network administrators
- Marketers
- 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, such as Page design, Caching strategies, Security strategies, data models, Page layout and content, and content management models.

Managing Users

WebCenter Sites administrators are responsible for creating users and enabling user access to the appropriate content management sites and interface functions. For each user, the administrator must define a user account, user profile, user attributes, roles, and access control lists (ACLs).

The user account gives the user access to the WebCenter Sites system and its database tables. The user profile is required for users who are working with WebCenter Sites applications, language packs, and workflow processes. The user attributes are used in those cases where actions and events are necessary beyond workflow. Roles are used to associate the user with a site and allow the user to log in to the site. If a user's assigned roles match the roles assigned to an application, the user gains access to the interface function associated with that role. ACLs are used to regulate access to WebCenter Sites database tables.

When you create a user, you must associate that user with a content management site and its associated applications (user interfaces) by means of roles. This enables the user to log in to the site and access the appropriate interfaces and interface functions associated with that site.

See Configuring Users, Profiles, and Attributes.

Managing Access Control Lists (ACLs) and Roles

The WebCenter Sites administrator assigns Access Control Lists (ACLs) and roles to users to enable access to content management sites and to perform content management functions within the WebCenter Sites interfaces (Admin, Community, and Contributor). WebCenter Sites uses both ACLs and roles to support a variety of users with clearly defined job descriptions who each contribute the parts necessary (from creating articles to developing page layouts) for building your company's website.

ACLs control user access to the database tables that represent interface functions and are defined in the user definition (user's account). ACLs serve as the foundation of the security and user management model in your Sites system by providing authorization functionality. 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. Roles control user access to the interface functions represented by database tables. They represent a job description or the title of individuals granted access to similar functions (for example, content contributor, editor, designer, and so on). As an administrator, you must verify that the user attempting the interface function is assigned the same ACL as the database table associated with it.

See Working with ACLs and Roles.

Managing Asset Types and Assets

Once you assign users to the site, you must then enable the asset types (content types) those users will be working with. Your developers create the asset types available on your system. It's your job to enable the asset types on the appropriate sites. Once you enable an asset type on a site, the asset type is made available to site users through interface functions (such as the menu bar in the Contributor interface), granted by the user's assigned roles.

In addition to enabling asset types and using roles to grant users permissions to interface functions, WebCenter Sites provides an additional level of security, called Access Permissions, which allows you to set permissions on a per-asset basis. Access Permissions enable you to define which roles have permissions to perform which functions on an asset. For example, you can specify which roles can perform the Copy function on an asset and which roles are denied the ability to copy an asset.

See Enabling and Managing Asset Types and Managing Access to Content Management Site Components.

Managing User Access to Applications

After creating sites and users, and enabling asset types for users to work with on those sites, you must enable the interfaces (for example, Contributor) and other applications WebCenter Sites users can access on the sites you assigned them to. For this task, you use the WEM Admin interface. The WEM Admin interface extends the use of sites and roles to control access to applications. Roles are used within applications to protect interface functions and regulate access to application content. From the WEM Admin interface, you assign applications and users to sites using roles. When a user and an application are assigned the same role, the user is granted access to the application on that site.

See Using the Web Experience Management Framework.

Managing Workflow

When you create a workflow process, you determine the tasks involved to move the asset from its creation point on the content management site to its publishing destination on the delivery system. Each task you include in the workflow process must be assigned a role, or group of roles, necessary for a user to complete it. When an individual asset is going through a workflow process, only the users who have the appropriate role are allowed to complete the task. 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. The movement of an asset between states is called a step.

See Creating and Managing Workflow Processes.

Managing the Publishing System

One of your main goals as a WebCenter Sites administrator is to make site content (created on the management system) accessible to website visitors (on the delivery system). To accomplish this, WebCenter Sites provides a variety of publishing features in the Admin interface that enable you to configure and monitor the publishing system.

> **Note:**
>
> If you assign a content contributor the Publisher role, that user can publish individual assets directly from the Contributor interface. See Approving and Publishing Content in *Using Oracle WebCenter Sites*.

As the administrator, you configure the publishing destinations for your system, and schedule publishing events. In addition, WebCenter Sites provides the Approval system which is used to approve content for publishing to the website. When a user approves content for publishing, that content is made available for the next publishing event. When a publishing event runs, all the approved content on your management system is made available on the website.

By default, WebCenter Sites supports the RealTime publishing method, which is a dynamic method of publishing content to the website. Dynamic publishing is the process of copying assets and their underlying schema from one WebCenter Sites system to another. See Using RealTime Publishing.

Administering the Performance of WebCenter Sites and Its Components

WebCenter Sites Administrators configure log files for WebCenter Sites in Enterprise Manager and enable logging of these types of messages: configuration, debug, and trace. Administrators are responsible for starting and stopping Visitor Services in Enterprise Manager, checking the status of these components, and monitoring their performance metrics.

For information about configuring and viewing log files, starting and stopping Visitor Services, and monitoring the performance of these components, see Administering WebCenter Sites and Its Components in Enterprise Manager.

Integrating WebCenter Content with WebCenter Sites

If you want to enable content integration between WebCenter Content and WebCenter Sites, you must first enable the **SitesIntegration** component and the **ConnectorAdmin** tree node. Once completed, you can then set up content synchronization between WebCenter Content and WebCenter Sites. This allows for you to synchronize content items from WebCenter Content in WebCenter Sites, where they are then managed by your content contributors as native assets in WebCenter Sites. These assets can be edited, deleted, previewed, revision tracked, published, and otherwise treated like any other asset that is created directly in WebCenter Sites.

See Integrating with Oracle WebCenter Content.

Sample Sites Used In This Guide

As part of your installation, we provide two sample sites (Avisports and FirstSiteII) to illustrate and help you understand how to use content management functions in the WebCenter Sites Contributor, Admin, and WEM Admin interfaces.

A user can access these sample sites only if they were installed with your WebCenter Sites system or if he is working with a JSK. Otherwise, the user can contact the WebCenter Sites Administrator.

The sample sites used in this guide are:

- **avisports:** A sports-centric sample site containing sports articles illustrated with images. In this guide, avisports illustrates the features in the interface such as creating and editing assets, linking assets, approving assets for publishing, finding and organizing assets, tracking assets with revision tracking, and creating a site's navigation.

- **FirstSite II:** An online electronics retail site that includes articles, images, a product catalog with several entries, documents, and sample user accounts. In this guide, FirstSite II illustrates workflow, multilingual, and preview features in the interface. FirstSite II is also used to describe and the process of creating and working with segments, recommendations, and promotions in the interface.

2
Exploring Content Management Concepts

As a general administrator, you're responsible for configuring and managing sites using Content Management (CM) sites. Let's get started with understanding CM sites and their relation to the online site.

The following topics provide overview information about content management concepts:

- Introduction to Oracle WebCenter Sites Concepts
- Websites
- Content Management Sites
- Configuration Components
- Content Management Models
- Oracle WebCenter Sites Environment

Introduction to Oracle WebCenter Sites Concepts

Learn about configuring the business users, linking them to content-entry forms and other authoring tools, and providing the users with publishing and delivery systems for serving the online site to browsers. This is required to build the foundation of your website that is itself a site or a collection of sites.

When you've configured the site foundation, users can log in to and create and manage content. The content they enter is saved to tables in the database. When ready for delivery, the content is pulled from the database tables, formatted, and served to browsers as the website.

defines the smallest unit that can function as a back end as a CM site. The CM site is where users create content that is saved to the database and invoke the publishing-delivery systems. 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 administrator, you work with CM sites—configuring them, replicating them, and managing their day-to-day use. Because they are a critical part of your work, you must understand how CM sites are defined, how they must be configured, and how they relate to the website.

Websites

An online site powered by Oracle WebCenter Sites is the set of pages that the site visitors interact with. The online site can be either password protected or made accessible to the general public. It can also be an exclusive site, such as a corporate network or departmental network, operating strictly within the private domain.

Online sites can be modeled by CM sites. 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.

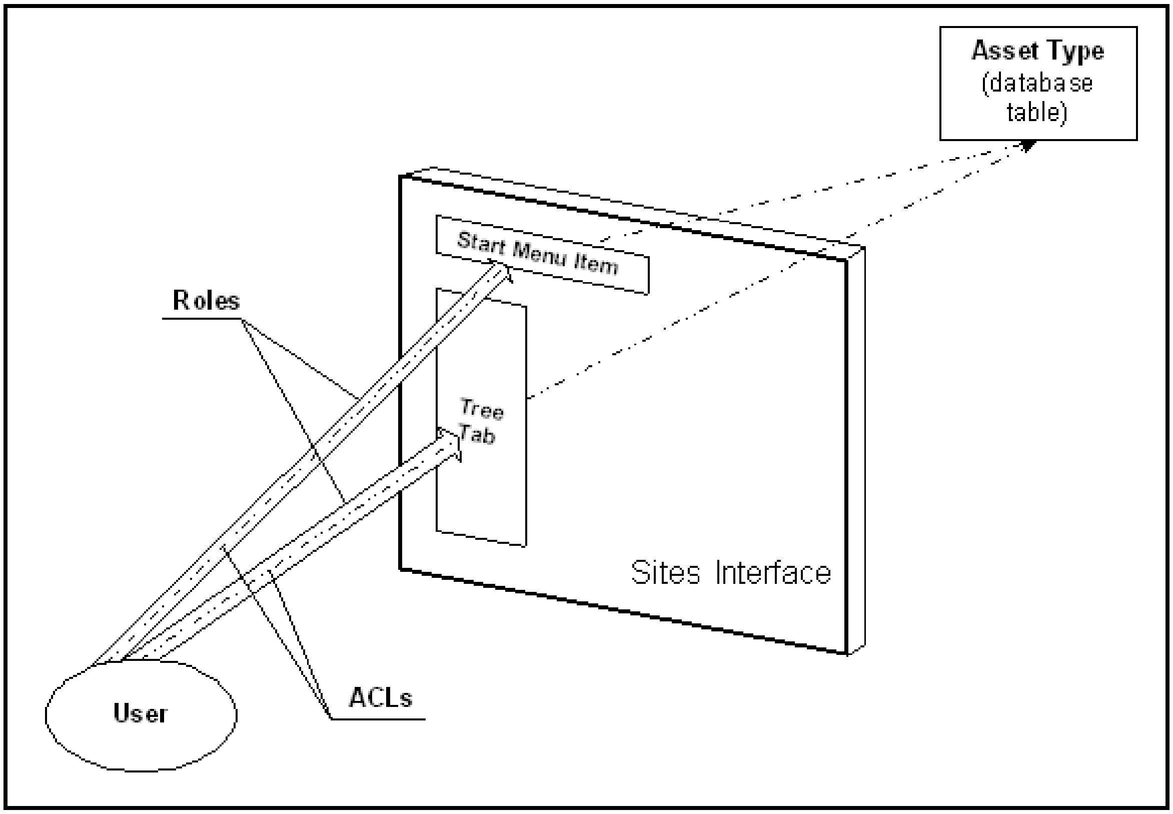
Content Management Sites

A Content Management (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 to first define the authors and managers of the online site, and then provide them with the permissions and content management tools they require: 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 the following figure.

Figure 2-1 User Roles and ACLs



The site is identified by a name (also configured by the administrator), stored in 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 to author and manage specific types of content.

An Oracle WebCenter Sites system typically has many such sites, each one unique in its function and composition. When users log in to 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, 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 WebCenter Sites environment.

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 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 Content Management Models.

Configuration Components

A Content Management (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.

The following table lists site components, most of which are required. The optional components vary according to business requirements 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 Building Your Data Model in *Developing with Oracle WebCenter Sites.*.

Table 2-1   Site Components

Component	Required	Created by
CM site definition	Yes	WebCenter Sites administrator
Data Model	Yes	Developers
Users	Yes	WebCenter Sites
Roles	Yes	WebCenter Sites administrator
Start menu items	Yes	WebCenter Sites
Workflow processes	No	WebCenter Sites administrator
Publishing system	Yes	WebCenter Sites
User interface options	No	WebCenter Sites

More information about these site components are provided in the following sections.

Content Management Site Definition

A CM site definition is a component 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, WebCenter Sites creates a node to represent the site in the Admin interface. It also appends default sub-nodes for linking components to the site, such as an Asset Types sub-node for linking the data model to the site, and a Users sub-node for linking users to the site.

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

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, 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, whose fields prompt users for information that is 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 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 several default asset types and allows developers to create their own. The asset types, their definitions, and the assets themselves are stored in the WebCenter Sites database as tables or table entries, and loosely referred to as content in this guide.

See Building Your Data Model in *Developing with Oracle WebCenter Sites*. We recommend that administrators read the guide to gain a basic understanding of asset types.

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 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. These users might have permission to publish assets, on-demand, from the Contributor interface.
- Content managers, who oversee the authoring, review, and publishing processes.

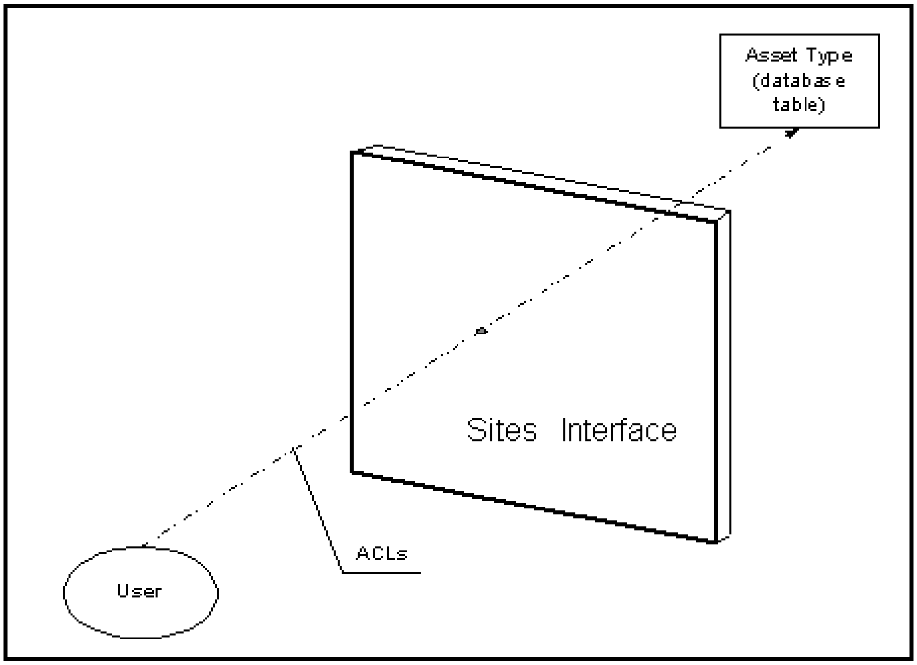
Each content provider must be identified to WebCenter Sites through a user account, which consists of a user name, a password, and Access Control Lists (ACLs), the foundation of Oracle WebCenter Sites security system.

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 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.

Figure 2-2 User ACLs



In general, WebCenter Sites uses ACLs at two levels:

- At the system security level, to provide authentication functionality and therefore prevent hackers from entering the 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.

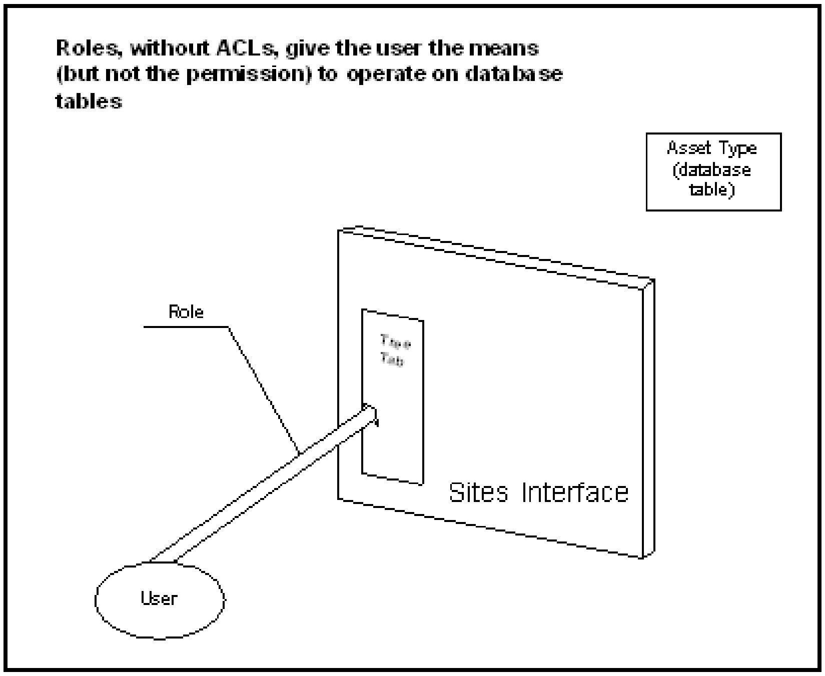
WebCenter Sites provides several default system ACLs and pre-assigns them to system tables. You can reuse the ACLs, or configure your own and assign them to custom tables, as necessary. For more information about ACLs, see Working with ACLs and Roles and System Defaults.

Users and 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.

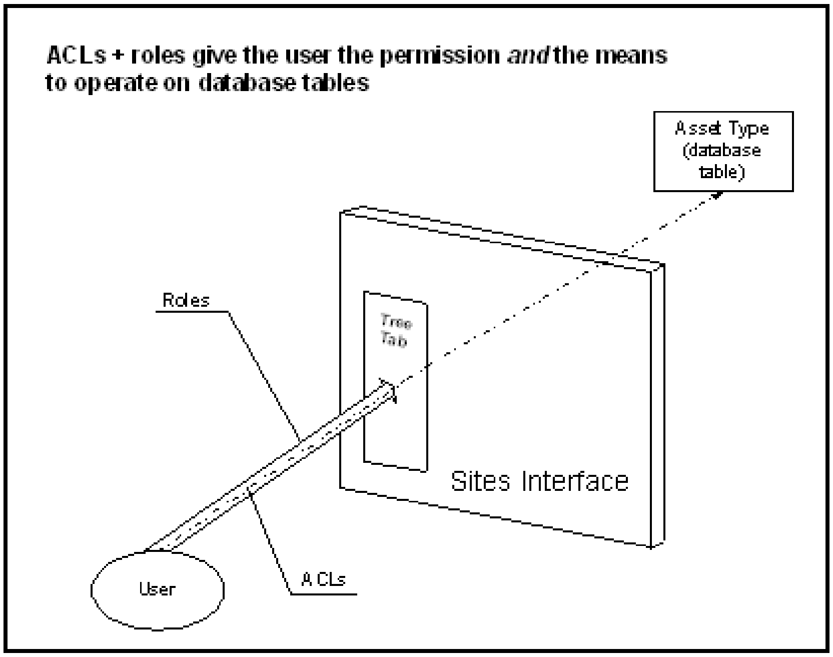
Roles are implemented in the same way as ACLs; that is, for a function to be displayed in the user's interface, the function and the user must be assigned the same role. To illustrate, we continue our previous scenario, where a user is given editorial permissions to the `SystemUser` table through the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

Figure 2-3 User Roles


Roles, without ACLs, give the user the means (but not the permission) to operate on database tables
Asset Type (database table)
Role
Tree Tab
Sites Interface
User

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** node. 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 2-4   User Roles and ACLs


ACLs + roles give the user the permission *and* the means to operate on database tables
Asset Type (database table)
Roles
Tree Tab
Sites Interface
User
ACLs

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 must have access to the **Admin** node. All the administrators can be assigned the `GeneralAdmin` role and thereby be given access to the **Admin** node.

WebCenter Sites defines several default system roles, all of which are pre-assigned to various functions in the Oracle WebCenter Sites interfaces. 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.

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.

defines two start menus: New, which allows the user to create assets on the site, and Search, which allows the user to look for and edit assets on the site.

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.

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

Workflow Processes

Workflow processes are optional constructs, mandated only by business requirements, 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. When approved, the asset is published to the delivery system by users and finally, delivered as content to the online site.

When 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 called by users as necessary. For more information about workflow processes, see Creating and Managing Workflow Processes.

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.

User Interface Options

WebCenter Sites supports an optional interface through which users can interact with content.

This optional interface is the Web Mode of the Oracle WebCenter Sites: Contributor interface, which supports the editing of content directly on the rendered page. When Web Mode is enables, users can make quick edits to content while viewing how the content will be displayed on the website once it is published.

Content Management Models

As the administrator, you must decide how to model the online site in the Oracle WebCenter Sites interfaces: as a single site or a set of sites. When making this decision, whether alone or with collaborators, be sure to consider the size of your online site and the nature of its content.

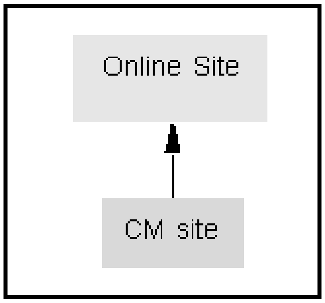
The following topics provide information about the content management models you can choose from:

- 1:1 Model
- 1:n Model
- x:n Model

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 2-5 1:1 Model

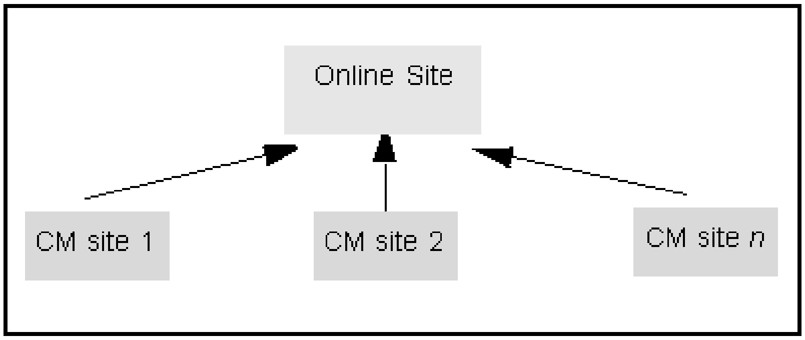


The 1:1 model works efficiently for small online sites, where content tends to be limited, uniform, and managed by few users. For enterprise-level sites, however, pages often number in the millions and differ significantly in subject matter, content, presentation, and scope. Organizing large sites requires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

1: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 2-6 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 requirements 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 must 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.)

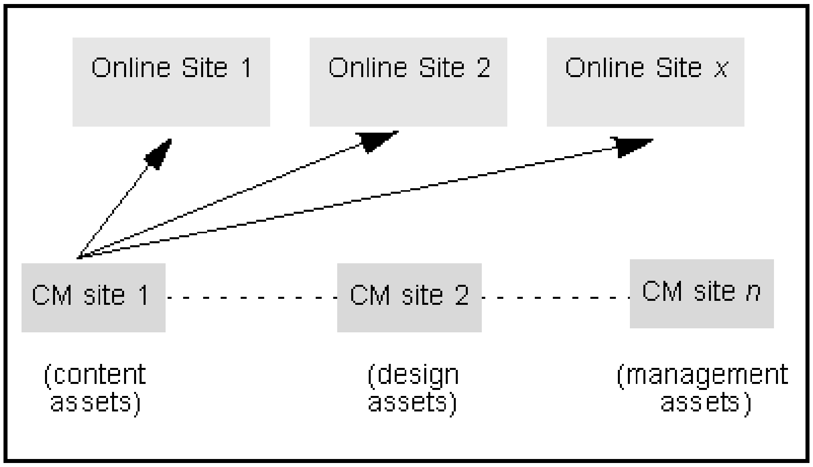
x:n Model

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 Logging In to Oracle WebCenter Sites.

Figure 2-7 x:n Model

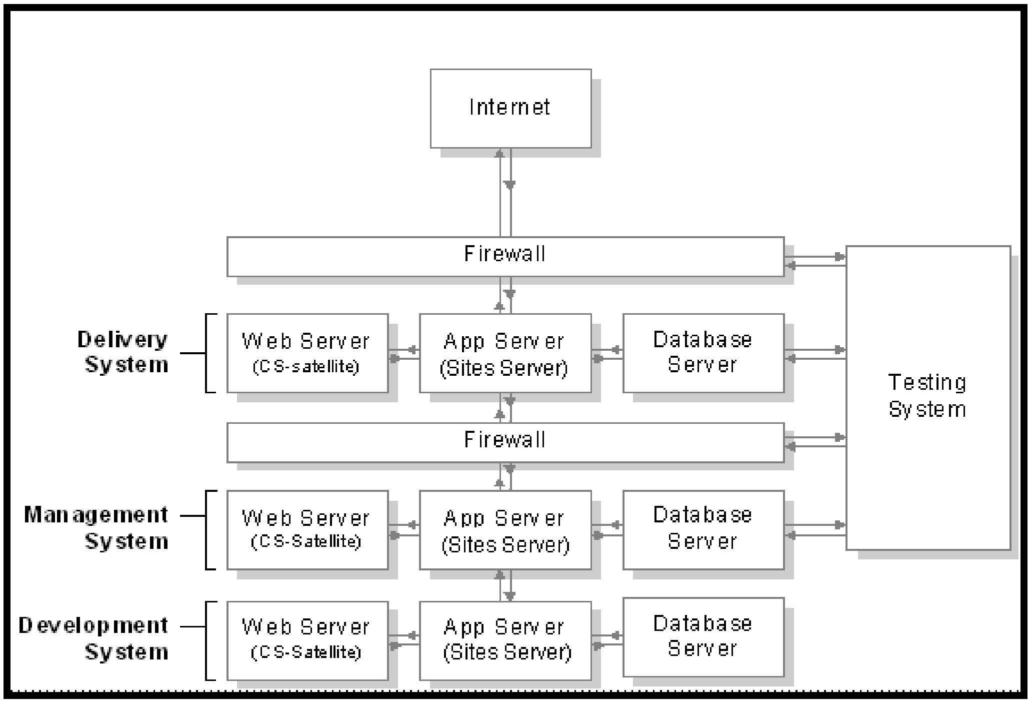


Oracle WebCenter Sites Environment

Oracle WebCenter Sites provides an environment rather than an isolated WebCenter Sites system. You can collaborate with developers and users to configure sites.
An enterprise-level environment typically consists of four different WebCenter Sites-powered systems: development, management, delivery, and testing, which run on their own database and at some point interacts with the other systems.

As the administrator, you tend to work with all of the systems at one time or another. Testing is an optional system that large organizations typically install for increased quality assurance. 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 2-8 Oracle WebCenter Sites Environment


Internet
Firewall
Delivery System
Web Server (CS-satellite)
App Server (Sites Server)
Database Server
Testing System
Firewall
Management System
Web Server (CS-Satellite)
App Server (Sites Server)
Database Server
Development System
Web Server (CS-Satellite)
App Server (Sites Server)
Database Server

Note:

The names of the systems in your WebCenter Sites environment can 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 is 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.

3 Logging In and Getting Started

You can log in as an administrator and set up accounts, as well as how to reset passwords for users.

- Logging In to Oracle WebCenter Sites
- How to Reset a WebCenter Sites Password
- How to Edit Your Oracle WebCenter Sites Profile
- Logging Out of WebCenter Sites

Logging In to Oracle WebCenter Sites

It's the responsibility of the general administrator to first set up an administrator account for themselves using the default system account. They must then set up accounts for all users based on their specific roles in the organization.

The following instructions show you how to log in to the Oracle WebCenter Sites Admin and WEM Admin interfaces as the general administrator.

> **Note:**
>
> If you are not the general administrator and must 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 your permissions to the Oracle WebCenter Sites interfaces are limited by your role.

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 system running WebCenter Sites, `<port>` is the port number for the server, and `<context>` is the context path of the 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.

   The WebCenter Sites login form opens.
4. Enter your user name and password into the login form.

> **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 Creating a User in the Admin Interface.
>
> As you have logged in as 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. For more information about user accounts and ACLs, see Configuring Users, Profiles, and Attributes.
>
> If you create general administrators then they must have the `xceladmin` ACL assigned to them, and must be part of the `REST_Admin` security group.

5. If you want WebCenter Sites to store your credentials and enter them automatically the next time you log in, select **Remember me**.
6. Click **Login**.

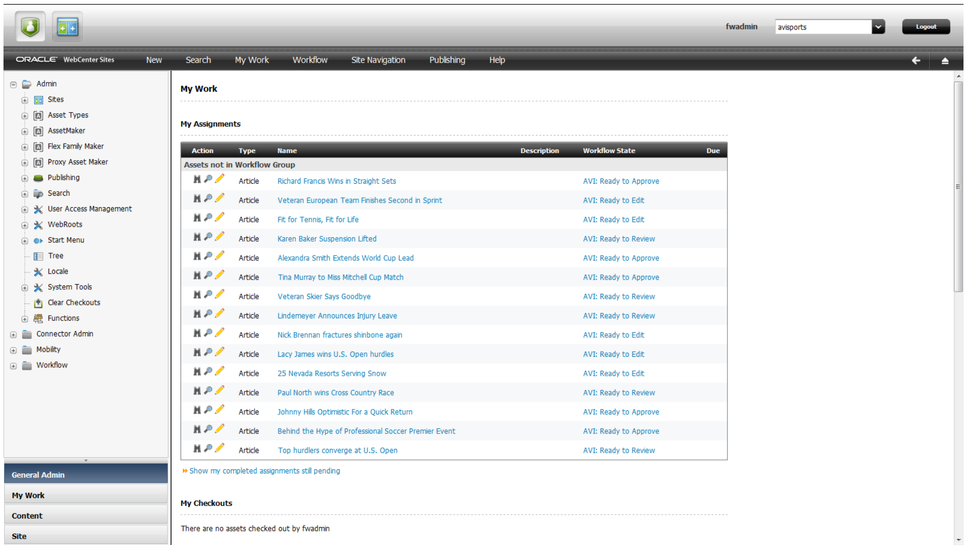
   If you are logging in for the first time, the site select form opens (see step 7); otherwise, you are logged in to the site and application you last visited.
7. In the **Site** drop-down list, select the site hosting the application in which you want to work. In this example, select **AdminSite**.

   When you select a site, the applications assigned to that site are displayed as icons under the **Site** field.
8. Select the icon representing the application you want to work with:
   - If you are a general administrator logging in to the AdminSite, you will see the **WEM Admin** and **Admin** icons. Select **WEM Admin** to access the WEM Admin interface. Select **Admin** to access the Admin interface.
   - If you are a general or site administrator logging in to a site other than AdminSite, you will see the **Admin** and **Contributor** icons. (You may also see other interface icons depending on which applications are enabled on the site.)
9. You are now logged in to the site and application you selected.
   - If you selected the Admin interface, see If You Selected the Admin Icon.
   - If you selected the WEM Admin interface, see If You Selected the WEM Admin Icon.

If You Selected the Admin Icon

When you select the **Admin** icon, the Admin interface loads (shown in the following figure). As an administrator, you can perform all administrative tasks from this interface.

Figure 3-1 Admin Interface



My Work

My Assignments

Action	Type	Name	Description	Workflow State	Due
Assets not in Workflow Group					
	Article	Richard Francis Wins in Straight Sets		AVI: Ready to Approve	
	Article	Veteran European Team Finishes Second in Sprint		AVI: Ready to Edit	
	Article	Fit for Tennis, Fit for Life		AVI: Ready to Edit	
	Article	Karen Baker Suspension Lifted		AVI: Ready to Review	
	Article	Alexandra Smith Extends World Cup Lead		AVI: Ready to Approve	
	Article	Tina Murray to Miss Mitchell Cup Match		AVI: Ready to Approve	
	Article	Veteran Skier Says Goodbye		AVI: Ready to Review	
	Article	Lindemeyer Announces Injury Leave		AVI: Ready to Review	
	Article	Nick Brennan fractures shinbone again		AVI: Ready to Edit	
	Article	Lacy James wins U.S. Open hurdles		AVI: Ready to Edit	
	Article	25 Nevada Resorts Serving Snow		AVI: Ready to Edit	
	Article	Paul North wins Cross Country Race		AVI: Ready to Review	
	Article	Johnny Hills Optimistic For a Quick Return		AVI: Ready to Approve	
	Article	Behind the Hype of Professional Soccer Premier Event		AVI: Ready to Approve	
	Article	Top hurdlers converge at U.S. Open		AVI: Ready to Review	

Show my completed assignments still pending

My Checkouts

There are no assets checked out by fwadmin

This figure shows the main parts of the Admin interface:

- The menu bar, at the top of the window, contains the following content management options:
  - **New**: Enables you to create assets whose asset types are enabled on the selected site. The Admin interface supports the creation of only high-level assets such as Templates and CSElements. All other types of assets are created in the Contributor interface. See What Can You Do in the Contributor Interface? in *Using Oracle WebCenter Sites*.
  - **Search**: Enables you to search for all high-level assets created for the current site. To search for content assets, you must access the Contributor interface.
  - **My Work**: Provides information about your workflow assignments, checkouts, and bookmarked assets.
  - **Workflow**: Enables you to create a workflow report, create a workflow group, view existing workflow groups, and view saved workflow reports. See Creating and Managing Workflow Processes.
  - **Site Navigation**: Available only if the site contains a site navigation. This option enables administrators to rename site navigations, re-order site navigations (shown on the **Site Tree** in the Contributor interface), and place and unplaced pages in a site navigation.
  - **Publishing**: Enables administrators to select a publishing destination, and view active publishes, scheduled publishes, and publish history.
- The navigation pane is located to the left of the window and allows you to perform administrative functions from the available tabs. Select a bar at the bottom of the pane to select a tree. Most administrative tasks can be performed from the **General Admin** tree. The **General Admin** tree contains data for all sites in the system, not just the site you are logged in to.
- The main window of the Admin interface is where you perform administrative functions. Any option you select from the menu bar or site navigation pane is displayed in the main window of the Admin interface. For example, when you

select **Search** from the menu bar, the Search form opens in the main window. See Administrator-Specific Trees and Nodes.

- The applications bar (at the top of the interface) shows the icons of the applications available to you on the current site, a link to your profile, a menu of sites accessible to you, the Logout button, and the Up/Down arrow you use to expand and collapse the applications bar.

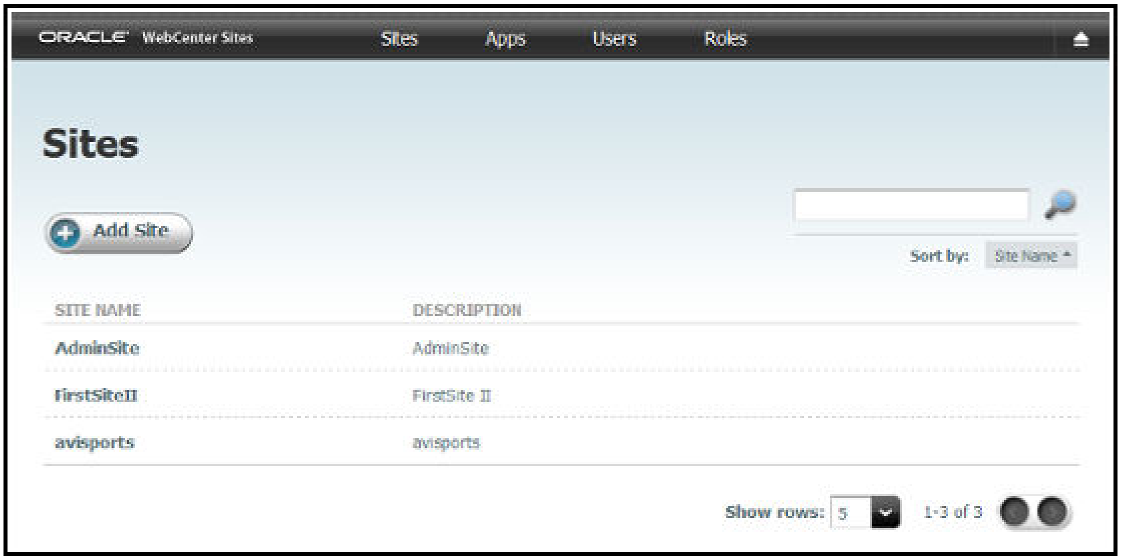
  When running more than five applications, the **Switch Apps** drop-down menu is shown for selecting the additional applications. When you select an application from the **Switch Apps** menu, the icon of the selected application replaces the icon of the least-used application's icon currently shown.

If You Selected the WEM Admin Icon

If you are a general administrator and you selected the WEM Admin icon, the WEM Admin interface loads. The WEM Framework determines your status as a general administrator or site administrator and enables only the forms and interface functions that you must work with.

The following figure shows the Sites form in the WEM Admin interface.

Figure 3-2 WEM Admin Interface: Sites Form



SITE NAME	DESCRIPTION
AdminSite	AdminSite
FirstSiteII	FirstSite II
avisports	avisports

The menu bar shown in this figure contains the following options you can use to manage your system's sites, applications, users, and roles:

- **Sites**: Opens the Sites form which lists all the sites in the system. Site administrators 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.
- **Apps**: Accessible only to general administrators. Opens the Apps form which enables you to edit application information (tooltip and description) and manage the roles assigned to assign an application to sites.

- **Users**: Accessible only to general administrators. Opens the Users form which enables you to create and manage users, including assigning users to sites.
- **Roles**: Accessible only to general administrators. Opens the Roles form which enables you to create and manage roles for your content management system.

See Using the Web Experience Management Framework.

How to Reset a WebCenter Sites Password

It's not unusual for users to forget their login passwords. As a general administrator you can reset these passwords.

To reset a user's password:

1. Log in to WebCenter Sites, select the **AdminSite**, and then select the **WEM Admin** icon.
2. In the WEM Admin interface, select **Users** from the menu bar.
3. In the Users form, point to the name of the user whose password you need to reset, the select **Edit**.
4. In the Edit User form, fill in the following fields:
   - **New Password**: Enter a password for the user.
   - **Confirm Password**: Re-enter the password you entered in the New Password field.
5. Click **Save**.

   You have successfully changed the user's WebCenter Sites password.
6. Notify the user about the new password. It is recommended the user log in with the new password and then immediately change it for security purposes. For instructions, point the user to Logging In in *Using Oracle WebCenter Sites*.

How to Edit Your Oracle WebCenter Sites Profile

Each WebCenter Sites user has a WebCenter Sites profile, which contains information about their user name, email, locale, time zone, preferred devices, and password.

1. Log in to WebCenter Sites, select a site, and an application icon.
2. In the menu bar, click the **down-arrow** to open the applications bar.
3. In the applications bar, click your user name.
4. In the Edit Profile form, you can make the following changes:
   - To change your profile image, in the **Image Preview** field, click **Browse** and then select the image you want to assign to your profile.
   - To add or change your email address, use the **Email** field.
   - To change your assigned locale, use the **Locale** drop down field.
   - To change your assigned time zone, use the **Time Zone** drop-down field.
   - To change your password, do the following:
     a. In the **Old Password** field, enter your current password.
     b. In the **New Password** field, enter your new password.

   c. In the **Confirm Password** field, enter your new password.

5. Click **Save and Close**.

   The Edit Profile form closes and the interface from which you accessed your profile is re-displayed.

Logging Out of WebCenter Sites

You can log out of WebCenter Sites to end your session or to log in as a different user.

Do the following:

1. If the applications bar is not open, click the down-arrow, located on the right of the menu bar, to open it.
2. Click **Logout**.

   You are now logged out of .

4 Roadmap for Administering WebCenter Sites

Although the order in which you complete each task in the site configuration process is flexible, we've provided a possible workflow that you can use. This section describes each of the tasks in detail as well as the information you need to obtain and decisions that must to be made before you begin the process.
When you have a better understanding of how sites are configured, you can modify the workflow to suit your requirements. One possible order is given in the proceeding procedure starting with Making Pre-Configuration Decisions.

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 Developing a Website in *Developing with Oracle WebCenter Sites*) where you can find detailed information about the particular step.

The following topics provide information about the steps administrators should follow to configure all the WebCenter Sites systems:

- Making Pre-Configuration Decisions
- Understanding Your Job On Each WebCenter Sites System
- Moving from a Test to a Production Environment
- Common Tasks for WebCenter Sites Administrators

Making Pre-Configuration Decisions

Before you begin working on any of the tasks in the site configuration process, there are a number of decisions that must be made first. What data model are you using? Who are the users? What assets need to be created? How about security? We've provided you with a list of all the questions that require answers before you can think about starting the site configuration process.

We assume that you and your 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'll complete the site configuration tasks, test the management system, and finally deploy the system.

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

- Determine the 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 be copied. Do the same for Template assets.
- Determine whether you are replicating the sites you are configuring:

If you plan to replicate the sites you are configuring, ensure template-coding and other requirements are satisfied for the source sites (the sites you are currently configuring). See Replicating Content Management Sites.

- Determine 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 must be created by developers?
- 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. See Users, Sites, and Roles in LDAP-Integrated Sites Systems. For information about configuring users in LDAP, see the LDAP product documentation.
- Determine user management models:
  - How many users are required.
  - Who the users are.
  - What permissions the users must be given to database tables:

    Regardless of which user management method you are 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 must create additional ACLs. Typically, the system ACLs are sufficient. For general information about ACLs, see Working with ACLs and Roles. For specifications, see System Defaults.
    - Which system ACLs the user must be assigned (without ACLs, no user accounts can be created).
    - Whether custom ACLs must be created.

      (Note that you may also have to create ACLs to assign to visitors of the website to restrict them from accessing certain database tables.)
  - How many roles are required:

    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 must create roles to cover the full range of users' responsibilities on any given site.

    For information about roles, see Exploring Content Management Concepts and Working with Roles for general information.
    - The types of roles that are required (writers, editors, illustrators). Ensure the roles are compatible with the users' ACLs.
    - To which users the roles should be assigned.
- Determine system security settings:

  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. See Setting Up External Security.

- Determine 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 require workflow processes
  - How must the workflow processes by designed
  - Which users and roles must participate in the workflow processes
- Determine publishing system options:
  - Which type of publishing must be configured
  - Whether publishing schedules must be approved
- User interface options. Determine which interfaces the user will require:
  - Contributor (Form Mode and Web Mode)
  - Admin
  - WEM Admin

Before beginning the configuration tasks, read Exploring Content Management Concepts for a general understanding of WebCenter Sites concepts, components, and content management models.

Understanding Your Job On Each WebCenter Sites System

Now that all the pre-configuration decisions have been established, you're ready to begin configuring your sites. As the administrator, you have a specific set of jobs to complete on all four systems (development, management, delivery, and testing); however, you'll find that you're going to spend most of your time on the management system.

When the specifications of the website and the content management sites are established, you are ready to begin your work. While developers are responsible for the data model and other code-based components, you are responsible for all other aspects.

As the administrator, you work on all systems in the WebCenter Sites environment during the development stage. After the website 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.

The following topics provide information about the jobs you perform on each WebCenter Sites system:

- Configuring the Development System
- Configuring the Management System
- Configuring the Delivery System
- Configuring the Testing System

Configuring the Development System

The development system is where coding takes place. Configuration steps at the development system are also given in detail in Coding with Developer Tools in *Developing with Oracle WebCenter Sites*.

When coding at the development system is complete, the data model is migrated to the management system, where site configuration is completed.

Your job as the administrator is to provide developers with the specifications of the management system and to assist with content management operations, such as creating sites.

The developer’s job is to:

- Create sites with the same names as those that are used on the management system. See 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 about this, see Configuring 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 Creating Collection Assets, Query Assets, and Page Assets in *Developing with Oracle WebCenter Sites*.
- Test the data model in accordance with your company's QA requirements.
- After the data model is complete and tested to the satisfaction of all collaborators, developers migrate the code to the management system.

  To migrate the data model, publish it to the management system. For instructions, see the following topics:

  – Understanding Publishing

  – Using RealTime Publishing

  – Scheduling Publish Events

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

After 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 requirements of the website visitors, content providers, marketers, and administrators.

Configuring the Management System

When the development system has been put into place, you can then configure the management system. Configuration tasks that require no coding are typically completed on 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 do not require 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.

On the management system, complete configuration steps in this order:

- Create the sites

  > **Note:**
  >
  > In this release of 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. See Replicating Content Management Sites.

- Configure the users (content providers and marketers):
  - Create their ACLs, as necessary. For procedures on creating ACLs, see 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.

  - Create user accounts, either through WebCenter Sites, or external user managers such as LDAP plug-ins. See Creating a User in the Admin Interface.
  - Create roles in WebCenter Sites, as necessary.
  - Create use profiles, if necessary:

    User profiles are required for users who are working with:

    * WebCenter Sites content applications (Oracle WebCenter Sites: Engage).
    * Language packs and setting a default language.
    * Workflow processes, in which email messages are 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. See Creating and Editing a User Profile.

- 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 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 must create workflow processes, do the following:

Table 4-1 Workflow for Asset Types

Workflow Process	Details of the Process
Create the Workflow tree node for the site, if one does not exist.	Creating a Tree Tab
Plan your workflow process by sketching it. Then see your sketch and notes throughout this section.	Creating and Managing Workflow Processes
If you have not done so, create the roles that are required for the workflow processes.	Creating a Role From the Admin Interface
Ensure that users are set up to participate effectively in the workflow: 1. 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. 2. 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.	Determining Roles and Participants
Create the email objects that you require for your actions and enable the `xcelerate.emailnotification` property in the `wcs_properties.json` file.	Exploring Content Management Concepts

Table 4-1 (Cont.) Workflow for Asset Types

Workflow Process	Details of the Process
Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions that you require.	Setting Up the Actions and Conditions
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 called in a timely way.	Setting Up the Timed Action Event
Create your states.	Setting Up the States
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 require.	Setting Up the Workflow Processes
Test your workflow processes.	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 Managing Start Menu Items.

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

  For information and procedures on creating sites, see Assembling Content Management Sites.

- Migrate the data model, if you have not done so:

  At this point, the data model is necessary 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.

- Assemble the sites:

  1. Grant users access to the sites. For instructions, see 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 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** 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 Managing Access to Asset Types Using Start Menus for additional instructions.
  2. Set access permissions on assets. For instructions, see How to Set Access Permissions to Assets.
  3. Give users access to the tabs in WebCenter Sites' tree. For instructions, see Options for Managing Access to the Tree (Admin Interface Only).
- Configure the WebCenter 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 about the RealTime publishing process, see the chapters on real-time publishing.
- Configure user interfaces:

  The user interface options are Oracle WebCenter Sites: Contributor, Admin, and WEM Admin. For instructions on making these interfaces available to content providers, see Configuring the User Interfaces.
- Enable asset types for search engines:

  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 Configuring the Lucene Search Engine.
- Set up revision tracking:

  Revision tracking prevents assets from being edited by multiple users 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 Managing Revision Tracking.
- Set up the delivery system. For more information, see Set Up the Delivery System in *Developing with Oracle WebCenter Sites*.
- Test all systems in accordance with your company's QA policies.

- Give users the URL and their log in information to the WebCenter Sites installation. After users log in, they can then control and manage the content as you've configured.

Configuring 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 WebCenter Sites installation. For Export to Disk and Export to XML delivery, the delivery system is simply the web server component of the WebCenter Sites installation.

Configuring the Testing System

The testing system is where QA analysts test the performance of both the management system and the delivery system, and 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.

Moving from a Test to a Production Environment

If your organization has more than one environment, you can switch one or more instances of WebCenter Sites or its component applications (Visitor Services, Satellite Server, or Site Capture) from a source environment to a target environment , like Test environment to Production environment.

See Moving from a Test to a Production Environment in *Administering Oracle Fusion Middleware*.

Common Tasks for WebCenter Sites Administrators

Once you’re in production, you’ll be performing most of your day-to-day tasks on the management system. For example, responding to user requests, adding new users, managing the publishing system, and so on.

In the management system, 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 requirements of online site visitors, content providers, and the e-business.

Part II

Managing the Site Foundation

The management of a site foundation involves assembling different site components to create a site and managing the different access rights for different users and roles. You can assist in these tasks to deliver a site foundation and keep it up and running.

Topics:

- Assembling Content Management Sites
- Working with ACLs and Roles

5 Assembling Content Management Sites

Your main goal as an administrator is to work with sites (create, edit, and delete sites); including managing site users (add, edit, and delete users) and asset types for the sites (enable and remove asset types). Content Management sites can be assembled after their basic components are created. The basic components are the users and the data model.

The following topics provide details about assembling content management sites:

- Working with Sites
- Assigning and Managing Site Users
- Enabling and Managing Asset Types

Working with Sites

Once you create a site, you must assign users and asset types, and configure site navigation, enable workflow processes, publishing destination, and so on.

The following topics provide instructions for creating sites, associating users to the sites, and assigning asset types to the sites:

- Creating a Site From the Admin Interface
- Obtaining Site Configuration Information
- Editing a Site in the Admin Interface
- Deleting a Site in the Admin Interface

Creating a Site From the Admin Interface

To create a site you specify a site name, description, and a method for previewing site content, as follows:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click **Add New**.

   The Add New Site form opens.

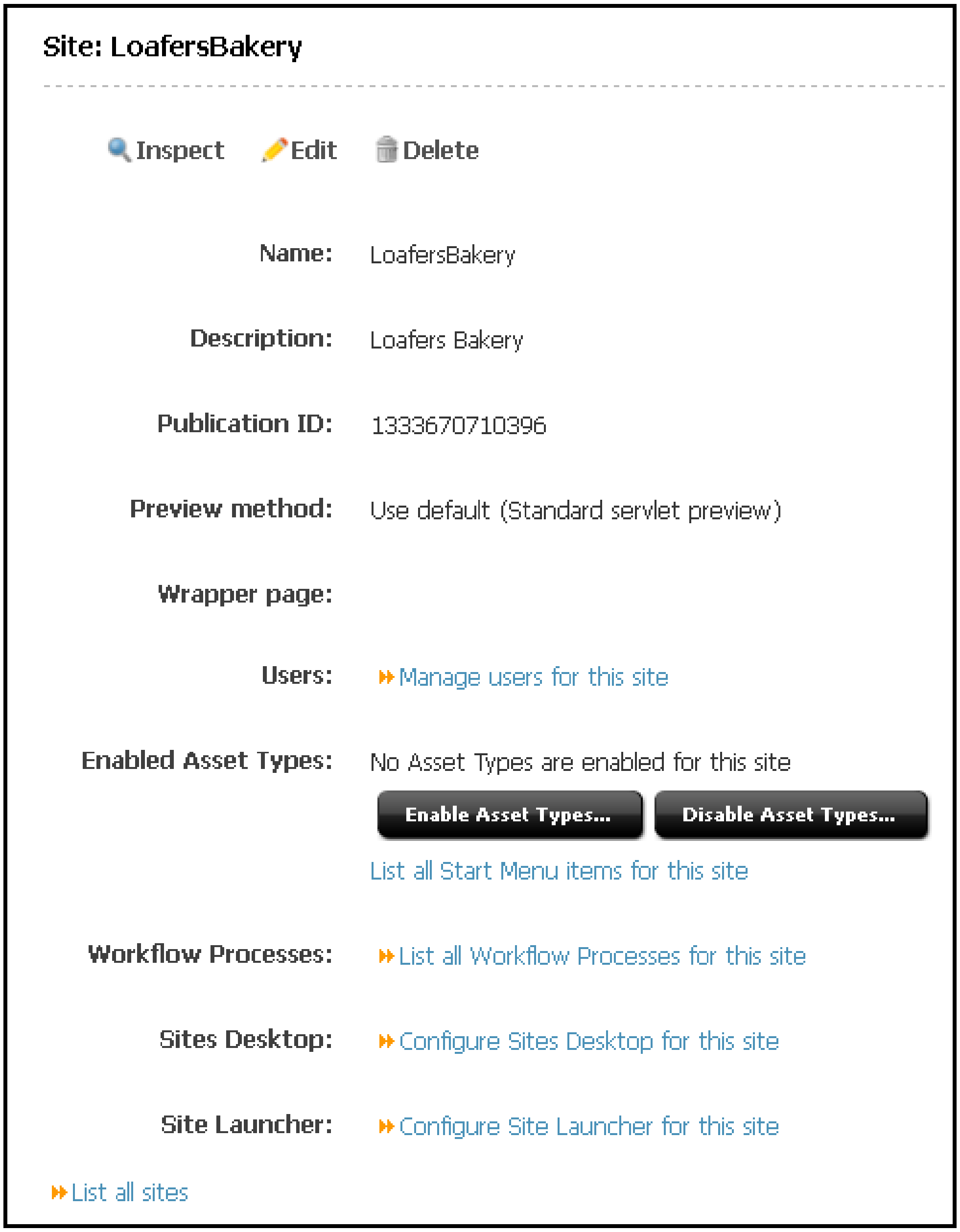
2. In the **Name** field, enter the name for the new site. Note the following conventions when naming the site:

   - The name must be between 1 and 255 characters.
   - The following characters are not allowed: single-quote ('), double quote ("), semicolon (;), colon (:), question mark (?), ampersand (&), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash.
   - The name cannot contain spaces.

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 site name is 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 opens.

Figure 5-1 Site Form



Site: LoafersBakery

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

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

This form is used to execute several operations. For example, inspect, edit, or delete the site, manage users, or configure Site Launcher to replicate the site.

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 show 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 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.

Obtaining Site Configuration Information

In certain situations, you will have 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 **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click the site whose configurations you want to view.

   The Site form opens.

2. In the Site form, select the option that provides the information you are looking for. For example:

   - In the **Workflow Processes** field, click **List all Workflow Processes for this site** to view the workflow processes associated with this site.

     In the Workflow Processes list, click **Add New Workflow Process** to configure an additional workflow process for the site.

   - In the **Site Navigation** field, click one of the following options:
     - Click **Manage Site Navigations** to view the site navigations associated with the site. Click **Add Site Navigation** to create a new site navigation for the site.
     - Click **Order Site Navigations** to rearrange the order in which the site navigations are displayed in the **Site Tree** in the Contributor interface.

- In the **Publish Destinations** field, click **Mirror site configuration for Delivery** to publish the content management site to the delivery system.
- In the **Site Launcher** field, click **Configure Site Launcher for this site** to share all site components with a new site.

Editing a Site in the Admin Interface

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

To edit a site description:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click the site you want to modify.

   The Site form opens.
2. In the **Site** form, click **Edit**.

   The Edit form opens.
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**.

Deleting a Site in the Admin Interface

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 and tree tabs are removed or unshared.

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

To delete a site:

1. In the **General Admin** tree, expand the **Admin** node , and then 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 shown to inform you that site components are deleted. The warning identifies the components and prompts you to confirm your intention to delete the site.

3. Click **Delete Site**.

Assigning and Managing Site Users

You can grant users access to a site, view current site users, change a users' role assignments for a site, and remove users from a site.

The following topics show you how to do such tasks:

- Granting Users Access to a Site (Assigning Roles to Users)
- Viewing Site Users and Reassigning Roles
- Deleting Users from a Site

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 a user's assigned role is also specified for an interface function on the site, the user gains access to the interface function.

> **Note:**
>
> It’s assumed you have created at least one user, a role, and a site. If you have to create one or another, follow the steps in Creating a User in the Admin Interface, Creating a Role From the Admin Interface, and Creating a Site From the Admin Interface.

To grant a user access to a site:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then expand the site.
2. Double-click **Users**.

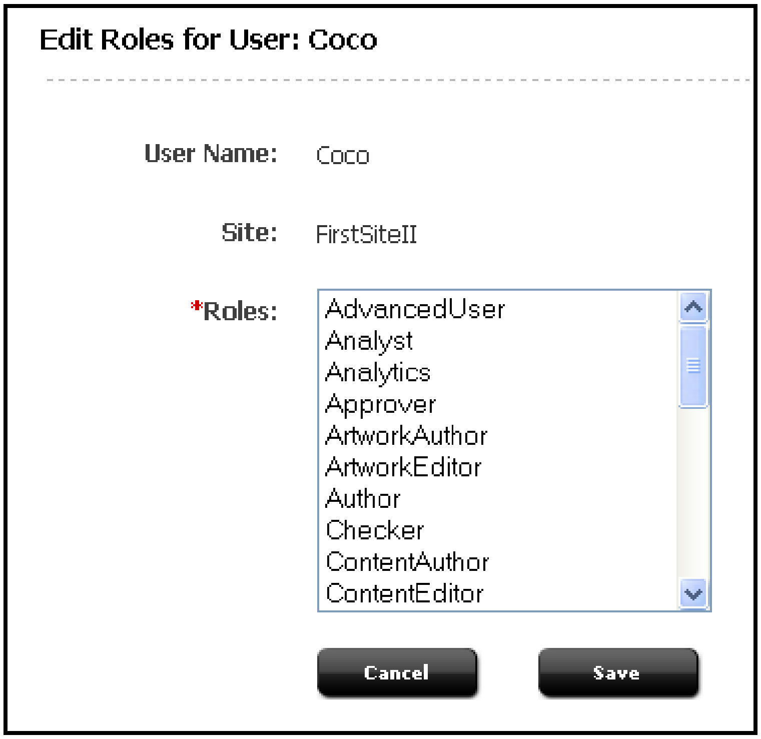
   The User Role Management form opens.
3. Enter the user name and click **Select**.

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

   > **Note:**
   >
   > If you require a list of system-wide users, open the list in this manner:
   >
   > 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.

   The Edit Roles form is shown.

Figure 5-2 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 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 Oracle WebCenter Sites: Contributor interface. In this instance, select the SitesUser role. For additional help with selecting system-defined roles, see 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 call 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 requires. The functions could be start menu items, tree tabs, workflow processes, and asset approval and publishing functions.
>
> For more information about assigning user roles, see Working with Roles.

6. Click **Save**.

7. Give the user access to site components by following procedures in Setting Access Permissions from the Admin Interface.

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 Users and Roles.

Viewing Site Users and Reassigning Roles

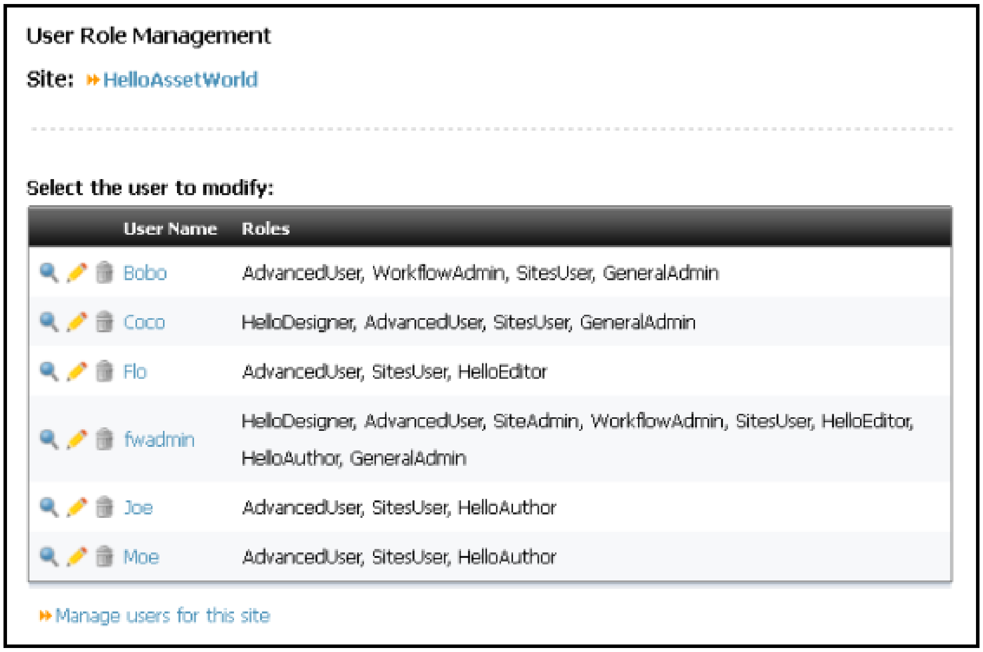
To view the list of users and their roles on a given site, and to reassign roles:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then expand the site.
2. Under the site, double-click **Users**.

   The User Role Management form opens.
3. In the **User Role Management** form, leave the **User Name** field blank and click **Select**.

   The User Role Management form shows a list of all the users currently assigned roles in the selected site.

Figure 5-3 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

4. (Optional) To reassign the user's roles, do the following:
   a. Navigate to the user name and click its **Edit** icon.

      The Edit Roles form opens.
   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.

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 **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then the site to which the user you want to delete is assigned.
2. Under the site, double-click **Users**.

   The User Role Management form opens.
3. In the User Role Management form, enter the user name and click **Select**.
4. In the form that opens, click the **Delete** icon next to the user.

   A warning message is shown.
5. Click **Delete User from Site**.

Enabling and Managing Asset Types

Every component in WebCenter Sites is an asset. You can add or remove and enable or disable an asset for a site. By doing such tasks, you can control the usage and access of assets to a site.

The following topics show you how add asset types to a site (enable the asset types), and delete them from a site:

- Enabling Asset Types for a Site
- Removing Asset Types from a Site

Enabling Asset Types for a Site

Enabling an asset type means associating the asset type with a chosen site or sites. After 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 node, as shown in Managing Access to Content Management Site Components.

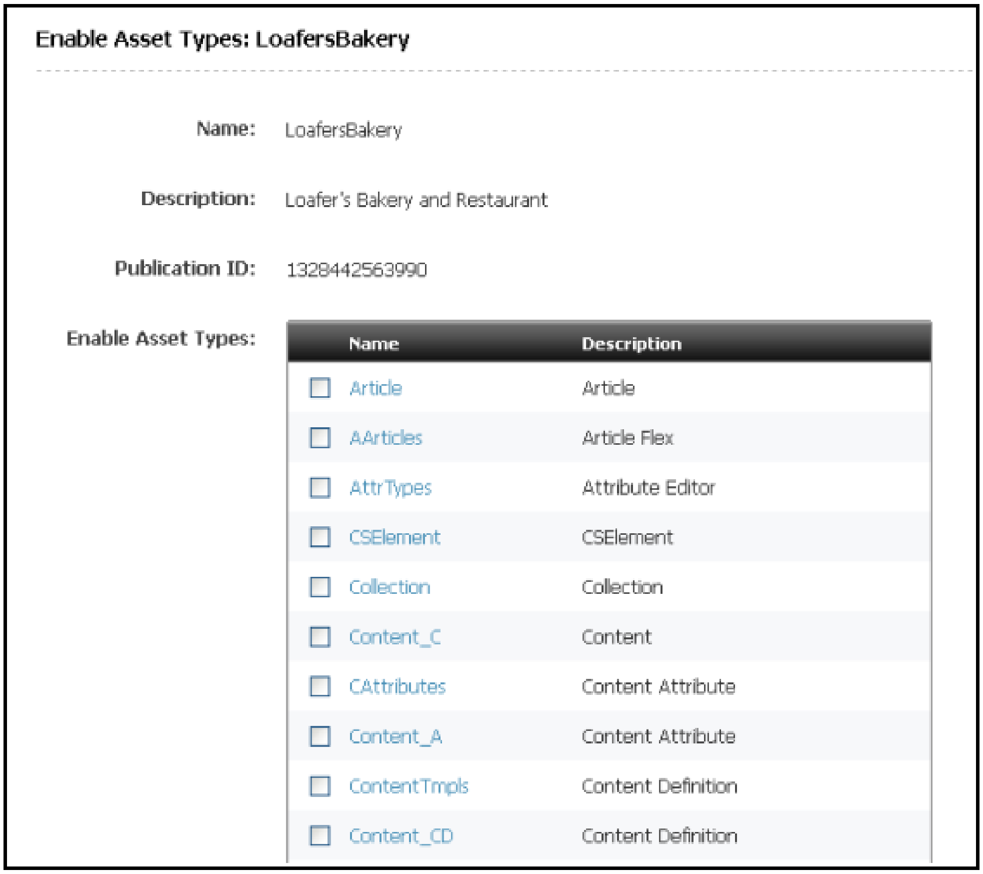
To enable asset types for a site:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then expand the site on which you want to enable asset types.

   > **Note:**
   >
   > If you have the `SiteAdmin` role, but not the `GeneralAdmin` role, select the **Site** tree, and then expand the **Site Admin** node.
2. Under the site, expand the **Asset Types** node, and then double-click **Enable**.

   The Enable Asset Types form is shown.

Figure 5-4 Enable Asset Types Form



Enable Asset Types: LoafersBakery

Name: LoafersBakery

Description: Loafer's Bakery and Restaurant

Publication ID: 1328442563990

Enable Asset Types:

	Name	Description
□	Article	Article
□	AArticles	Article Flex
□	AttrTypes	Attribute Editor
□	CSElement	CSElement
□	Collection	Collection
□	Content_C	Content
□	CAttributes	Content Attribute
□	Content_A	Content Attribute
□	ContentTmpls	Content Definition
□	Content_CD	Content Definition

The listed asset types have not yet been enabled in the selected site.

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

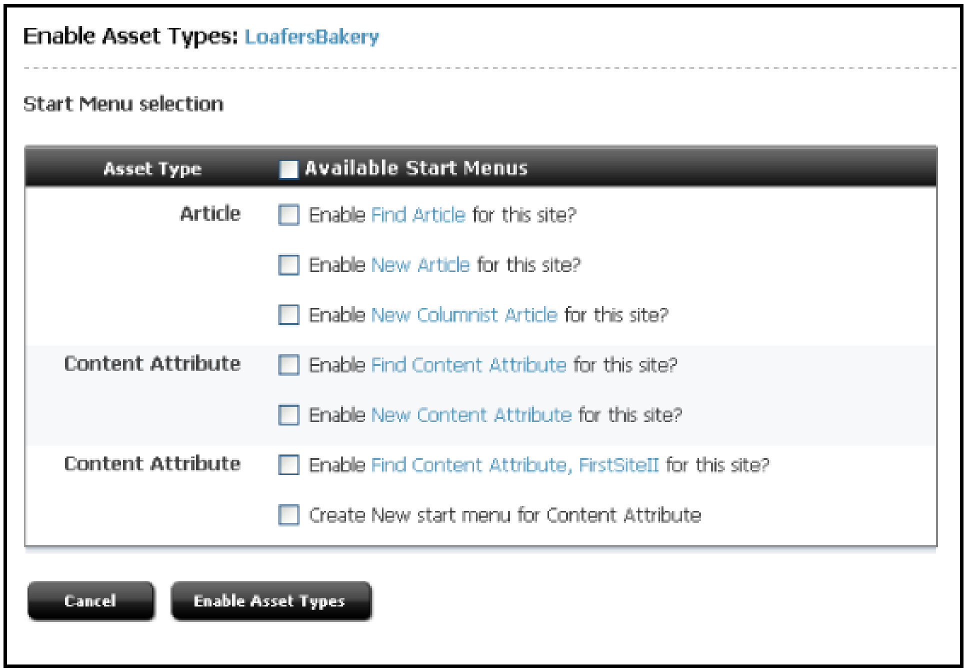
> **Note:**
>
> Content Query assets are not supported within the Admin interface.

4. Click **Enable Asset Types**.

   The Enable Asset Types - Start Menu Selection form is shown.

5. (Optional) In the Enable Asset Types - Start Menu Selection form, select the box next to each start menu item that you would like WebCenter Sites to create. See Managing Access to Asset Types Using Start Menus.

Figure 5-5   Enable Type Assets Form - Start Menu Selection



Enable Asset Types: LoafersBakery

Start Menu selection

Asset Type	□ Available Start Menus
Article	□ Enable Find Article for this site?
	□ Enable New Article for this site?
	□ Enable New Columnist Article for this site?
Content Attribute	□ Enable Find Content Attribute for this site?
	□ Enable New Content Attribute for this site?
Content Attribute	□ Enable Find Content Attribute, FirstSiteII for this site?
	□ 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 have to create start menu items for them, see Managing Access to Asset Types Using Start Menus.

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

Removing Asset Types from a Site

You can remove asset types from a site through the Admin interface.

To remove asset types from a site:

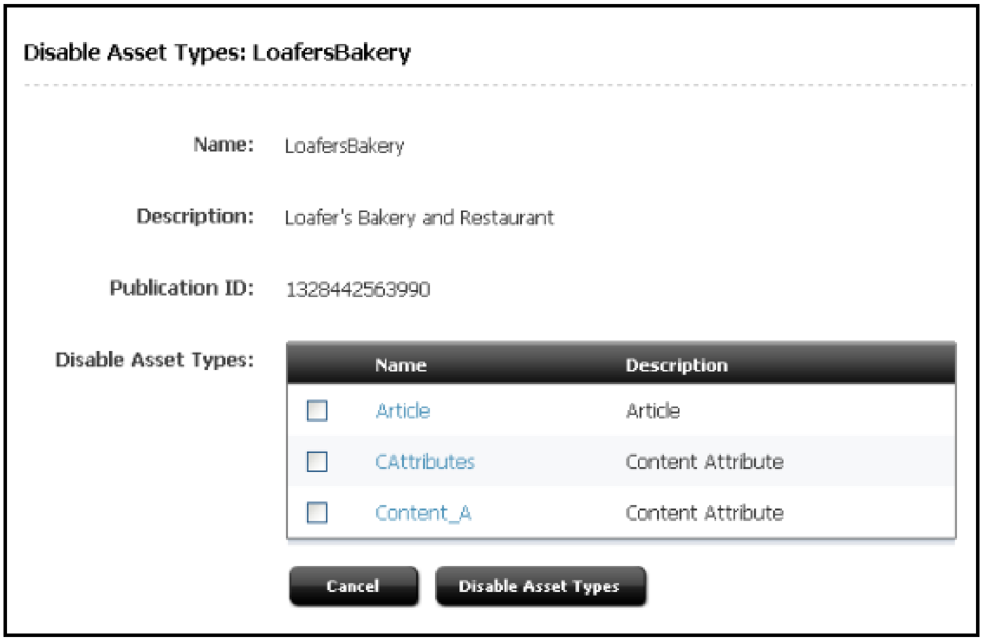
1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then the site from which you want to remove asset types..

> **Note:**
>
> If you have the `SiteAdmin` role, but not the `GeneralAdmin` role, select the **Site** tree and then expand the **Site Admin** node.

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

   The Disable Asset Types form is shown.

Figure 5-6 Disable Asset Types Form



Disable Asset Types: LoafersBakery

Name:	LoafersBakery
Description:	Loafer's Bakery and Restaurant
Publication ID:	1328442563990
Disable Asset Types:	

	Name	Description
□	Article	Article
□	CAttributes	Content Attribute
□	Content_A	Content Attribute

The asset types that have been enabled for the selected site are shown on the form.

3. Select the asset types 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.

6

Working with ACLs and Roles

You control access to WebCenter Sites with the use of Access Control Lists (ACLs), user accounts, and roles. You begin site configuration with the creation of ACLs, user accounts, and roles. This section provides an overview of ACLs and roles, and describes how to set up and manage users’ access on the management system. ACLs are used on both your management and your delivery systems. However, the main focus of the following topics is user management on the management system. For more information about user management on your delivery system, see The WebCenter Sites Development Process in *Developing with Oracle WebCenter Sites*.

- Overview of ACLs and Roles
- Access Control Lists (ACLs)
- Working with ACLs
- Roles
- Working with Roles

Overview of ACLs and Roles

You can control access to the WebCenter Sites system or database tables by assigning ACLs to users, database tables, and WebCenter Sites pages. To manage access to sites and their components you use roles. If you would like to enable interface functions or hide them, you can assign roles to users and interface functions on a site.

If a user's ACLs match the database ACLs, the user has permission (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.

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 ACLs and roles for the users. If the ACLs and roles do not exist, you must create them in WebCenter Sites, even if you intend to use LDAP plug-ins to create the user accounts.

Access Control Lists (ACLs)

Access Control Lists (ACLS), 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 WebCenter Sites systems starts with ACLs.

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

- Individual database tables

- Individual WebCenter Sites pages

Oracle WebCenter Sites and the WebCenter Sites content applications control a user's access rights to the database tables that represent those functions through ACLs. WebCenter Sites verifies 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 users or edit existing user information.

ACLs serve as the foundation of the security and user management model in your WebCenter 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 to view WebCenter Sites pages. However, additional ACL restrictions are enforced only when the `cc.security` property in the `wcs_properties.json` file (categorized under Core) is set to `true`. For more information about the `cc.security` property, see Core Properties in *Property Files Reference for Oracle WebCenter Sites*. Also see 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 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 required 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. Managing Users on the Management System in *Developing with Oracle WebCenter Sites* 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.

Topics:

- ACLs for User Accounts
- ACLs for Database Tables
- ACLs for Page Entries in the SiteCatalog Table
- System ACLs
- Custom ACLs

ACLs for 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 must 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.

Oracle WebCenter Sites and the WebCenter Sites content applications use several 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 System Defaults.

ACLs for 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** node (under the **General Admin** tree). Then, only those users with the same ACL have access to the data in that table.

If you assign multiple ACLs to a table, a user only requires 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 WebCenter 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. See Assigning ACLs to Custom Tables.

> **Note:**
>
> Do not add ACLs to database tables through the Oracle WebCenter Sites Explorer application. Instead, use the **Sites Database** option in the **System Tools** node of the **Admin** node.

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. See Foreign Tables in *Developing with Oracle WebCenter Sites*.

ACLs for Page Entries in the SiteCatalog Table

The SiteCatalog table holds page entries for all the pages shown for the content applications and the pages shown for your online site (that is, the site that you are delivering from the delivery system). 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 have to use ACLs to restrict access to your custom pages.

System ACLs

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

Custom ACLs

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

- If your site requires user registration, you may have to create a set of ACLs for site visitors.
- If your developers create functions and put them on new tabs to customize the management system, you might have 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 require and how to apply them.

Working with ACLs

> **Note:**
>
> When using an LDAP integration option, be aware of system response to user and site management operations. For information about system response, see Users, Sites, and Roles in LDAP-Integrated Sites Systems.

- Creating an ACL
- Editing a Custom ACL
- Deleting a Custom ACL
- Assigning ACLs to Custom Tables
- Assigning ACLs to WebCenter Sites Pages

Creating an ACL

> **Note:**
>
> When creating ACLs, consider the roles you are using to ensure that the ACLs are commensurate with the roles. For example, if you are 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:

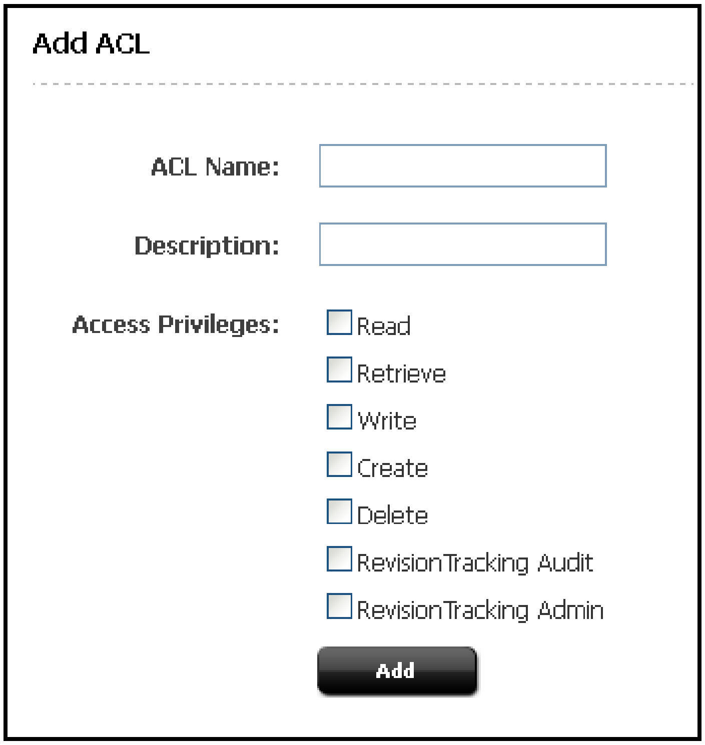
1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **ACLs**.

   The ACLs form opens.

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 opens.

Figure 6-1 Add ACL Form



Add ACL

ACL Name:

Description:

Access Privileges:
□ Read
□ Retrieve
□ Write
□ Create
□ Delete
□ RevisionTracking Audit
□ RevisionTracking Admin

3. In the **ACL Name** field, enter a unique name.

4. Select the access privileges you want to assign to this ACL.

5. Click **Add**.

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

   > **Note:**
   >
   > For a new ACL that is created in LDAP, in order to use it in Sites user interface, add an entry for the newly created ACL in SystemACL table.

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.

Editing a Custom ACL

> **Caution:**
>
> Never modify any of the system ACLs. For a list of these ACLs, see System ACLs and Their Permissions.

To edit a custom ACL:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **ACLs**.

   The ACLs form opens.

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 opens. Do not modify system ACLs.

4. In the **Modify ACL** form, make the changes to the Description and Access Privileges. For information about the options, see Permissions.
5. Click **Modify**.

   WebCenter Sites writes your changes to the `SystemACL` table.

Deleting a Custom ACL

> **Caution:**
>
> Never delete a system ACL. For a list of these ACLs.

To delete a custom ACL:

1. If you are using LDAP, delete (from your LDAP server) the group corresponding to the ACL you are deleting.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **ACLs**.

   The ACLs form opens.
3. In the **Select an ACL** field, select the ACL to delete.
4. Select the ACL from the drop-down list and click **OK**.

   A warning message opens.
5. Click **OK**.

   The ACL has been deleted.

Assigning ACLs to Custom Tables

If you or the site designers create tables, you might have to restrict access to those tables by assigning ACLs to them. Typically, you assign ACLs to new tables when you create those tables. (See Controlling User Access in *Developing with Oracle WebCenter Sites*.)

> **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 **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Sites Database**.

   The Sites Database form opens.
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 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 table.

   The Modify Catalog form opens.
5. In the **ACL** field, select the ACL(s) you want to assign to the selected table. To select multiple ACLs, **Ctrl-click** each 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 Indirect Data Storage with the WebCenter Sites URL Field in *Developing with Oracle WebCenter Sites*.

6. Click **Modify**.

Assigning ACLs to WebCenter Sites Pages

ACLs are nearly always set using the Explorer tool. However, administrators can assign ACLs to the page entry created for SiteEntry or template assets through a field in the Create or Edit form.

To assign ACLs to a SiteEntry asset:

1. Find and open the SiteEntry asset to modify.
2. In the **Access Control Lists** field, select the ACLs to assign to this asset.
3. Save the asset.

The Access Control Lists field is available when creating a new SiteEntry asset as well. Assign ACLs when creating SiteEntry assets in the same way.

To assign ACLs to a template asset:

1. Find and open the template asset to modify.
2. Select the **Element** section.
3. In the **Access Control Lists** field, select the ACLs to assign to this asset.
4. Save the asset.

The Access Control Lists field is available when creating a new template asset as well. Assign ACLs when creating template assets in the same way.

To assign ACLs to pages that are not associated with a SiteEntry asset or a template asset, use the Explorer tool.

Roles

Roles manage site-specific access to WebCenter Sites interface functions. They represent 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 jobs that user does 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 opened 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.

Understanding System Roles

Several system roles are installed by WebCenter Sites. One role is required for the WebCenter Sites content applications to function, and three are required for the WebCenter Sites administrators to function. See System Roles.

Understanding Custom Roles

Unlike ACLs, roles are objects that you will most likely have to create to account for the full range of users' responsibilities on your sites. To create roles, follow instructions in the section, Working with Roles.

Understanding Sample Roles

If you installed one or more sample sites, you will have access to several 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.

Working with Roles

You can create, edit and delete roles assigned to the users of a site to manage access to the site and its components.

The following topics show 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 more information about system responses, see Users, Sites, and Roles in LDAP-Integrated Sites Systems.

- Creating a Role From the Admin Interface
- Editing a Role From the Admin Interface
- Deleting a Role in the Admin Interface

Creating a Role From the Admin Interface

To create a new role:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **Roles** node, and then double-click **Add New**.

The Add New Role form opens.

2. In the **Name** field, enter a unique name of up to 32 characters.

> **Note:**
>
> The following characters are not permitted in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), and greater-than sign (>). Additionally, the **Name** field cannot end with a backslash (\).

3. In the **Description** field, enter a short and informative description of no more than 255 characters.

> **Note:**
>
> If you want to see/use a new role that is created in LDAP in Sites user interface, for example, workflow related operations or assign roles to tree tabs, add an entry for the newly created role in CCRoles table.

4. Click **Add New Role**.
5. Add the role to the default tree tabs (**Workflow**, **Bookmarks**, **Site Admin**, **Admin**, and **Site Plan**), as appropriate. For instructions, see Editing Trees and Nodes.

Editing a Role From the Admin Interface

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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **Roles**.

   A list of all roles opens in the main window.
2. In the list of roles, navigate to the role you want to edit and click its **Edit** icon.
3. In the **Edit Role** form, make your changes, then click **Save**.

Deleting a Role in the Admin Interface

> **Caution:**
>
> Do not delete any of the system default roles: `GeneralAdmin`, `SiteAdmin`, `WorkflowAdmin`, `AdvancedUser`, WebCenter Sites User.

To delete a role:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **Roles**.

   A list of all roles opens 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.

   A confirmation message opens.

3. Click **Delete Role**.

   The role is deleted.

Part III
Managing User Interfaces and Assets

Your portfolio includes managing users and their access to different sites, configuring Admin interface, Contributor interface, and WEM interface. You'll also configure various site components like start menu and search components in the Contributor interface.

Some of these features are alternate interfaces to the interfaces, such as the Web Mode of the interface. Others change the interfaces (locale settings), the behavior of the preview function, and the keyboard commands that are used to work with the Contributor interface.

Topics:

- Configuring the User Interfaces
- Creating and Authorizing Users
- Managing Access to Content Management Site Components
- Working with the Admin Interface
- Using the Web Experience Management Framework
- Administering WebCenter Sites and Its Components in Enterprise Manager

7
Configuring the User Interfaces

You need to configure the various WebCenter Sites interfaces before content providers can use them. You can customize the interfaces as the need arises.
Some of these features are alternate interfaces to the WebCenter Sites interfaces, such as the Web Mode of the Contributor interface. Others change the WebCenter Sites interfaces (locale settings), the behavior of the preview function, and the keyboard commands that are used to work with the Contributor interface.

Topics:

- Setting the Locale for the WebCenter Sites Interfaces
- Using Web Mode
- Maintaining Separate Browser Sessions for Preview
- Reconfiguring Keyboard Shortcuts for the Contributor Interface
- Setting the User Interface to Right-to-Left (RTL) Orientation

Setting the Locale for the WebCenter Sites Interfaces

As the administrator, you can configure a default locale or language for WebCenter Sites interface items. After WebCenter Sites is installed, English is set as the default locale for the system. You can change the default locale to suit your preferences and requirements.

There are several options and properties that you use to set the locale:

- The **Locale** item in the **Admin** node (under the **General Admin** tree) 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 `wcs_properties.json` file (categorized under Core) 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 `wcs_properties.json` file (categorized under Publish) to 4 or lower.

Topics:

- Using the System Default Locale for WebCenter Sites Users
- System Locale Properties
- Single-Language Restrictions
- Locale and Asset Types

- Locale and the Article Asset Type

Using the System Default Locale for 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 are multiple locales 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 Creating and Editing a User Profile to understand 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.
- If the user profile has a value for the `locale` user attribute but does not have the files and stylesheets necessary to support that language in the browser, WebCenter Sites uses the default locale that is set for the system. Ensure the files and stylesheets for the locale are located in the application server directory under the `CSS` folder (`/Xcelerate/data/CSS`).
- If the default locale does not have the files and stylesheets necessary to support the default locale set for the system, then the admin pages will be displayed without stylesheets and formatting. Ensure the files and stylesheets for the default locale are located in the application server directory under the `CSS` folder (`/Xcelerate/data/CSS`).

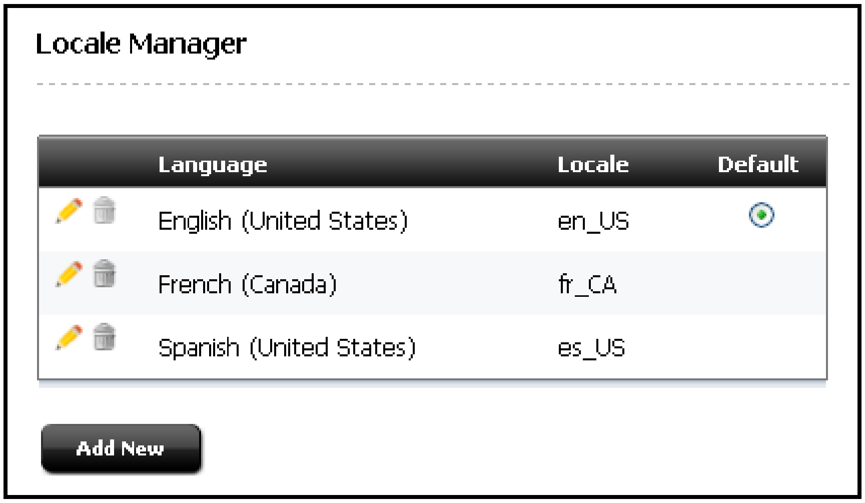
> **Note:**
>
> When you select a site and set a new system default language from that site, the language is automatically set as the global default language for all sites on your WebCenter Sites system.

To set the system default locale:

1. In the **General Admin** tree, expand the **Admin** node, and then double-click **Locale**.

   The Locale Manager form is displayed.

   Figure 7-1 Locale Manager Form

   Locale Manager

	Language	Locale	Default
✎ 🗑	English (United States)	en_US	◉
✎ 🗑	French (Canada)	fr_CA	
✎ 🗑	Spanish (United States)	es_US	

2. Click the **Edit** icon next to the locale.

   A radio button opens on the form in the Default column.

3. Select the **Default** radio button next to the locale.

4. Click **Save**.

   The next time users log in to WebCenter Sites, the system displays 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 Working with User Profiles and User Attributes for information about how to set a user's locale preference.

System Locale Properties

The properties in the table on System Locale properties should have been configured during the installation of the language pack on WebCenter Sites. They are listed here for troubleshooting. For example, if the workflow emails are displaying odd characters, perhaps the email properties were not set correctly.

For instructions on how to access the Property Management Tool in the Admin interface and use it to verify or modify property values, see Using the Property Management Tool in *Property Files Reference for Oracle WebCenter Sites*.

Table 7-1 System Locale Properties

Property	Category	Description
`cs.contenttype`	Core	Determines the HTTP header character set for all WebCenter Sites pages, and other HTTP interaction (Oracle WebCenter Sites Explorer, CatalogMover, and so on).This value is set to `text/html; charset=UTF-8` by default. This value must be set to an appropriate value for the online site that your WebCenter Sites system is delivering.
`cs.emailcharset`	Core	Determines the character set used in the subject lines of email messages sent by WebCenter Sites. (Typically these are workflow emails.)
`cs.emailcontenttype`	Core	Determines the character set used in the body of email messages sent by WebCenter Sites.
`cs.mirrorrowsperpost`	Publish	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`	Core	Determines the HTTP header character set used for all WebCenter Sites pages. This value must be set to UTF-8.

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 and individual items such as categories and source codes can have one name only. While much of the text on an individual WebCenter Sites form can be displayed in any of the languages that you have installed on your WebCenter 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 node 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.

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 WebCenter Sites system. However, there are several core columns and fields that WebCenter Sites creates for all asset types by default:

- `id`
- `name`
- `description`
- `status`
- `createdby`
- `createddate`
- `updatedby`
- `updateddate`
- `startdate`
- `enddate`
- `subtype`
- `filename`
- `path`
- `category`

When WebCenter Sites displays them as fields in its interfaces, it uses the language that is set as the system default language. However, there are certain field names that are always displayed in English. See User Interface Fields that are Exceptions to Localization.

> **Note:**
>
> If you are using SQL Server, category names cannot differ by case only. You cannot have a category named `MyCategory` and another category named `mycategory`. SQL Server will not differentiate case for attribute categories.

See Default Columns in the Basic Asset Type Database Table and Default Columns in the Flex Asset Type Database Table in the *Developing with Oracle WebCenter Sites*.

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 `wcs_properties.json` 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 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.

User Interface Fields that are Exceptions to Localization

There are certain field names in the interfaces of WebCenter Sites that are always displayed in English irrespective of the system default language.

The following list provides those field names:

- page
- collection
- template
- query
- site
- segment
- promotion
- custom

Using Web Mode

Use Web Mode if you are an infrequent user or if you need to only find, edit, and submit content directly from the rendered (Preview) version of an asset. Web mode is an alternate interface to the Contributor interface.

Enabling this feature requires three general steps:

- The administrator sets the `xcelerate.enableinsite` property in the `wcs_properties.json` file to `true` using the Property Management Tool in the Admin interface, on the WebCenter Sites management system.
- The developers code templates that call the Web Mode feature for the fields for 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 use Web Mode.

To enable Web Mode:

1. In the Admin interface, open the Property Management Tool.
2. In the **Name** field, enter `xcelerate.enableinsite`.
3. Click **Search**.
4. Select the name of the property. In the **Value** field, set its value to true.
5. Click **Save**.

To configure your servlet and host name used in Web Mode and Preview:

1. In the Admin interface, open the Property Management Tool.
2. In the **Category** drop-down menu, select **UI**.
3. Click **Search**.
4. Do one or both of the following:
   1. Select `xcelerate.previewhost`. In the **Value** field, set its value to `http://<server>:<port>/servlet/` and then click **Save**.
   2. Select `xcelerate.previewservlet`. In the **Value** field, change its value from `ContentServer` to `Satellite`, and then click **Save**. 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.previewservlet` property to specify the Satellite Server servlet rather than the WebCenter Sites servlet.

Maintaining Separate Browser Sessions for Preview

You can configure your WebCenter Sites management system so that the web server opens a new browser session for the previewed asset. A content provider can lose any unsaved work because without a new, separate browser session for preview, the WebCenter Sites session ends when the preview session begins.

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 WebCenter 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 WebCenter Sites management system so that the web server opens a new browser session for the previewed asset, configure properties in the `wcs_properties.json` file: `xcelerate.previewhost` and `xcelerate.previewservlet`. You use these properties to point to the URL for the WebCenter 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.previewservlet` property to specify the Satellite Server servlet rather than the WebCenter Sites servlet.

In the Admin interface, open the Property Management Tool, select **UI** from the **Category** drop-down menu, click **Search**, 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.previewservlet` property from `ContentServer` to `Satellite`.

For instructions about using the Property Management Tool to set property values, see Using the Property Management Tool

in the *Property Files Reference for Oracle WebCenter Sites*.

Reconfiguring Keyboard Shortcuts for the Contributor Interface

You can customize default keyboard shortcuts for the Contributor interface and instructions for reconfiguring these shortcuts.

- Default Keyboard Shortcuts
- Customizing Keyboard Shortcuts for the Contributor Interface

Default Keyboard Shortcuts

The Contributor interface enables you to use various keyboard shortcuts to both speed up your content management process and provide a completely accessible interface. To view the current keyboard shortcuts for the Contributor interface, press **Ctrl**+**Alt**+**L** from the Contributor interface to log the current keyboard shortcuts in your browser's web console.

For a list of the default keyboard shortcuts, see Keyboard Shortcuts in *Using Oracle WebCenter Sites*.

Customizing Keyboard Shortcuts for the Contributor Interface

To customize the default keyboard shortcuts for the Contributor interface, complete the following steps:

1. Open for editing the `<shared-install-location>\elements\UI\Config\GlobalHtml.jsp` file.
2. Add the `config` JavaScript object with a `shortcuts` property, in proper JSON format (`config.properties`), to the `GlobalHtml.jsp` file. In the `config` object, add a line for each keyboard shortcut you want to reconfigure along with the keys you want to associate with the shortcut, in the following syntax:

   ```
   config.shortcuts = {
       "ShortCutName" [Keystrokes]
       "ShortCutName2" [Keystrokes]
   }
   ```

   For the key command to be valid, one or more modifier keys or special keys must be used.

   - Valid modifier keys:
     - ALT
     - CTRL
     - SHIFT
     - META
   - Valid special keys:
     - ALT
     - BACKSPACE
     - CAPS_LOCK
     - CLEAR
     - CTRL
     - DELETE
     - DOWN_ARROW
     - DOWB_DPAD
     - END
     - ENTER
     - ESCAPE
     - F1 through F15
     - HELP
     - HOME
     - INSERT
     - LEFT_ARROW
     - LEFT_DPAD

- LEFT_WINDOW
- META
- NUMPAD_0 through NUMPAD_9
- NUMPAD_DIVIDE
- NUMPAD_ENTER
- NUMPAD_MINUS
- NUMPAD_MULTIPLY
- NUMPAD_PERIOD
- NUMPAD_PLUS
- NUM_LOCK
- PAGE_DOWN
- PAGE_UP
- PAUSE
- RIGHT_ARROW
- RIGHT_DPAD
- RIGHT_WINDOW
- SCROLL_LOCK
- SELECT
- SHIFT
- SPACE
- TAB
- UP_ARROW
- UP_DPAD

For example, to change the keyboard shortcut for viewing the current shortcut configuration in a browser's web console (`logShortcutConfiguration`), add the following to the `GlobalHtml.jsp` file:

```
config.shortcuts = {
    "logShortcutConfiguration": ["CTRL", "ALT", "SHIFT", "L"]
}
```

3. Save and close the `GlobalHtml.jsp` file.
4. Restart the server for your changes to take effect.

Setting the User Interface to Right-to-Left (RTL) Orientation

If a user's locale is Arabic, you can configure the orientation of the Contributor interface, such as the position of the navigation pane, search results, and asset tabs with the Right-to-Left (RTL) support.

To implement RTL support:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User Profiles**.

The User Profile Management form opens.

2. In the form, enter the user name of the user whose locale you want to change to Arabic, and then click **Select**.

   The User Profiles list opens.

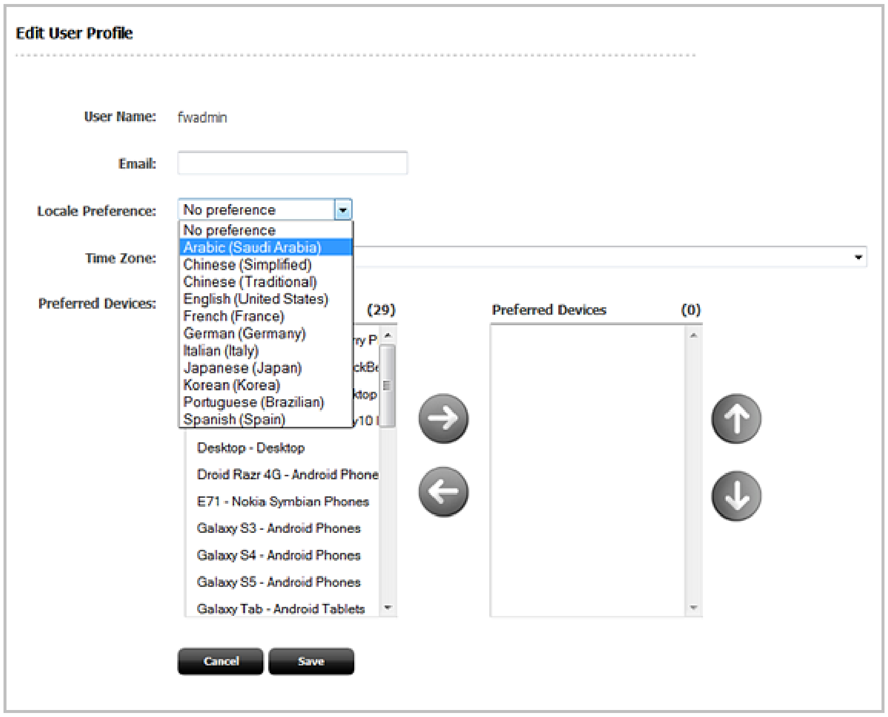
3. Select the name of the user.

   The User Profile Management form opens.

4. Click **Edit**.

5. In the **Local Preference** field, select **Arabic (Saudi Arabia)** from the **Locale Preference** drop-down list.

Figure 7-2 Edit User Profile: Select Arabic



Edit User Profile

User Name: fwadmin

Email:

Locale Preference: No preference

Time Zone:

Preferred Devices:

6. Click **Save**.

7. Restart WebCenter Sites for your changes to take affect across all WebCenter Sites instances and cluster members.

8

Creating and Authorizing Users

You can create a user in the WEM Admin interface, and authorizing that user to manage a site and the applications available on that site.

- Creating Users
- Authorizing Users to Work with Applications
- Authorizing a Predefined User
- Authorizing Developers to Register Applications
- Asking Your Developers About Available Applications

Creating Users

You can create general administrators, site administrators, and contributors as users of a site.

> **Note:**
>
> Only general administrators can create users. See Authorizing a Predefined User.

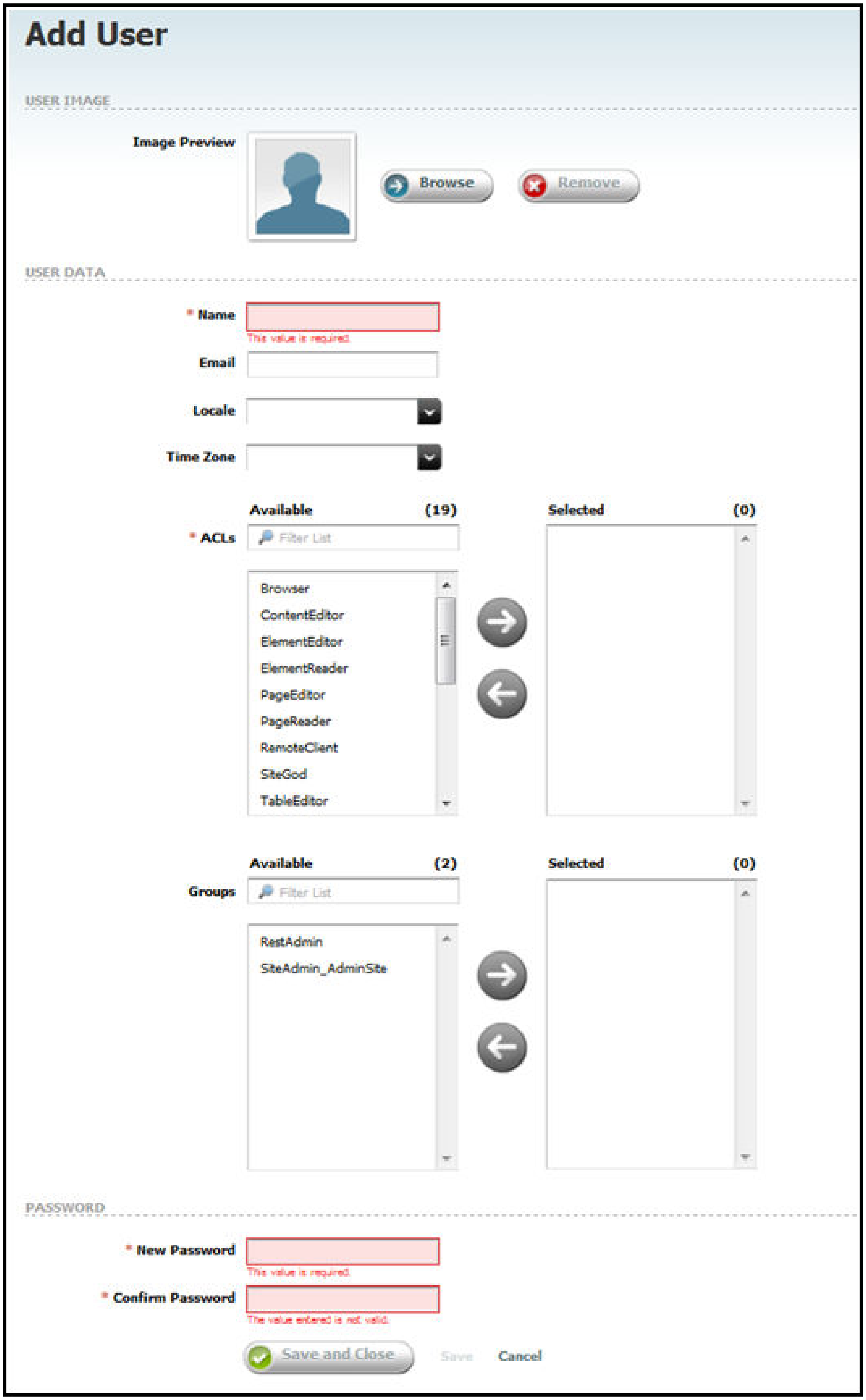
To create the user:

1. Log in to the WEM Admin interface using the general administrator credentials that you used during the WebCenter Sites installation process.
2. In the **Admin** menu bar, click **Users**.

   The Users form opens.
3. In the Users form, click **Add User**.

   The Add User form opens.

Figure 8-1 Add User Form



Add User

USER IMAGE

Image Preview

USER DATA

* Name

This value is required.

Email

Locale

Time Zone

	Available (19)		Selected (0)
* ACLs	Filter List		
	Browser ContentEditor ElementEditor ElementReader PageEditor PageReader RemoteClient SiteGod TableEditor		
	Available (2)		Selected (0)
Groups	Filter List		
	RestAdmin SiteAdmin_AdminSite		

PASSWORD

* New Password

This value is required.

* Confirm Password

The value entered is not valid.

Fill in the following fields:

- **Image Preview**: (Optional) Use the **Browse** button to associate a picture with the new user.
- **Name**: Enter a name with which the user logs in.
- **Email**: (Optional) Enter a valid, unique email address.
- **Locale**: (Optional) Select the user language preference. If you do not specify a preference, the WEM Framework uses the default locale that is set for the browser.
- **Time Zone**: (Optional) Select the user time zone preference from the menu. If you do not specify a time zone preference, the system automatically detects the time zone of the user from the browser.
- **ACLs**: ACLs regulate user 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 Oracle WebCenter Sites: Engage).
- **Groups**: Groups provide access to REST. You use groups to control access to application resources.
  - 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 (Authorizing Users to Work with Applications).
  - 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 Viewing REST Security Configurations.

- **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 indicates that the user does not have access to any sites.

To enable the user as an administrator or regular user:

1. Assign the user to a site:
   a. From the Users form, hold the cursor 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`.

    - 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 has 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 sites in which the user is assigned the `SiteAdmin` role.

    - 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.

2. To authorize a user to work with applications, see the next topic on authorizing users to work with applications.

Authorizing Users to Work with Applications

You can authorize users to work with applications, such as the Oracle WebCenter Sites: Contributor interface. After authorizing users, assign roles to them based on the application they need to access.

To authorize a user:

1. Select or create a site.
2. Assign an application to the site.
3. Assign a user to the same site and couple the user to the application.
4. Assign the user to a group to give permission to the REST application resources.

> **Note:**
>
> Both general and site administrators can authorize users.
>
> If you must have 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 that the user you want to authorize works with applications that do not specify a predefined user. See 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.

   The WEM Admin Sites Form opens.

   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. The Sites form lists only the sites you are allowed to manage.

2. Assign an application to the site:

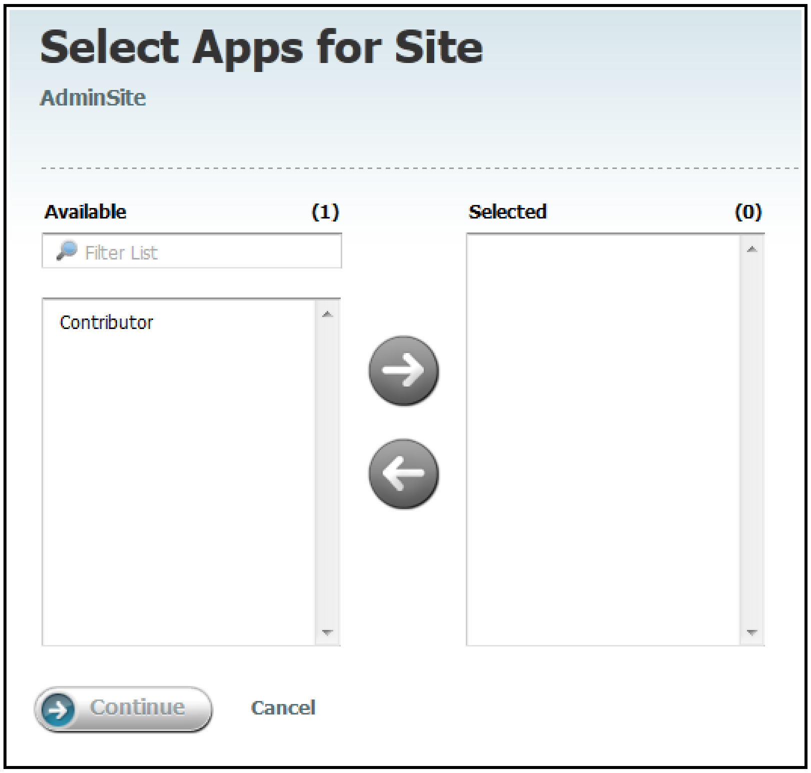
   a. In the Sites form, hold the cursor over the site name and click **Manage Site Apps**.

   b. Click **Assign Apps**.

      The Select Apps For Site form is shown.

      > **Note:**
      >
      > The **Assign Apps** button is dimmed if no applications are registered with the WEM Framework.

      Figure 8-2 Select Apps for Site Form

      Select Apps for Site

      AdminSite

   c. Select the application you want to assign to the site and move it to the **Selected** list.

      To search for an application, type its name in the **Filter List** field. The results appear in the **Available** list.

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.

> **Note:**
>
> If the application is the Admin interface, assign it the `AdvancedUser` role. If the application is the Contributor interface, assign it the `SitesUser` role.
>
> Take note of the roles you assign. You must 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:

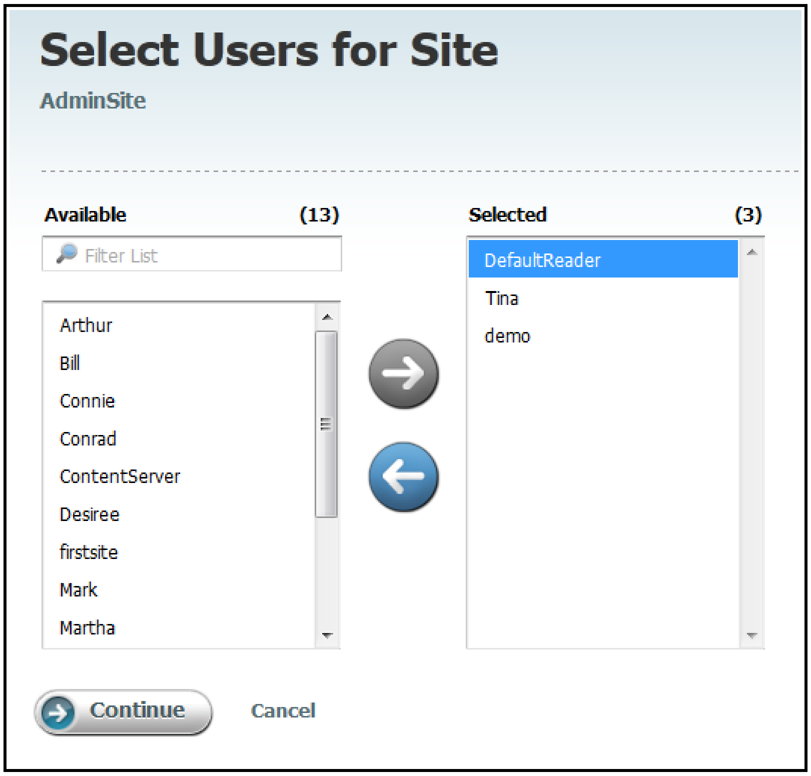
a. In the **Admin** menu bar, click **Sites**.

b. Hold the cursor over the new site name and click **Manage Site Users**.

c. Click **Assign Users**.

The Select Users for Site dialog opens.

Figure 8-3 Assign Users Form



Select Users for Site

AdminSite

Available (13)

Filter List

Arthur
Bill
Connie
Conrad
ContentServer
Desiree
firstsite
Mark
Martha

Selected (3)

DefaultReader
Tina
demo

d. In the Select Users for Site form, select the user you want to assign to the site and move the user to the **Selected** list.

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 role 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 Admin interface, you must assign the user the AdvancedUser role. If the application is the 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 function roles (specifications are available from application developers). The user is then fully authorized at the application level. However, the user cannot work with the application 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. For example, to grant a user permission to publish assets from the Contributor interface, you must assign the user the Publisher role in addition to the SitesUser role (see Authorizing Users to Publish Assets from the Contributor Interface). 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, hold the cursor over the user you want to authorize and click **Edit**.

c. In the Edit User form, select groups 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 Authorizing Users to Access Application Resources.

   d. Click **Save and Close**.

5. Verify the user can access the new application.

   The login dialog lists the new site (in the **Site** menu) and shows the application icon below the menu.

   The new site is also listed in the menu next to the name of the logged-in user, and the application icon opens in the upper left-hand corner.

6. As a reminder, if you have not yet authorized the user with permissions to REST, complete the steps in Using REST Security.

Authorizing a Predefined User

Site developers can specify predefined users in their applications to simplify administration of the authorization processes. Instead of authorizing each user individually at the REST level, you authorize the predefined user. When these predefined users log in to the sites, they can readily access the applications.

If an application is configured with a predefined user, complete the following steps in the WEM Admin application to make a predefined user available and authorized.

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**: ACLs regulate access to the database tables. You must assign the predefined user the ACLs of the logged-in users who will access the application. 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 Engage).
   - **Group assignment**: Groups authorize the user at the REST level (to manage application resources). You must assign the predefined user to a group with the security privileges that you would otherwise grant to the application users. For information about configuring REST security, see Using REST Security.

   For instructions on creating the user, see Creating Users.

2. Assign the predefined user to the application. For instructions, see Authorizing Users to Work with Applications.

3. Assign users to the application (using the procedure in Authorizing Users to Work with Applications), but skip their assignment to groups (step 4).

Authorizing Developers to Register Applications

If you want to display assets through REST services on the Apps page in the WEM Admin interface, register them or create them as assets. After registering the assets for a site, you can authorize users to work with the applications.

Typically, developers register the applications they create programatically. If developers 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`. (See Registering Applications Manually in WEM Framework in *Developing with Oracle WebCenter Sites*.)

To authorize a developer, make sure the developer is a general administrator (that is, has complete permissions to the system, including REST services). For instructions on creating a general administrator, see Creating Users.

Asking Your Developers About Available Applications

If you want to manage applications and users, gather information from your developers about the applications they have created for the WEM Framework.

- **Resources:** ask your developers about the resources that custom-built applications use; once you know which asset types, assets, and other resources users work with, you can determine which privileges (such as create, update) the users must have for those resources and assign the users to groups that have those privileges. Information about configuring groups and assigning users is available in Using REST Security.
- **Roles:** ask your developers if interface functions, such as Edit, 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 Authorizing Users to Work with Applications.
- **Users:** ask your developers if 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 are passed to logged-in users when they access the application. See Authorizing a Predefined User.

9

Managing Access to Content Management Site Components

You can configure and manage site components such as start menus of sites, roles to access assets, permissions to access and configure Sites trees, and so on.
Topics:

- Overview
- Managing Access to Asset Types Using Start Menus
- How to Set Access Permissions to Assets
- Other Options for Asset Types
- Options for Managing Access to the Tree (Admin Interface Only)

Overview

You can grant permissions to users to access a site, and then manage their roles to configure start menus, access assets, and configure site trees.

See Enabling and Managing Asset Types.

- 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 the tree in WebCenter Sites. You can control users' permissions to the tree, tabs in the tree, nodes in the tabs, and items in the nodes.

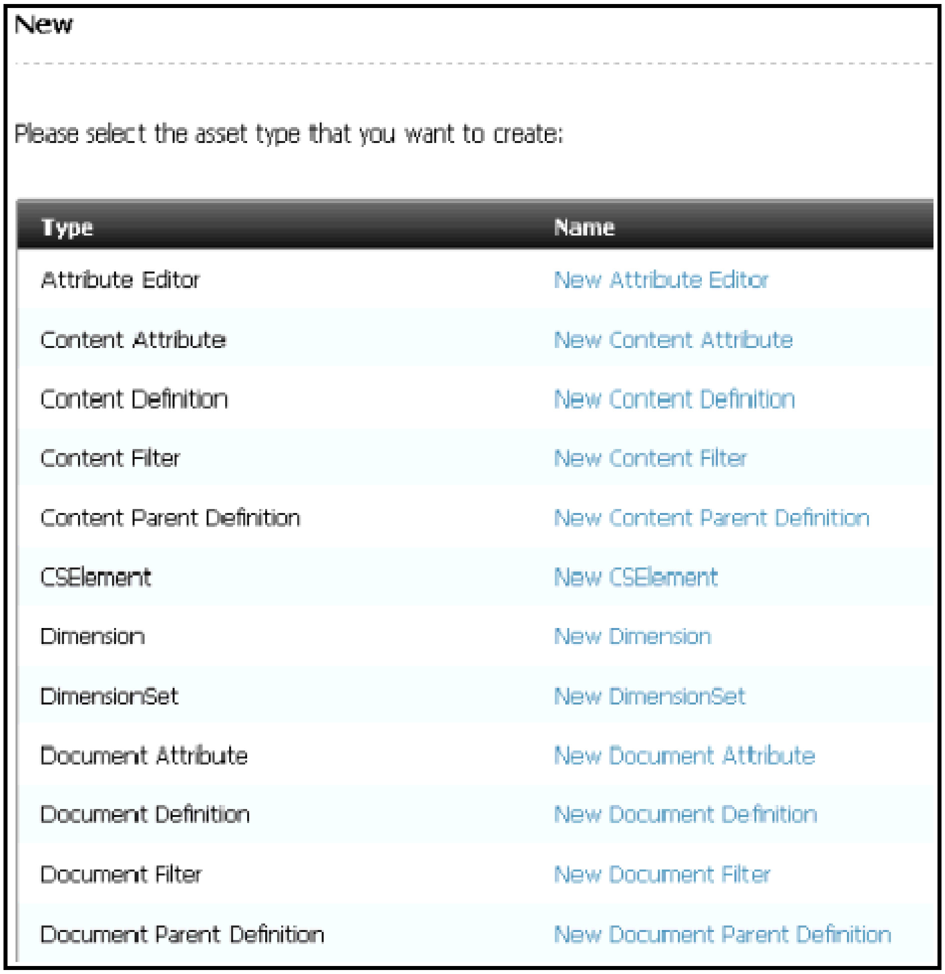
Roles are the components that you call to associate users with interface functions and the content that the users must manage.

Managing Access to Asset Types Using Start Menus

A start menu is a list of linked asset types of a site. You can manage start menus by modifying the asset components that can added from the Start menus, search and new.

See the following two figures for the New and Search menu forms in the Admin interface.

Figure 9-1 New Menu in the Admin Interface

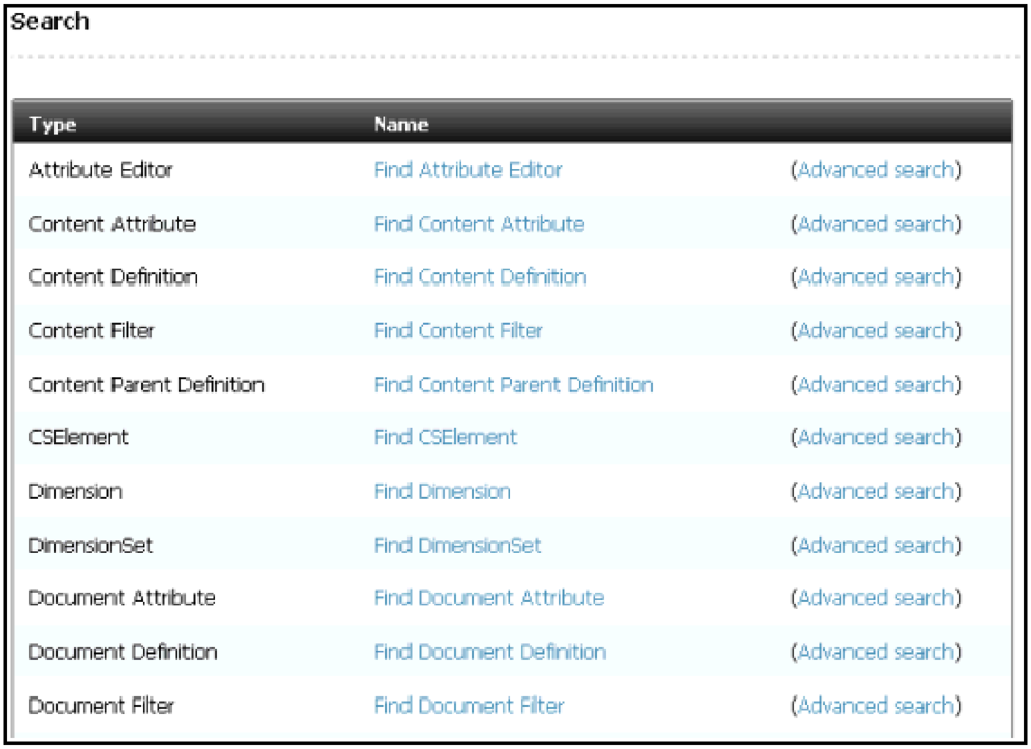


New

Please select the asset type that you want to create:

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

Figure 9-2 Search Menu in the Admin Interface



Search

Type	Name	
Attribute Editor	Find Attribute Editor	(Advanced search)
Content Attribute	Find Content Attribute	(Advanced search)
Content Definition	Find Content Definition	(Advanced search)
Content Filter	Find Content Filter	(Advanced search)
Content Parent Definition	Find Content Parent Definition	(Advanced search)
CSElement	Find CSElement	(Advanced search)
Dimension	Find Dimension	(Advanced search)
DimensionSet	Find DimensionSet	(Advanced search)
Document Attribute	Find Document Attribute	(Advanced search)
Document Definition	Find Document Definition	(Advanced search)
Document Filter	Find Document Filter	(Advanced search)

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 the preceding figures 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 covers the following topics:

- Working with Start Menu Items
- Managing Start Menu Items

Working with Start Menu Items

Each option in a start menu is a start menu item. A start menu item names the type of asset a user can create, copy, or search for. For example, the **New** start menu shown in Figure 9-1 belongs to the Admin Interface and shows 16 start menus items to the user authorized to work with the items. The user can create assets of the types listed in the menu.

In the Admin Interface, start menu items are shown 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 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 an asset with the shortcut.

  For example, for basic assets, it can 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 (in the Admin interface). If an asset type is shown on a tree, but there is no start menu item created for it, the only options listed in the right-click menu for the asset type are **Refresh**, **Edit**, and **Inspect**. To have the **New** option included in the right-click 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 an 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.

Managing 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 shown 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. See About Start Menu Items.
- **Search** – creates a link for the asset type on the list that is shown 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. See Creating Search Start Menu Items.

Topics:

- About Start Menu Items
- Creating Search Start Menu Items
- Editing Start Menu Items
- Showing a List of All Start Menu Items

About 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, complete the steps in Enabling Asset Types for a Site.
- To specify a default workflow process, create the workflow process first. See 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 interface.

- 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 requirements 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 multiple values.
- 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:

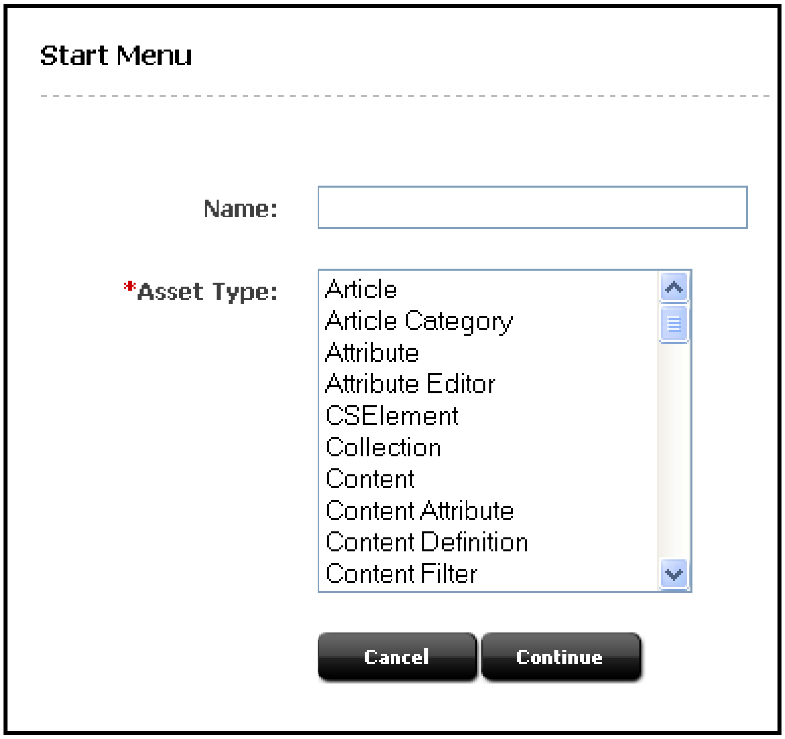
- Visitor Attribute
- CS Element
- Dimension
- Dimension Set
- Flex Attribute
- Flex Child Definition
- Flex Filter
- Flex Parent Definition
- History Attribute
- History Definition
- Page Attribute
- Page Definition
- Page Filter
- SiteEntry
- Template
- Attribute Editor

Creating a Start Menu Item

1. In the **General Admin** tree, expand the **Admin** node, expand the **Start Menu** node, and then double-click **Add New**.

   The Start Menu form opens.

Figure 9-3 Start Menu Form — Part I



Start Menu

Name:

*Asset Type:

2. 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.

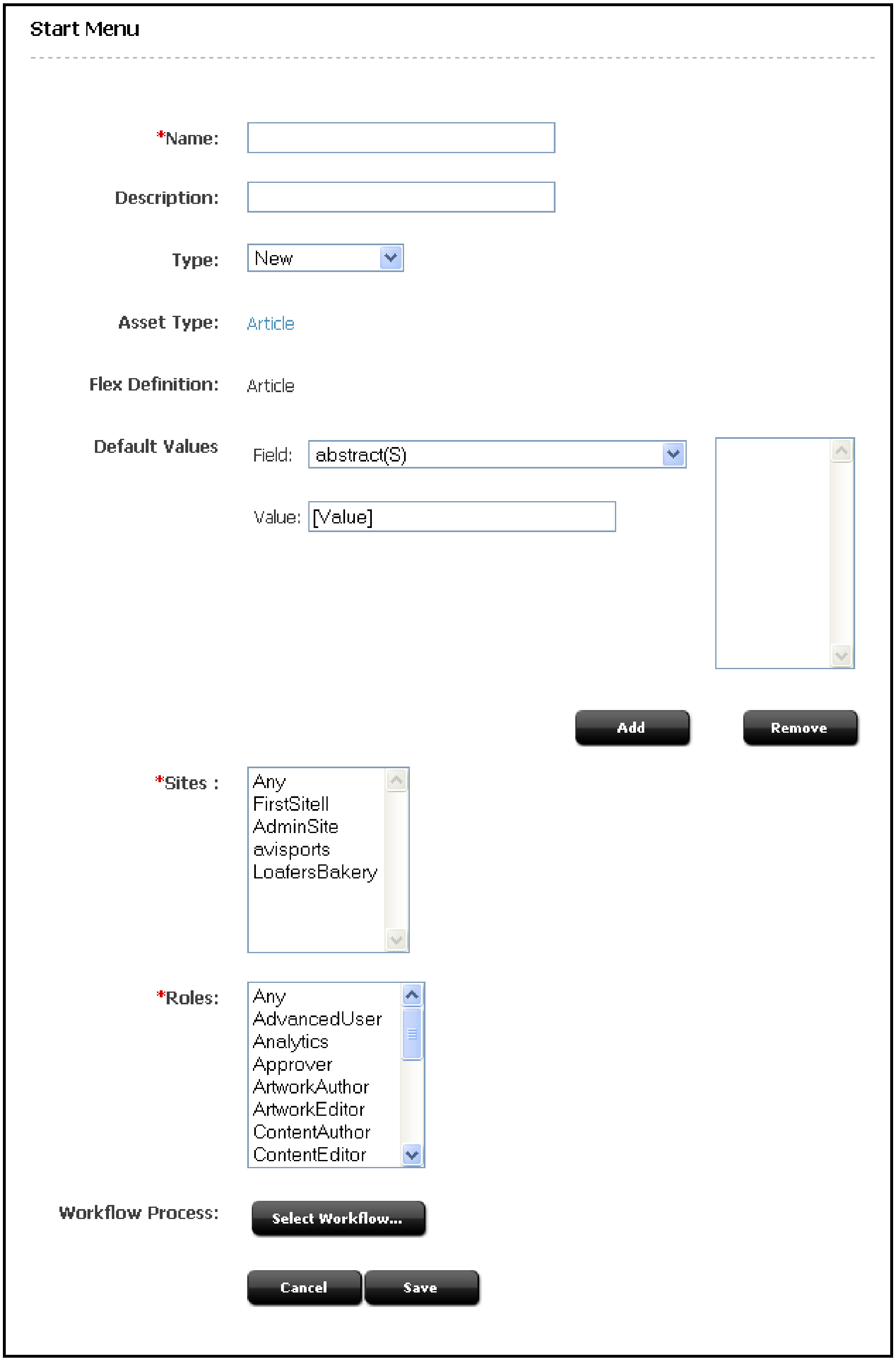
   > **Note:**
   >
   > The following characters are not allowed in the **Name** field: single quote ('), double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

   The name used in the **New** field will be shown on the **New** and **Search** forms if the **Description** field is left blank.

3. In the **Asset Type** field, select the type of asset for which you are creating the Start Menu item.
4. Click **Continue**.
5. 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 6.
   a. In the **Flex Definition** field, select the asset subtype.
   b. If the form does not automatically reload, click **Continue**.

   The Start Menu form continues with new fields.

Figure 9-4 Start Menu Form — Part II



Start Menu

*Name:	
Description:	
Type:	New
Asset Type:	Article
Flex Definition:	Article
Default Values	Field: abstract(S) Value: [Value]
*Sites :	Any FirstSiteII AdminSite avisports LoafersBakery
*Roles:	Any AdvancedUser Analytics Approver ArtworkAuthor ArtworkEditor ContentAuthor ContentEditor
Workflow Process:	

6. In the **Description** field, enter a short description, using up to 255 characters. The name entered in the **Description** field will be shown on the **New** and **Search** forms.

7. In the **Type** field, select **New**.

   Note that the **Asset Type** field is an information-only field that shows your choice of asset type from the previous form.

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 shows 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 multiple sites.

10. In the **Roles** field, select the roles that can have access to the start menu item. Use **Ctrl-Click** to select multiple roles.

    > **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 opens, and click **Set Participants**.
    d. Click **Continue**.

       The system saves the workflow information and re-shows the **Start Menu** form.

12. Click **Save**.

Creating Search Start Menu Items

To create a Search start menu item:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Start Menu** node, and then double-click **Add New**.

   The Start Menu form opens.

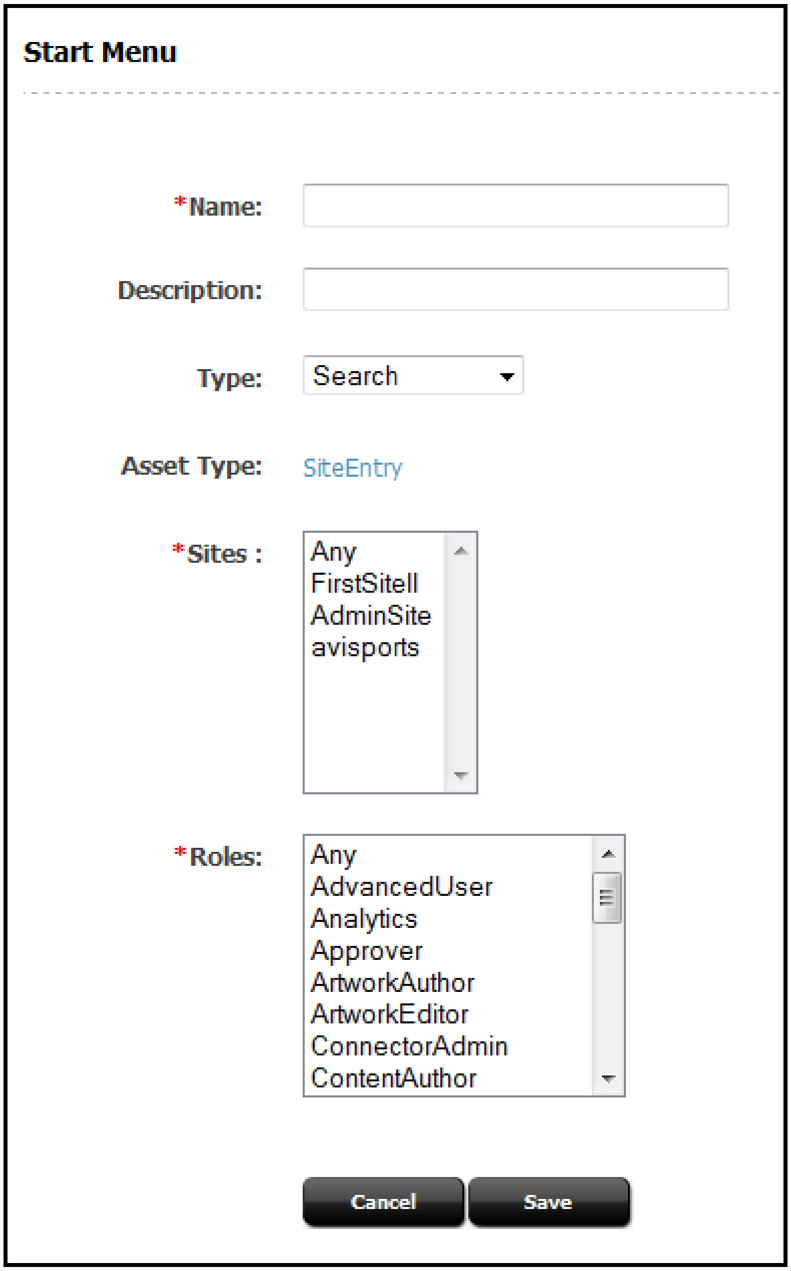
2. In the **Name** field, enter the name of the item, using up to 64 characters. Although it is not required, it is a good idea to start the name with the word "Find" (for example, "Find Article" or "Find Template"). This is the name that is shown on the **Search** list.

   > **Note:**
   >
   > The following characters are not allowed in the **Name** field: single quote (;), double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

3. In the **Asset Type** field, select the type of asset for which you are creating the Start Menu item.
4. Click **Continue**.
5. 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 7.
    a. In the **Flex Definition** field, select the asset subtype.
    b. If the form does not automatically reload, click **Continue**.

   The Start Menu form opens with new fields.

6. **Figure 9-5 Start Menu (for New Search)**



Start Menu

*Name:

Description:

Type: Search

Asset Type: SiteEntry

*Sites : Any, FirstSiteII, AdminSite, avisports

*Roles: Any, AdvancedUser, Analytics, Approver, ArtworkAuthor, ArtworkEditor, ConnectorAdmin, ContentAuthor

7. In the **Description** field, enter a short description, using up to 255 characters.
8. In the **Type** field, select **Search**.

   Note that the **Asset Type** field is an information-only field that shows your choice of asset type.
9. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl-Click** to select multiple sites.
10. In the **Roles** field, select the roles that can have access to the start menu item. Use **Ctrl-Click** to select multiple roles.

> **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. Click **Save**.

Editing Start Menu Items

To edit a Start Menu item:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Start Menu** node, and then expand the node the Start Menu item is located under. Double-click the Start Menu item.

   The Start Menu Item form opens.

2. Click **Edit**.

   The Start Menu form with the stored information for the item opens.

3. Make the changes to the item.
4. Click **Save**.

Showing 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 system, there is a top-level **Start Menu** node in the **Admin** tree.

To show a list of all start menu items, do one of the following:

- Use the **Start Menu** node in the **Admin** tree to see a list of all the start menu items in the system.
- In the Inspect form of any individual start menu item, there is a link you can click to list all the start menu items in the system.

How to Set Access Permissions to Assets

With WebCenter Sites, set access permissions to assets that aren’t part of a workflow process of a site. You can configure policies that define roles for such assets. These roles control the permissions to perform various functions such as checkout, copy, edit, delete, rollback, share, approve, and build on assets.

For definitions of the functions, see Determining Function Privileges. For example, a policy for the Copy function can specify which roles are allowed to copy an asset. A different policy for the Copy function can 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 are inherited by the children.

All of the above-listed functions are defined in the `wcs_properties.json` file, as values of the property `xcelerate.authorizefunction` (categorized under **Security**, if you are using the Property Management Tool).

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 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.

Topics:

- Understanding the Permission Structure
- Working with Authorization Types
- Setting Access Permissions

Understanding the 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, 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 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 user's roles denies the permission, and no 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. 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 must use the Property Management Tool in the Admin interface to examine the `wcs_properties.json` file for this permission. In the `wcs_properties.json` file, permissions for a particular function are set for all assets, rather than for a single asset.

  See Setting Access Permissions from Property Files.

- Asset permissions overrides property settings in the `wcs_properties.json` file.

  For example, if there is a permission `xcelerate.copy.grant` in the `wcs_properties.json` file and the permission to delete asset A is revoked, then the permission to delete asset A is denied.

Working with Authorization Types

Three types of authorization can be defined on a particular function and a particular asset: workflow function privileges, access permissions, and authorization properties in `wcs_properties.json`, all independent of each other.

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 `wcs_properties.json`.
3. If no 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:

a. The system checks the access permissions policy for the function, as defined in the asset's access permissions interface.

b. If no access permission is set for the function (including inheritance of the parent's policy), then the system checks the authorization properties in `wcs_properties.json`.

c. If no levels of authorization are defined, permission to perform the function on the asset is granted, by default.

The different types of authorization are independent of each other. 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 require a simple authorization policy across multiple assets, use the Property Management Tool to set authorization properties in the `wcs_properties.json` file.
3. If you require a simple authorization policy on a particular asset, set access permissions.
4. If you require a more complex type of authorization that allows workflow to control the permissions on an asset, set workflow function privileges.

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 Admin interface to set permissions on individual assets.
- Using the Property Management Tool to set permissions on all assets at one time in the `wcs_properties.json` file.
- Using the Oracle WebCenter Sites: Contributor interface to set permissions on individual assets. For instructions, see Setting Access Permissions from the Contributor Interface in *Using Oracle WebCenter Sites*.

> **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, Restricting Access to Assets While They Are in Workflow.

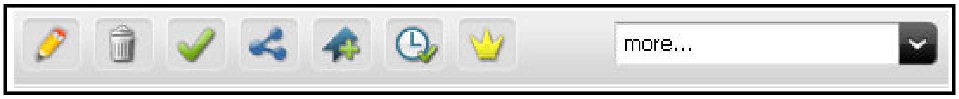
This section covers the following topics:

- The Asset Toolbar
- Setting Access Permissions from the Admin Interface
- Setting Access Permissions from Property Files

The Asset Toolbar

When an asset is viewed, a toolbar is located at the top to aid in working with the asset. This is where you can perform certain function with each asset.

Figure 9-6 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** node in the **My Work** tree.
- **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 Setting Access Permissions from the Admin Interface.
- **Drop-down menu**: this menu allows you additional ways to work with the asset, including using the asset to copy and create an asset, and create an asset and search for other assets.

Setting Access Permissions from the Admin Interface

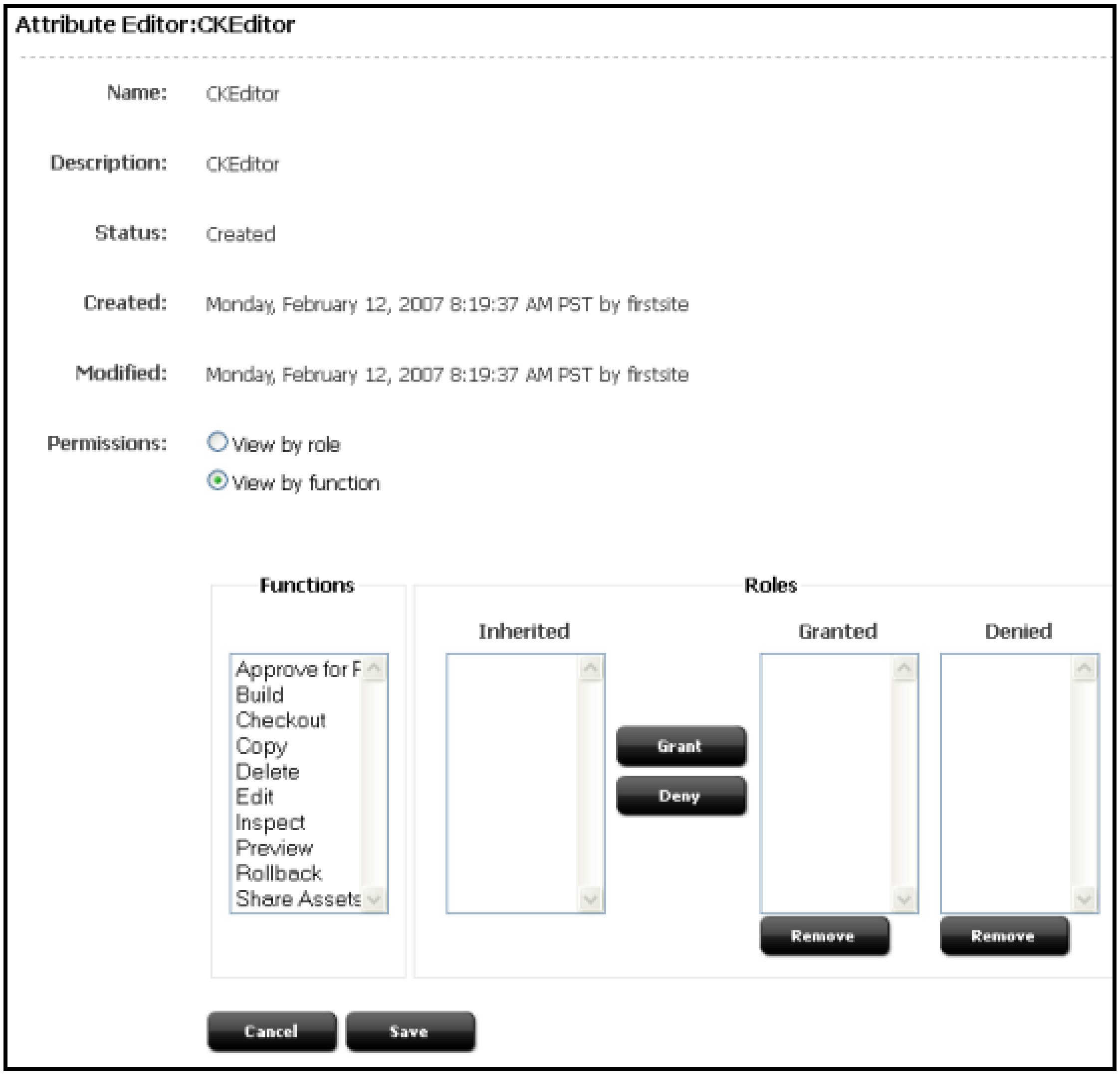
To set access permissions from the Admin interface:

1. Locate the asset for which you will set access permissions.

2. In the asset's **Inspect** form, select **Access Permissions** from the asset toolbar.

   The **Attribute Editor** form opens.

Figure 9-7 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:

Approve for F
Build
Checkout
Copy
Delete
Edit
Inspect
Preview
Rollback
Share Assets

3. Select a radio box, click **View by role** or **View by function**. In this example, **View by Function** is used.

   > **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 is 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 shown 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 beginning and end 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 through 6 for each function and role that must be granted or denied or which you want to grant or deny permission.
8. Click **Save** when finished.

   Users who now access this asset are able to perform the functions that their roles allow.

Setting Access Permissions from Property Files

Use the Property Management Tool to either grant or deny permissions by setting the following properties in `wcs_properties.json` file:

- `xcelerate.grant.`*`functionname`* = rolelist
- `xcelerate.deny.`*`functionname`* = rolelist

  where

  `functionname` is the name of the function, as shown in the `wcs_properties.json file` file, categorized under **Security** if you are using the Property Management Tool.

  `rolelist` is a comma-separated list of roles for which the permission is either denied or granted.

If the permission is granted or denied in `wcs_properties.json`, it is granted or denied for all assets. The following two examples provide more information.

Example A

```
xcelerate.grant.edit=
```

In the `wcs_properties.json` file, if the property `xcelerate.grant.edit` doesn't have any value, then the access permission specified from the Contributor user interface holds true.

Example B

```
xcelerate.grant.edit=RoleY
```

In the `wcs_properties.json` file, if the property `xcelerate.grant.edit` has value=RoleY, then the access permission specified from the Contributor user interface is used along with the value specified in wcs_properties.json file. Therefore, the role specified from the Contributor user interface is used with RoleY and both can have the edit access.

Other Options for Asset Types

Oracle WebCenter Sites provides site developers with various items in the **General Admin** tree, under the **Admin** node to configure asset types. They can design and implement asset types with the help of these items.

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 Creating Basic Asset Types in *Developing with Oracle WebCenter Sites.*

Options for Managing Access to the Tree (Admin Interface Only)

In the **General Admin** tree, under the **Admin** node provides various items to configure asset types. You can configure this **General Admin** tree according to your preferences and requirements.

You have several ways to configure the tree that is shown in the left-hand panel of the Admin interface:

- You can configure the tree to be either shown 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 nodes within the tree, adding items or functionality to them, and determining which users have access to them based on their roles.
- You can configure a tree node's sub nodes to specify the number of items that are shown under any given node.

WebCenter Sites supports two general categories of trees and nodes:

- System default trees and nodes, which provide administrative functionality (for example, **Admin**, **Site Admin**, and **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, Oracle WebCenter Sites: Engage assets (such as recommendations, segments and promotions), Query, Collection, and Page assets.

- Custom trees:
  - Trees that you create to make it more convenient for users in different roles to create the most commonly used asset types.
  - Trees that developers create to offer customized behavior or functionality.

> **Note:**
>
> The **Workflow Groups** node is a hybrid between a system default and a custom tab. Although the **Workflow Groups** node 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 node, your developers do not have to code an element for you.

The following topics describe how to add roles to the system default trees and nodes, how to create basic tree nodes that provide access to asset types, and how to configure whether the tree is shown by default. See Customizing the Content Tree in *Developing with Oracle WebCenter Sites*.

- Showing and Hiding the Tree
- About Creating Tree Tabs
- Editing Trees and Nodes
- Deleting Tree Tabs
- Changing Tree Tab Order
- Configuring the Number of Items Shown Under Nodes on Tabs

Showing and Hiding the Tree

Any user can toggle the tree on and off whenever he requires less or more space in the main window to show the assets that he is working on.

You can configure your system so that the tree is toggled off by default. You can also control whether users are allowed to toggle the tree back on.

The features for working with assets are available in right-mouse menus on the tabs and through the icons at the top of the window and the action bars shown in the **New**, **Edit**, and **Inspect** forms for individual assets managed solely in the Admin interface. Therefore, it is not necessary to show 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 the tree and toggling rights:

1. In the Admin interface, open the Property Management Tool.
2. In the **Category** drop-down menu, select **UI**.

3. Click **Search**.
4. To specify that the tree should be toggled off by default when users first log in to the Admin interface, do the following:
   a. Select the `xcelerate.showSiteTree` property.
   b. In the **Value** field, set its value to `false`.
   c. Click **Save**.
5. (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, do the following:
   a. Select the `xcelerate.restrictSiteTree` property.
   b. In the **Value** field, set its value to `true`.
   c. Click **Save**.
6. Restart the application server.

About Creating Tree Tabs

A tree node is a node (such as Admin) that opens in the left-hand panel of the Admin interface. Access to tree nodes is controlled through roles.

Several default tree nodes are core to WebCenter Sites. Before creating a tree node, review the default nodes to gain an understanding of the existing capabilities of WebCenter Sites. The default tree nodes are:

- **Bookmarks** – shows 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** – shows 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 node.
- **Mobility** – enables developers to configure the Mobility feature. By default, only users with the default system role named `GeneralAdmin` have access to this node.
- **Connector Admin** – shows the administrative functions for administering WebCenter Content.
- **Dev** — 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** – shows 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 have to create users or sites.
- **Visitors Admin** – enables developers to configure and manage Visitor Services through the WebCenter Sites Admin interface.

- **Site Navigation** – shows 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.
- **Bookmarks** – holds a list of bookmarked pages.
- **History** – shows the assets that you worked with during the current session.

This section covers the following topics:

- Creating a Tree Tab
- Creating the Workflow Groups Tab

Creating a Tree Tab

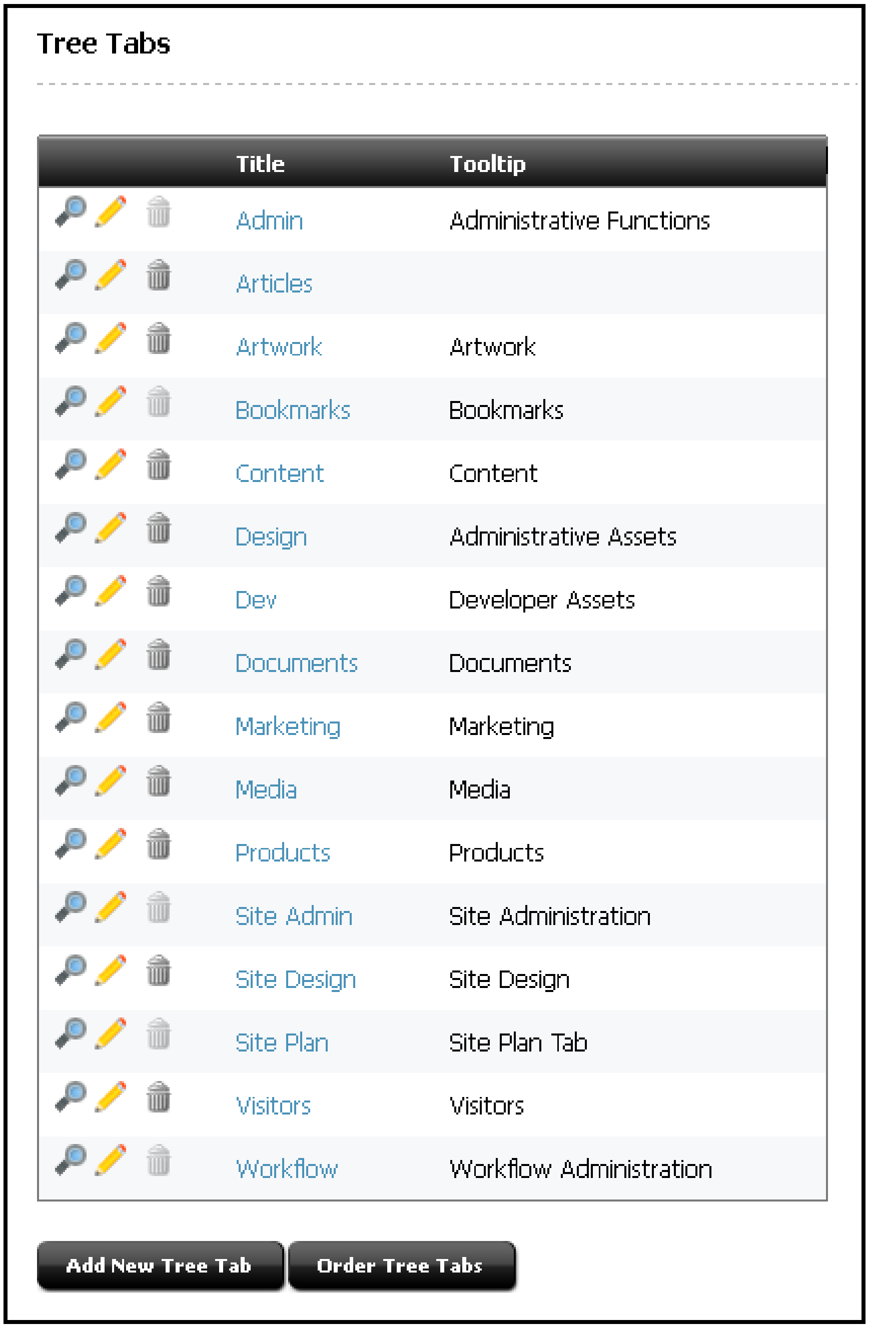
> **Note:**
>
> Before creating a tree tab, familiarize yourself with the default tree tabs to avoid duplicating their functions.

To create a tree tab:

1. In the **Admin tab,** double-click **Tree**.

   The **Tree Tabs** form opens.

Figure 9-8   Tree Tabs Form



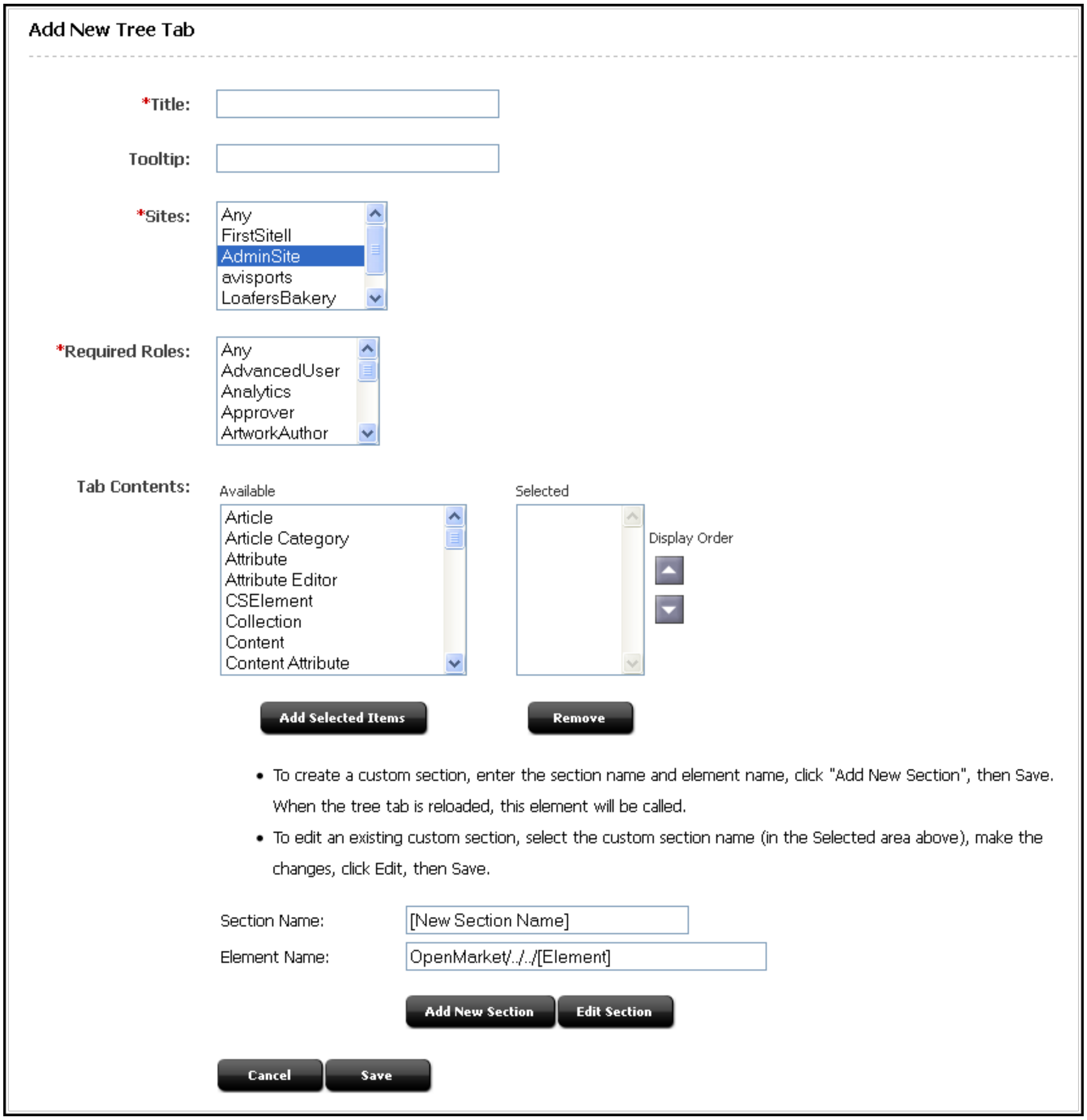
Tree Tabs

	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

2. In the **Tree Tabs** form, click **Add New Tree Tab**.

   The Add New Tree Tab form opens.

Figure 9-9 Add New Tree Tab Form



- 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: [New Section Name]

Element Name: OpenMarket/../../[Element]

3. In the **Title** field, enter a short, descriptive name of up to 64 characters.
4. In the **Tooltip** field, enter a short, informative description of the tab, up to 255 characters. This description is shown when the mouse hovers over the tab.
5. In the **Site** field, select the sites that will show the tab.
6. In the **Required Roles** field, select which roles can access the tab.
7. In the **Tab Contents** field, select the asset types and then click **Add Selected Items**.
8. (Optional) To add custom functionality to this tab, use the **Section Name** and **Element Name** fields at the bottom of the form. You will require help from your developers for this step. See Implementing a Custom Tree in *Developing with Oracle WebCenter Sites*.
9. Click **Save**.

The tab is shown in the tree (refresh if necessary).

Creating the Workflow Groups Tab

The **Workflow Groups** tab shows the workflow groups that exist for a site and the assets that are included in those groups. See Understanding Workflow Groups. This tab does not appear by default. To use it, you must create it.

To create the Workflow Groups tab:

1. In the **Admin** tab, double-click **Tree**.

   The Tree Tabs form opens.

2. In the form, click **Add New Tree Tab**.

   The Add New Tree Tab form opens.

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 show 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 `Groups`.
8. In the **Element Name** field, enter `OpenMarket/Gator/UIFramework/LoadWorkflowGroups`
9. Click **Add New Section**.

   The word Groups opens in the **Selected** column.

10. Click **Save**.

Editing Trees and Nodes

Editing a tree or node means that you can:

- Rename it.
- Edit its tooltip.
- Enable it for a different set of sites.
- Show it to a different set of roles.
- Select a different set of content for it, and/or rearrange the content's order.

> **Note:**
>
> A tree node is shown in the Contributor interface when two conditions are met: the tree node is enabled on the user's site, and both the tree node and the user have at least one role in common.
>
> When you create site users with roles, you must select the roles in the default tree nodes so that the users have access. (Default tree nodes are not automatically enabled for roles.).

To edit a tree or node:

1. In the **General Admin** tree, expand the **Admin** node, and then double-click **Tree**.

   The Tree Tabs form opens.
2. Navigate to the tab and click its **Edit** icon.

   The Edit Tree Tab form opens.
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**.

Deleting Tree Tabs

To delete a tree tab:

1. In the **General Admin** tree, expand the **Admin** node, and then double-click **Tree**.

   The Tree Tabs form opens.
2. In the Tree Tabs form, navigate to the tree tab and click its **Delete** icon.

   A warning message opens.
3. Click **Delete Tree Tab**.

   The tree tab has been deleted.

Changing Tree Tab Order

You can change the order in which the tree tabs (nodes) are shown.

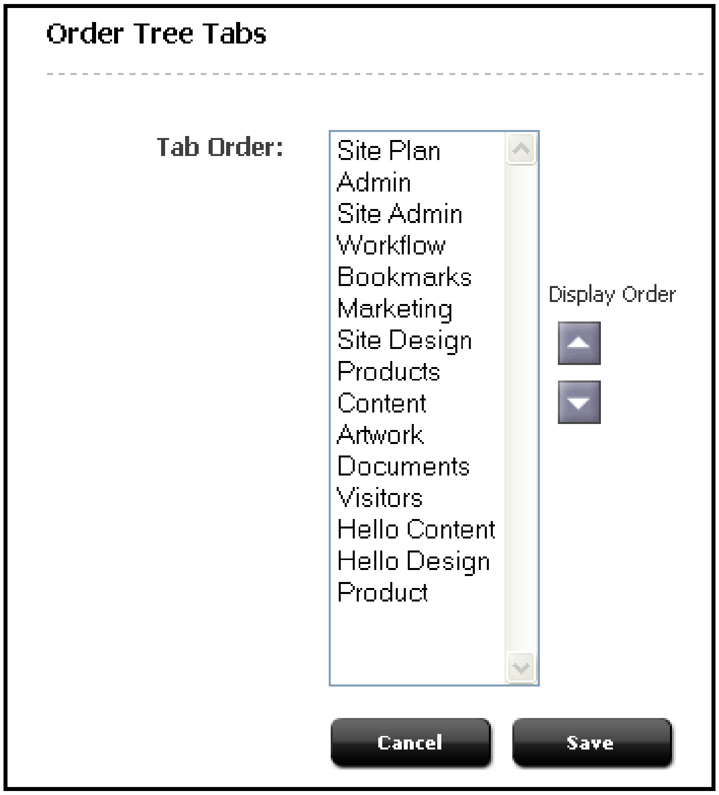
To change the order of tree tabs:

1. In the **General Admin** tree, expand the **Admin** node, and then double-click **Tree**.

   The Tree Tabs form opens.
2. At the bottom of the **Tree Tabs** form, click **Order Tree Tabs**.

   The Order Tree Tabs form opens.

Figure 9-10 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

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 to reposition.
4. Click **Save**.

   The new tab order is shown in the tree (refresh if necessary).

Configuring the Number of Items Shown Under Nodes on Tabs

To optimize the response time of , you may want to limit the number of assets that are listed under their respective node in a tree tab. When the database holds many assets of any one type, can take a long time to load all the assets and then list them under their node in the tab. Limit the number of items to be listed under the node by setting a threshold to minimize loading time. 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, consider a node contains more than 300 articles, and the threshold is set to 100. When a user expands the node icon that represents the articles, shows a small search criteria form that prompts the user to enter values to limit the number of articles that it returns.

To configure the number of items under a node in a tree tab:

1. In the Admin interface, open the Property Management Tool.
2. In the **Name** field, enter `xcelerate.treeMaxNodes`.
3. Click **Search**.
4. Select the name of the property. In the **Value** field, specify a value for it. By default, this property is set to 100, which means that up to 100 items can be shown under a node.

   > **Note:**
   >
   > When the number of items to display exceeds the value defined in the `xcelerate.treeMaxNodes` property, categorized assets (AviSports, Tennis Articles, Basketball Articles, and so on) are properly delimited with a Next and Previous button that allows the users to load the articles in sets of X assets. However, uncategorized assets are not delimited. However, the current tree design in contributor user interface doesn't allow this process at the root(tab) level as multiple tab contents are displayed at the root level.

5. Click **Save**.
6. Restart the application server.

10 Working with the Admin Interface

The Admin interface has a tree structure with nodes. With the Admin interface you can manage sites, configure its users and default settings, and manage security and workflow. The Developers use the Admin interface to build data models for assets, design and configure attribute editors of assets, and configure mobility and multilingual support for sites.
The following topics describe the three types of administrators and describe the Admin interface, its trees and nodes (tabs), that allow you to manage sites, and system defaults that are accessible either from the Admin interface or the back end:

- About the Admin Window
- Administrator-Specific Trees and Nodes
- Miscellaneous Trees and Nodes
- About System Defaults

About the Admin Window

Along with Developers, you can use the Admin interface to perform tasks and configure elements that help in defining and creating a functional site structure. Use the Admin interface to manage WebCenter Sites content management sites, asset types, users, and features such as publishing, workflow, and security.

Administrators use WebCenter Sites Admin to:

- Create and configure users accounts and profiles, including creating and assigning ACLs and roles to users.
- Configure and implement security for WebCenter Sites, including REST security, encryption key security, and external security.
- Configure and schedule publishing sessions.
- Create and manage workflow processes.
- Enable revision tracking on asset types.
- Enable asset types on content management sites and create start menu items which enable users to work with those asset types in the Contributor interface.
- Configure vanity URLs on a per-asset type basis by creating WebRoots and defining URL patterns.
- Manage the WebCenter Sites caching system (inCache framework).
- Use diagnostic utilities to troubleshoot the WebCenter Sites system.
- Integrate and manage WebCenter Content with WebCenter Sites.
- Configure the Lucene search engine for the Contributor interface.

Developers use WebCenter Sites Admin to:

- Build the data model by creating and defining asset types. For example:

  - Creating flex families, which includes defining flex attributes, flex parents, flex parent definitions, flex definitions, flex assets, and flex filters.
  - Creating basic asset types and database tables for those basic assets.
  - Creating proxy assets
- Create design assets for content management sites, such as Template, CSElement, SiteEntry, Wrapper, Filter, and so on.
- Design attribute editors and configure bundled attribute editors.
- Configure sites for multilingual support.
- Configure the Mobility feature. For example, create device repositories, device groups and suffixes, device assets, site navigations, and mobile templates.

For more information about the Admin interface, see the following topics:

- Administrator-Specific Trees and Nodes
- Miscellaneous Trees and Nodes
- About System Defaults

Administrator-Specific Trees and Nodes

You can perform your tasks as an administrator by using the nodes and sub nodes of the General Admin and Site trees. Users having appropriate ACLs and roles can view these trees and nodes.

The following table shows these trees and nodes and the ACLs and roles required to view them.

Table 10-1 Tab Permissions and Roles

Tree	Node	Sub Nodes	ACL	Role
General Admin	Admin	• Sites • Asset Types • Publishing • Search • User Access Management • WebRoots • Start Menu • Tree • Locale • System Tools • Clear Checkouts • Functions	xceladmin	GeneralAdmin
General Admin	Content Cloud Admin	• Console • History	xceladmin	General Admin
General Admin	Connector Admin	• Console • History • Orphaned Content • Configuration • Rules	xceladmin	ConnectorAdmin

Table 10-1 (Cont.) Tab Permissions and Roles

Tree	Node	Sub Nodes	ACL	Role
General Admin	Workflow	• States • Processes • Actions • Timed Action Events • Emails • Clear Assignments	xceladmin	WorkflowAdmin
Site	Site Admin	Modify the following for the site you are logged in to: • Users • Asset Types • Site Navigations	xceladmin	SiteAdmin
Site	Site Navigation	Contains a node for each site navigation associated with the site you are logged in to.	xceladmin	SiteAdmin

The following Administrative Trees are displayed based on the type of site you are logged in to:

- If you are logged in to a custom site, both **General Admin** and **Site** trees are displayed. The **General Admin** tree remains system-wide. The **Site** tree is specific to the site you are logged in to. It is therefore a sub-set of the **Admin** node in the **General Admin** tree.
- If workflow processes are created for any site (except the default site), the **Workflow** node is available in the **General Admin** tree. The **Workflow** node lists all workflow processes that have been configured for all sites in the system.

Administrative Nodes in the General Admin Tree

The **General Admin** tree is the main tree used to create, configure, and manage sites.

The **General Admin** tree displays the following administrator-specific nodes and sub nodes:

- **Admin**: Accounts for all content management components in the WebCenter Sites installation. It includes the following sub nodes specific to administrative functions:
  - **Sites**: Enables you to manage all content management sites on the WebCenter Sites system.

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

    - List of sites that are created in the system.
    - Enables you to define sites and associate the components from the system-wide configuration pool, below the **Sites** node. WebCenter Sites provides several sample sites as learning tools.

    The configuration pool consists of the following nodes:
  - **Asset Types**: Configure 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 Managing Revision Tracking.

- **Publishing**: Set up the publishing system on the development and management systems.
- **Search**: Configure asset types so that they are indexed correctly when you are using supported third-party search engine modules. See Configuring the Lucene Search Engine.
- **User Access Management**: Create ACLs and users, to enable or disable revision tracking for non-asset tables. This node includes the following:
    - **Users**: Manage WebCenter Sites users and the roles associated with each user ID.
    - **Roles**: Use roles for determining the recipients of workflow assignments and to control users' access to functions in the Oracle WebCenter Sites interfaces.
    - **ACLs**: Allow access to individual tables and sites.
    - **User Profiles:** Specify email addresses for your users.
    - **REST Security**: Configure the security groups and assigning users to those groups. Adding a user to the security groups is tied to the WebCenter Sites interface.
- **WebRoots**: Interpret a vanity URL. Click **Add New** to create WebRoots. You can edit existing WebRoots. See Working with WebRoots.
- **Start Menu**: Create links in the form that opens when users click the **New** button or **Search** button. You use the links to create assets or search for existing assets.

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

- **Tree**: Control access to the existing tree tabs, create tabs, or view assets in the tree in the Oracle WebCenter Sites: Contributor interface.
- **Locale:** Set the default locale for the WebCenter Sites system. See Configuring the User Interfaces.
- **System Tools**: Troubleshoot problems directly from the administrator's interface. Features include configuring log4j logging, accessing various types of system information, managing caches, searching logs, testing the performance of the shared file system, and to perform various other operations on WebCenter Sites database.
- **Clear Checkouts**: Unlock and check back in to the database the assets that are checked out by specific users. See Managing Revision Tracking.
- **Functions**: Administrators can enable or disable in the Oracle WebCenter Sites interfaces of all users or selected users.

- **Content Cloud Admin**: You can configure Content Cloud with WebCenter Sites that connects WebCenter Sites and Content Cloud .

  You can view the following sub nodes in the **Content Cloud Admin** node:

  - **Console**: Enables you to configure Content Cloud with WebCenter Sites.
  - **History**: Displays the synchronization logs between Content Cloud WebCenter Sites and Content Cloud, and helps you to identify the status of the synchronizations.

- **Connector Admin**: Shows the administrative functions for administering WebCenter Content. See Understanding WebCenter Content Integration.

  You can view the following nodes in the **Connector Admin** node:

  - **Console**: Displays the Recent Activity table. The table displays the recent connector activity.
  - **History**: View the archived connector sessions. You can also drill down to individual content items for detailed information about their processing.
  - **Orphaned Content**: Displays assets that you cannot delete when its counterpart content item is either deleted, expired, or disabled from synchronization. See Orphaned Content.
  - **Configuration**: Configure WebCenter Sites to connect with WebCenter Content.See Connecting to WebCenter Content.
  - **Rules**: Set of instructions enabled for the connector to follow when it runs. See Creating an Active Rule.

- **Workflow**: Workflows are configured from this tree node. Workflows are created from the workflow building blocks (states, assets, users) that were created and assigned using the subnodes under the Admin node. For more information about workflow, see Creating and Managing Workflow Processes.

  > **Note:**
  >
  > While the **Workflow** tab is site-specific, its content is not.
  >
  > - The tab: When an administrator logs in to a site, the **Workflow** tab opens 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**: Create asset status points in the workflow.
  - **Actions**: Create workflow building blocks called actions. Any workflow can use these actions.
  - **Timed Action Event:** Configure how often due dates are calculated for assets in workflow.

- **Processes:** Create a workflow process, specify global process information, and then create steps, assigning step actions to them as required. The steps in the process create the flow of the process by linking the states in a specific order.
- **Email**: Create the workflow email messages. Any workflow can use these emails.
- **Clear Assignments**: Clear workflow assignments.

About the Site Admin Node in the Site Tree

The **Site Admin** node in the **Site** tree is site-specific. Use it to manage the selected site.

The **Site Admin** node holds a subset of the administrative functions in the **General Admin** tree’s **Sites** node. It allows you to manage existing users of the site, enable or disable asset types. It does not allow you to create users, asset types, or sites. You use this tree when your administrative roles and ACLs limit you to administer one or a few sites, rather than when you have administrative rights across the entire system. If you have access to all sites, it would not matter if you used this tree or the **General Admin** tree.

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.

Developer-Related Trees and Nodes

Developers use the **General Admin** and **Content** trees in the left navigation pane of the Admin interface to create the framework for the content management site and its components. Only the users with coding knowledge and required ACLs and roles can perform the developer-specific tasks using the **General Admin** and **Content** trees.

The following table provides information about the permissions required to access developer-related tree nodes.

Table 10-2 Tree Permissions and Roles

Tree	Node	Sub Nodes	ACL	Role
General Admin	Admin	• AssetMaker • Flex Family Maker • Proxy Asset Maker	xceladmin	GeneralAdmin
General Admin	Mobility	• Device Groups • Devices • Device Repository • Image Properties	xceladmin	GeneralAdmin
General Admin	Visitors Admin	• Identity Providers • Access Providers • Profile Providers • Aggregation Templates • General Configuration	xceladmin	GeneralAdmin

About the Developer Nodes in the General Admin Tree

The developer-specific nodes in the **General Admin** tree are used to configure, integrate, and otherwise develop WebCenter Sites and its features.

The **General Admin** tree displays the following developer-specific nodes and sub nodes:

- **Admin**: Enables developers to create and manage the asset types on your content management sites. It includes the following sub nodes specific to development functions:
  - **AssetMaker**: for site developers to create asset types using the basic asset data model. See Creating Basic Asset Types in *Developing with Oracle WebCenter Sites*.
  - **Flex Family Maker**: for site developers to create new flex families and flex asset types. See Creating a Flex Asset Family in *Developing with Oracle WebCenter Sites*.
  - **Proxy Asset Maker**: for site developers to create proxy asset types. See Installing Sample Proxy Assets in *Developing with Oracle WebCenter Sites*.
- **Mobility**: Enables developers to configure the Mobility feature. It includes the following sub nodes:
  - **Device Groups**: for site developers to create and enable device groups for all content management sites on the WebCenter Sites system. See About Device Groups and Suffixes in *Developing with Oracle WebCenter Sites*.
  - **Devices**: for site developers to create device assets, which are then associated with device groups. See About Device Assets in *Developing with Oracle WebCenter Sites*.
  - **Device Repository**: Used by developers to upload the device repository, which is a file that WebCenter Sites uses to detect mobile devices. The device repository contains the properties of devices and uniquely identifies each device based on its user agent. Developers can either activate the default device repository or upload a third-party WURFL file. See About Device Repository in *Developing with Oracle WebCenter Sites*.
  - **Image Properties**: Used by developers to set the properties of device images for optimization. See How to Create Device Assets in *Developing with Oracle WebCenter Sites*.
- **Visitors Admin**: Enables developers to configure and manage Visitor Services through the WebCenter Sites Admin interface. The following options enable you to add, remove, and change aggregation templates and providers without restarting the server.
  - **Identity Providers**: Enables developers to create and configure a custom identity provider to replace Oracle Access Manager (OAM).
  - **Access Providers**: Enables developer to configure access providers. Access providers verify the credentials of components (WebCenter Sites and Engage) requesting profile information from Visitor Services through the Visitor Services API client.
  - **Profile Providers**: Enables developers to create and configure a customer profile provider. By default profile providers are provided for LDAP and Eloqua.

   - **Aggregation Templates**: Enables developers to create templates that identify the attribute each profile provider will supply and to identify and aggregate like attributes across visitor profiles even when attributes have different names.
   - **General Configuration**: Enables developers to configure an instance of Visitor Services with WebCenter Sites which, in turn, enables WebCenter Sites to request profiles from Visitor Services through REST APIs, Java APIs, and JavaScript APIs.

About the Developer Nodes in the Content Tree

The **Content** tree provides developers with shortcuts to access design assets such as Template, SiteEntry, and Attribute assets that have been created on your WebCenter Sites system.

Provides developers with access to the following types of assets:

- Attribute
- Attribute Editor
- Definition
- Parent Definition
- Page Attribute
- Page Definition
- Template
- CSElement

Developers can expand an asset type node to view all the assets that have been created from that type.

Miscellaneous Trees and Nodes

Apart from the administrative nodes in the trees of the Admin interface, there are miscellaneous trees and nodes such as the default tees and nodes and nodes and trees customized and configured by administrators.

See Options for Managing Access to the Tree (Admin Interface Only).

The default miscellaneous trees and nodes are:

- **My Work** tree:
   - **Bookmarks**: Holds a list of book marked pages. Frequently-used pages can be marked and accessed from this tab to make finding them easier.
   - **History**: Shows 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.
- **Site Navigation** node, under the **Site** tree: Represents the layout and overview of the site. This tab shows each site that is controlled by WebCenter Sites.

  It displays lists of **Placed** pages and the **Unplaced** pages. 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.

It displays nodes such as **Default**, **Touch**, or **NonTouch** based on the suffix you select when you associate device groups to the navigation. See How to Create Site Navigations in *Developing with Oracle WebCenter Sites*.

About System Defaults

The Admin interface of WebCenter Sites has several system defaults. Familiarise yourself with these system defaults to help you configure users and sites.

- Trees and tree nodes
- ACLs
- Roles
- Users
- Asset types
- System Tables

You must be familiar with the defaults to configure users and sites. Information about system defaults is given in System Defaults and WebCenter Sites Database Tables.

11

Using the Web Experience Management Framework

The Web Experience Management (WEM) Framework helps Use to create applications that can be integrated with WebCenter Sites, authorize users, and manage sites, users, roles, and profiles through the WEM interface.
Topics:

- About the WEM Admin Window
- Overview of the Web Experience Management Framework
- Administrative Roles and Permissions
- WebCenter Sites Applications
- Managing WebCenter Sites CM Sites in the WEM Framework
- Enabling Tree Tabs
- Managing the WEM Framework - Quick Tips and Roadmaps
- Common WEM Tasks for WebCenter Sites Administrators

About the WEM Admin Window

Use the WEM Admin interface that runs on WebCenter Sites to manage the sites, applications, users, and roles for WebCenter Sites systems. Developers use it to create applications and integrate them with WebCenter Sites, but its main function is to define roles for users to create applications.

WEM Admin supports only general and site administrator functions. To access this interface, the WEM Framework must be running on WebCenter Sites, you must have access to AdminSite, and you must be assigned either the `GeneralAdmin` or `SiteAdmin` role.

When WEM Admin opens, the first screen you see is the Sites list. General administrators see all the sites created on the WebCenter Sites system while site administrators see only the sites in which they are assigned the `SiteAdmin` role. The menu bar at the top of the WEM Admin interface displays the following options:

- **Sites**: Click this option to view a list of all the sites on the WebCenter Sites system in which you are assigned either the `GeneralAdmin` or `SiteAdmin` role. If you are a general administrator, there is an **Add Site** option displayed above the list of sites. Click the **Add Site** option to create a new content management site.
- **Apps**: Click this option to view a list of all the applications available to be assigned to the sites on your WebCenter Sites system.
- **Users**: Click this option to view a list of all the users created for your WebCenter Sites system. If you are a general administrator, there is an **Add User** option displayed above the list of Users. Click this option to create a new WebCenter Sites user.

- **Roles**: Click this option to view a list of all the roles that have been created for your WebCenter Sites system. If you are a general administrator, there is an **Add User Role** option displayed above the list of roles. Click this option to create a new user role for your WebCenter Sites system.
- **Down**/**Up** arrow: Located at the top-right of the menu bar. Click the arrow to either open or close the applications bar. From the applications bar you can switch to another site or application, edit your WebCenter Sites profile, and log out of WebCenter Sites.

For more information about working in the WEM Admin interface and the functions you can perform in WEM Admin as either a general administrator or site administrator, see the following topics:

- Overview of the Web Experience Management Framework
- Administrative Roles and Permissions
- WebCenter Sites Applications
- Managing WebCenter Sites CM Sites in the WEM Framework
- Enabling Tree Tabs
- Managing the WEM Framework - Quick Tips and Roadmaps
- Common WEM Tasks for WebCenter Sites Administrators

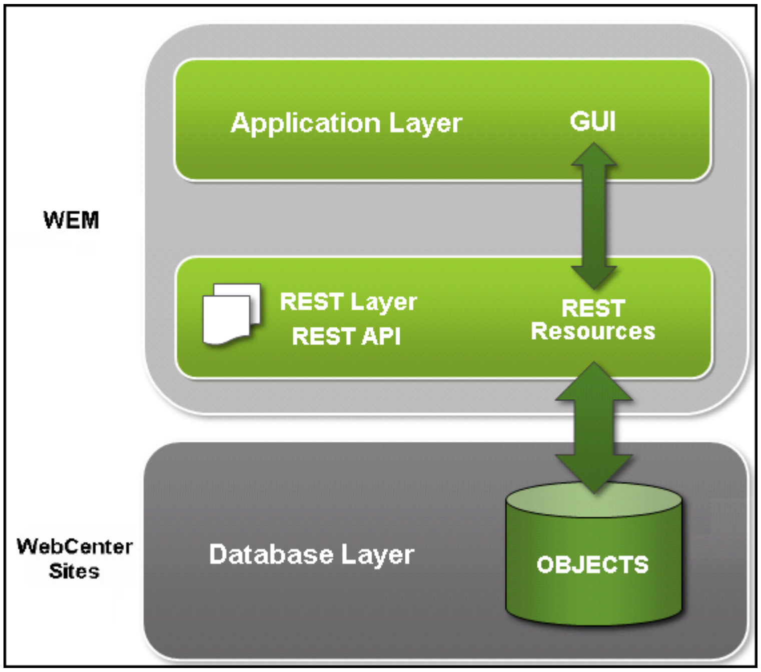
Overview of the Web Experience Management Framework

Use the WEM Framework's technology to develop applications and integrate them with Oracle WebCenter Sites. The WEM Admin interface enables you to manage applications centrally and authorizes users to use single sign-on facility to access the applications.

The WEM Framework requires a content management platform. The Framework runs on 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.

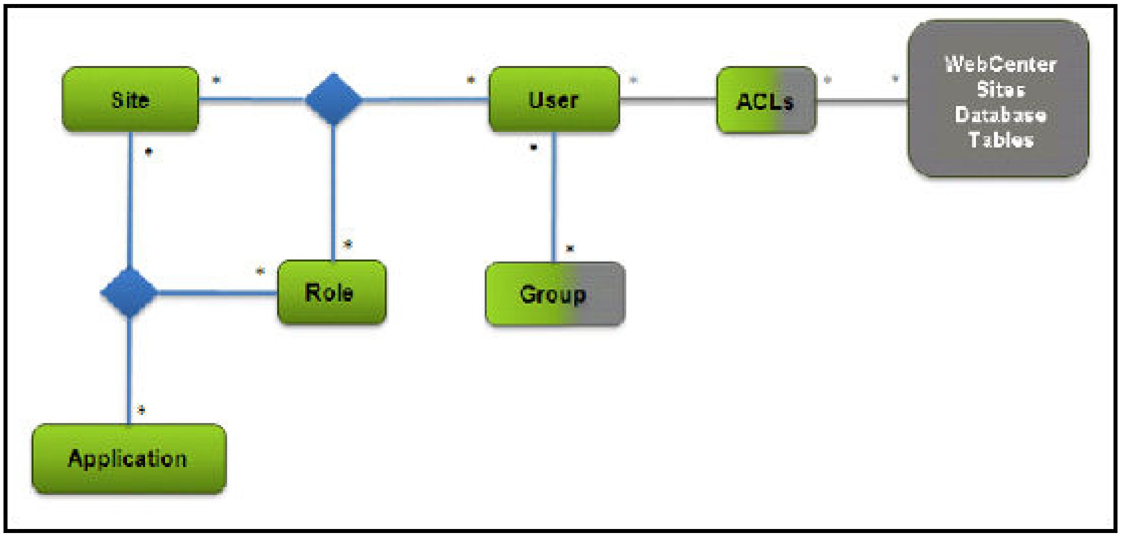
When implemented on the WEM Framework, applications communicate with the WebCenter Sites database through REST services. The applications appear in WEM Admin as list items on the Apps page. Administrators authorize users, which involves configuring access to applications and their resources.

Figure 11-1 WEM Framework



Coupling the items as shown in Figure 11-2 enables applications for users.

Figure 11-2 Authorization Model

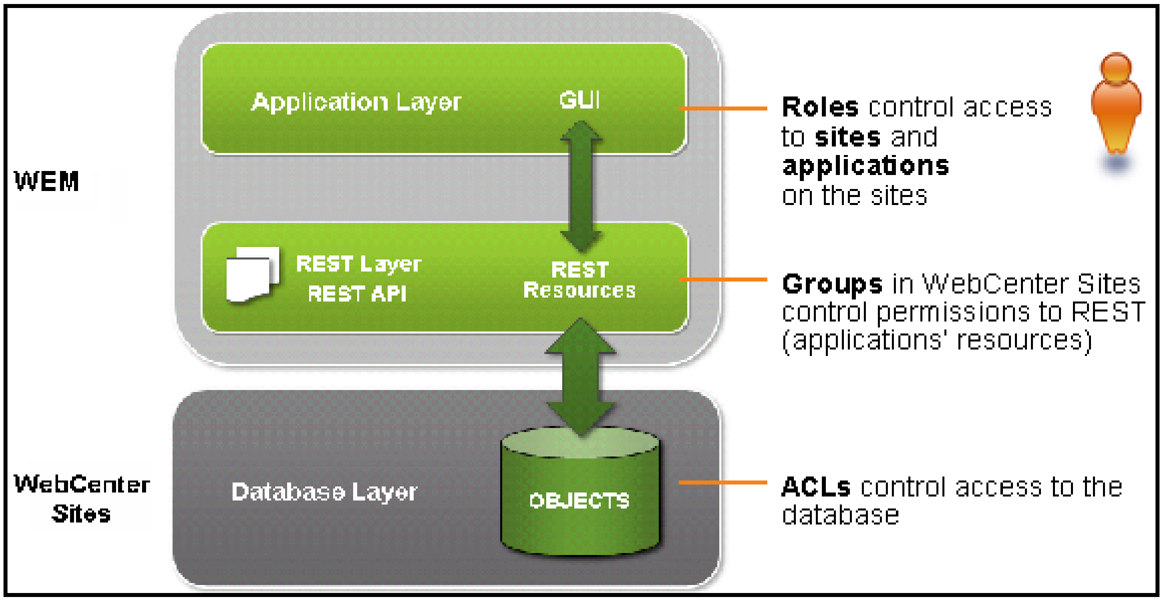


- Applications and users are assigned to sites through roles.
- Sharing a role to a user and an application on the same site grants the user access to the application on that site.
- ACLs are assigned to users, providing them with access to the system.

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.

After the coupling is complete, users are authorized at the database, REST, and application levels.

Figure 11-3 WEM Framework


WEM
WebCenter Sites
Application Layer
GUI
REST Layer
REST API
REST Resources
Database Layer
OBJECTS
Roles control access to sites and applications on the sites
Groups in WebCenter Sites control permissions to REST (applications' resources)
ACLs control access to the database

Experienced WebCenter Sites administrators 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 application content.

Unlike WebCenter Sites, WEM Admin does not expose the data model. The REST API does. In this respect, you can think of WEM Admin as strictly an authorization interface, supported by the WebCenter Sites Admin interface (for configuring ACLs and groups). The rest of this guide provides instructions on creating and authorizing users, and guidelines for managing sites.

Administrative Roles and Permissions

You can work in the WEM Admin interface either as a general administrator or a site administrator.

This section covers the following topics:

- General Administrator
- Site Administrator

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) is 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 (using the GeneralAdmin role). Instructions are available in Creating and Authorizing Users.

Site Administrator

General administrators assign specific administrators to selected sites, where the Site administrators 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. You cannot assign users the SiteAdmin role in AdminSite only.

In the WEM Admin interface, site administrators can access the Sites form only. They can perform the following administrative tasks:

- Assign and remove users to and from sites
- Assign and remove applications to and from sites
- Change the role assignments of site users
- Change the role assignments of applications on the sites

A site administrator cannot create, modify, or delete sites, users, or roles. As a site administrator you determine how they are connected within the parameters of your site. Your permissions determine which forms and interface functions WEM Admin you can access.

You must specially configure site administrators for WEM on WebCenter Sites systems running the WEM Framework. You must assign them to the SiteAdmin_AdminSite group, which is a default REST security group configured in the WebCenter Sites Admin interface.

WebCenter Sites Applications

Through the WEM Admin interface, you can enable the Admin interface and Contributor interface applications. Users can access these applications if you authorize them at the application level.

The Contributor and Admin interfaces are registered applications. They are listed on the Apps page of the WEM Admin interface.

Managing WebCenter Sites CM Sites in the WEM Framework

You can configure the WebCenter Sites applications to run on the active WebCenter Sites CM sites to access these sites quickly. It is also important to remember that deleting sites from the WEM Admin interface deletes the sites from the system and from any applications where the sites are in use.

Before deleting or modifying sites from the WEM Admin interface, ensure that the sites are not active content management sites in the WebCenter Sites platform. When you attempt to delete a site you are prompted to confirm your decision. It is assumed that you have determined the status of the site.

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 are presented with a list of dependencies which you must clear before you can delete the role.

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. In the Admin interface, in WEM Admin, your site is listed on the Sites page.
2. Navigate to your site and click **Manage Site Apps**.

   With no assigned applications, the site is considered inactive.
3. When you assign the WebCenter Sites Admin application and yourself to the site, the site is listed as an option in the site menu, located in the applications bar, (refresh the display, if necessary).
4. Select your site from the menu to open the login dialog. The site is listed in the menu.

   After you log in, the WebCenter Sites Admin application icon is on the top left and the name of the site is in the drop-down menu on the top right. Subsequent access does not require login.

Enabling Tree Tabs

Though the sites created in the Admin interface are exposed in the Admin interface, you can update the sites outside the Admin interface. You can enable users to perform operations on the different nodes of the Sites trees from other interfaces such as WEM Admin interface and Contributor interface.

To enable the WebCenter Sites Admin interface tree and tabs for a WEM site:

1. Log in to the Admin interface as a general administrator.
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.

   c. Individually select each tab you want to enable. For example, suppose you want to enable the **Site Design** tab:

      i. From the list of tree tabs, click **Site Design**.

      ii. In the Tree Tab form, click **Edit**.

         The Edit form for the **Site Design** tab opens.

      iii. In the **Sites** selection box, select sites on which the tab must be enabled.

      iv. Select roles that are allowed to access the tab.

      v. Select content for the tab.

      vi. Click **Save**.

      vii. To enable additional tabs, click **List all Tree Tabs** at the bottom of the Edit form and repeat the steps above.

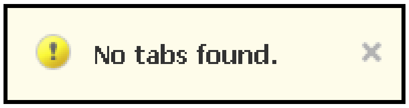
3. When you finish enabling the **WebCenter Sites** tree and tabs on the sites, continue to the next section to make sure the tree and the tabs are rendered properly.

To verify that 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 Admin interface as a user who shares role 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 see the following error message.

   Figure 11-4 Error Message

   No tabs found.

   Verify the site and role assignments for the user and the tab.

Managing the WEM Framework - Quick Tips and Roadmaps

To manage the WEM Framework that provides the technology to develop and implement applications for WebCenter Sites, you must follow few tips and instructions.

Before following instructions in the rest of this section, take note of a few tips:

- (Optional) If you are experimenting with sites, users, and roles you may want to distinguish them from dedicated CM sites, users, and roles. For example, you can add a description for the site/user/role, or add a prefix, such as "", to the name. Note that after you create a site, user, or role, you cannot change its name.
- ACLs are required for user accounts. Only general administrators can create ACLs and only in the **General Admin** tree under the **Admin** node.

- When you assign a role to a user and an application, you grant the user access to the application in that site.
- Applications can have role-protected interface functions. When you assign a role to a user and an interface function, you grant the user access to the interface function.
- Groups provide access to REST resources. You must configure groups in the Admin interface.
- Groups are used to control access to application 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.

Common WEM Tasks for WebCenter Sites Administrators

Use the WEM Admin interface to create and manage sites, applications, users, and roles.

- Managing Sites in the WEM Admin Interface
- Managing Applications
- Managing Users in the WEM Admin Interface
- Managing Roles From the WEM Admin Interface
- Managing Profiles From the WEM Admin Interface

Managing Sites in the WEM Admin Interface

Only general administrators can create, edit, and delete sites. The Sites form is accessible to general and site administrators.

To manage sites in the WEM Admin interface, see the following topics:

- Creating a Site in the WEM Admin interface
- Editing a Site in the WEM Admin interface
- Deleting a Site in the WEM Admin Interface
- Adding an Application to a Site From the Sites List in the WEM Admin Interface
- Assigning Users to a Site From the Sites List in the WEM Admin Interface
- Reassigning Roles to a User from the Sites List in the WEM Admin Interface
- Reassigning Roles to an Application from the Sites List in the WEM Admin Interface
- Removing a User From a Site in the WEM Admin Interface Sites List
- Removing an Application From a Site in the WEM Admin Interface Sites List

Creating a Site in the WEM Admin interface

1. In the WEM Admin interface, select the **Sites** option in the menu bar.

2. In the Sites list, click **Add Site**.
3. In the Add Site form, enter a name and description for the site.
4. Click **Save and Close**.

   The name of the site is listed in the Sites list.

Editing a Site in the WEM Admin interface

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, point to the site you want to edit and click the **Edit** option.
3. In the Edit Site form, change the name or description of the site.
4. Click **Save and Close**.

   The edited site is listed in the Sites list.

Deleting a Site in the WEM Admin Interface

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, hold the cursor over the site you want to delete, and then click **Delete**.
3. In the Warning dialog, click **Delete**.

   The site is removed from the Sites list in the WEM Admin interface, the **Admin** tree in the Admin interface, and from the WebCenter Sites system.

Adding an Application to a Site From the Sites List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, hold the cursor over the site you want to add an application to, select **Manage Site Apps**, and then **Assign Apps**.
3. In the Select Apps for Site form, select the applications you want to add to the site and move them to the **Selected** list.
4. Click **Continue**.
5. In the Assign Roles to App form, select the roles you want to assign to the applications and move them to the **Selected** list.
6. Click **Save and Close**.

> **Note:**
>
> For general admins who want an alternative way to add an application to a site, see Managing Applications.

Assigning Users to a Site From the Sites List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, hold the cursor over the site you want to assign users to, select **Manage Site Users**, and then select **Assign Users**.

3. In the Select Users for Site form, select users to assign to the site and move them to the **Selected** list.
4. Click **Continue**.
5. In the Assign Roles to Users form, select the roles you want to assign to the users and move them to the **Selected** list.
6. Click **Save and Close**.

> **Note:**
>
> For general admins who want an alternative way to assign users to a site, see Managing Users in the WEM Admin Interface.

Reassigning Roles to a User from the Sites List in the WEM Admin Interface

Caution: Reassigning user roles on a site in WEM can uncouple the user from certain applications on that site.

To reassign roles to a user:

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, hold the cursor over the site whose user you want to modify, and select **Manage Site Users**.
3. In the Manage Site Users form, hold the cursor over the user whose site roles you want to modify, and then select **Assign Roles to User**.
4. In the Assign Roles to User form, assign new roles to the site user, or unassign roles, as necessary.
5. Click **Save and Close**.

Note: For general admins who want an alternative way to modify user roles in a site, see Reassigning Roles to a User from the Users List in the WEM Admin Interface.

Reassigning Roles to an Application from the Sites List in the WEM Admin Interface

Caution: Reassigning application roles on a site in WEM can uncouple the application from certain users on that site. To reassign roles to an application:

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the Sites list, hold the cursor over the site whose application you want to modify, and then select **Manage Site Apps**.
3. In the Manage Site Apps form, hold the cursor over the application whose site roles you want to modify, and then select **Assign Roles to App**.
4. In the Assign Roles to App form, assign new roles to the application, or unassign roles, as necessary.
5. Click **Save and Close**.

Note: For general admins who want an alternative way to modify application roles in a site, see Managing Applications.

Removing a User From a Site in the WEM Admin Interface Sites List

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the **Sites** list, hold the cursor over the site you want to remove the user from, and then select **Manage Site Users**.
3. In the Manage Site Users form, hold the cursor over the user you want to remove from the site, and then click **Remove**.
4. In the Warning dialog, click **Remove**.

Note: For general admins who want an alternative way to remove a user from a specific site, see Managing Users in the WEM Admin Interface.

Removing an Application From a Site in the WEM Admin Interface Sites List

1. In the WEM Admin interface, select the **Sites** option in the menu bar.
2. In the **Sites** list, hold the cursor over the site you want to remove an application from, and then select **Manage Site Apps**.
3. In the Manage Site Apps form, hold the cursor over the application you want to remove from the site, and then click **Remove**.
4. In the Warning dialog, click **Remove**.

Note: For general admins who want an alternative way to remove an application from a site, see Managing Applications.

Managing Applications

Only general administrators can modify an application. The Apps form is accessible to general administrators only.

The following topics provide information about managing applications from the WEM Admin interface:

- Modifying an Application
- Assigning an Application to a Site in the WEM Admin Interface
- Reassigning Roles to an Application from the Apps List in the WEM Admin Interface
- Removing an Application From a Site in the WEM Admin Interface Apps List

Modifying an Application

1. In the WEM Admin interface, select the **Apps** option in the menu bar.
2. In the **Apps** list, hold the cursor over the application you want to modify, and then select **Edit**.
3. In the Edit form, update the following fields:
   - **Name**: Cannot be modified.
   - **Tooltip**: The name that opens when you hold the cursor over an application icon.
   - **Description**: Short explanation describing the application.

4. Click **Save and Close**.

Assigning an Application to a Site in the WEM Admin Interface

1. In the WEM Admin interface, select the **Apps** option in the menu bar.
2. In the **Apps** list, hold the cursor over the application you want to assign to a site, select **Manage App**, and then select **Assign to Sites**.
3. In the Select Sites for App form, select the sites to which you want to assign the application and move them to the **Selected** list.
4. Click **Continue**.
5. In the Assign Roles to App form, select the roles you want to assign to the application and move them to the **Selected** list.
6. Click **Save and Close**.

> **Note:**
>
> For an alternative way to add an application to a site, see Managing Sites in the WEM Admin Interface.

Reassigning Roles to an Application from the Apps List in the WEM Admin Interface

Caution: Reassigning application roles on a site in WEM can uncouple the application from certain users on that site.

To reassign roles to an application:

1. In the WEM Admin interface, select the **Apps** option in the menu bar.
2. In the **Apps** list, hold the cursor over the application whose roles you want to modify for a select site, and then select **Manage App**.
3. In the Manage App form, hold the cursor over the site in which you want to modify the application roles, and then select **Assign Roles to App**.
4. In the Assign Roles to App form, add roles to an application, or remove roles, as necessary.
5. Click **Save and Close**.

Note: For an alternative way to modify application roles in a site, see Reassigning Roles to an Application from the Sites List in the WEM Admin Interface.

Removing an Application From a Site in the WEM Admin Interface Apps List

1. In the WEM Admin interface, select the **Apps** option in the menu bar.
2. In the Apps list, hold the cursor over the application you want to remove from the site, and then select **Manage App**.
3. In the Manage App form, hold the cursor over the site you want to remove the application from, and then click **Remove**.
4. In the Warning dialog, click **Remove**.

Note: For an alternative way to remove an application from a site, see Managing Sites in the WEM Admin Interface.

Managing Users in the WEM Admin Interface

Only general administrators can create, edit, and delete users. The Users form is accessible to general administrators only.

The following topics provide information about managing users from the WEM Admin interface:

- Creating a User in the WEM Admin Interface From the Users List
- Enabling a User From the Users List in the WEM Admin Interface
- Assigning a User to a Site From the Users List in the WEM Admin Interface
- Editing a User From the Users List in the WEM Admin Interface
- Reassigning Roles to a User from the Users List in the WEM Admin Interface
- Assigning a User to a Group from the WEM Admin Interface
- Removing a User From a Site in the WEM Admin Interface Users List
- Deleting a WebCenter Sites User from the Users List in the WEM Admin Interface

Creating a User in the WEM Admin Interface From the Users List

1. In the Admin interface, select the **Users** option in the menu bar, then click **Add User**.
2. In the Add User form, update the following fields:

   ACLs: ACLs determine user access permissions to the database.

   Note: You cannot create ACLs in the Admin interface.

   Groups: Groups determine user access permissions to REST application resources. Two default groups are configured: `RestAdmin` (a general administrator group) and `SiteAdmin_AdminSite` (a site administrator group).
3. Click **Save and Close**.

Note: When you create a user, the user is able to log in. To access sites and applications, you must enable the user.

Enabling a User From the Users List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the Users list, hold the cursor over the user you want to enable, then select **Manage User**, and then **Assign to Sites**.
3. 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.

4. Click **Continue**.
5. 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`.
6. Click **Save and Close**.

Assigning a User to a Site From the Users List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user you want to assign to the site, then select **Manage User**, and then **Assign to Sites**.
3. In the Select Sites for User form, select the sites to assign to the user and move them to the **Selected** list.
4. Click **Continue**.
5. In the Assign Roles to User form, select the roles you want to assign to the user and move them to the **Selected** list.
6. Click **Save and Close**.

> **Note:**
>
> For an alternative way to add a user to a site, see Managing Users in the WEM Admin Interface.

Editing a User From the Users List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user you want to modify, and then click **Edit**.
3. In the Edit User form, update the fields, and then click **Save and Close**.

   Note: You cannot modify the user name (shown in the **Name** field) after the user is saved.

Reassigning Roles to a User from the Users List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user whose roles you want to modify, and then select **Manage User**.
3. In the Manage User form, hold the cursor over the site in which you want to modify user roles, then click **Assign Roles to User**.
4. In the Assign Roles to User form add roles to a user, or remove roles, as necessary.

5. Click **Save and Close**.

Assigning a User to a Group from the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user you want to assign to a group, and then select **Edit**
3. In the **Groups** field, select the appropriate groups and move them to the **Selected** list.
4. Click **Save and Close**.

Note: Groups provide access to REST. You use groups to control access to application resources.

Removing a User From a Site in the WEM Admin Interface Users List

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user you want to remove from a site, and then select **Manage User**.
3. In the Manage User form, hold the cursor over the site you want to remove the user from, and then click **Remove**.
4. In the Warning dialog, click **Remove**.

> **Note:**
>
> For an alternative way to remove a user from a site, see Removing a User From a Site in the WEM Admin Interface Sites List.

Deleting a WebCenter Sites User from the Users List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user you want to remove from the system, and then click **Delete**.
3. In the Warning dialog, click **Delete**.

Managing Roles From the WEM Admin Interface

Only general administrators can create, edit, and delete roles. The Roles form is accessible to general administrators only.

The following topics provide information about managing roles from the WEM Admin interface:

- Adding a Role to the WebCenter Sites System From the WEM Admin Interface
- Editing a Role From the Roles List in the WEM Admin Interface
- Deleting a Role from the Roles List in the WEM Admin Interface

Adding a Role to the WebCenter Sites System From the WEM Admin Interface

1. In the WEM Admin interface, select the **Roles** option in the menu bar.
2. In the **Roles** list, click **Add User Role**.
3. In the Add Role form, enter a name and description for the role, and then click **Save and Close**.

Editing a Role From the Roles List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Roles** option in the menu bar.
2. In the **Roles** list, hold the cursor over the role you want to modify, and then click **Edit**.
3. In the Edit Role form, update the description field, and then click **Save and Close**.

   Note: You cannot modify a role name after the role is saved.

Deleting a Role from the Roles List in the WEM Admin Interface

1. In the WEM Admin interface, select the **Roles** option in the menu bar.
2. In the **Roles** list, hold the cursor over the role you want to remove, and then click **Delete**.
3. In the Warning dialog, click **Delete**.

If the role is assigned to users, applications or both, you cannot delete it until you unassign the role from the user and applications. Review the table of dependencies in the Role Dependencies form. **Delete all Dependencies** removes the role from its users and applications and deletes the role from the system.

Managing Profiles From the WEM Admin Interface

General administrators can modify user profiles.

The following topics provide information about modifying user profiles:

- Modifying a User Profile
- Modify Your Own Profile

Modifying a User Profile

1. In the WEM Admin interface, select the **Users** option in the menu bar.
2. In the **Users** list, hold the cursor over the user name whose profile you want to modify, and then click **Edit**.
3. In the Edit User form, update the desired fields, and then click **Save and Close**.

> **Note:**
>
> You cannot modify the value in the **Name** field.

Modify Your Own Profile

1. In the Admin interface, expand the applications bar, and then click your user name.
2. In the Edit Profile form, update the fields, and then click **Save and Close**.

> **Note:**
>
> You cannot modify the value in the **Name** field.

Part IV

Managing Users and Security

You can create users through the WebCenter Sites native user manager (in the Admin interface) or through LDAP. However, to configure user profiles and attributes, you must use the Admin interface. After creating the users, configure external security protocols on both the management and delivery systems to manage user access to WebCenter Sites resources.

Topics:

- Configuring Users, Profiles, and Attributes
- Setting Up External Security

12
Configuring Users, Profiles, and Attributes

You can create users in WebCenter Sites, as well as configure their profiles and user attributes.
For more information about creating users in external sources, see the product documentation:

- About Creating WebCenter Sites Users
- Understanding User Management Options
- Configuring Users in WebCenter Sites
- Working with User Accounts
- Working with User Profiles and User Attributes

About Creating WebCenter Sites Users

You can create users in WebCenter Sites 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 are working with:
  - WebCenter Sites applications
  - Language packs and setting a default language
  - Workflow processes, in which email messages are 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.

After users are created and configured, they must be associated, by means of roles, with the sites they are to work in. (Procedures for associating users with sites are given in Assembling Content Management Sites.)

Understanding User Management Options

To connect to external directory servers or user managers that contain authentication information, user information, and so on; use WebCenter Sites Directory Services API to enable your WebCenter Sites system.

The following connection options are available:

- **Central Authentication Service (CAS)** —the WebCenter Sites native user manager, which uses the native WebCenter Sites user management tables SystemUsers and SystemUserAttrs.

- **Oracle Access Management**—WebCenter Sites integrates with Oracle Access Management to provide a seamless single-sign on experience. For maximum security, Oracle highly recommends using this authentication method on production environments.
- **LDAP directory**—with this option, user names and attributes are stored in an LDAP directory server rather than in the WebCenter Sites database. For maximum security, Oracle highly recommends using this authentication method as an alternative to Oracle Access Management integration on production environments.

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

Information about switching to an external authentication solution is given in Switching to External Authentication in *Installing and Configuring Oracle WebCenter Sites*. Properties that configure the plug-ins are located in the `wcs_properties.json` and `ldap.ini` files. The files are described in Properties in the User Category in *Property Files Reference for Oracle WebCenter Sites*.

> **Note:**
>
> This guide uses the native (CAS) WebCenter Sites user manager throughout.

The following topics provide information about your user management options:

- Understanding Native WebCenter Sites User Manager
- Understanding the LDAP Plug-In

Understanding Native WebCenter Sites User Manager

If you are using the native WebCenter Sites user manager, follow the guidelines in Roadmap for Administering WebCenter Sites to create and configure users, and then grant them access to the management system.

Understanding the 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.

Configuring Users in WebCenter Sites

You can configure users in WebCenter Sites and thereby edit and maintain a user account, user profile, and, if necessary, user attributes.

Familiarize yourself with the concepts of user account, user profile, and, if necessary, user attributes as follows:

- A user account is required for anyone who is to work with WebCenter Sites.

- A user profile is required for users who work with WebCenter Sites modules and products, setting a default language, and participating in workflow processes in which email messages are sent.
- User attributes, in addition to the locale and email attributes in the user profile, could be required for your operation. If so, the additional attributes can be created.

When 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. See 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 system running WebCenter Sites. Depending on how the system was set up, you might also have to include the port number— `server:8080` for example; and

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

See the following topics for instructions on creating modifying and deleting user accounts, profiles, and attributes:

- Working with User Accounts
- Working with User Profiles and User Attributes

Working with User Accounts

To control and manage user accounts, you can create, edit, and delete users in the WebCenter Sites administrator's interface.

The following topics provide procedures for creating, editing, and deleting users in the WebCenter Sites administrator's interface.

> **Note:**
>
> If you are using LDAP, see the LDAP product documentation. If you are using WEM Admin, see Managing Users in the WEM Admin Interface.
>
> 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.

- Creating a User in the Admin Interface
- Creating Common User Types
- Editing a User From the Admin Interface
- Deleting a User from the System Using the Admin Interface

Creating a User in the Admin Interface

The following task provides information and instructions about creating a user in the Admin interface.

> **Note:**
>
> If you are using LDAP, see the LDAP product documentation.

Before creating a user, determine the user's:

- User name.
- Password.
- ACLs, which regulate the user's access to WebCenter Sites database tables.
  - To determine the user's required ACLs, see Required ACLs for Custom Users.
  - To determine the user's additional system ACLs, see System ACLs and Their Permissions.
  - To determine which ACLs are assigned to the database tables the user must access, follow the steps in Assigning ACLs to Custom Tables.

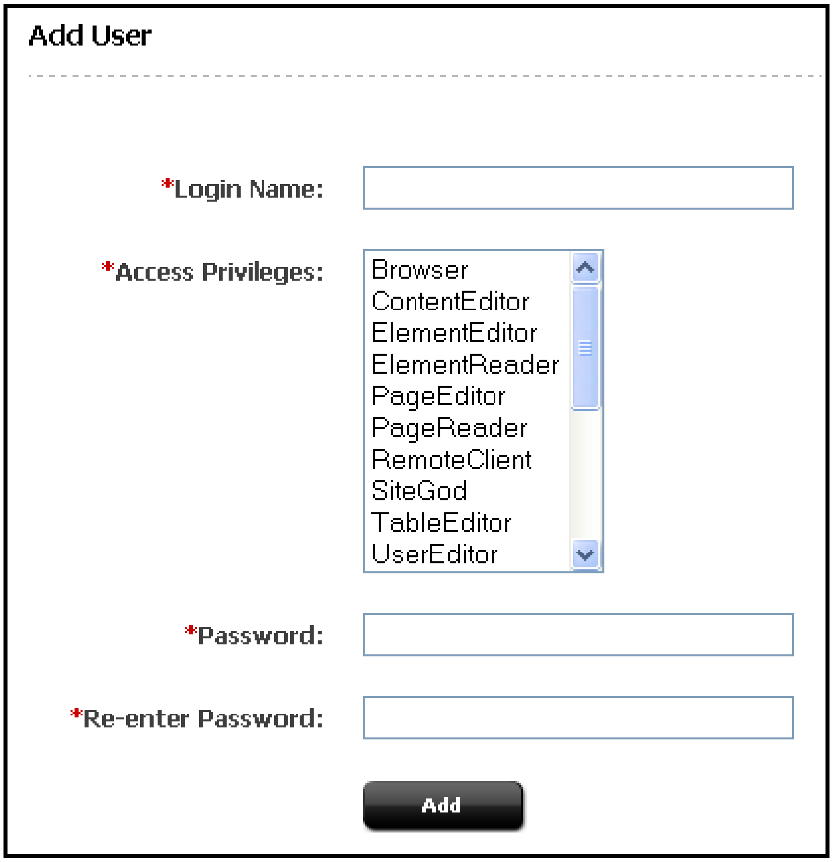
To Create a User in the Admin Interface

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User**.
2. In the User form, 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.

   The Add User form opens.

Figure 12-1 Add User Form



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

   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 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 is created.

You can create a general administrator. For more information about making the user an administrator or a content contributor, see Creating Common User Types.

To Follow Up With Post-Creation Procedures

1. If the user implements any of the following options, create a profile for the user:

   - Oracle WebCenter Sites products such as Oracle WebCenter Sites: Engage.
   - Language packs and different languages.
   - Workflow processes that send email messages.

   See 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 Modifying, Adding, and Deleting User Attributes.
3. After the user has been completely defined, you must associate the user with a site by means of roles.
   a. If you have not done so, create roles for the user, following instructions in Creating a Role From the Admin Interface.
   b. To associate the user to the site, create the site and add the user to the site. For instructions, see Creating a Site From the Admin Interface and Granting Users Access to a Site (Assigning Roles to Users).

Creating Common User Types

When creating a user, it is important to consider the roles that the user will have on separate websites. In many cases, a user could be an administrator on one site and a content contributor on another. The following topics detail how to make each user a specific type of user.

- Making the User a General Administrator
- Making the User a Site Administrator
- Making the User a Workflow Administrator
- Making the User a Content Contributor

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, ensure that it has the correct ACLs assigned to it.

   To determine the ACLs for a user, see 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. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Assign Users to Groups** node, and then double-click **Add New**.

   The Assign Groups to User form opens.
3. Select the user to make a general administrator, and then select **RestAdmin** in the **Groups** field.

   You can use **Ctrl-Click** to select multiple users or groups, 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. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then select the site to add the user to.
5. Click **Users**.

   The User Role Management form opens.

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

6. Enter the login information for the user, and then click the **Edit** icon.

   The Edit Roles for User form opens.

7. Select **GeneralAdmin** and any other roles you wish the user to have on this website.

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

   When adding a general administrator to a site, consider if you will want the same person to have different roles on a different site. You will have 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 website (or websites) where they have the general administrator role.

Making the User a Site Administrator

1. For the newly created user, ensure that it has the correct ACLs assigned to it.

   To determine the ACLs for a user, see 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. In the **General Admin** tree, expand **Admin**, then **Sites**, and then expand the site to add the user to.

3. Double-click **Users**.

   The User Role Management form opens.

   This form is also available under the **Site** tree, by expanding **Site Admin**, and then double-clicking **Users** under the site you are logged in to.

4. In the **Username** field, enter the name of the user, and then click **Select**.

5. In the User Role Management form, click the **Edit** icon next to the user’s name.

   The Edit Roles for User form opens.

6. In the **Roles** field, select **SiteAdmin** and **Advanced User** and any other roles you wish the user to have on this website. The SiteAdmin role grants the user access to the **Site Admin** tab, the AdvancedUser role grants access to the Admin interface.

7. Click **Save**.

8. In the **General Admin** tree, expand **Admin**, then **User Access Management**, then **REST Security**, and then **Assign Users to Groups**.

9. Click **Add New**.
10. Select the user to make a site administrator, and then select **SiteAdmin_AdminSite** in the **Groups** field.

    You can use **Ctrl-Click** to select multiple users or groups, 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.
11. Click **Save**.

> **Note:**
>
> The names of roles used in this documentation are the names of the roles used in the avisports sample site that comes installed. You can choose to use different roles or role names with your published website.

When adding a site administrator to a site, consider if you will want the same person to have different roles on a different site. You will have 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** tree will only appear when they are logged into the website (or websites) where they have the site administrator role.

Making the User a Workflow Administrator

1. For the newly created user, ensure that it has the correct ACLs assigned to it.

   To determine the ACLs for a user, see 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. In the **General Admin** tree, expand the **Admin** node, and then expand the site to add the user to.
3. Double-click **Users**.

   The User Role Management form opens.

   This form is also available under the **Site** tree by expanding **Site Admin** node, and then double-clicking **Users** node if you are logged in to the specific site.
4. In the **Username** field, enter the name of the user, then click **Select**.
5. In the User Role Management form, click the **Edit** icon next to the user’s name.

   The Edit Roles for Users form opens.
6. In the **Roles** field, select **WorkflowAdmin** and **AdvancedUser**, and any other roles you wish the user to have on this website. The `WorkflowAdmin` role grants the user access to the **Workflow** node (under the **Admin** node in the **General Admin** tree), and 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 website that comes installed. You can choose to use different roles or role names with your published website.

When adding a workflow administrator to a site, consider if you will want the same person to have different roles on a different site. You will have 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 website (or websites) where they have the workflow role.

7. Click **Save**.

Making the User a Content Contributor

For the newly created user, ensure it has the correct ACLs assigned to it.

To determine the ACLs for a user, see Required ACLs for Custom Users. Looking at the table, you can see that the ACLs for all users are `Browser`, `ElementReader`, `PageReader`, `UserReader`, and `xceleditor`. You may want to add additional ACLs required 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 about the specific ACLs and how they impact the contributor's editing access, see System Defaults.

The user does not require 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 **General Admin** tree, then expand the **Site** node, and then expand the site to add the user.
2. Click **Users**.

   The User Role Management form opens.

   This form is also available under **Site** tree by expanding **Site Admin** and then double clicking **Users** under the name of the site you are logged in to.
3. In the **Username** field, enter the name of the user, and then click **Select**.
4. In the User Role Management form, click the **Edit** icon next to the user’s name.
5. In the **Roles** field, select the **SitesUser** 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 and the AdvancedUser role for the same website. The SitesUser role grants access to the Oracle WebCenter Sites: 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 website. You can choose to use different roles or role names with your published website.

Most content contributors do not require roles on the website 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.

6. Click **Save**.

Editing a User From the Admin Interface

The following steps show you how to edit users in the WebCenter Sites Admin interface. If you are using LDAP, see 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User**.
2. In the form that opens, 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 changes, then click **Modify**.

Deleting a User from the System Using the Admin Interface

The following steps shows you how to delete users from the WebCenter Sites system using the WebCenter Sites Admin interface. If you are using LDAP, see the LDAP product documentation.

Note:

Before deleting a user(internal, that is stored in Sites database or external, that is LDAPs), ensure that all the resources belonging to the user have been released. For example, if there are any assets checked-out by the user, they should be released. Also, if the user is part of any workflow step, he should be removed from workflow.

Caution:

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

1. Delete the user profile, as shown in Deleting a User Profile in the Admin Interface.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User**.
3. In the form that opens, 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**.

   A warning message opens.
7. Click **OK**.

   The user is deleted from the WebCenter Sites system.

Working with User Profiles and User Attributes

A user profile defines a user and its access and roles and user attributes make up a user profile.

A user profile is required for any user who works with the following:

- Sites modules and products.
- Language packs.
- Workflow processes in which email messages are 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. However, to use these values in the WebCenter Sites interfaces requires you to customize the elements that display the user profile forms. See Customizing Oracle WebCenter Sites in *Developing with Oracle WebCenter Sites.*

This section covers the following topics:

- Creating and Editing a User Profile
- Deleting a User Profile in the Admin Interface
- Modifying, Adding, and Deleting User Attributes

Creating and Editing a User Profile

If you are using LDAP, be aware of system responses to user and site management operations. For more information about these system responses, see Users, Sites, and Roles in LDAP-Integrated Sites Systems.

To create or edit a user profile:

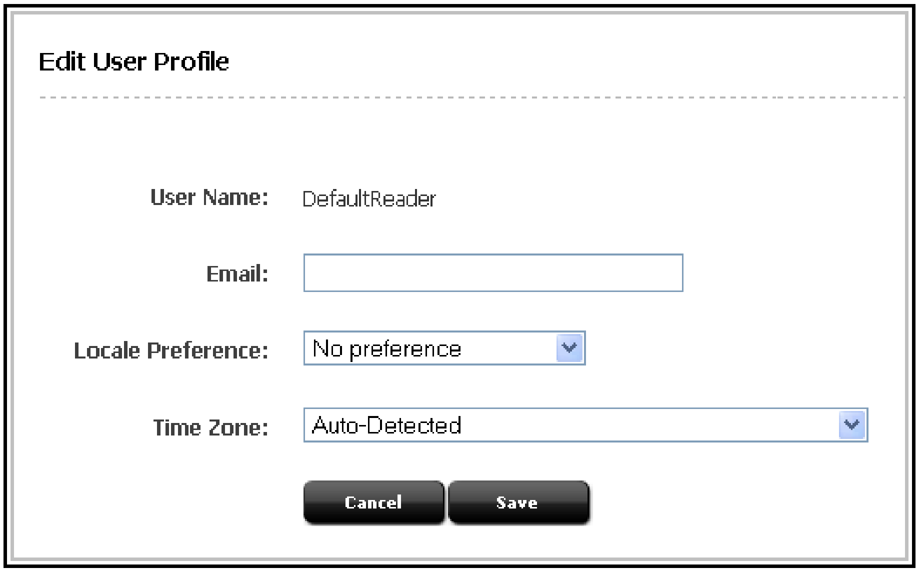
1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User Profiles**.

   The User Profile Management form opens.
2. In the form, enter the user name and click **Select**.

   The profile of the selected user opens.
3. Click the **Edit** icon.

   The Edit User Profile form opens.

Figure 12-2 Edit User Profile Form



Edit User Profile

User Name: DefaultReader

Email:

Locale Preference: No preference

Time Zone: Auto-Detected

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 display in the language set for the WebCenter Sites system.

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 must to work with. For instructions, see Granting Users Access to a Site (Assigning Roles to Users).

Deleting a User Profile in the Admin Interface

To delete a user profile:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User Profiles**.

   The User Profile Management form opens.
2. Click **Delete**.

   A warning message opens.
3. Click **Delete User Profile**.

   A message confirming the deletion opens.

   > **Note:**
   >
   > Before deleting a user(internal, that is stored in Sites database or external, that is LDAPs), ensure that all the resources belonging to the user have been released. For example, if there are any assets checked-out by the user, they should be released. Also, if the user is part of any workflow step, he should be removed from workflow.

Modifying, Adding, and Deleting User Attributes

By default, the only user attributes that the WebCenter Sites content applications require 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 Creating and Editing a User Profile. If you have to, you can store and use additional user attributes for your users in this table, even if you are using LDAP.

> **Note:**
>
> If LDAP is integrated with WebCenter Sites, user attributes such as locale, timezone, and email are stored in LDAP; and these attributes are not editable in WebCenter Sites.

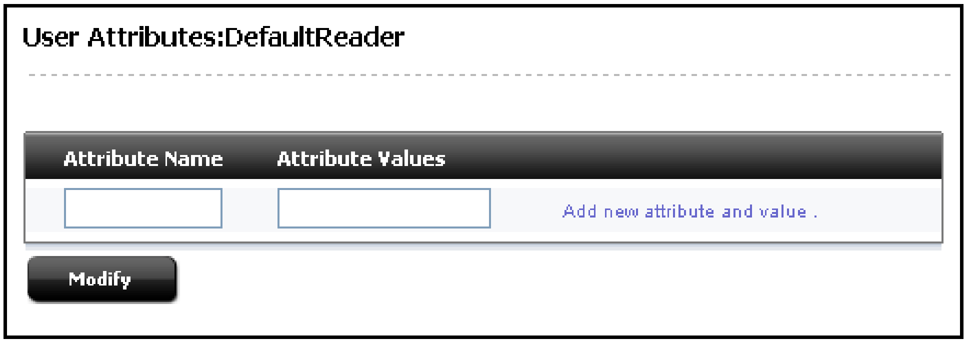
The Modify User Attributes option allows you to modify the attributes that are used in the user profile. It also allows you to add and delete attributes.

To modify a user's attributes:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User**.
2. In the form that opens, 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. Click the user name in the list that whose attributes you want to modify.

   The User Attributes form opens.

   Figure 12-3 User Attributes Form

   User Attributes:DefaultReader

Attribute Name	Attribute Values	
		Add new attribute and value .

4. In the fields of the **User Attributes** form, complete at least one 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).
5. Click **Modify**.

   Your changes are committed to the database.

13 Setting Up External Security

Security standards help organizations to prevent compromise of important information. WebCenter Sites provides you with robust security features to protect your data and attributes through security protocols, encryption of assets, and REST security.

The following topics describe implementing security protocols.

- Setting Up External Security
- Implementing Cryptography and Encryption Key Security
- Using REST Security

Setting Up External Security

Before the developers start coding templates and designing asset types, , you must decide with them on how to configure security on both the management and the delivery systems.

The following topics provide information about implementing and testing security on the WebCenter Sites systems:

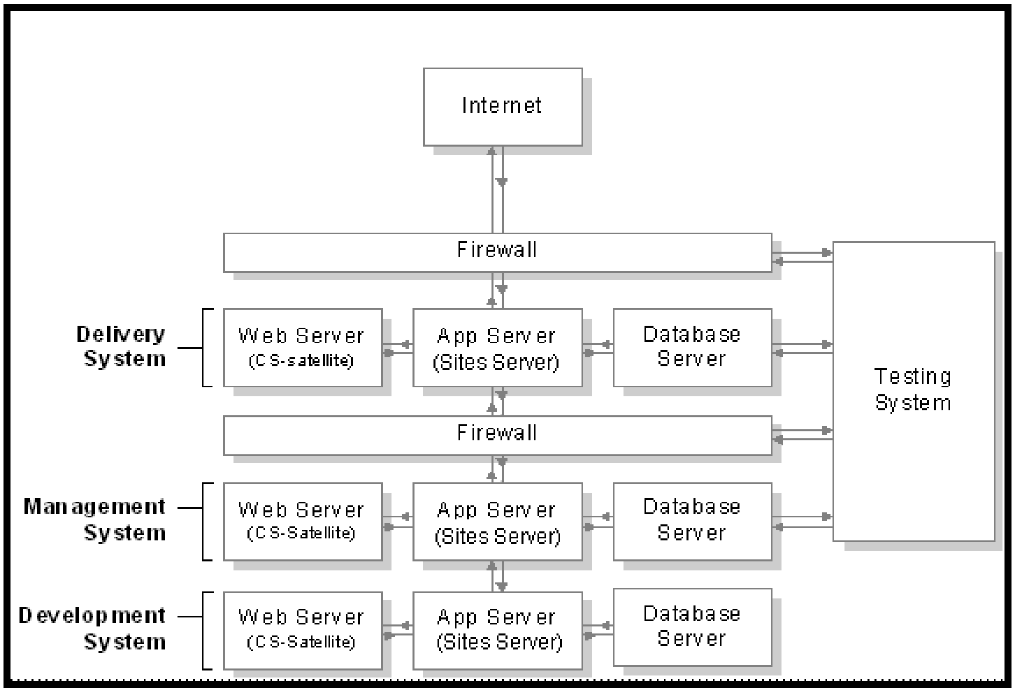
- Introduction to WebCenter Sites Security
- Implementing Security
- Testing Security

Introduction to WebCenter Sites Security

Before developers design the online site or make changes to the user interface on the management system, you must implement your security protocols. The decisions you make about security configuration affects the way that you develop your online site.

At regular intervals, you should review your systems to determine whether they are working properly. The following figure shows an example of a secure WebCenter Sites system.

Figure 13-1 WebCenter Sites System


Internet
Firewall
Delivery System
Web Server (CS-satellite)
App Server (Sites Server)
Database Server
Firewall
Management System
Web Server (CS-Satellite)
App Server (Sites Server)
Database Server
Development System
Web Server (CS-Satellite)
App Server (Sites Server)
Database Server
Testing System

The following topics provide overview information about the WebCenter Sites security measures available to you:

- Understanding ACLs and Security
- Understanding DefaultReader, secure.CatalogManager, and secure.TreeManager
- Understanding BlobServer and Security
- Implementing Security Goals

Understanding ACLs and Security

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 `wcs_properties.json` file are set to `true`. These properties are set to `true` by default. If you have to disable security for any reason, open the Property Management Tool from the Admin interface 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 System Defaults), and determine whether you require additional ACLs. You may have to create an ACL with a different combination of permissions than any of the system ACLs if your developers plan to create tables or 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 WebCenter Sites content application table.

Understanding DefaultReader, secure.CatalogManager, and secure.TreeManager

WebCenter Sites and the WebCenter Sites content applications are delivered with several default user accounts. The DefaultReader 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 Oracle WebCenter Sites: 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, use the Property Management Tool in the Admin interface to set the following properties in the `wcs_properties.json` 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:

For more information about permissions that are associated with each ACL, see Table 39-3. For more information about the descriptions of the ACLs, see Table 39-4.

Understanding 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 using the Property Management Tool in the Admin interface to set the `bs.security` parameter in the `wcs_properties.json` 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.

To protect the blob header names from being vulnerable, provide valid blob header names in the bs.validheadernames parameter in the wcs_properties.json file. Once this property value is set, it doesn't allow the user or hacker to inject any more blob header names through URL parameters. If this property's value is left empty, it allows all values for blob header names through URL parameters and might lead to HTTP header injection vulnerability.

For web pages that are stateless, that is without any session id, use Vanity URLs. Vanity URLs make it possible for images to have friendly links without query parameters. See Configuring Vanity URLs.

Implementing Security Goals

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

See the following topics for information about implementing security on each WebCenter Sites system:

- Implementing Security Goals for the Development System
- Implementing Security Goals for the Management System
- Implementing Security Goals for the Delivery System

Implementing Security Goals for the Development System

Even though development systems are commonly behind firewalls, you should implement the same security configuration on the development system that is in place on the system the developers are designing for (management or delivery). This ensures 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.

Implementing Security Goals for the Management System

Security on the management system encompasses two main concepts:

- Ensuring that only valid users can access the system (described in this chapter).
- Ensuring that those valid users can access only the functions that are appropriate for them.

Implementing 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 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 servlets.

Implementing Security

The following topics describe configuring security measures: 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 topic states which steps should be performed on which system or systems.

- Using Properties to Configure Security Settings
- Working with Users and Passwords
- Working with SSL and Digital Certification
- Working with URLs and the Web Server (Delivery System Only)
- Working with WebCenter Sites Forms and Pages (Delivery System Only)

Using Properties to Configure Security Settings

The properties that implement various kinds of security are located in the WebCenter Sites `wcs_properties.json` file.

Use the Property Management Tool to verify that the following properties are set to `true`:

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

If you plan to use BlobServer security, set `bs.security` to `true`.

To ensure that the session timeout value is appropriate for each system, set the `cs.timeout` property as follows:

- On the development and management systems, set the timeout value for if 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. Also, ensure the following:
  - Remove the entire `futuretense_cs` directory from the web server. Do not remove it from the application server. If you do, use the Property Management Tool to set the `cs.wrapper` property in the `wcs_properties.json` 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 system error messages.
  - Go to `SiteCatalog/OpenMarket/Xcelerate/UIFramework` and rename all of the pages. At the very least, rename `LoginPage` and `LoginPost`.

In addition, 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`.

Working with Users and Passwords

Be sure that the default user accounts were made secure after WebCenter Sites was installed on all systems. Oracle recommends using OAM and LDAP for strong user accounts and password policies.

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

- Change the default password for the `fwadmin` user account (in the **General Admin** tree in the Admin interface, expand the **Admin** node, expand the **User Access Management** node, and then double-click **Users**).
- Change the default administrator user name/password for your application server.
- Change the default administrator user name/password for your web server.
- If you have the sample sites installed, a mirror user was created for the Mirror to Server publishing method (username: `mirroruser`). Change the password for this user.
- For any user accounts that have the SiteGod ACL, change the password frequently (in the **General Admin** tree in the Admin interface, expand the **Admin** node, expand the **User Access Management** node, and then double-click **User**). 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.

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.

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.

Working with 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.

Working with URLs and the Web Server (Delivery System Only)

On the web server of the delivery system, be sure to give access to the following servlets only. Map only their URLs to the application server:

- ContentServer
- BlobServer
- Satellite
- CookieServer

Do not map URLs to the application server for any of the other 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

Working with WebCenter Sites Forms and Pages (Delivery System Only)

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

- Remove the entire `futuretense_cs` directory from the web server. Do not remove it from the application server. If you do, use the Property Management Tool to set the `cs.wrapper` property in the `wcs_properties.json` 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 system error messages.
- Go to `SiteCatalog/OpenMarket/Xcelerate/UIFramework` and rename all of the pages. At the very least, rename `LoginPage` and `LoginPost`.

Testing Security

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

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

1. Ensure that you modify the passwords for the user accounts `DefaultReader`, `ContentServer`, `editor`, and `fwadmin`.
2. After you log in using the `DefaultReader` user account, set the properties `secure.CatalogManager` and `secure.TreeManager` to `true`.
3. Verify that the sample site users do not exist on the management or delivery systems.

4. Verify that you cannot log in with `ContentServer/password` using a CatalogManager URL:

   ```
   http://<server>:<port>/<context>/CatalogManager?
   ftcmd=login&username=ContentServer&password=<password>
   ```

5. 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
   ```

6. Verify that you cannot log in to the application server as the default administrator user.
7. Verify that you cannot log in to the database as the default administrator user.
8. Verify that you cannot log in to the web server as the default administrator user.

Implementing Cryptography and Encryption Key Security

With WebCenter Sites, you can use secure server access to protect your system from security attacks. All access to assets is protected through cryptographic encryption.

- WebCenter Sites Security Overview
- About Security Configurations in WebCenter Sites
- About the SitesSecurityContext.xml Configuration File
- Working with WebCenter Sites Security Utilities
- Working with ESAPI Security Validation

WebCenter Sites Security Overview

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.

supports the Oracle Platform Security Services (OPSS) credential store for increased security. This 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. 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.

For components that require cryptographic support across application server contexts or systems or both, the same encryption keys must be available. In other words, a data value encrypted at one location must be decrypted using the same key at the destination location. You must use copies of the central key store at each place where encryption and decryption are necessary.

About Security Configurations in WebCenter Sites

Configuration of the security is controlled by the `SitesSecurityContext.xml` configuration file. This file must reside on the classpath of external programs that require access to OPSS credential store (see OPSS documentation for configuration details). 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. For any key store type, these four associative names are used:

1. **masterkey** – reserved for future use
2. **tokenskey** – the encryption key used by the 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

To enhance security, the set of four encryption keys is unique for every installation of .

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 across all environments. It 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.

About the SitesSecurityContext.xml Configuration File

The `SitesSecurityContext.xml` file controls all aspects of the security environment which 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 server. It is an integral part of security between the various system components.

```
<?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 -->
     <!-- Default Crypto package for WebLogic -->
     <bean id="cryptoPackage"
class="com.fatwire.security.common.OracleSecurityCrypto" />

      <!-- Credential store support package -->
      <!-- Default Credential Store for WebLogic -->
      <bean id="credentialPackage" 
class="com.fatwire.security.common.WeblogicCredentialStore" >
          <property name="securityMetaInfo" ref="securityMetaInfo" />
      </bean>

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

     <bean id="securityMetaInfo" 
class="oracle.fatwire.security.common.SecurityMetaInfo" >
         <property name="uniquePrefix" value="Sites" />
     </bean>

     <!-- Security Basis -->
     <bean id="securityBasis" 
class="com.fatwire.security.common.SecurityBasis" >
            <property name="securityCrypto" ref="cryptoPackage" />
            <property name="securityToken" ref="tokenPackage" />
            <property name="credentialStorage" 
ref="credentialPackage" />
         <property name="securityMetaInfo" ref="securityMetaInfo" />
     </bean>

     <!-- Security Context -->
     <bean id="securityContext" 
class="COM.FutureTense.Security.Context.AdvancedSecurityContextImpl" >
            <property name="securityBasis" ref="securityBasis" />
            <property name="convertCredentials" value="true" />
     </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 server (tokenPackage and securityContent). When modified, the revised file must be placed on the classpath of all components that require it.

The beans in this file are described as:

- `cryptoPackage` – this bean specifies which cryptographic package is initialized.
  - `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.
  - `com.fatwire.security.common.WeblogicCredentialStore` – WebLogic specific CSF implementation.
- `tokenPackage` –this bean is required for the 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.

- securityMetaInfo – This bean is required to specify a prefix for an application that uses the credential store. The prefix is added to keys and passwords so that different applications can use the same credential store. The prefix for WebCenter Sites is Sites.
- 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 server.
- securityContext – this bean is required for 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 classpath 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 for the applications that require access to OPSS credential store (see OPSS documentation for configuration details).

Working with WebCenter Sites Security Utilities

There are several utilities available with the WebCenter Sites system to aide in the creation and maintenance of encryption key storage. These utilities are located in site-security.jar and are executed by calling by the specific class names from the Java command. When running, it is necessary to define the classpath for all the necessary jars required by the utilities.

Any Java SE environment trying to access Weblogic CSF for getting keys needs to be setup for some configuration stated as follows, and the programs that access or manipulate contexts of the credential store must be granted permission.

Create folders called config, fmwconfig and lib. Place the following files in the folders created.

- config: ‘java.policy’ (<weblogic_Home_Path>\oracle_common\modules\oracle.jps\domain_config\jse) 'SitesSecurityContext.xml'(from config of Sites)
- fmwconfig: 'bootstrap', 'keystores.xml' and 'jps-config-jse.xml' (<weblogic_Home_Path>\user_projects\domains\<domain_name>\config\fmwconfig)
- lib: The JARS mentioned below need to be here. You can get the JAR files at WEB-INF/lib under the deployed sites instance or at ${ORACLE_HOME}/wcsites/webcentersites/sites-home/lib.

The following is a sample shell script that runs a utility class when the class name is supplied as the first argument:

```
#!/bin/sh
# Set JAVA_HOME
JAVA_HOME="/fmw/jdk/bin"
# Set CLASS_PATH
weblogicPath=<Path to weblogic>
fmwConfigPath=./fmwconfig
oracleCommonPath=${weblogicPath}/oracle_common
oracleJps=${oracleCommonPath}/modules/oracle.jps

jpsConfigPath=${fmwConfigPath}/jps-config-jse.xml
```

```
javaPolicy=./config/java.policy

jpsManifestJar=${oracleJps}/jps-manifest.jar


javaOpts=-cp .:./config
javaOpts=${javaOpts}:./lib/sites-security.jar

javaOpts=${javaOpts}:${jpsManifestJar}
javaOpts=${javaOpts}:./lib/commons-codec-1.7.jar
javaOpts=${javaOpts}:./lib/commons-configuration-1.9.jar
javaOpts=${javaOpts}:./lib/commons-lang-2.5.jar
javaOpts=${javaOpts}:./lib/commons-logging-1.1.3.jar
javaOpts=${javaOpts}:./lib/log4j-1.2.17.jar
javaOpts=${javaOpts}:./lib/quartz-2.2.1.jar
javaOpts=${javaOpts}:./lib/sites-request-authenticator.jar
javaOpts=${javaOpts}:./lib/sites-cs.jar
javaOpts=${javaOpts}:./lib/spring-core-4.3.0.RELEASE.jar
javaOpts=${javaOpts}:./lib/xercesImpl-2.11.0.jar
javaOpts=${javaOpts}:./lib/xml-apis-1.4.01.jar
javaOpts=${javaOpts}:./lib/javaee-api-7.0.jar


$JAVA_HOME/java ${javaOpts} -Doracle.security.jps.config=$
{jpsConfigPath} -Dcommon.components.home=${oracleCommonPath} -
Djava.security.policy=${javaPolicy} com.fatwire.security.util.$1 $2 $3
```

Description of Java Classes

`com.fatwire.security.util.CreateDefaultCredentialStore`: This utility populates the OPSS store with a randomly generated set of Oracle encryption keys. The replace argument should be set to yes if the utility should replace any WebCenter Sites encryption keys that exist.

Arguments

`product=sites`

`replace=no | yes` (default is no)

`com.fatwire.security.util.ExportFromCSF` extracts the salts for each of the WebCenter Sites keys in the current store and writes the values to the export.keys file in the working directory. The export file is used as input to an import utility program listed below.

Arguments: (optional)

`export.keys`

`com.fatwire.security.util.ImportToCSF.class` takes the extracted salts from the `export.keys` file, creates each of the WebCenter Sites encryption keys using the associated salt, and adds to the output credential store. 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`)

Working with ESAPI Security Validation

The server employs the Enterprise Security API to check for and prevent security vulnerabilities that could occur from injection of malicious web data. The validation expressions contained in the `ESAPI.properties` file can be modified - this file is in the `WEB-INF/classes` folder when you install the server.

Further information about ESAPI is available at `https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API`.

Using REST Security

Through REST security, you can authenticate users to access assets and manage different roles of users with respect to application resources.

The following topics provide information about REST security:

- REST Authorization
- Authorizing Users to Access Application Resources
- REST Security Configuration Reference

REST Authorization

REST authorization grants privileges to perform REST operations on application 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 after the application is deployed and registered with the WEM Framework.

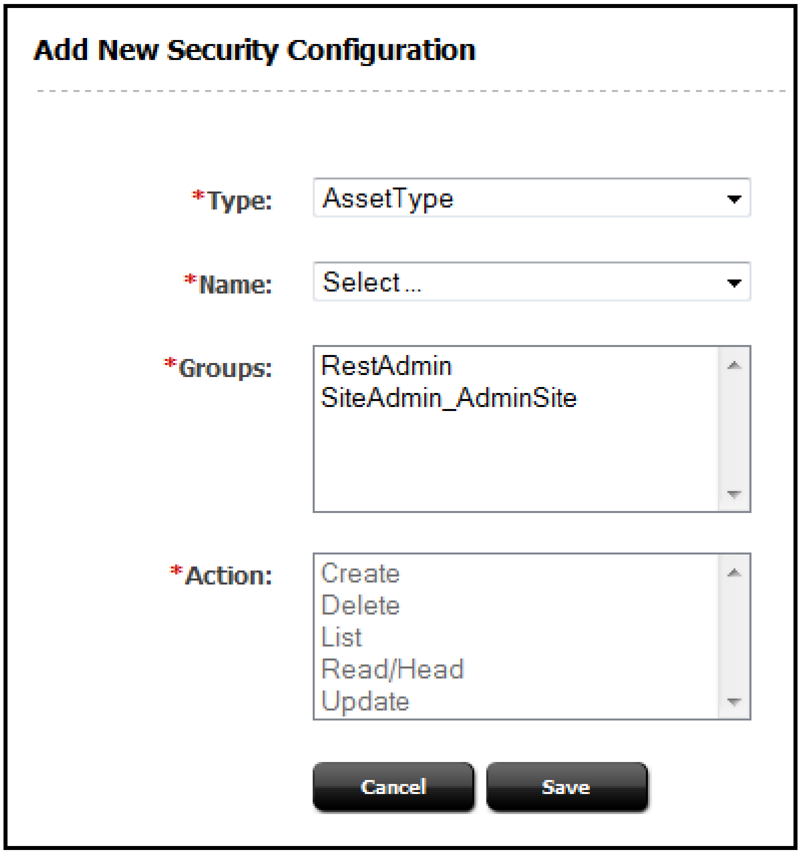
The following topics provide information about REST authorization:

- Security Model
- Privilege Resolution Algorithm

Security Model

The security model is based on objects, groups, and actions. Objects and groups are predefined in (objects in map to REST resources in the Framework). Actions are privileges (READ, CREATE) that you create in . You must configure security per object type in the Admin interface:

Figure 13-2 Add New Security Configuration Form



Add New Security Configuration

*Type: AssetType

*Name: Select ...

*Groups: RestAdmin
SiteAdmin_AdminSite

*Action: Create
Delete
List
Read/Head
Update

- An *object* is 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 manage multiple user's permissions to operate on objects.
- 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.
- 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 are created directly in , but not over REST).

A security configuration is an array that specifies:

- The protected object type and objects
- Groups that are able to access the objects
- Actions that groups (and their members) can perform on the objects

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 avisports site allows users in the group to modify avisports site details in WEM.

The `Asset` object type requires you to specify the site to which the security setting applies, because assets are always accessed from a particular site. You can refine the `AssetType` object type by specifying a subtype. For example, if you set the `DELETE` privilege on asset type `Content_C`, you perform a `DELETE` request on the REST resource `/types/Content_C` (that is, to delete the `Content_C` asset type from the system).

Because you can grant privileges to groups only, the total privileges for a user are not obvious until they are computed across all of the groups to which the user belongs. 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 group from step 1 is in the list of groups from step 2, then access is granted. Otherwise, access is denied.

Authorizing Users to Access Application Resources

Before authorizing users to access application resources, see REST Authorization for background information relating to the steps provided in the following topics:

- Viewing REST Security Configurations
- Creating a Group
- Adding Users to a Group
- Configuring Security for REST Resources.

> **Note:**
>
> To configure security for REST resources, you must assign privileges to groups. To view the list of groups, continue with Viewing REST Security Configurations. To create groups, see Creating a Group.

Viewing REST Security Configurations

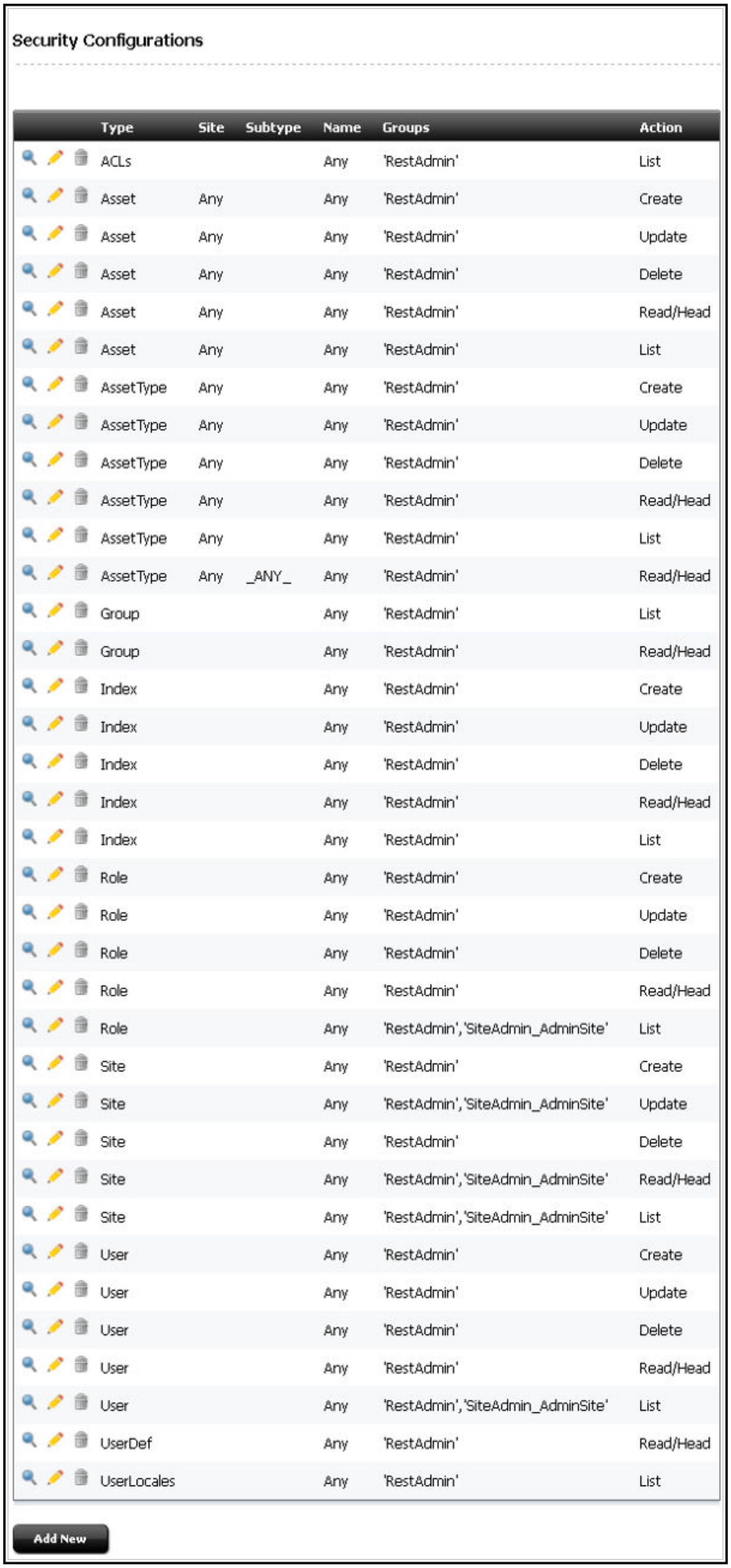
A security configuration identifies which groups have which permissions to which REST resources. WebCenter Sites defines security configurations for two default groups. They are `RestAdmin` and `SiteAdmin_AdminSite`.

To view REST security configurations:

1. Log in to the Admin interface as a general administrator.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, and then double-click **Configure Security**.

   The Security Configurations window opens in the main window.

Figure 13-3 Security Configuration Form



Security Configurations

	Type	Site	Subtype	Name	Groups	Action
🔍 ✏ 🗑	ACLs			Any	'RestAdmin'	List
🔍 ✏ 🗑	Asset	Any		Any	'RestAdmin'	Create
🔍 ✏ 🗑	Asset	Any		Any	'RestAdmin'	Update
🔍 ✏ 🗑	Asset	Any		Any	'RestAdmin'	Delete
🔍 ✏ 🗑	Asset	Any		Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	Asset	Any		Any	'RestAdmin'	List
🔍 ✏ 🗑	AssetType	Any		Any	'RestAdmin'	Create
🔍 ✏ 🗑	AssetType	Any		Any	'RestAdmin'	Update
🔍 ✏ 🗑	AssetType	Any		Any	'RestAdmin'	Delete
🔍 ✏ 🗑	AssetType	Any		Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	AssetType	Any		Any	'RestAdmin'	List
🔍 ✏ 🗑	AssetType	Any	_ANY_	Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	Group			Any	'RestAdmin'	List
🔍 ✏ 🗑	Group			Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	Index			Any	'RestAdmin'	Create
🔍 ✏ 🗑	Index			Any	'RestAdmin'	Update
🔍 ✏ 🗑	Index			Any	'RestAdmin'	Delete
🔍 ✏ 🗑	Index			Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	Index			Any	'RestAdmin'	List
🔍 ✏ 🗑	Role			Any	'RestAdmin'	Create
🔍 ✏ 🗑	Role			Any	'RestAdmin'	Update
🔍 ✏ 🗑	Role			Any	'RestAdmin'	Delete
🔍 ✏ 🗑	Role			Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	Role			Any	'RestAdmin','SiteAdmin_AdminSite'	List
🔍 ✏ 🗑	Site			Any	'RestAdmin'	Create
🔍 ✏ 🗑	Site			Any	'RestAdmin','SiteAdmin_AdminSite'	Update
🔍 ✏ 🗑	Site			Any	'RestAdmin'	Delete
🔍 ✏ 🗑	Site			Any	'RestAdmin','SiteAdmin_AdminSite'	Read/Head
🔍 ✏ 🗑	Site			Any	'RestAdmin','SiteAdmin_AdminSite'	List
🔍 ✏ 🗑	User			Any	'RestAdmin'	Create
🔍 ✏ 🗑	User			Any	'RestAdmin'	Update
🔍 ✏ 🗑	User			Any	'RestAdmin'	Delete
🔍 ✏ 🗑	User			Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	User			Any	'RestAdmin','SiteAdmin_AdminSite'	List
🔍 ✏ 🗑	UserDef			Any	'RestAdmin'	Read/Head
🔍 ✏ 🗑	UserLocales			Any	'RestAdmin'	List

3. Depending on your requirements, continue as follows:
   - To create a group, see Creating a Group.
   - To add users to a group, see Adding Users to a Group.

     > **Note:**
     >
     > To complete this step, ensure that the required groups exist. You must assign privileges to the groups.

   - To configure security privileges for REST resources, see Configuring Security for REST Resources.

Creating a Group

1. Log in to the Admin interface as a general administrator.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Groups** node, and then double-click **Add New**.
3. In the Add New Group form, enter a name and brief description about the group you want to create.
4. Click **Save**.

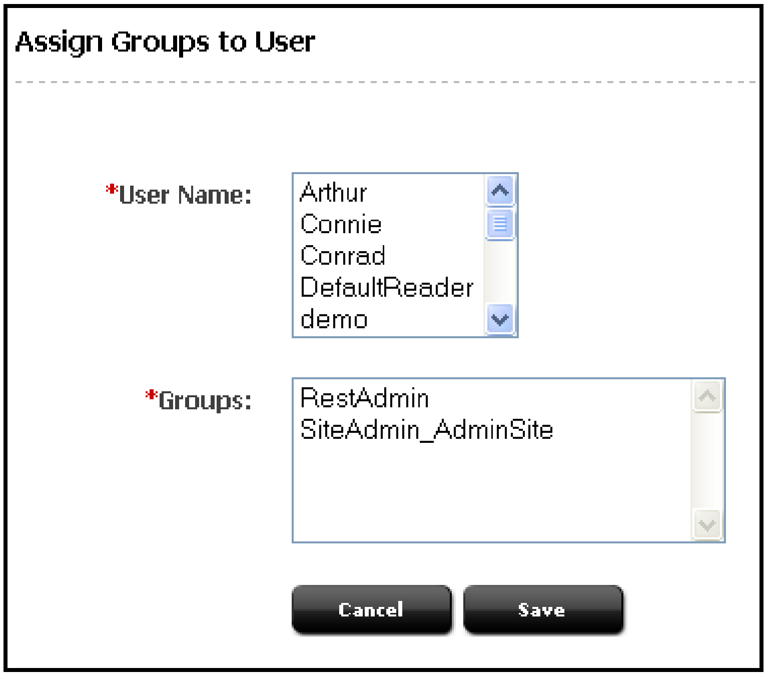
   The group you created is now listed under the **Groups** node.
5. Now that you have created a group, you can:
   - Add users to the group.
   - Configure REST security for the group.

Adding Users to a Group

Add users to a group to determine the permissions that they will have to operate on REST resources associated with applications.

1. Log in to the Admin interface as a general administrator.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Assign Users to Groups** node, and then 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 13-4 Assign Groups to User Form



Assign Groups to User

*User Name: Arthur
Connie
Conrad
DefaultReader
demo

*Groups: RestAdmin
SiteAdmin_AdminSite

> **Note:**
>
> If the user you want to assign to the group is not listed, that user is 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. To view the groups that include a particular user, double-click the user name.

5. (Optional). If the name of the user you want to assign to a given group is not displayed in the **User Name** field, do the following:

   a. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand **Assign Users to Groups**, and then double-click the name of the user you want to assign to another group.

   b. In the Inspect form, click **Edit** to open the Edit User Groups form.

   c. In the **Groups** field, select the groups to which you want to assign the user, and then click **Save**.

6. Now that you have added users to a group, you can do the following:

   - Create a group. For more information, see Creating a Group.
   - Configure security for a group. For more information, see Configuring Security for REST Resources.

Configuring Security for REST Resources

When configuring security, you 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 Admin interface as a general administrator.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.
3. In the Add New Security Configuration form, set security for object types and objects.

   The following tables show a summary of possible security configurations.

Table 13-1 Available Actions (Security Privileges)

Action	Description
Create	Group members can create specified resources.
Delete	Group members can delete specified resources.
List	Group member can retrieve specified resources.
Read/Head	Group members can read specified resources. Read returns the requested resources. Head returns metadata describing the requested resources.
Update	Group members can 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 13-2 Summary of Possible Security Configuration Options

Object Type	Name	Subtype	Site	Possible Actions
ACLs	Any	N/A	N/A	List
Application (see note 1)	Any	N/A	N/A	Create, Update, Delete
Application	*AppName*	N/A	N/A	Update, Delete
Asset	Any	N/A	Any	List, Read/Head, Create, Update, Delete
Asset	Any	N/A	*SiteName*	List, Read/Head, Create, Update, Delete
Asset	*AssetType*	N/A	*SiteName*	List, Read/Head (see note 2), Create, Update, Delete

Table 13-2 (Cont.) Summary of Possible Security Configuration Options

Object Type	Name	Subtype	Site	Possible Actions
Asset	*AssetType* and *AssetName*	N/A	*SiteName*	Read/Head, Update, Delete
AssetType	Any	N/A	N/A	List, Read/Head, Create, Delete
AssetType	*AssetType*	N/A	N/A	Read/Head, Delete
AssetType	*AssetType*	Any	N/A	List
AssetType	*AssetType*	*Subtype*	N/A	Read/Head
Group	Any	N/A	N/A	List
Group	*GroupName*	N/A	N/A	Read/Head
Index	Any	N/A	N/A	List, Read/Head, Create, Update, Delete
Index	*IndexName*	N/A	N/A	Read/Head, Update, Delete
Role	Any	N/A	N/A	List, Read/Head, Create, Update, Delete
Role	*Role*	N/A	N/A	Read/Head, Update, Delete
Site	Any	N/A	N/A	List, Read/Head (see note 3), Create, Update, Delete
Site	*SiteName*	N/A	N/A	Read/Head, Update, Delete
User	Any	N/A	N/A	List, Read/Head, Create, Update, Delete
User	*UserName*	N/A	N/A	Read/Head, Update, Delete
UserDef	Any	N/A	N/A	List
UserLocales	Any	N/A	N/A	List

See REST Security Configuration Reference for configuring the different object types.

Note:

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.

REST Security Configuration Reference

The following reference topics support Table 13-2. The topics provide details of the tabulated security configurations.

- Configuring REST Security for ACL Resources
- Configuring REST Security for Application Resources
- Configuring REST Security for Asset Resources
- Configuring REST Security for Asset Type Resources
- Configuring REST Security for Engage Resources
- Configuring REST Security for Group Resources
- Configuring REST Security for Indexed Asset Type Resources
- Configuring REST Security for Role Resources
- Configuring REST Security for Site Resources
- Configuring REST Security for User Resources
- Configuring REST Security for UserDef Resources
- Configuring REST Security for UserLocale Resources
- Configuring REST Security for Visitor Resources

Configuring REST Security for ACL Resources

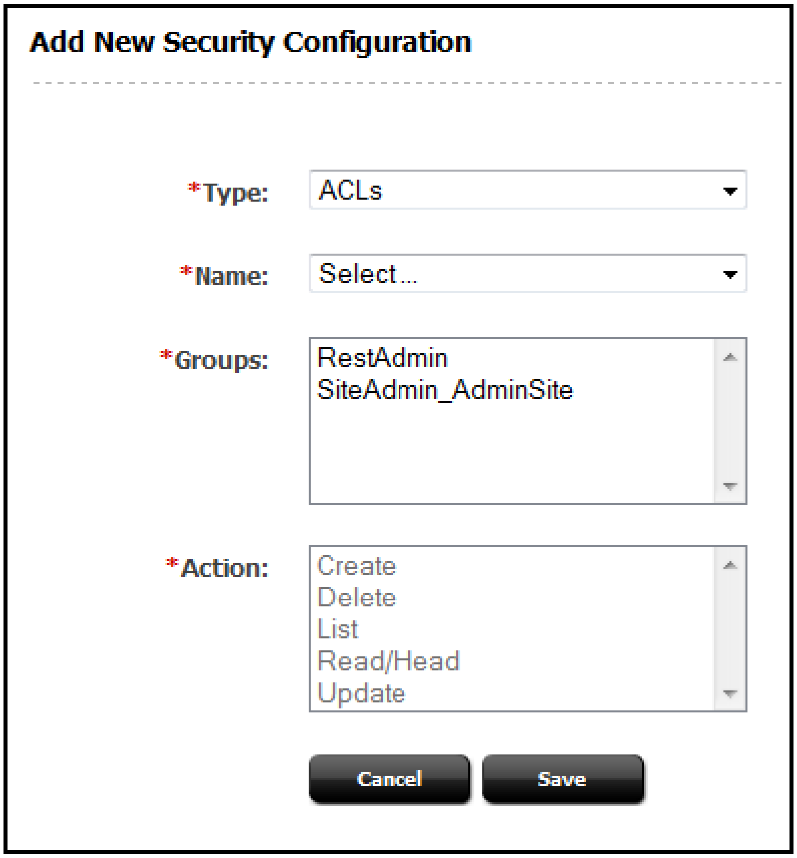
When you assign security privileges to ACLs, you determine which groups can view the ACL resource list.

To configure group security for ACLs:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **ACLs**.
3. In the **Name** field, select **Any**.

Figure 13-5 Add New Security Configuration for ACLs



Add New Security Configuration

*Type: ACLs

*Name: Select...

*Groups:
RestAdmin
SiteAdmin_AdminSite

*Action:
Create
Delete
List
Read/Head
Update

4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Configuring REST Security for Application Resources

When you assign 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 **General Admin** tree expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **Application**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-3 Add New Security Configuration Field Names

Field	Definition
Name	Select the name of the application you want to make available to the groups, or select **Any** to make all applications available to the groups.
Groups	Select the groups that have privileges to operate on the applications.
Action	Assign the security privileges to the groups. Your options depend on your selections in the previous fields. For example, if you selected **Any** and **Create**, members of your selected groups can create the assets which make the applications accessible in WEM.

Configuring REST Security for Asset Resources

When you assign 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **Asset**.
3. In the **Site** field, select the appropriate site.
4. In the **Name** field, select **Any**.
5. Select the **Groups** and **Actions** as required.
6. Click **Save** to save the configuration.

Table 13-4 Add New Security Configuration Form Field Names

Field	Definition
Site	Select the site associated with the asset you want to make available to the groups, or select **Any** to make all assets, system wide, available to the groups.
Name	Select the asset type associated with the asset you want to make available to the groups, or select **Any** to make all assets available to the groups. To make a specified asset of the selected asset type available to the groups, click the **Browse** button.
Groups	Select the groups that have privileges to operate on the assets.
Action	Assign the security privileges to the groups. 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 can perform searches in the specified site for assets of the specified asset type.

Configuring REST Security for Asset Type Resources

When you assign security privileges to asset types, you determine which groups can perform which operations on the specified asset types.

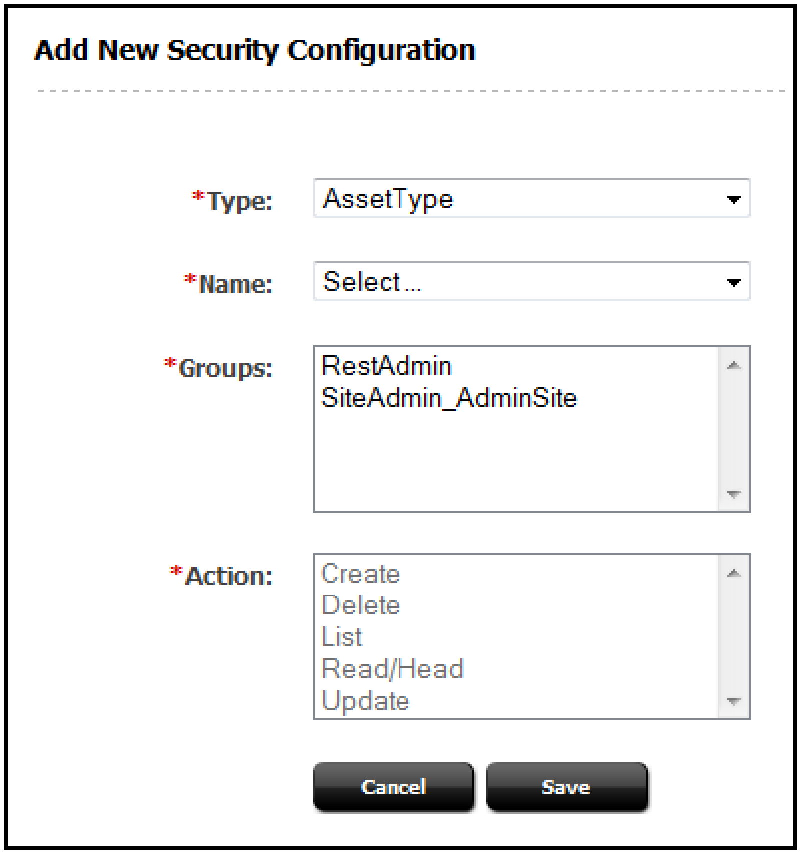
To configure group security for asset types:

1. In the **General Admin** tree expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.

2. In the **Type** field, select **Asset Type**.
3. In the **Name** field, select **Any**.

Figure 13-6 Add New Security Configuration Form for Field Names



Add New Security Configuration

*Type: AssetType

*Name: Select ...

*Groups: RestAdmin
SiteAdmin_AdminSite

*Action: Create
Delete
List
Read/Head
Update

4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-5 Add New Security Configuration Field Names

Field	Definition
Name	Select the asset types you want to make available to the groups, or select **Any** to make all asset types available to the groups.
Subtype	(Optional) Select the subtype of the asset type you want to make available to the groups. **Note:** If you selected the **Any** option in the **Name** field, then the **Subtype** field is not displayed.
Groups	Select the groups that have privileges to operate on the asset types.

Table 13-5 (Cont.) Add New Security Configuration Field Names

Field	Definition
Action	Assign the security privileges to the groups. Your options depend on your selections in the previous fields. For example, if you selected **Any** and **Create**, members of your selected groups can create asset types.

Configuring REST Security for Engage Resources

When you assign security privileges to Engage resources, you determine which groups can perform which operations on Engage.

To configure group security for groups:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **Engage**.
3. In the **Site** field, select the appropriate site.
4. In the **Name** field, select **Any**.
5. Select the **Groups** and **Actions** as required.
6. Click **Save** to save the configuration.

Table 13-6 Add New Security Configuration Form Field Names

Field	Definition
Site	Select the site associated with the asset you want to make available to the groups, or select **Any** to make all assets, system wide, available to the groups.
Name	Select the asset type associated with the asset you want to make available to the groups, or select **Any** to make all assets available to the groups. To make a specified asset of the selected asset type available to the groups, click **Browse**.
Groups	Select the groups that have privileges to operate on the assets.
Action	Assign the security privileges to the groups. 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 can perform searches in the specified site for assets of the specified asset type.

Configuring REST Security for Group Resources

When you assign security privileges to groups, you determine which groups can perform which operations on the specified groups.

To configure group security for groups:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

The Add New Security Configuration form opens.

2. In the **Type** field, select **Group**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-7 Add New Security Configuration Field Names

Field	Definition
Name	Select the groups you want to make available to the groups, or select **Any** to make all groups available to the groups.
Groups	Select the groups that have privileges to operate on the groups.
Action	Assign the security privileges to the groups. Your options depend on your selections in the previous fields. For example, if you selected **Any** and **List**, members of your selected groups can view a listing of the groups in the system.

Configuring REST Security for Indexed Asset Type Resources

When you assign 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." See Setting Up Search Indices.

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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **Group**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-8 Add New Security Configuration Form Field Names

Field	Definition
Name	Select the name of the indexed asset type you want to make available to the groups. Select **Any** to make all indexed asset types available to the groups. Select **Global** to make all indexed asset types associated with the "Global Search" available to the groups.
Groups	Select the groups that have privileges to operate on the indexed asset types.
Action	Assign the security privileges to the groups. Your options depend on your selections in the previous fields. For example, if you selected **Any** and **List**, members of your selected groups can search for assets of all types that are indexed on the system.

Configuring REST Security for Role Resources

When you assign 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **Role**.
3. In the **Name** field, select **Any**.
4. Select the Groups and Actions as required.
5. Click **Save** to save the configuration.

Table 13-9 Add New Security Configuration Form Field Names

Field	Definition
Name	Select the name of the role you want to make available to the groups, or select **Any** to make all roles available to the groups.
Groups	Select the user groups that have privileges to operate on the roles.
Action	Assign the security privileges to the groups. Your options depend on your selections in the previous fields. For example, if you selected **Any** and **Create**, members of your selected groups can create roles.

Configuring REST Security for Site Resources

When you assign 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.

2. In the **Type** field, select **Site**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-10 Add New Security Configuration Form Field Names

Field	Definition
Name	Select the name of the site you want to make available to groups, or select **Any** to make all sites available to groups.
Groups	Select the user groups that have privileges to operate on the sites.
Action	Assign the security privileges to the groups. Your menu options depend on your selections in the previous fields. For example, if you selected **Any** and **Create**, members of your selected groups can create sites.

Configuring REST Security for User Resources

When you assign 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **User**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-11 Add New Security Definition Field Names

Field	Definition
Name	Select the name of the user you want to make available to groups, or select the **Any** option to make all users available to groups.
Groups	Select the groups that have privileges to operate on the users.
Action	Assign the security privileges to the groups. Your menu options depend on your selections in the previous fields. For example, if you selected **Any** and **Create**, members of your selected groups can create users.

Configuring REST Security for UserDef Resources

When you assign security privileges to user definitions, you determine which groups can view the user definitions for the system.

To configure group security for userdef resources:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **UserDef**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-12 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 groups that have privileges to view user definitions.
Action	The only available security privilege you can assign to the groups is **Read/Head**, which enables the members of your selected groups to view the user definitions on the system.

Configuring REST Security for UserLocale Resources

When you assign 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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.
2. In the **Type** field, select **UserLocales**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configuration.

Table 13-13 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 groups that have privileges to view user locales.
Action	The only available security privilege you can assign to the groups is to view a listing of the user locales on the system.

Configuring REST Security for Visitor Resources

When you assign security privileges to visitors, you determine which groups can view the Visitor assets.

To configure group security for visitor resources:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **REST Security** node, expand the **Configure Security** node, and then double-click **Add New**.

   The Add New Security Configuration form opens.

2. In the **Type** field, select **Visitor**.
3. In the **Name** field, select **Any**.
4. Select the **Groups** and **Actions** as required.
5. Click **Save** to save the configurations.

Table 13-14   Add New Security Configuration Field Names

Field	Definition
Name	The only available option is to make all visitors available to groups.
Groups	Select the groups that have privileges to view a list of visitors.
Action	The only available security privilege you can assign to the groups is to view a listing of the visitors.

Part V

Managing Publishing

To be able to publish site content for your site visitors, you need to know the prerequisites, publishing destinations, publishing sessions, approval system, realtime publishing, and publishing events and templates.

Topics:

- Understanding Publishing
- Using RealTime Publishing
- Scheduling Publish Events

14

Understanding Publishing

Publishing, which involves copying content from the management system to the delivery system, helps you to make content available to the visitors of your online site. WebCenter Sites provides an approval system that determines which content gets published. WebCenter Sites' scheduling function enables you to set the publication time and a utility helps you to configure site-specific publishing destinations.
The following topics describe the publishing system and approval system:

- Prerequisites for Publishing Assets
- Publishing Destinations
- Authorizing Users to Publish Assets from the Contributor Interface
- Understanding the Approval System
- About Scheduled Publishing Sessions
- Understanding the Publishing Session
- Obtaining Information About a Publishing Session

Prerequisites for Publishing Assets

Before you can start publishing assets, you need to determine publishing method, publishing destination, assets to be approved for publishing, time of publishing, assets to be published, and user permissions to publish.

Consider the following before publishing any asset:

- Which publishing method WebCenter Sites should use. The recommended publishing method is RealTime Publishing, which is described in Using RealTime Publishing. Also, see Alternate Publishing Methods.

  As an administrator, you specify the publishing method for the destination when you configure the publishing destination.
- The publishing destination WebCenter Sites should publish to during a given session.

  As an administrator, you configure the publishing destinations for your system. See 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 ready for publication, a content provider marks it as approved for a specific publishing destination. The approval system validates the approved label to determine if publishing the asset can 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. 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. See Understanding the Approval System and Learning Export to Disk Publishing Terminology.

- 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 regular intervals. You can also override the schedule and publish on demand. See About Scheduled Publishing Sessions.

- Which users should be granted permissions to run on-demand publishes on individual assets from the WebCenter Sites: Contributor interface.

  To grant Contributor interface users the permission to publish assets from the Contributor interface, you must assign each user the Publisher role. For instructions, see Authorizing Users to Publish Assets from the Contributor Interface.

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 for other administrative tasks, as the publishing process runs in the background. See Understanding the Publishing Session and Obtaining Information About a Publishing Session.

Publishes triggered from the Contributor interface can also be viewed from the Contributor interface’s Publish Console.

Publishing Destinations

When you publish sites from WebCenter Sites, you can choose either the WebCenter Sites database to which you are mirroring content, or a directory to which you are exporting content, as the publishing destination.

A publishing destination is expressed as a named object that defines the following parameters:

- A publishing method.
- 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, categorized under Publish, in the `wcs_properties` 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 multiple publishing destinations. Depending on how your sites are designed by the developers, you may have 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 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 WebCenter Sites system to any other WebCenter 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.

See Configuring an Export Destination and Creating a Mirror Destination.

Authorizing Users to Publish Assets from the Contributor Interface

Typically, publishing is scheduled and run from the Admin interface. However, the Contributor interface provides an on-demand publish feature which users, granted the appropriate role, can use to publish individual assets directly from the Contributor interface.

To authorize a user to publish assets from the Contributor interface:

1. From the WEM Admin interface, click **Sites** on the menu bar.
2. In the **Sites** list, hold the cursor over the site name and click **Manage Site Users**.

> **Note:**
>
> The Contributor application must be assigned to the site you select. See Authorizing Users to Work with Applications.

3. In the **Manage Site Users** list, point to the user you want to authorize to publish assets directly from the Contributor interface, and select **Assign Roles to User**.
4. In the Assign Roles to User form, select the **Publisher** role from the **Available** list box and move it to the **Selected** list box.
5. Click **Save and Close**.
6. Verify the user has permission to publish from the Contributor interface on the selected site.
   a. Log in to the site with the user’s credentials and select the **Contributor** icon.
   b. In the Contributor interface, find and open an asset you want to publish. See Finding and Organizing Assets in *Using Oracle WebCenter Sites*.
   c. In the asset’s toolbar, point to the **Approve** icon. If the tooltip says “Publish,” then you have successfully granted the user publishing permissions in the Contributor interface.

Understanding 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 calling the approval system. The approval system runs a pre-publication process that protects the site against possible 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 — 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 could 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. See Evaluating Approval Dependencies and Approving and Publishing the Assets to the Delivery System in *Developing with Oracle WebCenter Sites*.

Detailed information about the workings of the approval system is given in Understanding the Approval System.

About Scheduled Publishing Sessions

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 one 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 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.
- The publishing feature in the Contributor interface is disabled during a scheduled publish session.

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, categorized under Publish, in the `wcs_properties.json` file to identify the batch user account for your WebCenter Sites system.

> **Note:**
>
> Publishing also requires a mirror user, different from the batch user. See Working with Users and Mirror to Server Publishing.

Understanding the Publishing Session

When the publishing system begins publishing approved assets, 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.
- Calls 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.

When the publishing session concludes, the publishing system notifies the approval system of which assets were published and then it updates the page cache.

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 opens 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 Troubleshooting.

> **Note:**
>
> You must have the xcelpublish ACL to view the Publish Console.

15

Using RealTime Publishing

RealTime publishing is a dynamic publishing process that shows you status updates on the progress of publishing, in real time. You can monitor sessions in progress, review publishing schedules, inspect history records, and explore logs.
When you configure the RealTime publishing process, you define the publishing destination and mirror site configuration data to that destination. You can also set the publishing schedule and use the RealTime process to synchronize site data.

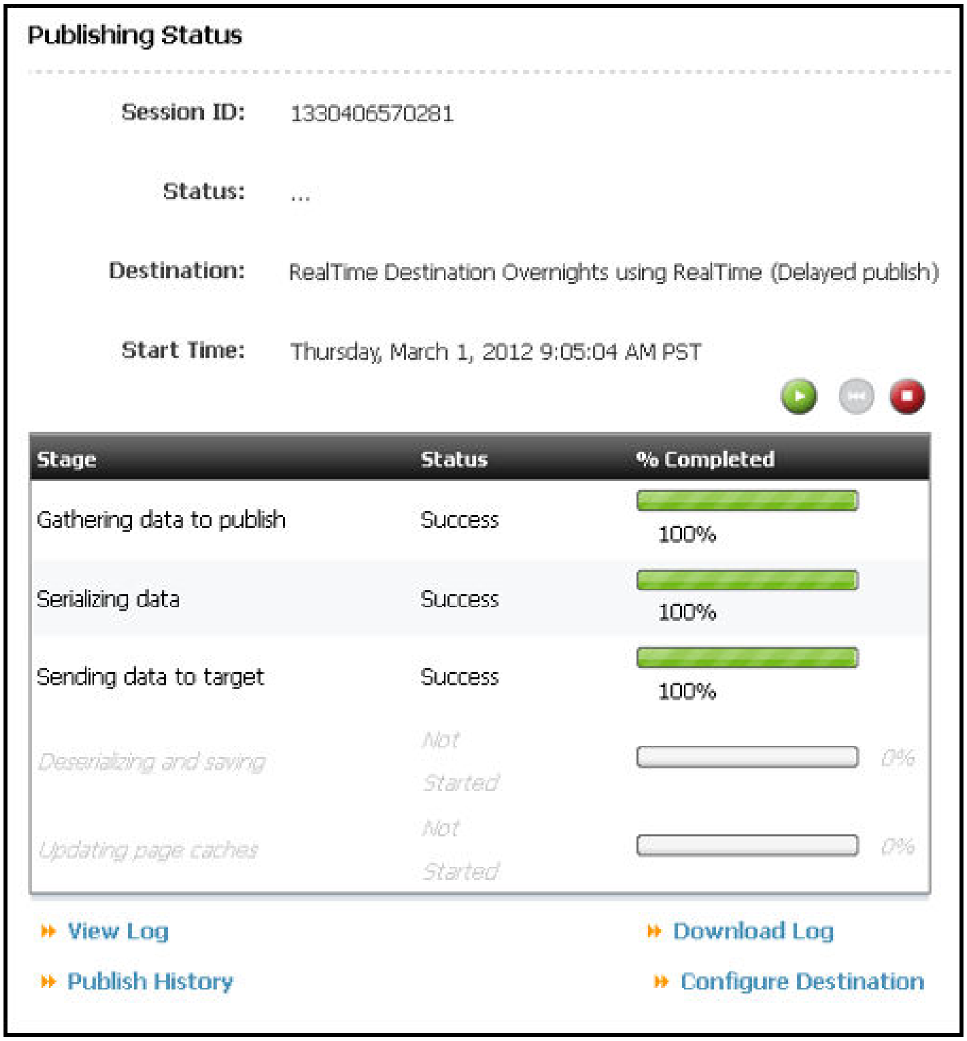
The following topics describe how to configure RealTime publishing and monitor the feedback from publishing processes:

- RealTime Publishing Overview
- Working with RealTime Publishing
- Configuring Your System for RealTime Publishing
- Adding a New RealTime Destination Definition
- About Synchronizing Site Configuration Data
- How to Differentiate Complete and Delayed Publishing
- About the Publish Console
- Working with Active Publishing Sessions
- Working with Scheduled Publishing Sessions
- Working with Prior Publishing Sessions
- Working with Publishing Logs

RealTime Publishing Overview

RealTime publishing is a dynamic publishing method named for its user interface, as publishing status messages update in real time. Progress bars and mouse-over tooltips illustrate the progression of a publishing session with status information from both source and destination systems.

Figure 15-1 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	100%
Serializing data	Success	100%
Sending data to target	Success	100%
Deserializing and saving	Not Started	0%
Updating page caches	Not Started	0%

The RealTime user interface is also interactive. 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 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 website.

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.

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 shows the five stages of publishing. These stages are described in detail in Working with RealTime Publishing.

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.

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.

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 for Publishing

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 `wcs_propeties.json` file.
- For large publishes (such as the initial publication of a site), the session timeout may have to be increased beyond the default 900 seconds (for example, 9000 seconds). To do so, open the Property Management Tool in the Admin interface

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

> **Note:**
>
> When WebCenter Sites is integrated with Oracle Traffic Directory (OTD), change the following setting in the server.xml file of the OTD while publishing:
>
> ```
> <http>
>     <server-header>default</server-header>
>     <max-unchunk-size>102400000</max-unchunk-size>
>     <unchunk-timeout>-1</unchunk-timeout>
>     <io-timeout>600</io-timeout>
> <http>
> ```

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 for Publishing

The destination system requires the following items to be configured:

- A **RealTime user account,** which has publishing privileges on the destination.
- **Mirrored site configuration data** from the source system for the sites to publish:
  - 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.

If you will not publish from the destination system, we recommend turning off asset invalidation.

Working with RealTime Publishing

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 required 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 Working with Active Publishing Sessions.

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 (using `java.io.temp`) 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), the space required is can exceed what WebCenter Sites is using, particularly when large binaries are included.

3. **Sending data to the target:**

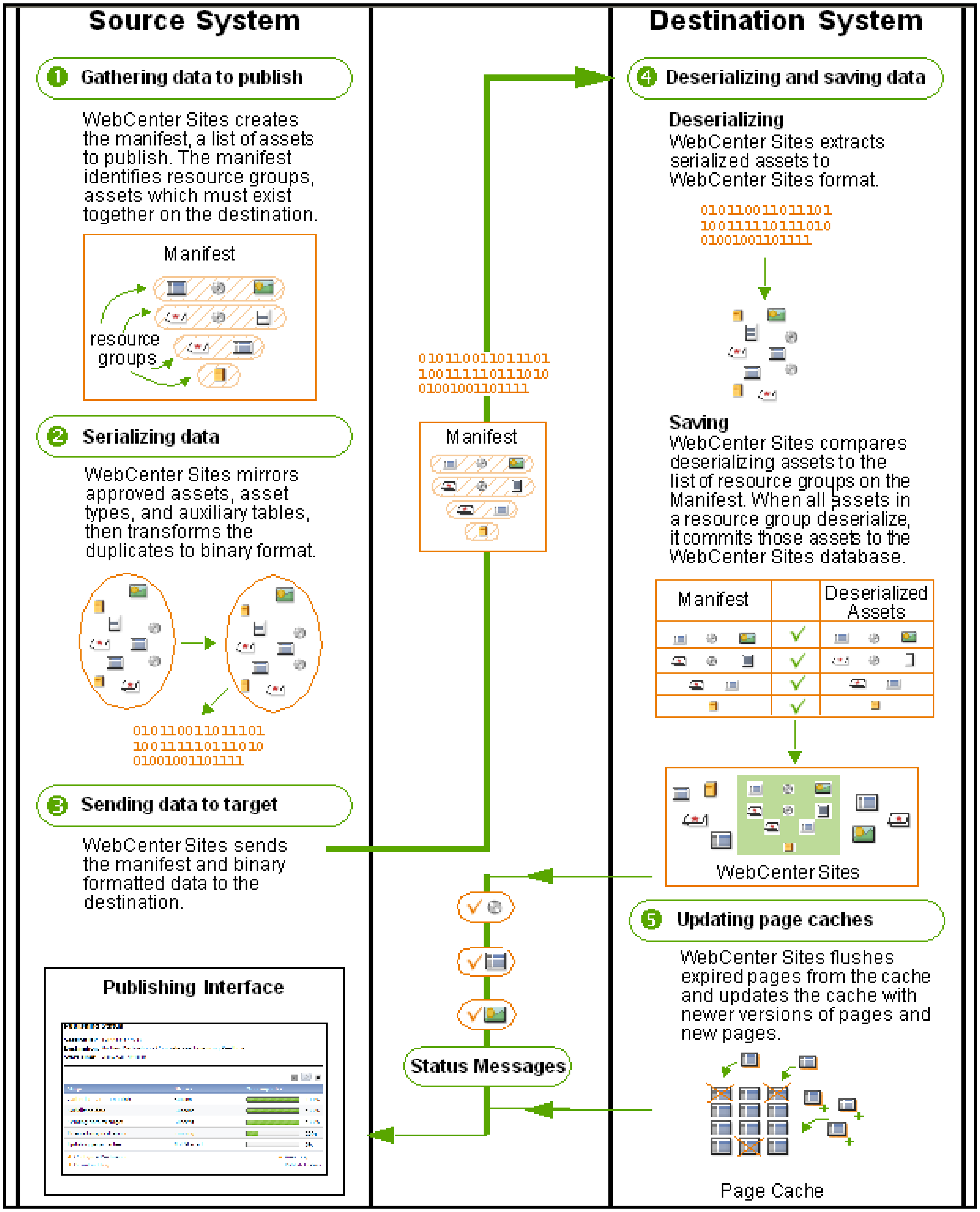
   a. The destination WebCenter Sites database locks assets approved for this publishing session. When assets on the target are locked, they cannot be edited.
   b. The WebCenter Sites source system sends the manifest and all serialized data to the destination. The data is not yet been committed to the 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:
      i. 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.
      ii. 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 15-2 The Five Stages of Publishing



WebCenter Sites creates the manifest, a list of assets to publish. The manifest identifies resource groups, assets which must exist together on the destination.

WebCenter Sites mirrors approved assets, asset types, and auxiliary tables, then transforms the duplicates to binary format.

WebCenter Sites sends the manifest and binary formatted data to the destination.

WebCenter Sites extracts serialized assets to WebCenter Sites format.

WebCenter Sites compares deserializing assets to the list of resource groups on the Manifest. When all assets in a resource group deserialize, it commits those assets to the WebCenter Sites database.

WebCenter Sites flushes expired pages from the cache and updates the cache with newer versions of pages and new pages.

Configuring Your System for RealTime Publishing

The following topics provide instructions for configuring a RealTime publishing environment with information specific to your WebCenter Sites installation and your business practices:

- Creating the Batch User Account (If One Does Not Exist)
- Setting Up the Destination System
- Identifying a Proxy Server to the Source System
- Creating a RealTime Destination Definition on the Source System
- Initializing the Destination Database
- Mirroring the Site Configuration Data to the Destination Database
- Testing the Publishing Process
- Testing the Published Site
- Setting Up the Schedule
- Turning Off Asset Invalidation on the Delivery System

Creating the Batch User Account (If One Does Not Exist)

This procedure must be performed only one time on each source system, regardless of the number and types of publishing destinations you are configuring.

If you have not done so, create the batch user account. A batch user account is required before any type of publishing can take place. If a batch user account exists on your source system, skip to Setting Up the Destination System.

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 are stored.

> **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.

To create the batch user account:

1. Log in to the Admin interface on your source system.
2. In the **General Admin** tree, expand the **User Access Management** node, and then 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 require help with this step, see Creating a User in the Admin Interface.

4. Open the Property Management Tool:

   In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double click **Property Management**.

   > **Note:**
   >
   > For information about using the Property Management Tool, see the Properties in the Publish Category in *Property Files Reference for Oracle WebCenter Sites*.

5. In the **Category** drop-down menu, select **Publish**.
6. Click **Search**.
7. 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.

   - `request.folder`

     Point this property to the directory 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 is shown in the **Publish History** tab of the Publish Console.

8. Restart the application server.

Setting 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 ensure that the structure of auxiliary tables on the source database is reproduced in the destination database.

The destination also requires a mirror user to complete database transactions on the destination system during publishing. You 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 using 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 (in the `wcs_properties.json` file, set from the Property Management Tool). Be sure to restart the application server if you make changes to database schema properties.

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 could 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 require help with creating the mirror user, see Creating a User in the Admin Interface.

Identifying a Proxy Server to the Source System

This procedure must be performed only one time on each source system, regardless of the number and types of publishing destinations you are configuring.

If the local proxy has been identified to your source system, or a proxy server is not used, skip to Creating a RealTime Destination Definition on the Source System.

To identify a proxy server to the source system, see one of the following topics:

- Identifying the Local Proxy Server to the Source System for All Destinations
- Identifying the Local Proxy Server to the Source System Per Destination

Identifying the Local Proxy Server to the Source System for All Destinations

1. In the Admin interface, open the Property Management Tool.
2. In the **Category** drop-down menu, select **Publish**.
3. Click **Search**.
4. Set values for and save 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.
5. Restart the application server.

Identifying the Local Proxy Server to the Source System Per Destination

1. In the Admin interface, select the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, expand the **Destinations** node, and then double-click to open the destination.

The destination opens in the main form.

2. Click **Edit**.

   The Edit Destination form for the destination opens.

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**.

Creating 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.

1. On the source system, log in to the administrator's interface.
2. In the **General Admin** tree, expand the **Admin** tree, expand the **Publishing** node, and expand the **Destinations** node.
3. Under the **Destinations** node, double-click **Add New**.

   The Add New Destination form opens.

4. Fill in the Add New Destination form. For guidelines, use the information below.
   - **Name**: Enter a unique name for the destination.

     > **Note:**
     >
     > The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), greater-than sign (>), percent (%), and question mark (?). Additionally, the name cannot end with a backslash (\).

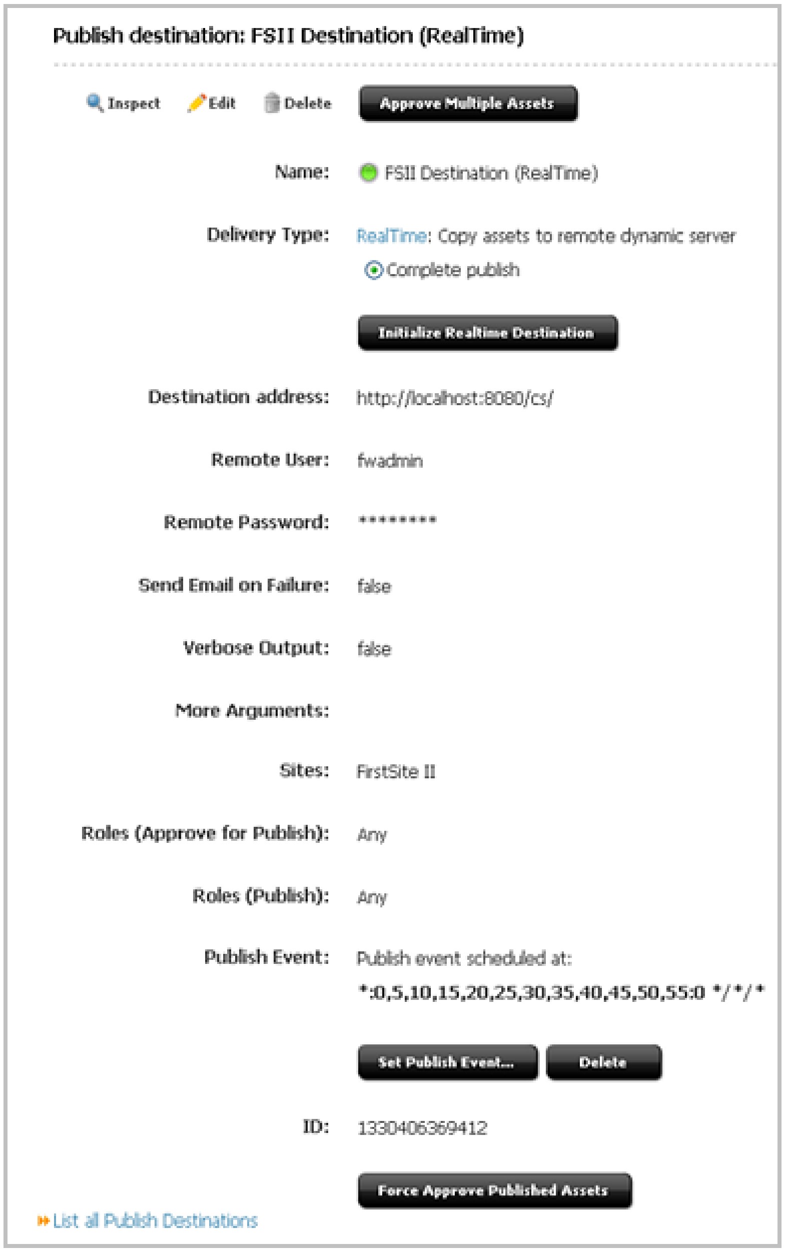
   - **Delivery Type**: Select **RealTime: Copy assets to remote dynamic server**, then choose an option:
     – **Complete publish** to copy assets to the destination address without interruption to the publishing process.
     – **Delayed publish** to pause the publishing process before assets are committed 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). See Resuming a Delayed Publishing Session.

- **Destination Address**: Enter the URL in the format shown. For `[targetserver:port]`, delete the brackets, enter the host name or IP address of the destination system, and specify the port to be used by the destination. (A slash is required after the URL because this URL is appended dynamically.)
- **Remote User**: Enter the name of the RealTime user that you created in Setting Up the Destination System. This user is called 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 check box and fill in the field that opens:
  - **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 can 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 are able to approve assets.
- **Roles (Publish)**: Select the roles to which you are assigning publish privileges. All users assigned these roles are 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 15-3   Publish Destination Form



Publish destination: FSII Destination (RealTime)

Name:	● FSII Destination (RealTime)
Delivery Type:	RealTime: Copy assets to remote dynamic server ● Complete publish
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 */*/*
ID:	1330406369412

Initialize Realtime Destination
Set Publish Event...
Delete
Force Approve Published Assets
List all Publish Destinations

Note:

A red button next to the destination name means that the target server cannot be located. Common causes are the target URL is incorrect or the target server is not running.

6. 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, Initializing the Destination Database. Begin at step 3 of that procedure.

Initializing the Destination Database

You must initialize the destination database before you can publish to it. To initialize, you mirror two constructs: WebCenter Sites, and data in auxiliary tables (which support the types of assets users are publishing)

The `Publication` and `PublicationTree` tables are updated with the site names and table data you have chosen to mirror.

To initialize the destination:

1. In the **General Admin** tree on the source system, expand the **Admin** node, expand the **Publishing** node, expand the **Destinations** node.
2. Under **Destinations** node, double-click the RealTime destination you want to initialize.
3. In the Publish Destination form, click **Initialize RealTime Destination**.

   The Initialize RealTime Destination form opens.
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 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 have 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 through 6 for the remaining tables.
6. Click **Mirror**.

   If initialization is successful, WebCenter Sites displays a confirmation message (otherwise, an error message opens).

   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.

Note:

Adding a new page under an existing page asset in `SitePlanTree` will mark the parent page as modified (that is, it needs approval). However, the dependency is not reflected in the cache, and the child page asset can be published without publishing the parent. This can result in the parent page asset not flushed from the cache, and the child page asset will not show up on any navigation the parent may have.

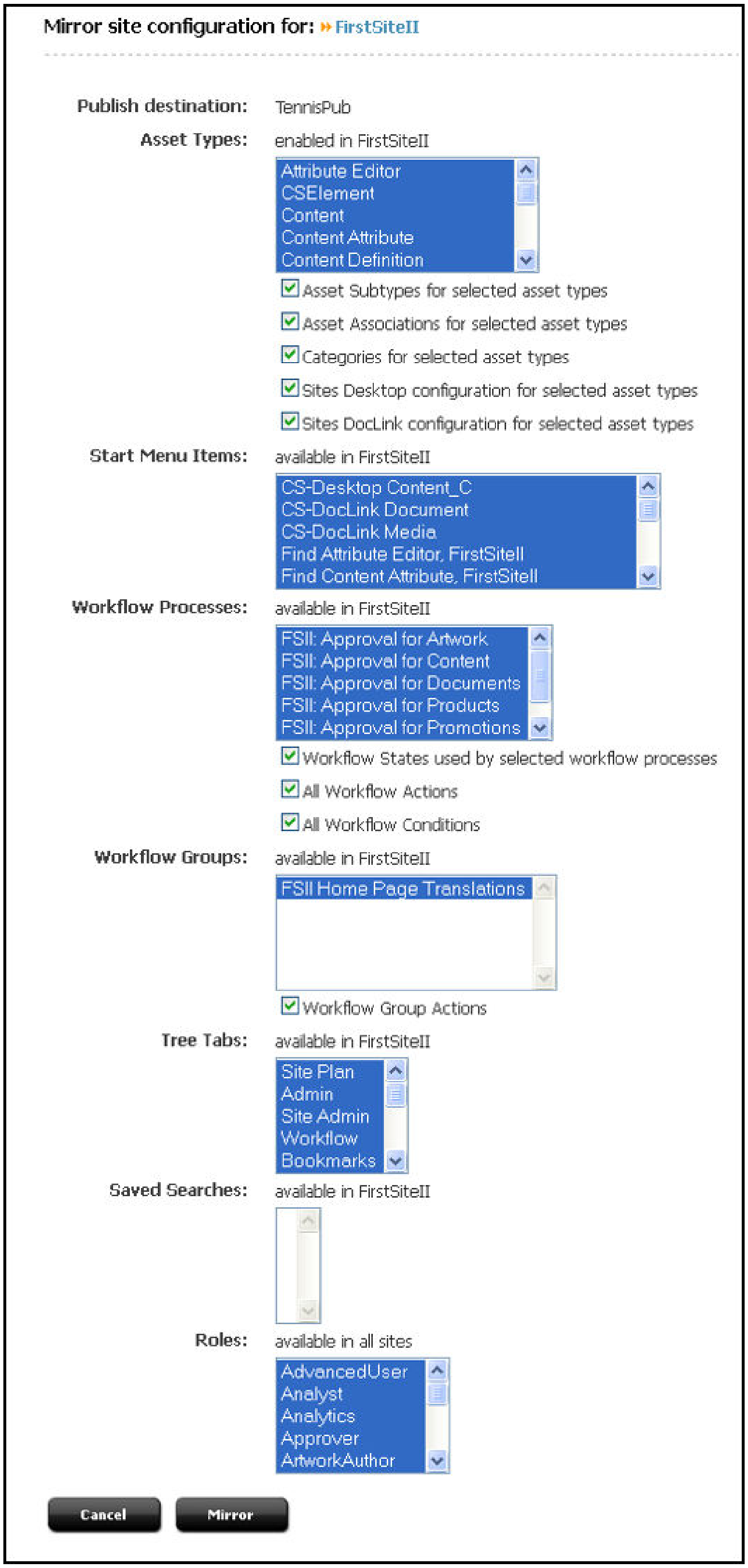
Mirroring the Site Configuration Data to the Destination Database

In this step, you will mirror the source's site configuration data (such as asset types and start menu items) to the destination database:

1. In the **General Admin** tree on the source system, expand the **Admin** node, expand the **Sites** node, and double-click the site whose data you want to publish to the destination.
2. In **Publish Destinations** field (near the bottom of the form), click **Mirror site configuration for destination**.

   The Mirror Site Configuration form opens.

Figure 15-4   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 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 is mirrored only if you select **Asset associations for selected asset types** and then the relevant asset types.
- **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 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 the workflow configuration options you selected in step 3, appropriate rows from the workflow tables are copied to the destination.
5. Repeat steps 1 through 4 for each site whose assets you want to publish to the destination.

Testing the Publishing Process

You can test your publishing process by using all publishable assets or a smaller collection. To publish assets to a test environment, you must first approve the assets you want to publish. The following topics provide instructions for approving and publishing assets:

- Approving Assets
- Publishing the Assets

Approving Assets

To approve assets, complete either of the following:

- To fully test your publishing process, you must approve all the assets that you intend to publish. See Approving Multiple Assets.
- If you wish to perform a simple test of your configuration, temporarily create a home page asset with just a few dependents.

Publishing 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 list and click **Select Destination**.

   WebCenter Sites displays information about the assets that are ready to be published to this destination.
3. If 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 Initializing the Destination Database). This creates the basic site information on the destination.
4. Click **Publish**.

   The publishing system mirrors all the approved assets for this destination to the WebCenter Sites database on the destination system.

Testing the Published Site

To test your published site, point your browser to the home page asset on the destination system and examine the site.

If you have not 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>
```

In this example:

- `<targetserver:port>` is the host name or IP address (and sometimes the port number) of the destination system.
- `<cgipath>` is the value of the `ft.cgipath` property in the `wcs_properties.json` file. For example, `<cgipath>` is `/servlet/` for Oracle WebLogic Server 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` (for information about these variables, see Configure SiteEntry in *Developing with Oracle WebCenter Sites*).

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. In the Admin interface, open the Property Management Tool. For help with this step, see Properties in the Core Category in *Property Files Reference for Oracle WebCenter Sites*.
   a. In the **Category** drop-down menu, select **Core**, and then click **Search**
   b. Locate the `ft.cgipath` and `ft.approot` properties and note their values.
2. Open a text editor and assemble the URL:
   a. Enter `http://` or `https://` followed by: `<targetserver:port>/<cgipath>/`

      For example:

      ```
      http://example.com:8080/servlet/
      ```

      (for Oracle WebLogic Server and IBM WebSphere)
   b. After the string, add the following text:

      ```
      ContentServer?pagename=<your_home_page>
      ```

      Now the URL should look similar to the following example (for Oracle WebLogic Server and IBM 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.
>
> 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.

Setting Up the Schedule

When you have ensured that your destination is correctly set up, finish configuring your source's publishing system.

- On the source system, schedule times running publishing sessions to the destination. For help with this step, see Scheduling a Publish Event.

- If your site uses image files that are stored on the web server as images, not as assets, plan how you will move the image files from the CM system web server to the delivery system web server. For example, FTP.
- 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 CatalogMover in *Developing with Oracle WebCenter Sites*.

Turning Off Asset Invalidation on the Delivery System

By default, the destination invalidates chain publishing. That is, the destination publishing system marks each published asset as changed, solely because the asset was published. Hence, an asset must be approved from the current destination before it can be published to yet another destination.

Asset invalidation should be turned off for assets on a final destination, such as a delivery system, because assets are not published from it. To turn off asset invalidation, do the following:

1. Open the Property Management Tool, in the Admin interface, on the final destination system. (For instructions on accessing the Property Management Tool, see Properties in the Publish Category in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Name** field, enter `xcelerate.publishinvalidate`.
3. Click **Search**.
4. In the Properties section of the form, select the name of the property.
5. Set the value of this property to `false`.
6. Click **Save**.

Adding a New RealTime Destination Definition

After you have configured your source system to publish to one RealTime destination, you can define other RealTime destinations, if necessary. Use only the following steps:

1. Setting Up the Destination System
2. Creating a RealTime Destination Definition on the Source System
3. Initializing the Destination Database
4. Mirroring the Site Configuration Data to the Destination Database
5. Testing the Publishing Process
6. Testing the Published Site
7. Setting Up the Schedule
8. Turning Off Asset Invalidation on the Delivery System

About 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.

For instructions on using the Site Configuration form, see Mirroring the Site Configuration Data to the Destination Database.

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 have to troubleshoot your configuration. If you can successfully initialize a RealTime destination, the source and the destination systems are communicating.

For instructions on using the Initialize Mirror Destination form, see Initializing the Destination Database.

How to Differentiate Complete and Delayed Publishing

A publishing session can run in one of two modes: complete or delayed. The mode must be explicitly specified.

- 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.

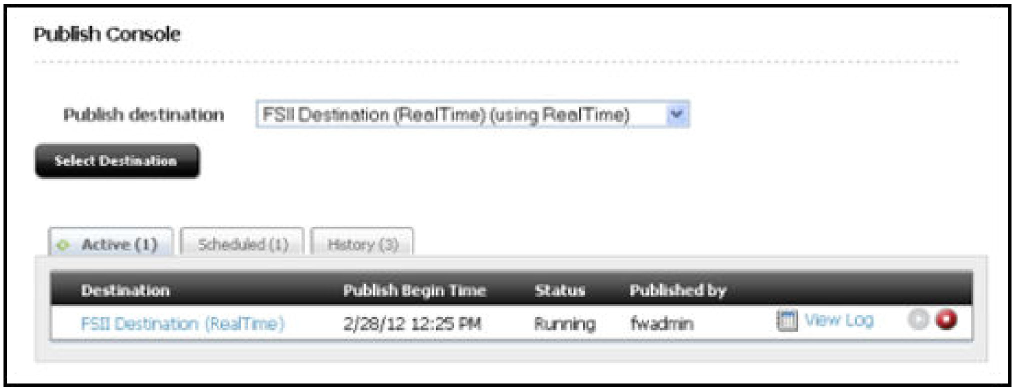
This chapter contains instructions for monitoring complete and delayed publishing sessions, adjusting publishing schedules, and retrieving the histories of past sessions.

About the 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 Admin interface and click **Publishing** on the button bar.

Figure 15-5 Publish Console



Publish Console

Publish destination: FSII Destination (RealTime) (using RealTime)

Destination	Publish Begin Time	Status	Published by		
FSII Destination (RealTime)	2/28/12 12:25 PM	Running	fwadmin	View Log	

- The **Active tab** shows a summary of ongoing publishing sessions that were triggered from the Admin and Contributor interfaces. 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 logs. Upon completion of the publishing session, you can download text logs. See Working with Active Publishing Sessions and Viewing and Searching Logs.
- The **Schedule tab** displays the number of approved assets for each destination and the schedule of publishing sessions. See Working with Scheduled Publishing Sessions.
- The **History tab** shows summary information on prior publishing sessions. See Working with Prior Publishing Sessions. From this tab, you can also access logs of prior publishing sessions. See Working with Publishing Logs.

Working with Active Publishing Sessions

The **Active** tab of the Publish Console shows information about all publishing sessions, triggered from the Admin and Contributor interfaces, 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.

The following topics provide information about working with active publishing sessions:

- Monitoring a Publishing Session
- Resuming a Delayed Publishing Session
- Canceling a Publishing Session

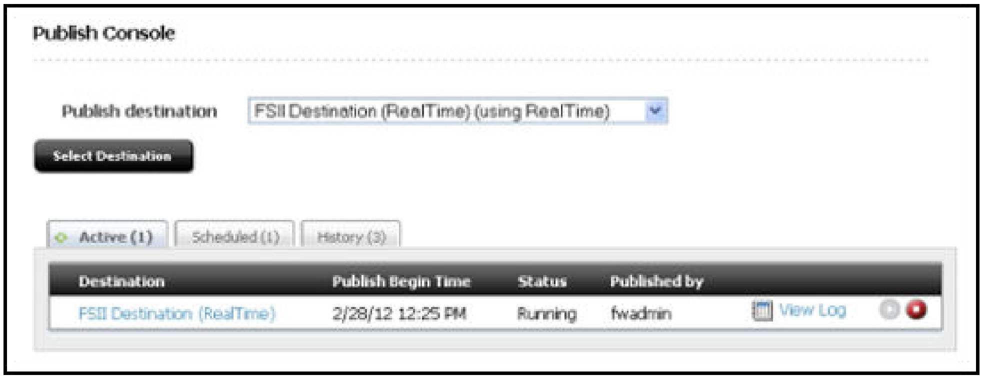
> **Note:**
>
> When a delayed publishing session pauses, it remains listed in the **Active** tab. **Canceled** sessions are moved to the **History** tab.

Monitoring a Publishing Session

1. Open the Publish Console (in the Admin interface, click **Publishing** in the menu bar).

   The Publish Console opens with the **Active** tab at the front.

   Figure 15-6 Publish Console

   Publish destination: FSII Destination (RealTime) (using RealTime)

Destination	Publish Begin Time	Status	Published by	
FSII Destination (RealTime)	2/28/12 12:25 PM	Running	fwadmin	View Log

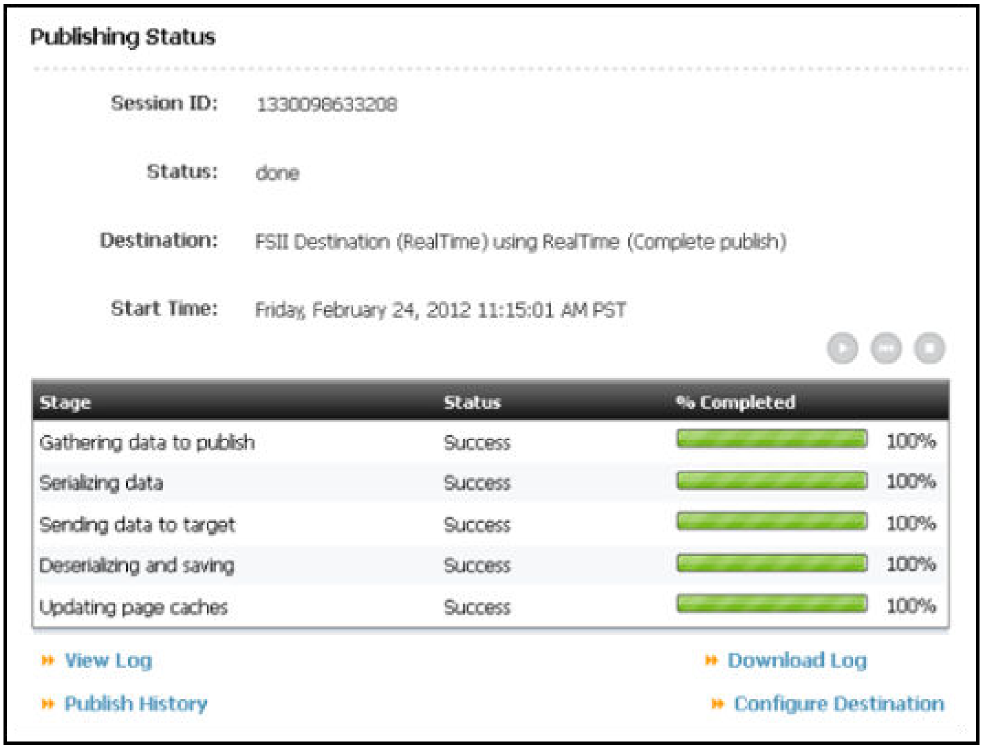
   There are some actions to note on this screen:

   - The circling green arrows on the **Active** tab indicate that at least one session is in progress. The number in parenthesis indicates the number of sessions in progress.
   - Each destination shown in the Destination column is an active link. Click the destination to open its Publish Status form.
   - Select the View Log check box to view the log as it is being created.
   - Click the red button to cancel the session.

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 15-7 Publishing Status



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	100%
Serializing data	Success	100%
Sending data to target	Success	100%
Deserializing and saving	Success	100%
Updating page caches	Success	100%

Notice the five status bars, one for each publishing stage. 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, click **View Log** to do so. Click **Download Log** to download the log for viewing once complete.

The publishing jobs per each target of the multi-target publish can complete in different orders because they are independently run on each target during the transport phase. For example, CacheFluster for target 1 might complete before Unpacker for target 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 publishing session:

1. Open the Publish Console (in the Admin interface, click **Publishing** in the menu 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.)
3. Read the destination's Status column and complete either:
   - 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).
   - 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.

        The Resume button is located between the **Start** and **Cancel** buttons.
     d. When the publishing session is complete, all five stages display "Success" in the Status column.
4. When 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 Working with Prior Publishing Sessions.

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.

For instructions about canceling a publishing session, see the following topics:

- Canceling a Publishing Session from the Publish Console
- Canceling a Publishing Session from the Publishing Status Form

Canceling 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 is not rerun. Otherwise, if you plan to rerun the session, cancel the session from the Publishing Status form.

   The Publish Console opens, 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.

   A message that the cancellation request has been sent opens above the list of active publishing sessions. 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.

Canceling a Publishing Session from the Publishing Status Form

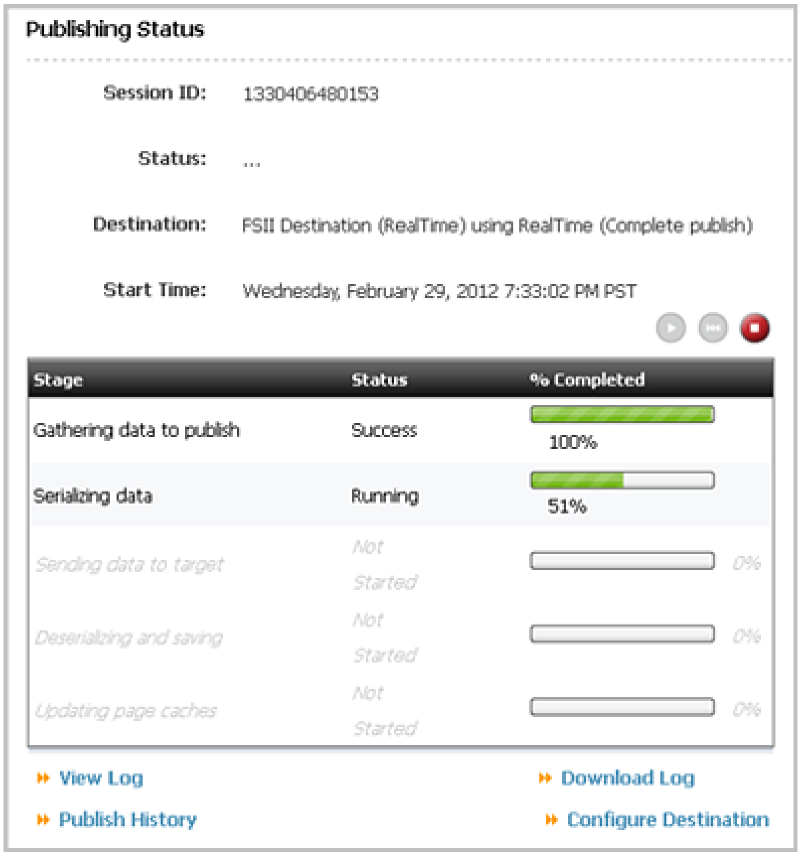
1. Open the Publish Console (in the Admin interface, click **Publishing** in the menu bar).

   The Publish Console opens, 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 before that stage invalidates rerunning the session.
>
> The option to redo a session is available if these conditions are met:
>
> - 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 15-8 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%
Sending data to target	Not Started	0%
Deserializing and saving	Not Started	0%
Updating page caches	Not Started	0%

A message opens 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.

5. When you are ready to rerun the session, select the **History** tab, navigate to the session, and click the **Redo** button. See Redoing a Publishing Session.

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 Scheduling a Publish Event.

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 Admin interface, click **Publishing** in the menu bar).
2. Click the **Scheduled** tab.

   The **Scheduled** tab displays only destinations with a publishing schedule.

   > **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 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 disable the schedule.)

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 Viewing Past Publishing Sessions.
- Access the final Publishing Status form for completed sessions.
- Redo a canceled session. See Redoing a Publishing Session.
- Access session logs.
- Clear the entire history list or delete selected sessions.

Viewing Past Publishing Sessions

Previous publishing sessions, triggered from both the Admin or Contributor interfaces, are stored in a history. Accessing the history will bring up the logs of each publishing session. The history can be cleared.

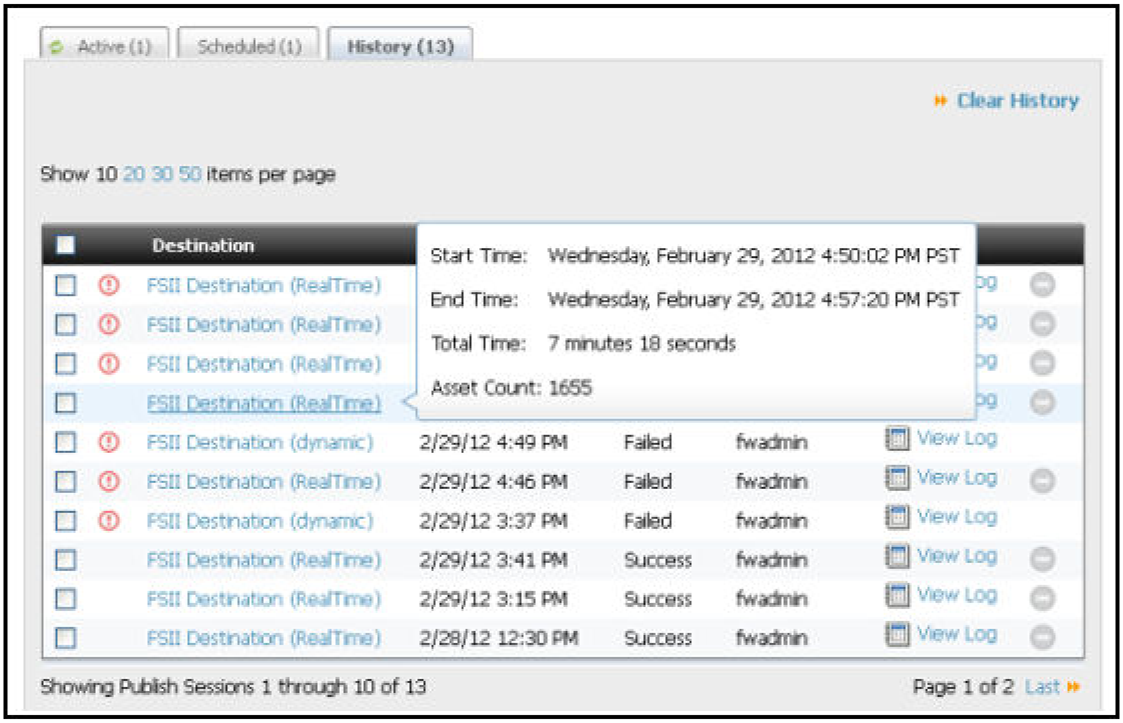
To view past publishing sessions:

1. Open the Publish Console (in the Admin interface, click **Publishing** in the menu bar).
2. Click the **History** tab.
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 in Figure 15-9.

   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 15-9 History Tab on Publish Console Form, Showing Context Window



Show 10 20 30 50 items per page

Start Time: Wednesday, February 29, 2012 4:50:02 PM PST
End Time: Wednesday, February 29, 2012 4:57:20 PM PST
Total Time: 7 minutes 18 seconds
Asset Count: 1655

Destination				
FSII Destination (RealTime)				og
FSII Destination (RealTime)				og
FSII Destination (RealTime)				og
FSII Destination (RealTime)				og
FSII Destination (dynamic)	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

Showing Publish Sessions 1 through 10 of 13

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.

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.

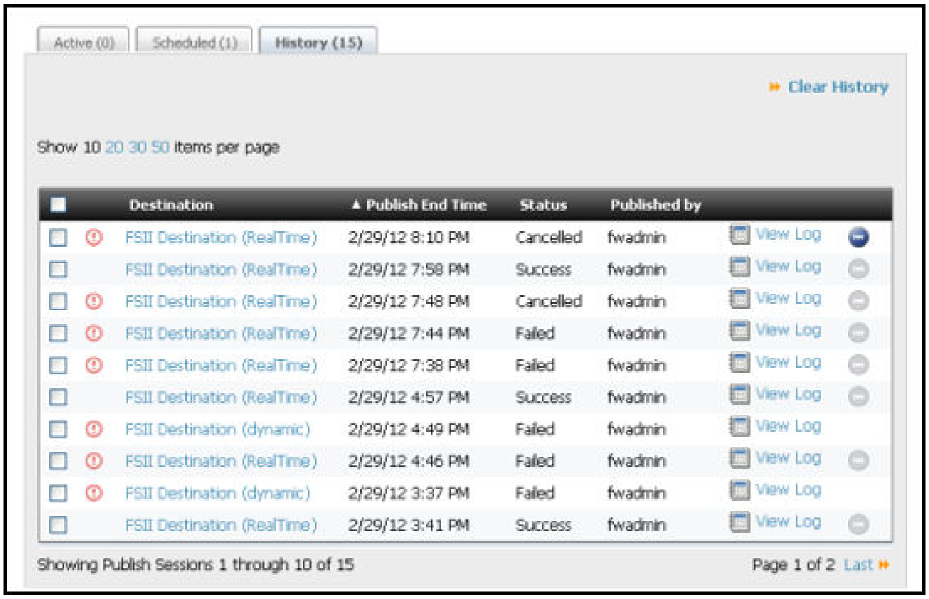
To redo a publishing session:

1. Open the Publish Console (in the Admin interface, click **Publishing** on the menu bar).
2. Click the **History** tab to view past publishing sessions.

The list of past publishing sessions opens. 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. In the following figure, **Redo** button is visible at the end of the line detailing the first destination. Click the **Redo** button to begin a publishing session at the point of sending data to the target.

Figure 15-10 History Tab, showing Redo Button



Show 10 20 30 50 items per page

□		Destination	▲ Publish End Time	Status	Published by		
□	!	FSII Destination (RealTime)	2/29/12 8:10 PM	Cancelled	fwadmin	View Log	●
□		FSII Destination (RealTime)	2/29/12 7:58 PM	Success	fwadmin	View Log	○
□	!	FSII Destination (RealTime)	2/29/12 7:48 PM	Cancelled	fwadmin	View Log	○
□	!	FSII Destination (RealTime)	2/29/12 7:44 PM	Failed	fwadmin	View Log	○
□	!	FSII Destination (RealTime)	2/29/12 7:38 PM	Failed	fwadmin	View Log	○
□		FSII Destination (RealTime)	2/29/12 4:57 PM	Success	fwadmin	View Log	○
□	!	FSII Destination (dynamic)	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	○

Showing Publish Sessions 1 through 10 of 15

Page 1 of 2 Last

3. Complete either:
   - 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 opens 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. (See Working with Active Publishing Sessions.)

Working with Publishing Logs

The following topics provide information and instructions about viewing, downloading, and deleting publishing logs:

- Viewing and Searching Logs
- Downloading Logs

- Deleting Publish History

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 required (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 Creating a RealTime Destination Definition on the Source System.

To view publishing logs:

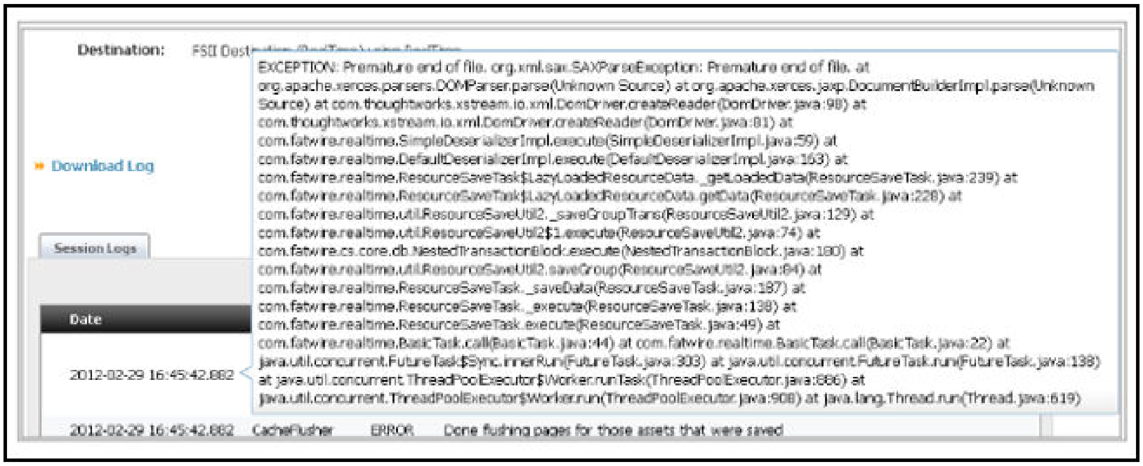
1. Open the Publish Console (in the Admin interface, click **Publishing** on the menu bar).

   The Publish Console opens with the **Active** tab open, displaying all currently running RealTime sessions.

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.
   - If a log entry exceeds 200 characters, an ellipsis (three periods) will appear at the end. To see the entire entry, hover the mouse cursor 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 15-11   Pop-Over Window Showing Information for the Log Event



```
EXCEPTION: Premature end of file. org.xml.sax.SAXParseException: Premature end of file. at
org.apache.xerces.parsers.DOMParser.parse(Unknown Source) at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown
Source) at com.thoughtworks.xstream.io.xml.DomDriver.createReader(DomDriver.java:98) at
com.thoughtworks.xstream.io.xml.DomDriver.createReader(DomDriver.java:81) at
com.fatwire.realtime.SimpleDeserializerImpl.execute(SimpleDeserializerImpl.java:59) at
com.fatwire.realtime.DefaultDeserializerImpl.execute(DefaultDeserializerImpl.java:163) at
com.fatwire.realtime.ResourceSaveTask$LazyLoadedResourceData._getLoadedData(ResourceSaveTask.java:239) at
com.fatwire.realtime.ResourceSaveTask$LazyLoadedResourceData.getData(ResourceSaveTask.java:228) at
com.fatwire.realtime.util.ResourceSaveUtil2._saveGroupTrans(ResourceSaveUtil2.java:129) at
com.fatwire.realtime.util.ResourceSaveUtil2$1.execute(ResourceSaveUtil2.java:74) at
com.fatwire.cs.core.db.NestedTransactionBlock.execute(NestedTransactionBlock.java:180) at
com.fatwire.realtime.util.ResourceSaveUtil2.saveGroup(ResourceSaveUtil2.java:84) at
com.fatwire.realtime.ResourceSaveTask._saveData(ResourceSaveTask.java:187) at
com.fatwire.realtime.ResourceSaveTask._execute(ResourceSaveTask.java:138) at
com.fatwire.realtime.ResourceSaveTask.execute(ResourceSaveTask.java:49) at
com.fatwire.realtime.BasicTask.call(BasicTask.java:44) at com.fatwire.realtime.BasicTask.call(BasicTask.java:22) at
java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303) at java.util.concurrent.FutureTask.run(FutureTask.java:138)
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886) at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908) at java.lang.Thread.run(Thread.java:619)
```

```
2012-02-29 16:45:42.882
2012-02-29 16:45:42.882   CacheFlusher   ERROR   Done flushing pages for those assets that were saved
```

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).

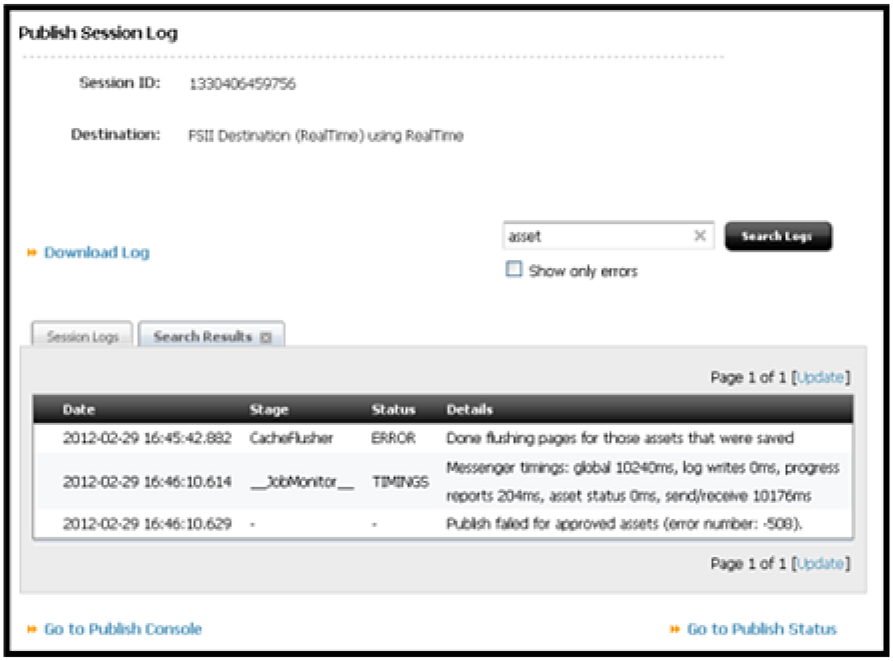
     > **Note:**
     >
     > The search utility recognizes terms that are used in the following columns: Stage, Status, and Details. Wildcards and Boolean operators are not recognized.

   - To view only error messages, select the **Show only errors** check box and leave the **Search** field empty.
   - To filter error messages by specific terms, select the **Show only errors** check box and enter a search term in the **Search** field.

   a. Click **Search**.

      The **Search Results** tab opens (next to the **Session Logs** tab).

Figure 15-12   Search Results Tab on the Publish Session Log Form



Publish Session Log

Session ID: 1330406459756

Destination: FSII Destination (RealTime) using RealTime

Page 1 of 1 [Update]

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).

Page 1 of 1 [Update]

b. To exit the **Search Results** tab, click the **X** on the tab.

   You are returned to the **Session Logs** tab.

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 following steps:

1. Open the Publish Console (in the Admin interface, click **Publishing** on the menu bar).

   The Publish Console opens with the **Active** tab at the front. The **Active** tab lists publishing sessions currently in progress.

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.

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.

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 do 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:

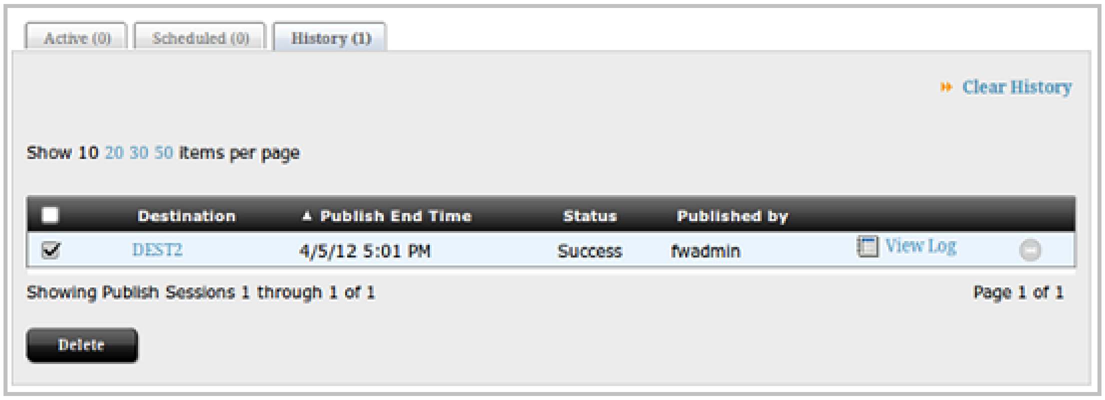
1. Open the Publish Console (in the Admin interface, click **Publishing** on the menu bar).

   The Publish Console opens with the **Active** tab at the front.

2. Select the **History** tab.

   All available logs are displayed.

Figure 15-13 History Tab on Publish Console Form



Active (0)
Scheduled (0)
History (1)
Clear History
Show 10 20 30 50 items per page

☐	Destination	▲ Publish End Time	Status	Published by		
☑	DEST2	4/5/12 5:01 PM	Success	fwadmin	View Log	⊖

Showing Publish Sessions 1 through 1 of 1
Page 1 of 1
Delete

3. Select the box next to each log or logs you want to delete.
4. Click **Delete**.

   The selected logs are deleted.

> **Note:**
>
> As data is 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.

16
Scheduling Publish Events

You can set up publishing events for all of your different publishing destinations. You can schedule a publishing event that repeats in an hourly or weekly manner. 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.

- Scheduling a Publish Event
- Reading the Schedule Abbreviations
- Editing Publish Events
- Overriding the Schedule
- Assigning Approval or Preview Templates
- Monitoring Sessions in the Publishing Console
- Verifying Publishing Readiness
- About Publish History
- About Session History
- About Publishing All Approved Assets
- Troubleshooting

Scheduling a Publish Event

When you schedule a publish event, there are two things you should be aware of:

- 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 **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
2. Double-click the destination to schedule a publish event for.
3. In the Publish Destination form, click the **Set Publish Event** button.

   The Edit Publish Event form opens.

Now you select days, months, hours, and minutes to set a schedule.

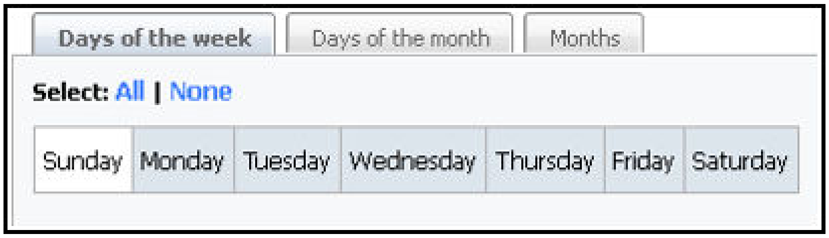
The form for setting a publish event is similar to the form for setting a timed action event in workflow. See Setting Up the Timed Action Event.

Example schedule:

For example, 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 **Monday** and drag to **Saturday** to select all days of the week except Sunday. The days highlight in gray as you select them.

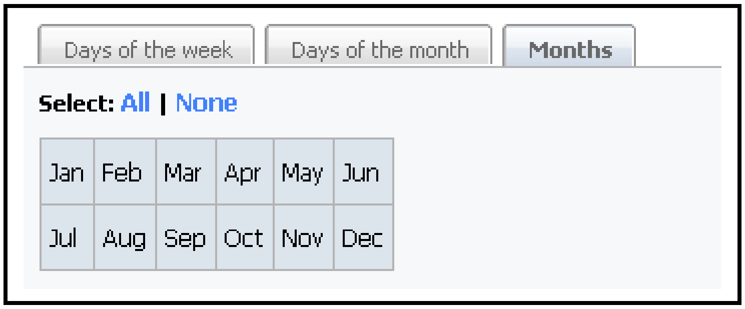
Figure 16-1 Days of the Week Tab



Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

2. Skip the **Days of the month** tab, since you have selected the days you want.
3. In the **Months** tab, click **All** to select all months. All months become highlighted in gray.

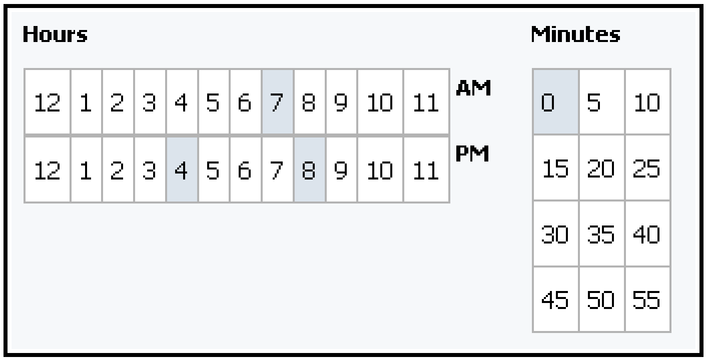
Figure 16-2 Months Tab



Jan	Feb	Mar	Apr	May	Jun
Jul	Aug	Sep	Oct	Nov	Dec

4. In the Times of Recurrence section, **Hours** selector, click **7am**, **4pm,** and **8pm**.
5. In the **Minutes** selector, click **0**.

Figure 16-3 Hours and Minutes Section



Hours

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

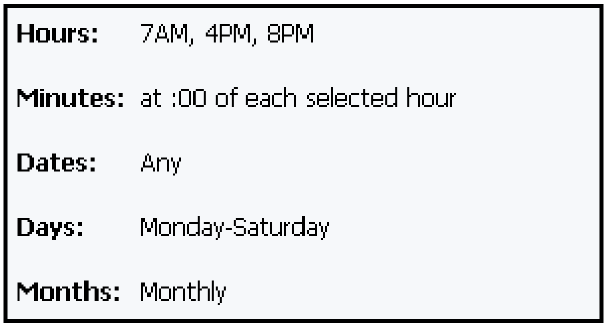
Minutes

0	5	10
15	20	25
30	35	40
45	50	55

6. Ensure the dates and times selected in the Enabled Dates and Times section of the form.

   It should look as follows:

Figure 16-4 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 opens in the Publish Event section of the Publish Destination form.

Reading the Schedule Abbreviations

When scheduled events are created, these events are displayed in string format in the WebCenter Sites interface.

In the Edit Publish Events form the schedule opens 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 opens 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 opens 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 example schedule (in Scheduling a Publish Event), 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 multiple 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 is in the field in that case.
  - If multiple days are 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 is 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/*/
- After 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 is 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/*/*

Editing Publish Events

Publishing schedules can be edited as necessary. This helps keep the number of schedules low, revising a schedule rather than incorporating newer schedules. To edit a publish event:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
2. Double-click the destination to schedule a publish event for.
3. In the Publish Destination form, click **Set Publish Event**.

   The Publish Event form is displayed. The text at the top of the form describes how the event is currently configured.
4. Edit the event as required and click **Schedule**. To clear your selections, click **Cancel**.

Overriding the Schedule

Schedule overrides are used when it is necessary to immediately publish a site. To start a publishing session immediately:

1. In the Admin interface, click the **Publishing** button in the toolbar.
2. In the Publishing Console, from the list, select the destination 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.

   If there are assets that can be published to the destination, the **Publish** button opens. If there are no assets to publish, there is no **Publish** button.
4. Click **Publish**.

   The publishing session starts.

> **Note:**
>
> If a publishing session for this destination is 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.

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 users approve article assets for the site, they should use the template that contains the most representative set of dependencies for all of the templates to determine the dependencies for the article. If the template that contains the most representative set of dependencies is not the template to assign to the asset, it should be set 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 **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.

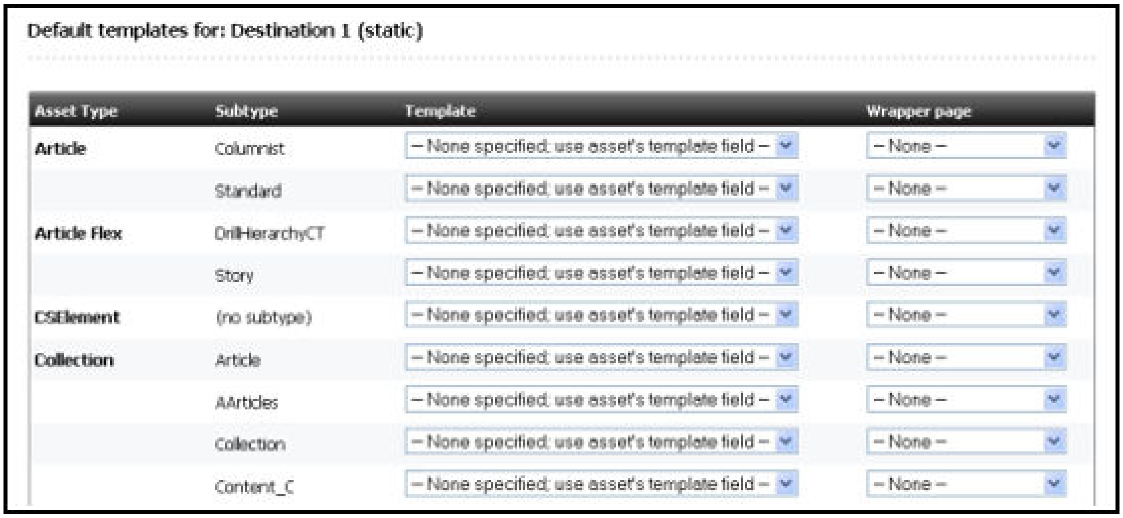
2. Expand the destination to configure default templates for and then select the **Set Default Templates** item.

   The Default Templates form is displayed.

3. In the Default Templates form, click **Edit Configuration**.

   The system displays a form similar to the following figure:

Figure 16-5 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 –

4. For each asset type that you have to configure, select the template and wrapper page for 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.

5. Click the **Save** button.

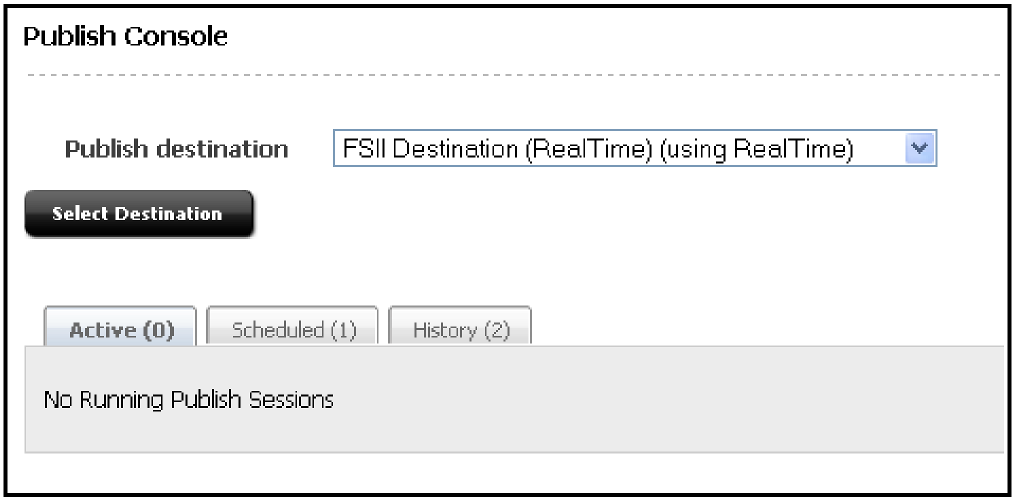
   You are returned to the Default Templates form for that destination.

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 interface. The Publish Console form will open.

The console displays a summary of all the publishing activity, including any sessions that are currently running, scheduled, or completed.

Figure 16-6 Publish Console Form



Publish Console

Publish destination: FSII Destination (RealTime) (using RealTime)

No Running Publish Sessions

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 list.
2. Click the **Select Destination** button.

   A summary form opens. It lists any assets that have to be published but are not ready (they are held) and assets that can be published.
3. Click the link for the number of assets help for publish or the number of assets ready for publish to see the details.

   A list of assets that are either approved or approved and still held for dependency reasons is displayed. On one tab is the list of Approved Assets, and the other tab are the assets that are in the On-Demand Queue.

   If you have 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.
- See Table 17-4.
- For procedures that describe how to work with assets, see Working with Assets in *Using Oracle WebCenter Sites*.

About Publish History

Publishing sessions that are completed are listed in the Publish History section of the Publish Console, which opens 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), which 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.

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 Troubleshooting.

About Session History

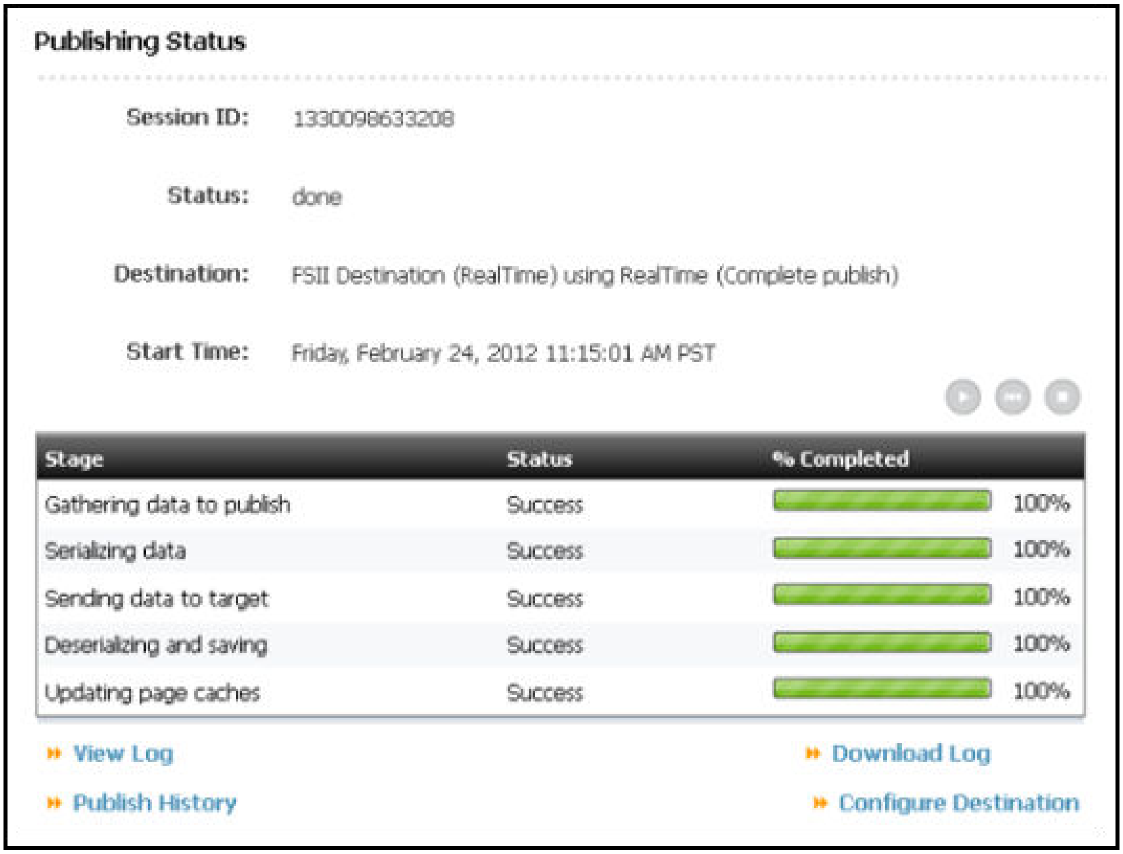
The publishing history log files are displayed in the **History** tab of the Publishing Console as summaries of publishing sessions.

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 16-7 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	100%
Serializing data	Success	100%
Sending data to target	Success	100%
Deserializing and saving	Success	100%
Updating page caches	Success	100%

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 opens as the history instead. If you see an error message in the history for a publishing session, see Troubleshooting and attempt to resolve the problem.

About 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 have 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 been published and have not changed since they were published.

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:

- Understanding Publishing System Error Messages
- Understanding Numeric Messages
- Indicating Other System or Configuration Issues

Understanding 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 opens.

  The `request.folder` property in the in the `wcs_properties.json` file, categorized under Publish, 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.

Understanding Numeric Messages

Table 16-1 provides descriptions and suggested corrective actions for the numeric error messages that the publishing system could report in any of the log files mentioned in the list above.

> **Note:**
>
> The table does not contain 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 *Tag Reference for Oracle WebCenter Sites*.

Table 16-1 Numeric Error Messages

Error Number	Description	Corrective Action
-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.	Correct the mirror user information. For help, see the procedure in Identifying the Mirror User to the Source System.
-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 requires.	Edit the mirror user's account and assign it the appropriate ACLs. See the procedure in Setting Up the Destination System for a list of the ACLs that the mirror user requires. 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.

Table 16-1 (Cont.) Numeric Error Messages

Error Number	Description	Corrective Action
-103	This error indicates that a table 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.	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. 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 Oracle WebCenter Sites Explorer.
-611	This error message is a generic mirror publishing error message (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.	Examine the `sites.log` file and application server logs on the destination system to look for additional messages.
-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.	Log in to the WebCenter Sites interface on the source, examine the definition of the destination, and determine that it is correct. For help, see Creating a Mirror Destination.
-12011	This error message is a generic message (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.	Examine the messages in the `PubMessage` table and the `sites.log` file on the source to look for additional error messages.
-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.	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).
-12045	This error indicates that the publishing system was able to start the session, but it then failed in some way.	Examine the `sites.log` file on the destination system and look for error and exception messages in the log.

Table 16-1 (Cont.) Numeric Error Messages

Error Number	Description	Corrective Action
-12046	This error indicates that there was a problem after 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.	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.
-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.	Examine the `sites.log` file on the destination and look for error exception messages and exceptions that describe the problem.
-13054	This error occurs for only complex and flex assets. 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.	Check the `sites.log` file on the source system. If the problem was the data itself, there are 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 is an additional message in the log.
-13055	This error occurs for only complex and flex assets. 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 multiple sources to the same destination, there can be ID collisions and other problems with data integrity that will cause this error.	Examine the `sites.log` file 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.

Indicating Other System or Configuration Issues

This section describes symptoms your WebCenter Sites system may exhibit when the publishing system discovers configuration errors or system problems.

This section covers the following topics:

- Publishing Session Does Not End and Displays an Odd Status
- The System Stops Altogether
- Pages Are Not Refreshed After the Publishing Session Ends
- DB2 Systems, Troubles When Publishing Assets with Associations

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 Creating 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.

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 must have enough disk space in your file system to store all the files that are being generated.
- For Mirror to Server publishing, you must have 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.

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 user names and passwords."

  This message indicates that there is something wrong with the values specified for the `cs.satellitehosts`, `cs.satelliteusers`, and `cs.satellitepassword` properties in the `wcs_properties.json` 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:

In the Property Management Tool, examine the `cs.Pastramiengine` and `satellite.blob.cachecontrol.default properties`, and note the values set for the user names, passwords, and host names. Then, note the values specified for the `cs.satellitehosts`, `cs.satelliteuser`, and `cs.satellitepassword` properties. Compare these values with the values set for the `username` and `password` properties in the `wcs_properties.json` file for remote satellite server (located under the `<WCSites_Shared_Dir>` for clustered environments and under `<WCSites_Product_Home>` for non-clustered environments). They must match.

Remember that the order in which you specify host names in the `wcs_properties.json` file must match the order in which you identify user names and passwords.

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:

Oracle 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.

Part VI

Managing Approval and Workflow

You need to protect the integrity of published content and verify if the assets are ready to be published. So, you should approve assets before you publish them. You should also manage workflow processes for assets to ensure that assets created by multiple contributors are routed to the appropriate contributor at the appropriate time.

The following topics provide information about managing approval and workflow:

- Understanding the Approval System
- Creating and Managing Workflow Processes

17 Understanding the Approval System

You can publish only those assets that you’ve approved for publishing. Assets of enterprise sites have complex interdependencies, To prevent broken links, approve assets along with their dependant assets.
The following topics describe the basic workings of the 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.

- About the Approval System
- Understanding Dependency Analysis
- Understanding Publishing Terms and Definitions
- Using Rules for Dependency Analysis
- Understanding Approval Dependencies and Parent-Child Relationships
- Working with Export to Disk Publishing
- Working with Mirror to Server Publishing and Export to XML
- Ensuring Version-Matched Content
- Using Exists-Exact Dependencies in Mirror to Server and Export to XML Publishing
- Using Exists-Exact Dependencies in Export to Disk Publishing
- Working with Exists/Exact Dependency Rules
- Evaluating Published Content
- Approval System Example
- Using Approval States: Reference

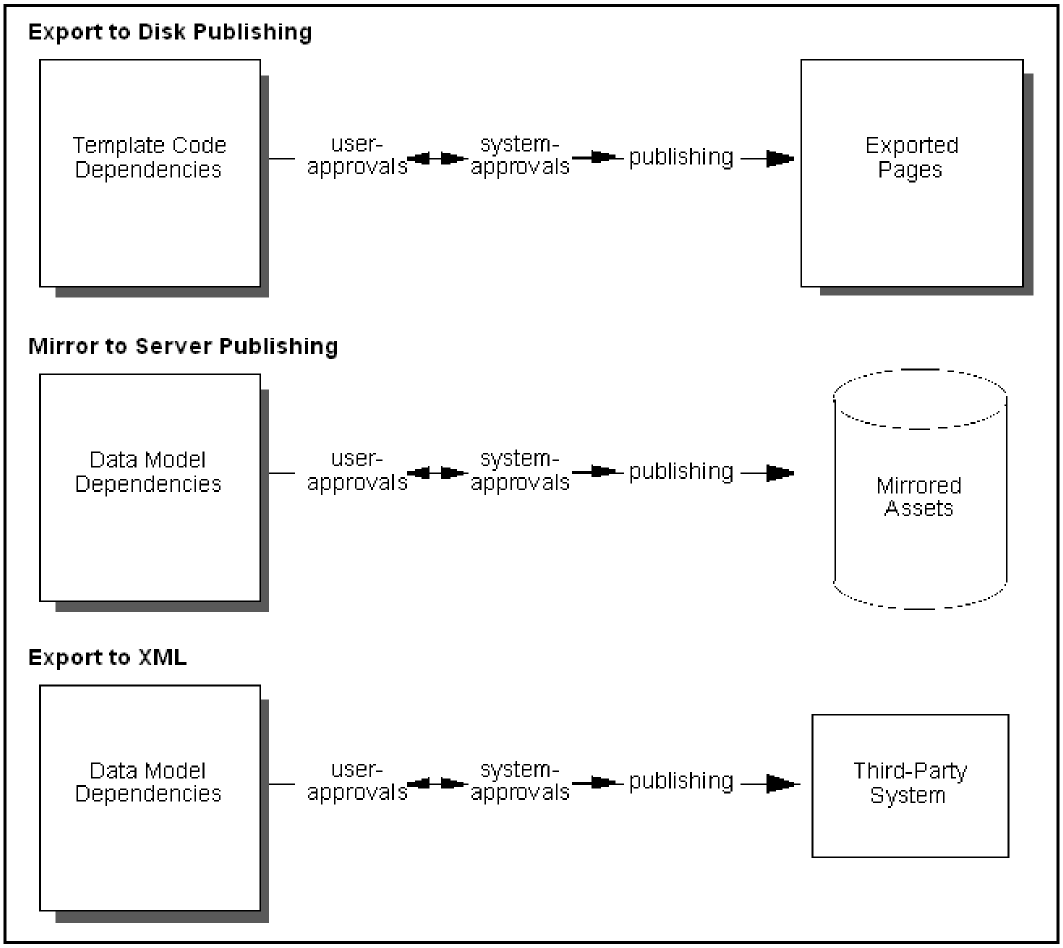
About the Approval System

Before a publishing session can begin, the user must approve assets for the publishing destination. This task is more intricate than it seems. Enterprise sites host vast 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 to be published. Unapproved assets compromise the integrity of 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. The approval system does not automatically allow the publication of user-approved assets. Instead, it evaluates the condition of the user-approved assets to determine if they are 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. The dependencies are then 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 17-1 Approval System in Publishing Methods



Understanding 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 17-2 and Figure 17-3:
   - In Export to Disk publishing, the approval system tests dependencies (among approved assets) against template code. 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. See Evaluating Approval Dependencies.

- 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 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 administrator. If so, user-approved assets must match the versions of published assets 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, categorized under Oracle WebCenter Sites: Engage, in the `wcs_properties.json` file. For Export to Disk publishing it is specified in the template code. See Ensuring Version-Matched Content.

Figure 17-2 Export to Disk Publishing Methods and Approvals

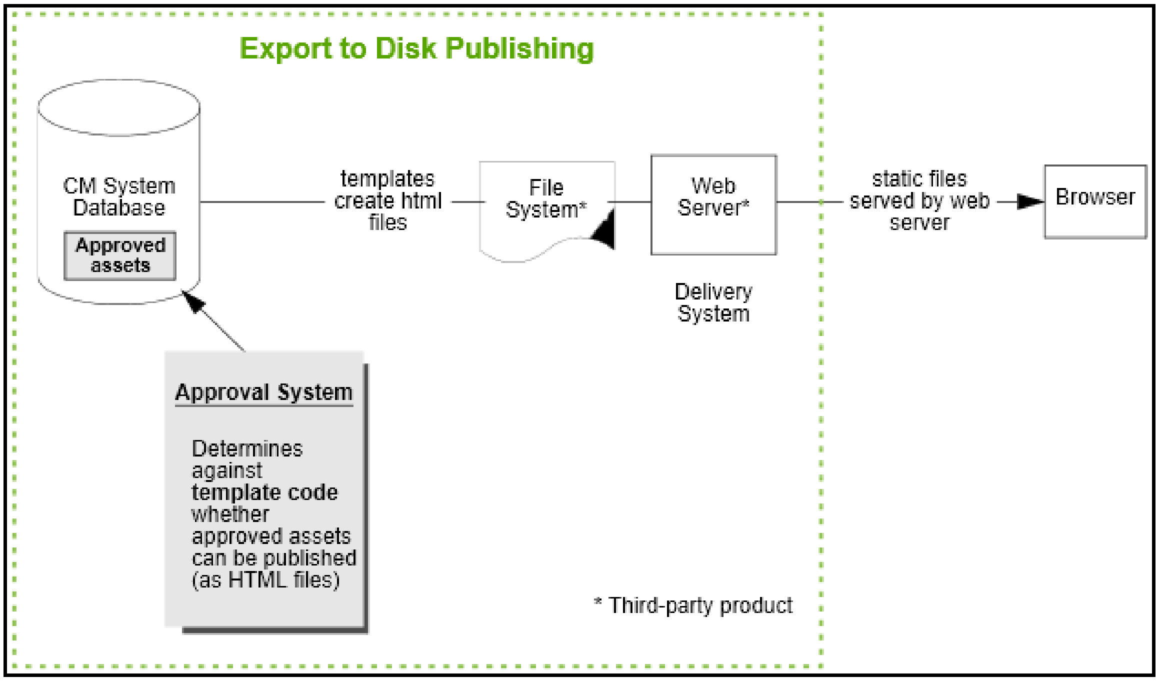
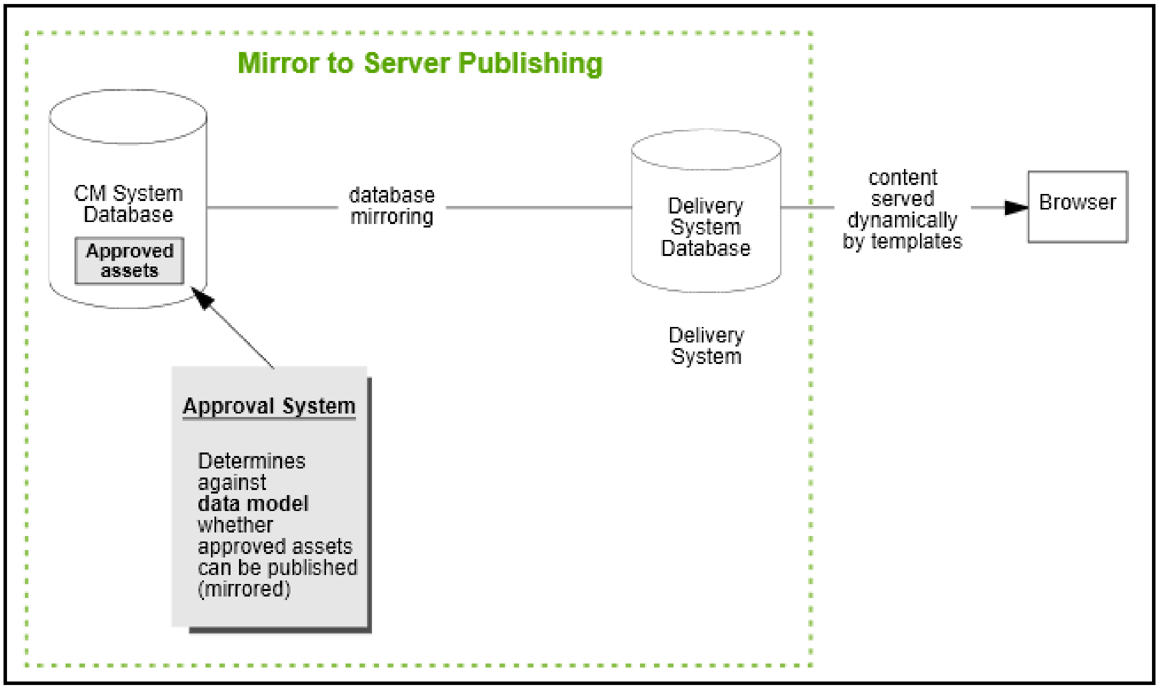

Export to Disk Publishing
CM System Database
Approved assets
templates create html files
File System*
Web Server*
Delivery System
static files served by web server
Browser
Approval System
Determines against template code whether approved assets can be published (as HTML files)
* Third-party product

Figure 17-3 Mirror to Server Publishing Methods and Approvals


Mirror to Server Publishing
CM System Database
Approved assets
database mirroring
Delivery System Database
Delivery System
content served dynamically by templates
Browser
Approval System
Determines against data model whether approved assets can be published (mirrored)

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. Thus dependencies among approved assets are 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 described and explained in Understanding Publishing Terms and Definitions, along with the equally important terms, "approvals" and "approval states."

Understanding Publishing 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, and the WebCenter Sites publishing interface.

This section covers the following topics:

- Understanding Approval: Intent to Publish vs. Permission to Publish
- Understanding Approval Dependencies
- Working with Approval States

Understanding 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.

  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 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 treats the approval as tentative and requiring verification. Only when the approval system completes the dependency test and confirms that the Approved state is valid does the system approve (release) the assets for publication.

If 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 conditional on the user also approving the supporting assets. Conditional approvals are caused by approval dependencies.

- 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 changed after it is approved for publication, the approval queue handles this change by unapproving the asset, rejecting it from the publishing queue.

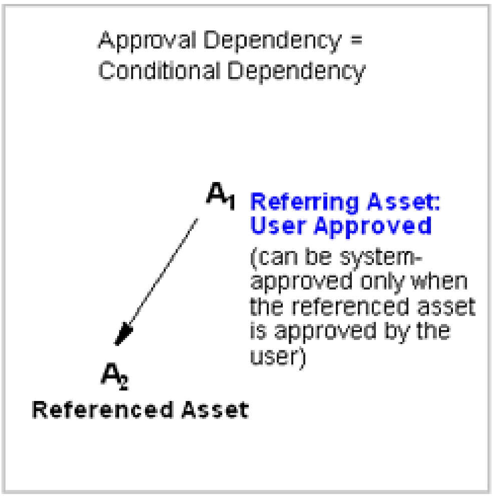
  The approval queue is not accessible to the user. It runs, by default, every 5 minutes. However, when you call a feature that uses approval functionality (such as the **Publishing** tab), the approval queue runs before the form is shown to keep approval information up to date.

Understanding Approval Dependencies

Before reading this topic, make sure you understand the distinction between the terms user-approval and system-approval. See Understanding Approval: Intent to Publish vs. Permission to Publish.

An approval dependency is a conditional dependency and results in a conditional approval:

Figure 17-4 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. 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.

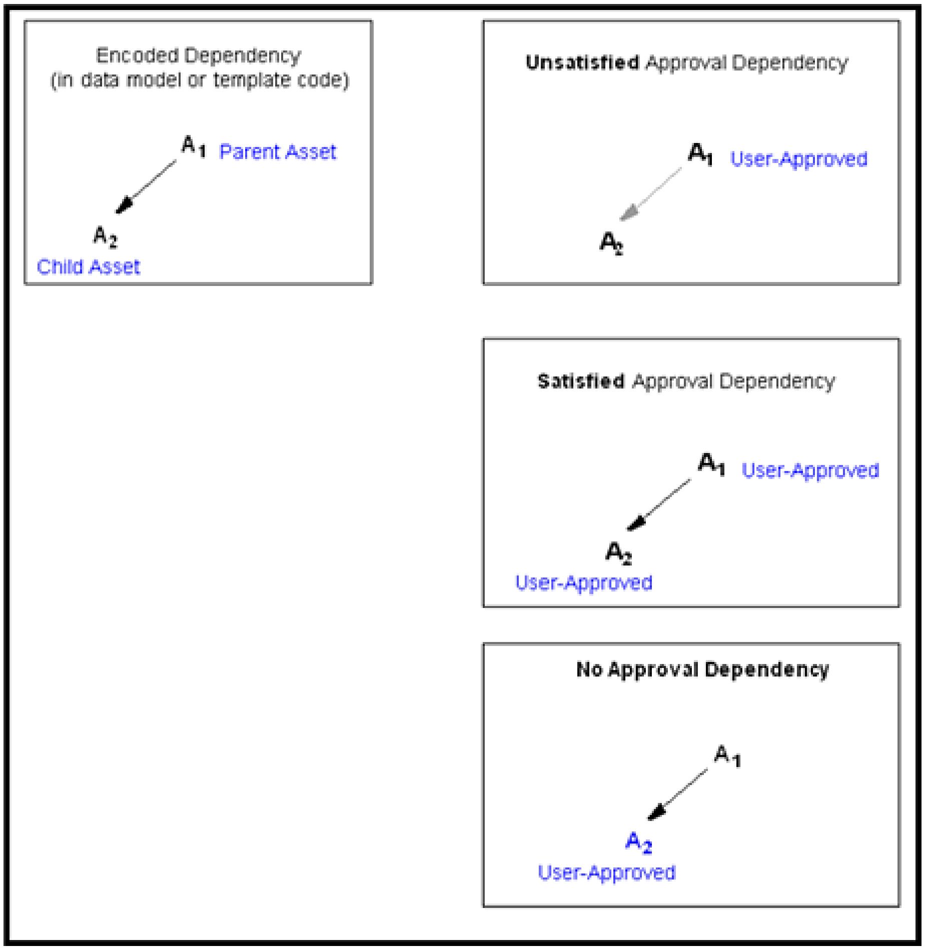
> **Note:**
>
> From this point forward, we define "referring asset" to be the parent asset. The "referenced asset" is the child asset. For simplicity, we also assume that parent assets are solely parents, and child assets are solely child assets.

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. See Examples of Approval Dependencies.

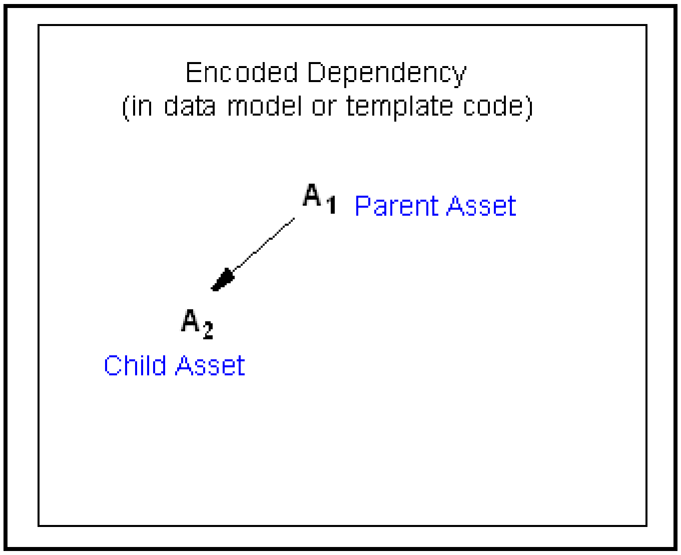
Figure 17-5 Encoded Dependencies and Approval Dependencies



Examples of Approval Dependencies

This example supplements the previous section by illustrating in greater depth how approval dependencies are determined.

Figure 17-6 Encoded Dependencies

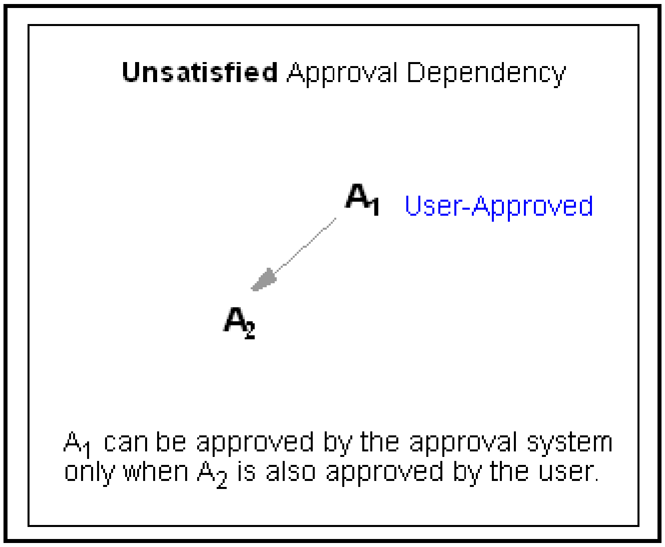


The preceding figure illustrates a two-asset data model. Asset A1 refers to asset A2 for content. While the data model specifies one dependency, the user has three different ways to approve the assets and, therefore, create different approval dependencies:

- **Unsatisfied Approval Dependency:**

  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 requires approval.

  Figure 17-7 Unsatisfied Dependencies

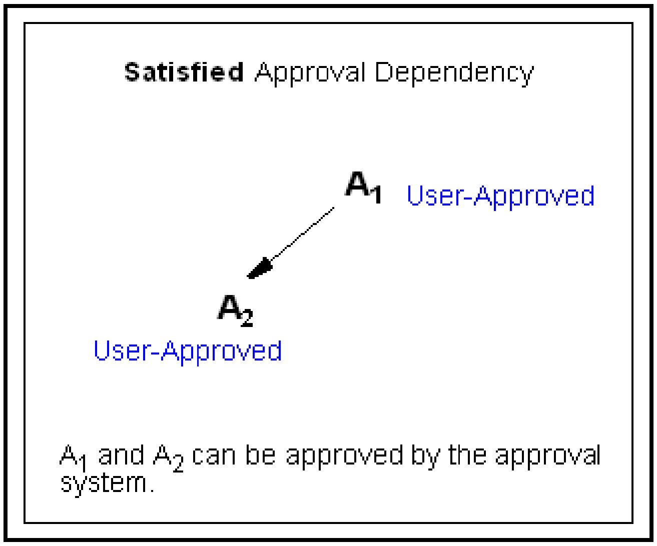
  
  Unsatisfied Approval Dependency
A1 User-Approved
A2
A1 can be approved by the approval system only when A2 is also approved by the user.

The preceding figure illustrates the system's approval of A1 as 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 17-8 Approval Dependencies

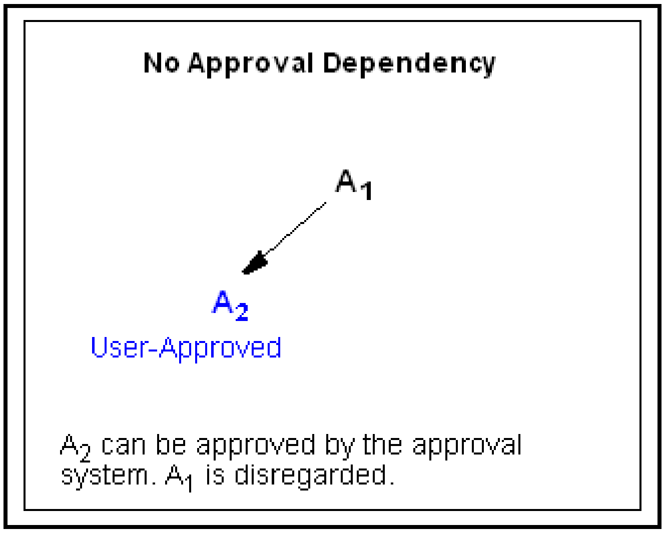
  
  Satisfied Approval Dependency
A_1
User-Approved
A_2
User-Approved
A_1 and A_2 can be approved by the approval system.

  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 is 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 is published (but A1 will not).

Figure 17-9 No Approval Dependencies


No Approval Dependency
A1
A2
User-Approved
A2 can be approved by the approval system. A1 is disregarded.

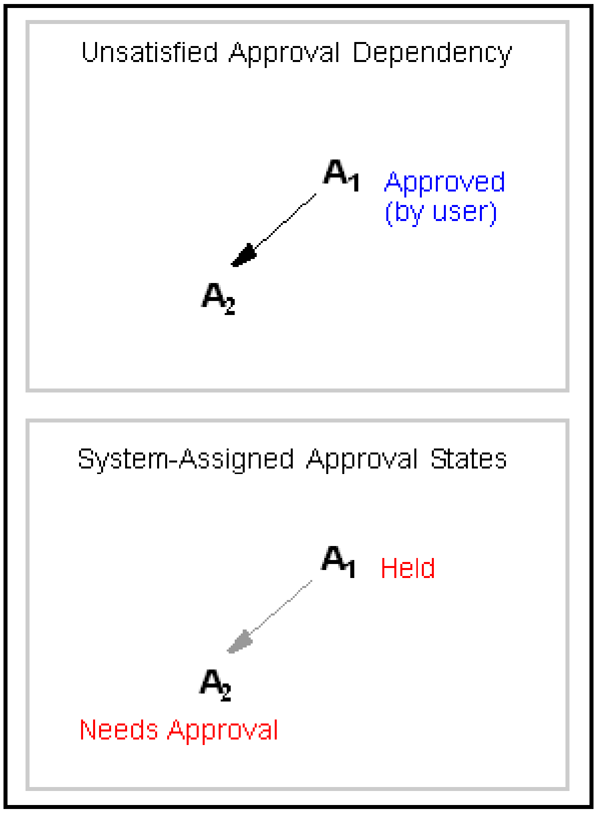
Notes About Approval Dependencies

- 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. See Using Rules for Dependency Analysis.
- 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 because the combination of assets approved varies from one approval attempt to another.

Working with 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 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 17-4.

Figure 17-10 Approved and Unapproved Approval States

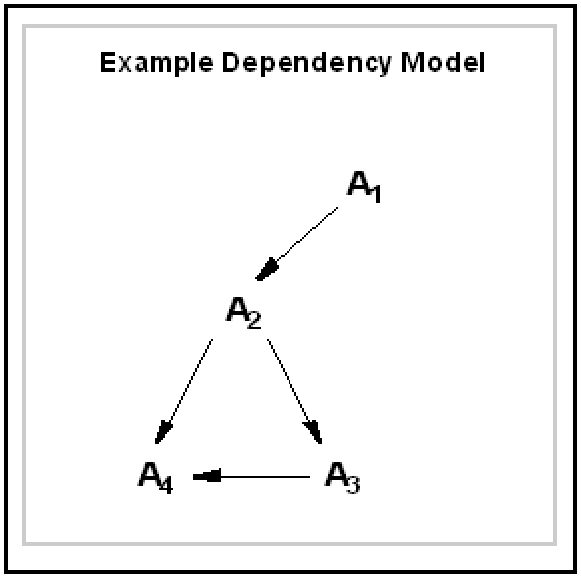

Unsatisfied Approval Dependency
A1
Approved
(by user)
A2
System-Assigned Approval States
A1
Held
A2
Needs Approval

In Figure 17-10, the user marked asset A1 as Approved. However, the system 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.

The list of approval states can (and typically does) vary from one approval attempt to another. This is because the combination of assets approved by the user usually varies from one approval attempt to another. The following figure illustrates a simple dependency model and three approval scenarios, all involving one and the same user.

Figure 17-11 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

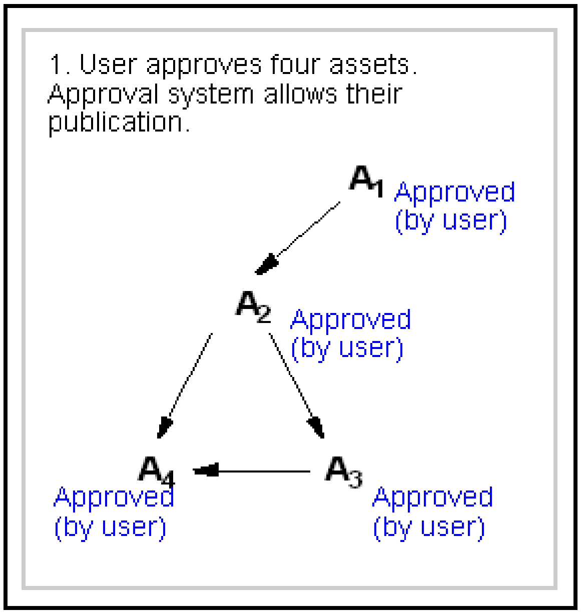
A3-to-A4

(No 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 one time. 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. This is illustrated in the following figure.

Figure 17-12 Approval System - All Approved



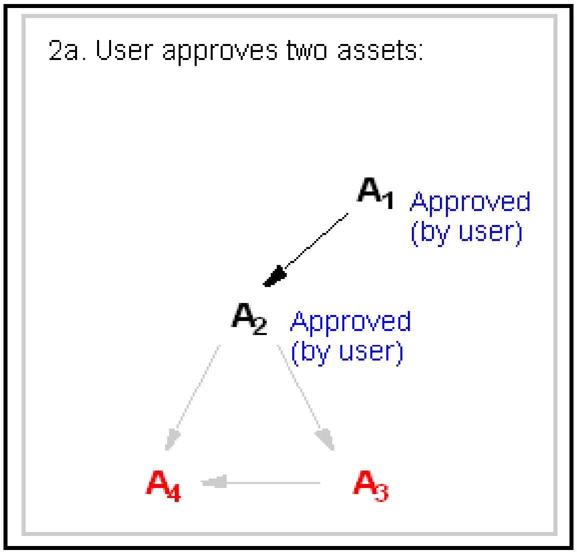
2. In the second scenario (illustrated through the following figure), the user approves two assets—A1 and A2—which satisfies one of four approval dependencies. Three approval dependencies remain to be satisfied:

   A2-to-A3

   A2-to-A4

   A3-to-A4

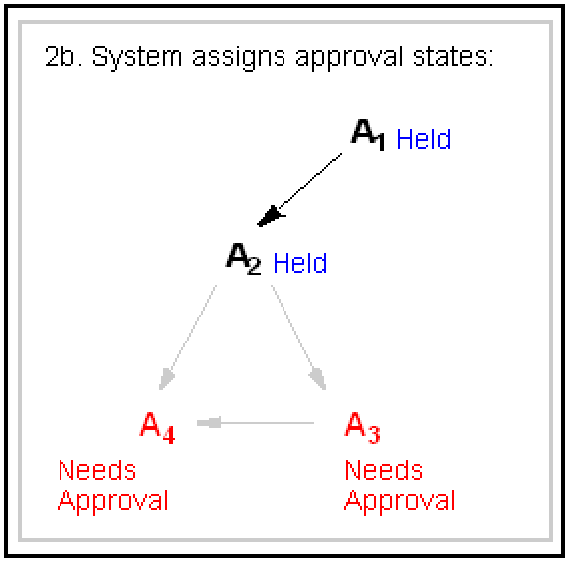
Figure 17-13 Dependency Model



3. The following figure shows that the approval system logs three unsatisfied approval dependencies, determines that no 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 17-14   Assets in Held State Waiting for Approval



4. In the third scenario (illustrated through the following figure), the user also approves two assets—A1 and A3—but this time, satisfies no approval dependencies:

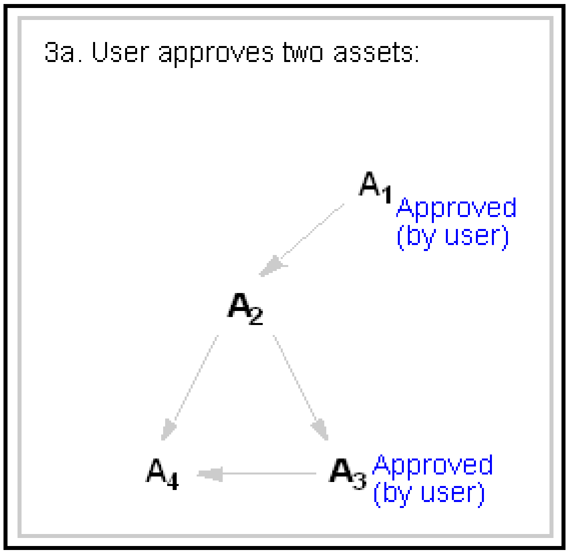
   A1-to-A2

   A2-to-A3

   A2-to-A4

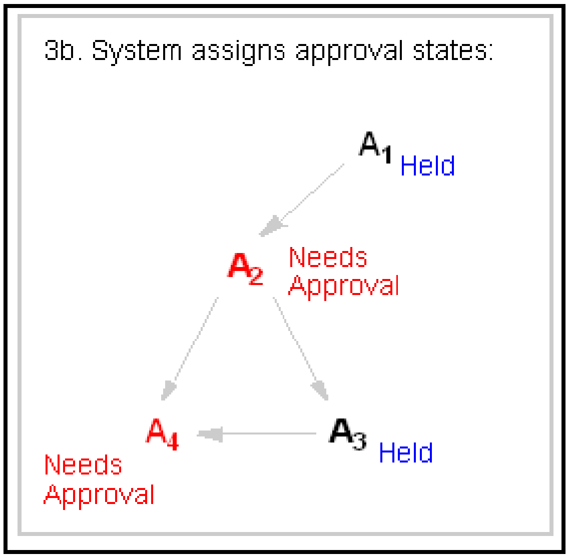
   A3-to-A4

Figure 17-15   Partial Approval State



5. The approval system logs four unsatisfied approval dependencies, and determines that no assets are ready for publication (illustrated through the following figure). The approval system returns a list of assets and approval states different from the list in scenario 2.

Figure 17-16 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.
>
> See Understanding Approval Dependencies.

Using 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:

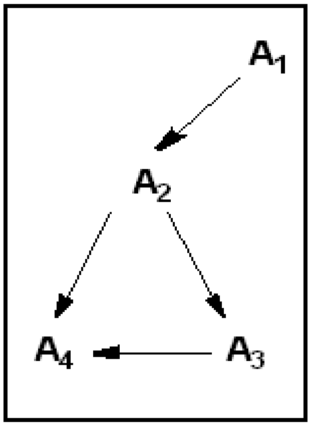
- The approval system simplifies all approval dependencies by treating them as parent-child relationships. See Understanding Approval Dependencies and Parent-Child Relationships.
- In Export to Disk publishing, the approval system tests approval dependencies against template code. See Working with Export to Disk Publishing.
- In Mirror to Server publishing and Export to XML, the approval system tests approval dependencies against the data model. See Working with Mirror to Server Publishing and Export to XML.
- If version matching must be controlled, the administrator can set an exists, exact, or none dependency to impose or cancel version matching. See 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. See Evaluating Published Content.

Understanding 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. The following figure 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 17-17 Parent-Child Relationships

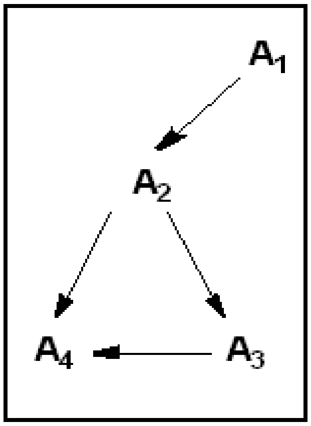


As with the previous illustrations, A1 is a parent asset, A2 is a child asset of A1 and a parent asset of both A3 and A4, A3 is a child asset of A2 and a parent asset of A4, and A4 is a child asset of A2 and A3.

- 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 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. See Ensuring Version-Matched Content.
- The approval system approves child assets for publication independently of their parent assets. On the live site, the link from parent to child does not appear to be broken; it simply does not appear. (In this respect, parent data can be inadvertently dropped from the live site.)

The following figure shows 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 17-18 Approval Dependencies



- For A1 to be system-approved, A2, A3 and A4 must exist as approved assets (or published assets on the destination).
- For A2 to be system-approved, A3 and A4 must exist as approved assets (or published assets on the destination).
- For A3 to be system-approved, A4 must exist as an approved asset (or published asset on the destination).
- A4 can be published alone.

If an element has a child asset added to it, this causes the parent asset to be considered modified. In this instance, if the parent asset was previously approved, adding a child asset means the parent page is no longer approved. This can hold publishing of changes using the approved parent asset.

Working with 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. Export to Disk publishing is described in Learning Export to Disk Publishing Terminology.

This section covers the following topics:

- Evaluating Approval Dependencies
- Understanding Test Rendering

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 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 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 link 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 in Learning Export to Disk Publishing Terminology.

Understanding Test Rendering

When an asset is user-approved to an 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 where 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 because the Export to Disk publishing method actually publishes an asset, but 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 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 Assigning Approval or Preview Templates.

Working with 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. See Understanding Test Rendering and Working with Compositional Dependencies.

Ensuring Version-Matched Content

Any publishing session (such as the one in Figure 17-18) can be made more stringent by the administrator requiring parent assets to match child assets in terms of version. This requirement will ensure self-consistent content. Dependency on version is not encoded in either the data model or the template code:

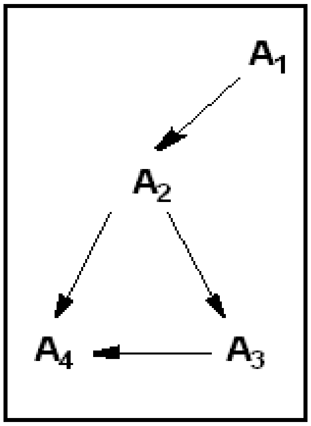
- For Mirror to Server and Export to XML publishing, it is specified in a property—`mwb.conservativedependencies` (in the `wcs_properties.json` file)—by setting the value of the property to `exists` or `exact`.
- For Export to Disk publishing, it is specified by setting the `deptype` attribute in the relevant tags to `exists`, `exact`, or `none` (the tags are listed in 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 must 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.

The following figure summarizes and illustrates exists and exact dependencies.

Figure 17-19   Exists/Exact Dependencies



Exists dependencies do not require version matching:

- For A1 to be system-approved, A2, A3 and A4 must exist as approved assets (or published assets on the destination).
- For A2 to be system-approved, A3 and A4 must exist as approved assets (or published assets on the destination).
- For A3 to be system-approved, A4 must exist as an approved asset (or published asset on the destination).
- A4 can be published alone.

Exact dependencies require version matching:

- System approval of A1 is the same as an exists dependency, but the version of A1 must match the version of its child asset, A2.
- System approval of A2 is the same as an exists dependency, but the version of A2 must match the version of its child assets A3 and A4.
- System approval of A3 is the same as an exists dependency, but the version of A3 must match the version of its child asset A4.
- A4 can be published alone.

Using 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 `wcs_properties.json` 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.

Using 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 Understanding 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 one time, you are not required to approve it again 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.

Working with 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 Understanding Approval Dependencies and Parent-Child Relationships.

This section covers the following topics:

- Understanding Dependency Rules for Basic Asset Types
- Understanding Dependency Rules for CSElement and SiteEntry Assets
- Understanding Dependency Rules for Flex Families
- Understanding Dependency Rules for Engage Visitor Data Assets

Understanding Dependency Rules for Basic Asset Types

For basic assets, the approval system follows the dependency rules described in the following table:.

Table 17-1 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
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. See Basic Assets That Can Have Embedded Links in *Developing with Oracle WebCenter Sites*.

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 TEXTAREA in *Developing with Oracle WebCenter Sites*.

Understanding Dependency Rules for 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.

Understanding Dependency Rules for Flex Families

For flex family members, the approval system follows the dependency rules described in the following table:

Table 17-2 Flex Family Dependency Rules

Approved Asset	Flex Parent Definition	Flex Parent	Flex Definition	Flex Attributes	Flex Filter	Attribute Editor	Template
Flex Parent Definition	exists	N/A	N/A	exists	N/A	N/A	N/A
Flex Parent	exists	exists	N/A	exists	N/A	N/A	N/A
Flex Definition	exists	N/A	N/A	exists	N/A	N/A	N/A
Flex Asset	N/A	exists	exists	exists	N/A	N/A	exists
Flex Attribute	N/A	N/A	N/A	N/A	exists (by default)	exists	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 Understanding Dependency Rules for Flex Families. For more information about the functions of the flex family members, see The Flex Family in *Developing with Oracle WebCenter Sites.*

Understanding Dependency Rules for Engage Visitor Data Assets

The approval system uses the following rules to determine approval dependencies for the visitor data asset types:

Table 17-3 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).

Evaluating Published Content

If the approval system determines the existence of an unsatisfied approval dependency, it searches for published content to avoid redundant approvals.

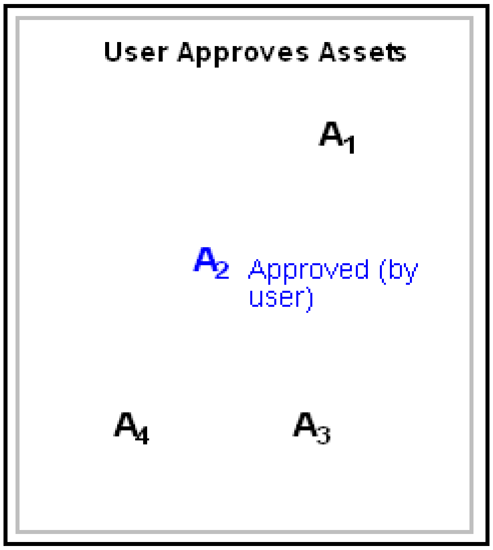
Specifically, the approval system reads the PublishedAssets table. If assets requiring approval are listed as published to the given destination, the approval system considers it unnecessary to approve and re-publish the same assets, assuming version-matching is not a requirement. If an exact dependency is specified in the data model or template code, the approval system checks the versions of the published assets. For more information about enforcing version-matched content, see Ensuring Version-Matched Content.

Approval System Example

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.

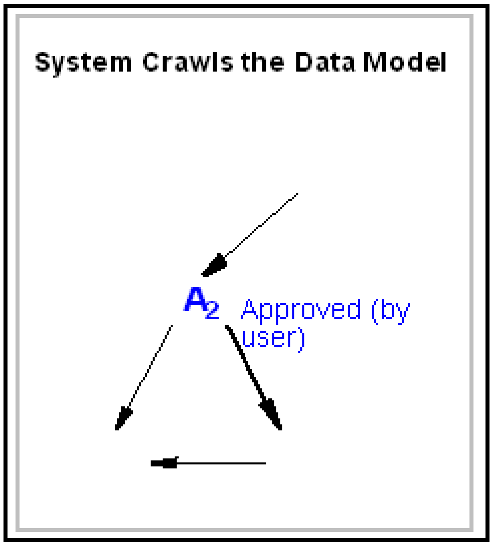
1. In this scenario (illustrated in the following figure), a user edits and approves a single asset, A2 (out of a possible four), to be published dynamically under an `exists` dependency. Only A4 has been previously published to the target destination.

Figure 17-20 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 following figure).

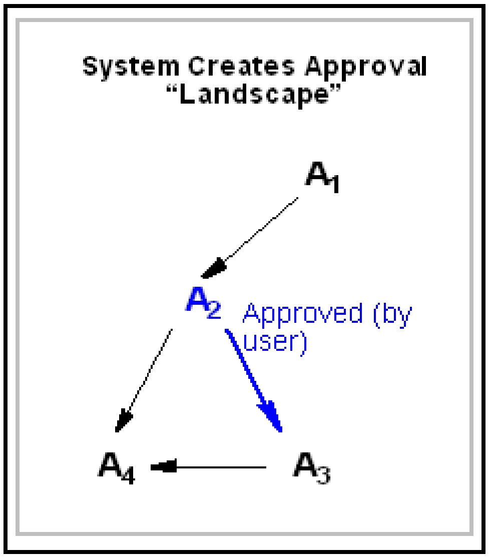
Figure 17-21 System Crawling Data Model



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 remaining assets' links to and from still 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.

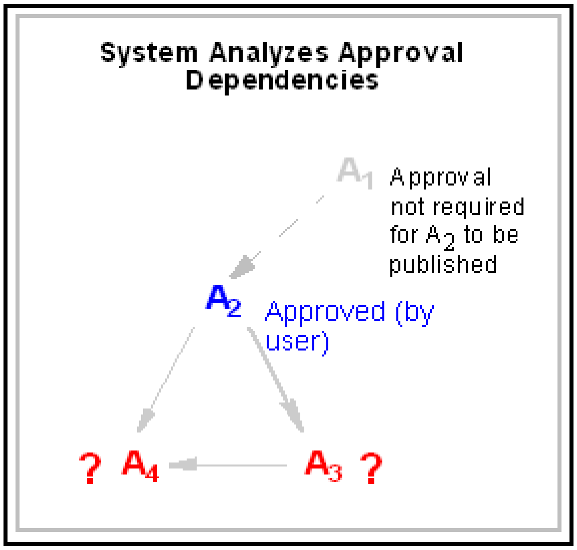
3. From the results of step 1, the approval system constructs an approval landscape, (illustrated in the following figure) which identifies all the interdependent assets and relationships among them.

Figure 17-22 System Creating Approval



4. The approval system tests for approval dependencies and their impact on the publishing session. The following figure illustrates the approval system determination process.

Figure 17-23 System Analyzing Approval Dependencies



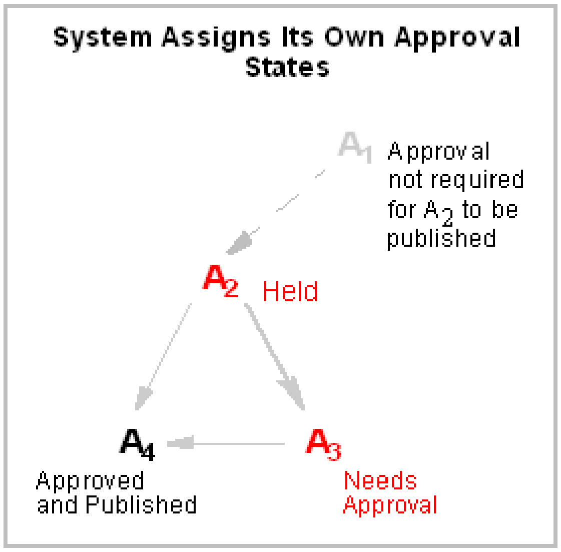
In this example, the approval system determines the following:

- 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 further 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 assigns its own approval states to the assets. The following figure illustrates the system assigning approval states.

Figure 17-24 System Assigning Approval States

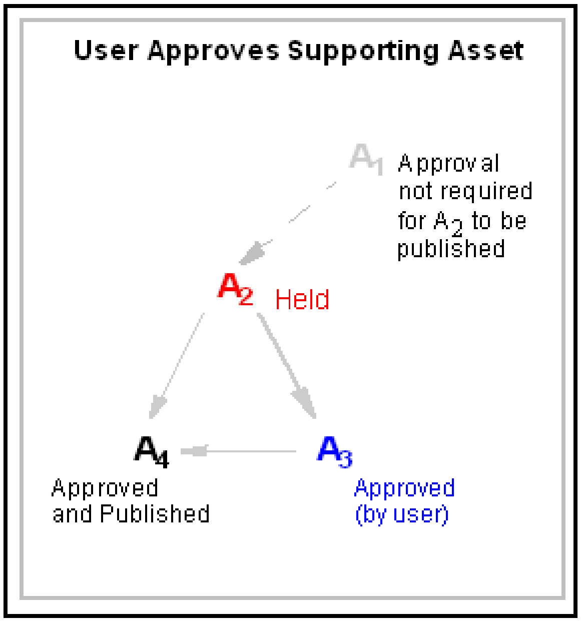

System Assigns Its Own Approval States
A_1
Approval not required for A_2 to be published
A_2
Held
A_4
Approved and Published
A_3
Needs Approval

- The A2-to-A3 approval dependency must be satisfied (by the user approving A3). The system assigns A3 the **Needs Approval** state and A2 the **Held** state.
- A3-to-A4 must be satisfied (by the user approving A3, which is 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 17-4.)

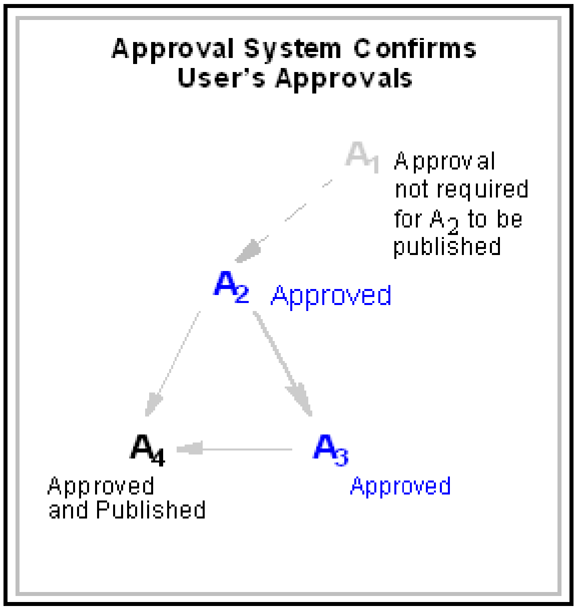
6. Finally, the approval system shows the list of assets and their approval states (shown in the preceding figure). In response, the user approves the asset that was assigned the **Needs Approval** state (A3).

Figure 17-25 User Approving Supporting Asset


User Approves Supporting Asset
A1 Approval not required for A2 to be published
A2 Held
A4 Approved and Published
A3 Approved (by user)

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 (illustrated in the following figure) for publication in the next publishing session.

Figure 17-26 System Confirmation of User Approval


Approval System Confirms User's Approvals
A1 Approval not required for A2 to be published
A2 Approved
A4 Approved and Published
A3 Approved

Using Approval States: Reference

The following table lists the possible approval states that can be assigned by the approval system to assets in an approval dependency and shown to the user.

Note:

In the following table, bear in mind that in the context of an approval dependency, the term "dependent asset" is equivalent to the "parent asset." See Understanding Approval Dependencies and Parent-Child Relationships.

Table 17-4 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. Not 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: • 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.
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 of the following conditions is true: • 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.

Table 17-4 (Cont.) Approval States Assigned by the Approval System

Approval State	Description
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.

18

Creating and Managing Workflow Processes

With the workflow feature of WebCenter Sites, you can route assets through whatever series of tasks you deem necessary in a workflow process from creation to approval. These tasks may be many or few as required. Through workflow management, you can manage an asset with multiple contributors efficiently.

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.

- About the Workflow Feature
- Planning Your Workflow Processes
- Configuring Your Workflow Processes
- Setting Up Email Objects
- Setting Up the Actions and Conditions
- Configuring Approve for Publish Step Actions
- Setting Up the Timed Action Event
- Setting Up the States
- Setting Up the Workflow Processes
- Creating Workflow Processes
- Testing Your Workflow Process
- Moving Your Work
- Clearing Workflow Assignments
- Creating and Viewing Workflow Reports
- Creating Workflow Groups

About the Workflow Feature

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 multiple asset types.

For more information about the workflow feature, see the following topics:

- Understanding Workflow Participants
- Understanding Workflow States
- Moving Assets from State to State
- Working with Multiple Paths for the Asset
- Managing Deadlocks
- Understanding Workflow Groups
- Delegating and Clearing Assignments
- Placing an Asset in Workflow
- Restricting Access to Assets While They Are in Workflow
- Working with Deadlines
- Scheduling a Deadline Calculation
- Ending a Workflow Process

Understanding Workflow Participants

The job titles of the people who participate in a workflow process are considered roles in the 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.

Understanding Workflow States

Tasks performed on assets 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.

The movement of an asset between states is called a step. Creating the steps in a process links together states in a specific order, and creating the steps in your workflow process is what organizes your process.

An asset that a participant is working on is called an assignment. A user's assignment list is shown In the **My Assignments** form in the administrator's interface. Assets are assigned to users as soon as it enters a state the user to fulfills as workflow participant.

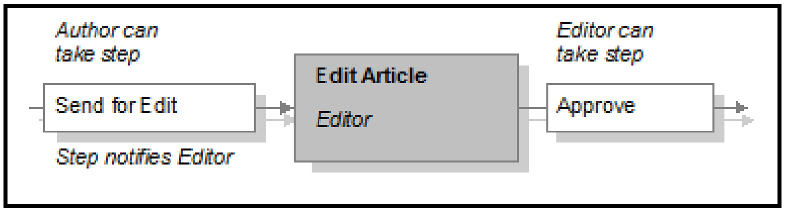
Assign a deadline to the state to keep the asset in that state for a specific amount of time. You can configure the workflow process to send email messages that remind a participant when the deadline is approaching or missed. These email messages are examples of timed actions. You can create one or more timed actions for each state.

Moving Assets from State to State

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.

A step places an asset in a state and notifies participants from the appropriate roles. This operation creates the assignment. You then have 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 18-1 Steps In Editing



When a user selects the **Finish My Assignment** option for an asset that is assigned to him or her, that option calls the step, moves the asset to the next state, and assigns the asset to the next participants. When multiple participants are 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 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 called 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.

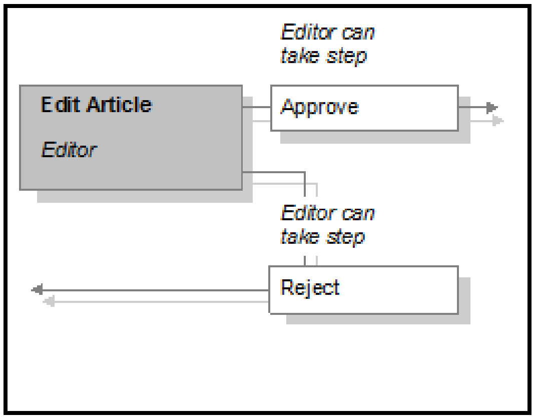
Step actions are events that occur when the step is completed. You can specify one or more step actions for each step. An example of a step action is the ApproveForPublish action. It is a default action delivered with that approves the asset. Typically you use this action in the final step of a workflow process.

Working with Multiple Paths for the Asset

When a workflow process includes a state where people review an asset, typically there are multiple paths possible for an asset in that workflow because the reviewer can either accept it the way it is or reject it.

In these cases, 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.

Figure 18-2 Steps in Workflow



If multiple participants are reviewing the same asset in the same state and they choose different steps, what happens 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.
- 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.

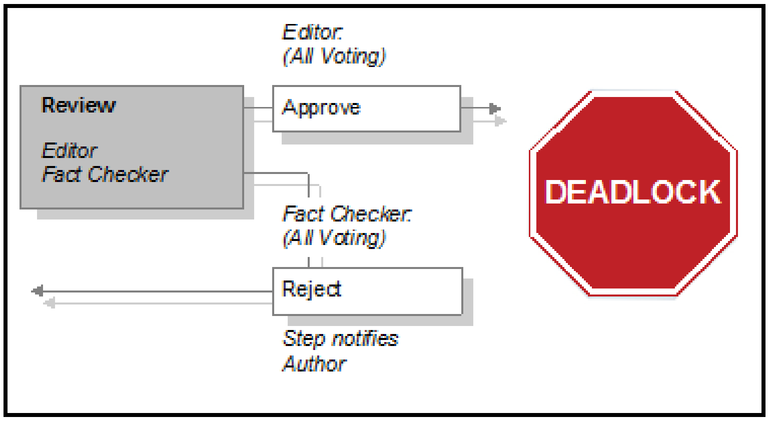
Managing Deadlocks

A deadlock occurs when the following conditions are true:

- There are multiple steps 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 multiple participants.
- The participants select different all-voting steps when they finish the assignment.

The following figure illustrates a deadlock.

Figure 18-3   Deadlocked Workflow



Note that if even one step is not all-voting and a participant selects that step, the asset will not become deadlocked.

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 either 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.

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 have 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.

If you do have 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 sends to the appropriate participants when a step causes a deadlock.

Understanding Workflow Groups

Workflow groups you treat several assets as one unit of work. Assets in a workflow group must be approved at the same time.

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.

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, must know what kind of assets will be included in workflow groups so you can configure the workflow processes appropriately. See Creating Workflow Groups.

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.

Each workflow process can have a workflow administrator. The administrator of a workflow process can delegate assignments on behalf of the other participants. You can specify one or more delegate actions, such as email notification, for each workflow process.

If an asset no longer must be assigned or if it is easiest to clear the assignment and then start over, you can use **Clear Assignments** in the **Admin** tab.

Placing an Asset in Workflow

An asset begins its participation in a workflow process in either 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 calls 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 calls 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. 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.

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, just because the asset is assigned to a specific user does not stop other users from modifying or even deleting that asset.

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.

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.

Function privileges restrict access to a function from the user interface only. You can program step actions that call 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 calls 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 call those functions in a highly controlled way.

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 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.

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.

Working with 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.

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.

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.

Scheduling a Deadline Calculation

There are two kinds of actions:

- Actions that are called by a step. These actions are events that completes when a step is taken.
- Actions that are triggered by a deadline. These actions are queued and are triggered only after calculates the deadlines for the assets and determines which timed actions (if any) should be called.

You specify how often the deadlines are calculated by configuring the Timed Action Event. This is an event that calls a background calculation process of all deadlines. It is similar to the publishing event that calls 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 called 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 shown 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.

Ending a Workflow Process

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.

Planning Your Workflow Processes

When you create a workflow process, you create steps that link together the states for that process. 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. To plan a workflow process, do the following:

- Starting with a Sketch
- Determining Roles and Participants
- Determining the Email Objects, Actions, and Conditions
- Determining States
- Determining Steps
- Determining Function Privileges
- Determining Additional Workflow Process Details

Starting with a Sketch

Where do you begin? With sketches of the business processes that you must 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, see your sketches as you use the administrator's interface to create your workflow processes, described in Configuring Your Workflow Processes.

Determining 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.

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 WebCenter 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 have 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 require 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 Creating a Role From the Admin Interface.

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 are selected when you configure the steps.

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 `wcs_properties.json` file, categorized under UI. See Properties in the UI Category in *Property Files Reference for Oracle WebCenter Sites*.

Determining the Email Objects, Actions, and Conditions

An action is an event that is triggered in one of three ways:

- A step calls 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 calls 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 requirements cannot be met by the default workflow elements that are provided, your developers can code the functionality that you require. See Customizing Workflow in *Developing with Oracle WebCenter Sites*.

WebCenter Sites provides the following default workflow elements:

Table 18-1 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.
`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 Workflow Actions in *Developing with Oracle WebCenter Sites.*

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.

Create workflow email objects to determine the value of the `emailname` variable. The names of the email objects you create in the **Admin** tab are the names that you specify as the arguments with the `emailname` variable for actions that send email messages.

This section covers the following topics:

- Understanding Email Objects
- Understanding Timed Actions
- Understanding Delegate Actions
- Understanding Step Actions
- Understanding Step Conditions
- Understanding Deadlock Actions
- Understanding Group Deadlock Actions

Understanding Email Objects

Email objects are building blocks separate from the actions so that you can use them with multiple actions. 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.

There are several workflow email 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 18-2 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.

Table 18-2 (Cont.) Default Workflow Variables

Variable Name	Description
`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`.

The following are the default workflow email objects:

- 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).

To determine the email objects that you must 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 multiple timed actions or step actions.

Typically, you 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 if 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 must have for that workflow.

Understanding 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 multiple timed actions 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.

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.

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 must 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 **General Admin** tree, expand the **Workflow** node, then expand **Actions**, and 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 must have for that process near the list of the email messages.

Understanding 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. There are no default delegate actions provided.

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.)

Understanding 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 function, the action does not check the function privileges of the participant taking the step. You can restrict access to a function from the user interface and require participants to use a step in a workflow to complete a specific function (approve for publish, for example).

You can specify one or more step actions for each step.

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 (if 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).

To determine what kind of step actions you must create, you must consider the following workflow components:

- The steps that you will create for the process. Do people have to be notified that a step has assigned an asset to them? If so, you require 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 require 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.

Understanding 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. To use conditions in your workflow processes, talk to your developers about the conditions you must have them to implement through elements.

You can specify one or more conditions for each step.

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.

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 require next to the appropriate steps.

Understanding 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.

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.

When thinking about possible deadlock actions, you have to consider the steps that you will create for your workflow processes. If you plan to design your steps so that deadlocks cannot occur, you do not have 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.

Understanding 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 actions for each group.

The group deadlock actions are called in addition to any other actions that are associated with the states or the steps in the workflow process. There is one default group deadlock action: Send Deadlock Email. It is identical to the default deadlock action.

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.

Determining 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.

To determine the states that you have 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 multiple asset types? If so, it's possible that you can reuse the same state in multiple workflow processes.
- 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 multiple workflow processes or share the workflow process with another site.
- If there is a deadline, should 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, 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 no longer 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.

Determining 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 either 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 shown. 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 multiple users (assignees), 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 is 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.

When determining what your steps should be, consider the following questions:

- Every process requires 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 require between each state? This answer determines how many steps you must have.

- Do you have 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 multiple people are 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 multiple possible paths from a state? (That is, there must be multiple steps 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 step that leads 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 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 have 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 are necessary?

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.

Determining 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 shown to site users with those roles. Users can call 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 (Optional) Configure Function Privileges for the Workflow Process.

The following table lists the functions that you can allow or disallow users to perform during a workflow process.

Table 18-3 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. Shown as an option in the **Workflow Commands** drop-down menu on the asset's **Status** form.
Approve for Publish	Users can select an asset as approved for publishing. Shown 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. Shown as an option in the **more...** drop-down menu on an asset's **Inspect** and **Status** forms.
Checkout	Users can check out an asset, if revision tracking is enabled for the asset type. Shown as the **Check Out** button at the top of an asset's **Edit** and **Status** forms.
Copy	Users can copy an asset. Shown 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). Shown as an option in the **Workflow Commands** drop-down list on the asset's **Status** form.
Delete	Users can delete an asset. Shown as an icon in the action bar on an asset's **Inspect** and **Status** forms. Also shown as an icon next to the asset in lists.
Edit	Users can open an asset in its **Edit** form. Shown as an icon in the action bar on an asset's **Inspect** and **Status** forms. Also shown 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 shows 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.) Shown as the **Make Master** link in the **Translations** field of a multi-lingual asset's **Inspect** form.
Place Page	Users can place a page on the **Site Plan** tab. Shown as an option in the context 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. Shown as the **Preview** icon in the asset's toolbar.
Remove from Group	Users can remove the asset from a workflow group. Shown as an option in the **Workflow Commands** drop-down list on the asset's **Status** form.

Table 18-3 (Cont.) Functions and User Action Allowances

Function	If the Function Privilege is Granted...
Remove from Workflow	Users can remove the asset from a workflow process. Shown 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 shown 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. Shown 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 shown only if the workflow is configured to allow its users to set a process deadline. This function then is 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 shown only if the workflow is configured to allow its users to set an assignment deadline. This function then is 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. Shown 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 multiple CM sites.
Show Participants	Users can view a list of the participants in the workflow process that is currently assigned to the asset. Shown 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. Shown as an option in the **more...** drop-down menu on an asset's **Inspect** and **Status** forms.
Show Version	Users can view information about each version of the asset that the revision tracking system is storing. Shown 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.

Table 18-3 (Cont.) Functions and User Action Allowances

Function	If the Function Privilege is Granted...
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.) Shown as an option in a multi-lingual asset's **more...** drop-down menu, on the **Edit**, **Inspect**, and **Status** forms.

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 to enforce next to the appropriate states on your sketches of your workflow processes.

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. If the roles don't match, the result can be a condition in which no one can move the asset out of a 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.

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.

Implementing Simplified Access Control

To implement access control:

1. Create a state. It requires a name and a description. It does not require 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 to be able to create assets of this type and assign this workflow to.
4. Configure the function privileges 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.

Determining 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 Determining 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 Determining Function Privileges.

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 must have access to the **Admin** tab, which means that you require the `GeneralAdmin` role and the `xceladmin` ACL. To create workflow states and workflow processes, you must have access to the **Workflow** tab, which means that you require the `WorkflowAdmin` role.

For more information about access rights in the WebCenter Sites interfaces, see Working with ACLs and Roles and Configuring Users, Profiles, and Attributes.

Before you create a workflow process, you must create the individual workflow components that you require for that process. Here are the general steps that you must take presented in the order you perform them:

Table 18-4 System Locale Properties

Workflow Processes	Workflow Components
Plan your workflow process by drawing it. Then see your sketch and notes throughout this topic.	Planning Your Workflow Processes
Create the roles that you require for your workflow processes. 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:** Create the roles that you require for your workflow processes. 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.	Determining Roles and Participants

Table 18-4   (Cont.) System Locale Properties

Workflow Processes	Workflow Components
Create the email objects that you require for your actions and enable the `xcelerate.emailnotification` property in the `wcs_properties.json` file.	Setting Up Email Objects
Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions necessary.	Setting Up the Actions and Conditions
Create your states.	Setting Up the States
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 necessary.	Setting Up the Workflow Processes
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 called in a timely way.	Setting Up the Timed Action Event
Test your workflow processes.	Testing Your Workflow Process
Set up your start menu shortcuts so that workflow processes are assigned automatically to assets when they are created.	Managing Start Menu Items

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 must have, and then use the procedures in this section to create and edit them.

Your user account must give you access to the **Admin** node, under the **General Admin** tree, in order for you to create email objects.

To ensure that your workflow process can successfully send email messages, the following conditions must be true:

- The following properties must be configured to provide information about your email server:
  - `cs.emailaccount`
  - `cs.emailauthenticator`
  - `cs.emailcharset`
  - `cs.emailcontenttype`

  - `cs.emailhost`
  - `cs.emailpassword`
  - `cs.emailreturnto`
- The `xcelerate.emailnotification` property must be set to `true`.

  See the Properties in the Core Category in *Property Files Reference for Oracle WebCenter Sites*.
- The workflow participants must have email addresses specified in their user profiles.

The following topics provide instructions for working with email objects:

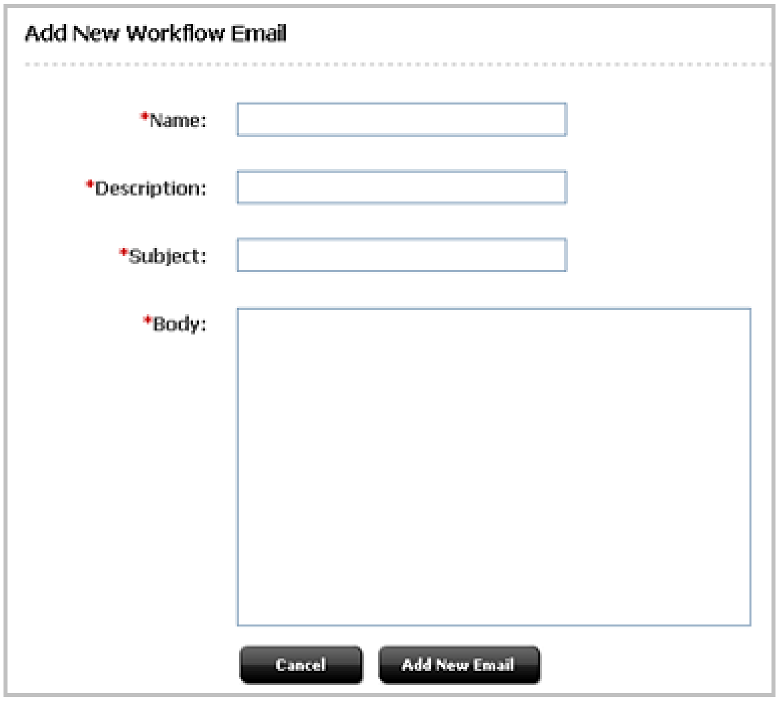
- Creating Email Objects
- Editing Email Objects
- Deleting Email Objects

Creating Email Objects

1. In the **General Admin** tree, expand the **Workflow** node, expand the **Emails** node, and then double-click **Add New**.

   The Add New Workflow Email form opens.

   Figure 18-4 Add New Workflow Email Form



Add New Workflow Email

*Name:

*Description:

*Subject:

*Body:

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.

> **Note:**
>
> The double quote ("), less-than sign (<), and greater-than sign (>) characters are not allowed in the **Name** field. Additionally, the name cannot end with a backslash (\).

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.
5. In the **Body** field, enter the text for the message.
6. Click **Save**.

Editing Email Objects

1. In the **General Admin** tree, expand the **Workflow** node, expand the **Emails** node, and then double-click the email object.
2. In the action bar, click **Edit**.

   The Edit Workflow Email form opens.
3. Make the changes necessary and click **Save**.

Deleting Email Objects

1. In the **General Admin** tree, expand the **Workflow** node, expand the **Emails** node, and then double-click the email object.
2. In the action bar, click **Delete**.

   A warning message opens.
3. Click **Delete Email**.

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 must have. 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 require additional email messages, consult Setting Up Email Objects and create the email messages that you must have.

The following procedures describe how to create, edit, and delete the workflow actions —step, timed, delegate, deadlock, and group deadlock action—and step conditions:

> **Note:**
>
> Your user account must give you access to the **Workflow** tab in order for you to create actions or conditions.

- Creating Actions and Conditions
- Editing Actions and Conditions
- Deleting Actions and Conditions

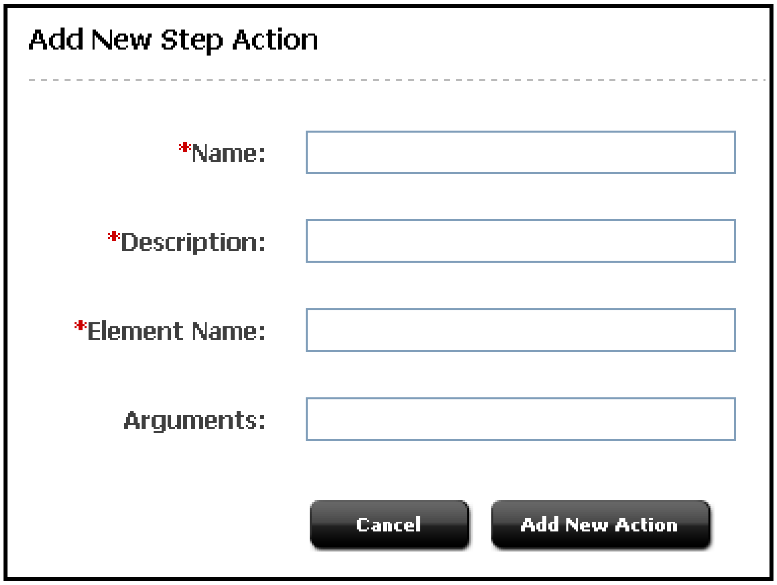
Creating Actions and Conditions

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**.

   The Add New form corresponding to the type of action you selected opens.

   Figure 18-5 Add New Step Action Form

   Add New Step Action

   *Name:

   *Description:

   *Element Name:

   Arguments:

   This example shows the **Add New Step Action** form.
3. In the **Name** field, enter a unique name of up to 40 characters.

> **Note:**
>
> The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

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 requires 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 multiple variables, separate the name/value pairs with the ampersand (&) character. For example:

   ```
   name1=value1&name2=value2
   ```
7. Click **Add New Action**.

Editing Actions and Conditions

1. In the **Workflow** node, 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, and then click **Save**.

Deleting 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**.

   A warning message opens.

5. Click **Delete Action**.

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 need an additional action for each publishing destination.

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 must have.

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 system.

For example:

```
targets=serverX.
```

To specify multiple publishing destinations, 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.

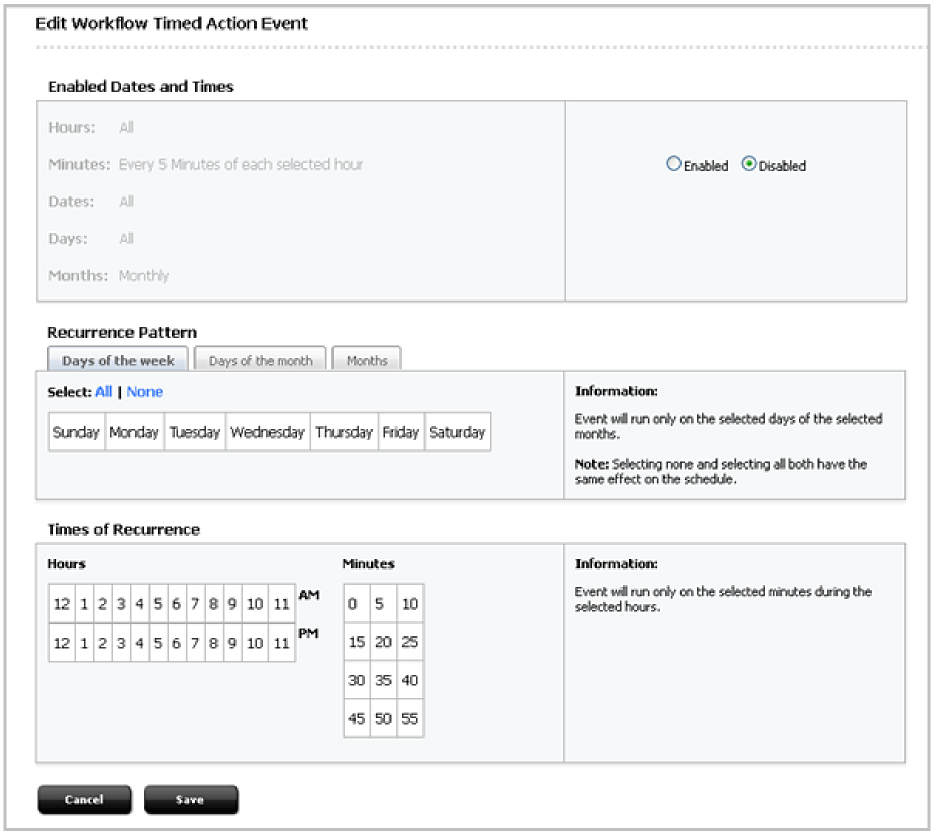
Setting Up the Timed Action Event

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.

1. In the **Workflow** tab, double-click **Timed Action Event**.
2. In the action bar, click **Edit**.

   The Edit Workflow Timed Action Event opens.

Figure 18-6 Edit Workflow Timed Action Event



Edit Workflow Timed Action Event

Enabled Dates and Times

Hours: All

Minutes: Every 5 Minutes of each selected hour

Dates: All

Days: All

Months: Monthly

○ Enabled ● Disabled

Recurrence Pattern

Select: All | None

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

Information:

Event will run only on the selected minutes during the selected hours.

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 are 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.

4. In the **Enabled Dates and Times** field, select **Enabled**.

5. Click **Save**.

   A summary of the schedule opens.

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 Reading the Schedule Abbreviations.

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. Before you create your workflow states, be sure that you have created the timed actions that you must have.

The following procedures describe how to create, edit, and delete workflow states:

> **Note:**
>
> To work with workflow states, your user name must be assigned the `WorkflowAdmin` role for the site that you are working with.

- Creating Workflow States
- Editing Workflow States
- Deleting Workflow States

Creating Workflow States

1. In the **Workflow** tab, expand **States** and double-click **Add New**.

   The Add New Workflow State form opens.

2. In the **Name** field, enter a unique, meaningful name of up to 40 characters.

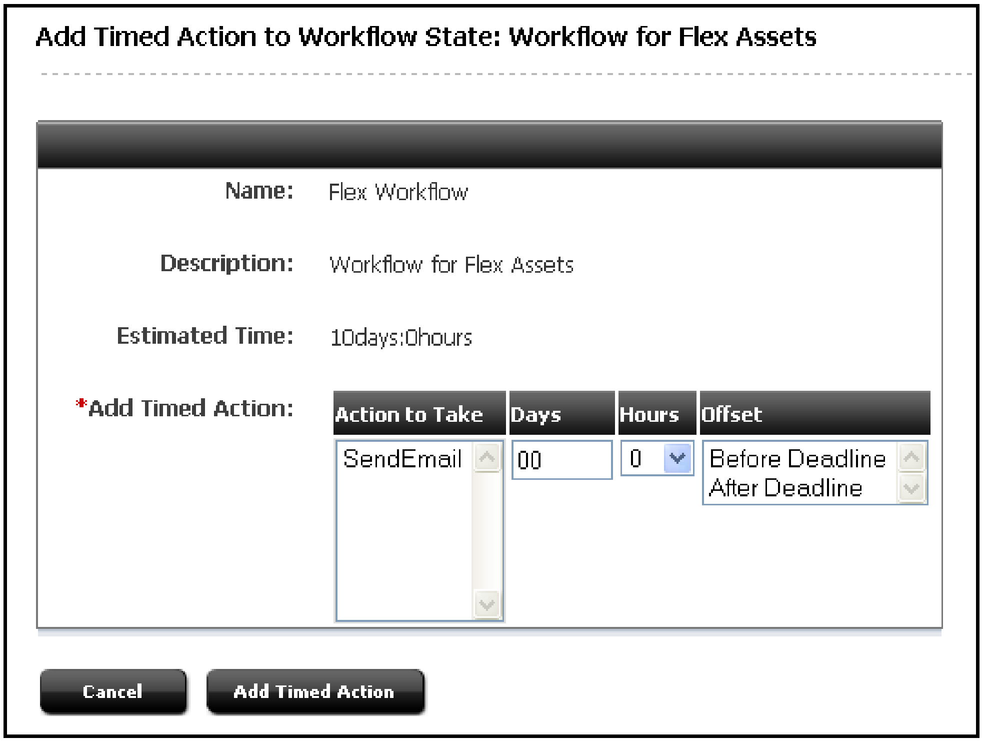
   > **Note:**
   >
   > The following characters are not allowed in the **Name** field: double-quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

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 18-7 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	Offset
SendEmail	00	0	Before Deadline After Deadline

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 the action is to be triggered.

d. Click **Add Timed Action**.

e. Repeat this entire step for each timed action to set up for this state.

6. Click **Add New State**.

Editing Workflow States

1. In the **Workflow** tab, expand **States** and double-click the state to edit.
2. In the action bar, click **Edit**.

   The Edit form opens.
3. Make the 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**.

Deleting 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**.

   A warning message opens.
3. Click **Delete State**.

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 required. 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 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 you are working with.

> **Note:**
>
> If your content providers will use workflow groups, be sure to enable the **Workflow Groups** tab for them. See Creating the Workflow Groups Tab.

The following procedures provide instructions for setting up a workflow process:

- Editing Workflow Processes
- Editing Workflow Steps
- Editing Function Privileges
- Copying Workflow Processes
- Deleting Workflow Processes
- Deleting Workflow Steps
- Deleting Function Privileges

Editing 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**.

   The Workflow Process form opens.
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 Workflow Steps.

- To edit the function privileges, see Editing Function Privileges.

4. When you are finished, click **Save**.

Editing 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**.

   The Workflow Process form opens.
3. Click **Add/Edit Steps**.

   The Steps for Workflow Process form opens.
4. Click **Edit** next to the step you want to edit.

   The Edit Workflow Process Step form opens.
5. Make your changes, and then click **Save**.

   The process is saved and the Inspect form opens.

Editing Function Privileges

To edit a function privilege, you must delete it and then re-create it.

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**.

   The Workflow Process form opens.
3. Click **Add/Edit FunctionPrivs**.

   The Functions for Workflow Process form opens.
4. Click **Remove** next to the function to change.

   A confirmation dialog opens.
5. Click **OK**.
6. Click **New** next to the function you just removed.

   The Add Function Privilege form opens.
7. Make your function privilege selections and click **Save**.

   The Functions for Workflow Process form opens.
8. Click **Save**.

   The workflow process is saved and the Inspect form opens.

Copying Workflow Processes

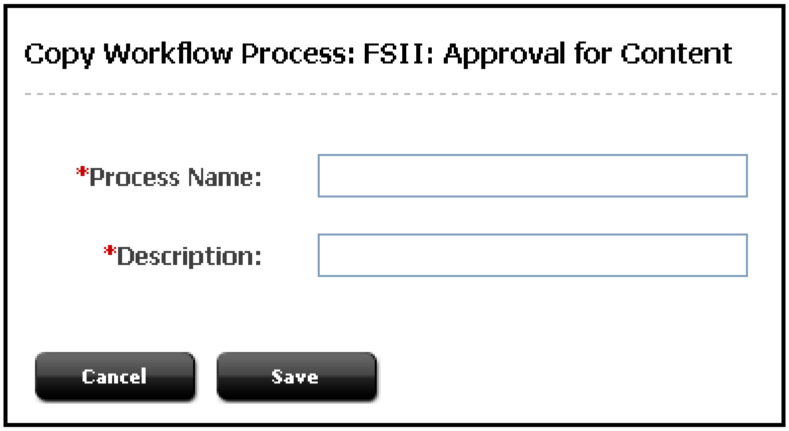
You can copy a workflow process, which can save you some steps in configuring additional processes.

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.

   The Copy Workflow Process form opens.

   Figure 18-8 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.

   > **Note:**
   >
   > The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

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 Workflow Steps
   - Editing Function Privileges

Deleting Workflow Processes

1. In the **Workflow** tab, expand **Processes** and double-click the workflow process you want to delete.
2. In the action bar, click **Delete**.

   A waning message opens.
3. Click **Delete Process**.

   The process has been deleted.

Deleting 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**.

   The Edit Process form opens.

3. Click **Add/Edit Steps**.

   The Steps for Workflow Process form opens.

4. Click **Remove** next to the step you want to delete.

   A dialog box opens.

5. Click **OK**.

6. Click **Save**.

   The workflow process is saved, and the Inspect form opens.

Deleting 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**.

   The Workflow Process form opens.

3. Click **Add/Edit FunctionPrivs**.

   The Functions for Workflow Process form opens.

4. Click **Remove** next to the function privilege you want to delete.

5. Click **Save**.

   The workflow process is saved, and the Inspect form opens.

Creating Workflow Processes

When you create a workflow process, you configure three categories of information: global process settings, steps, and function privileges.

To create a workflow process, perform the instructions in the following procedures:

- Name and Set Global Settings for the Workflow Process
- Create the Start Step for the Workflow Process
- Create the Subsequent Workflow Steps
- (Optional) Create the End Step for the Workflow Process
- (Optional) Configure Function Privileges for the Workflow Process

Name and Set Global Settings for the Workflow Process

1. In the **Workflow** tab, expand **Processes** and double-click **Add New**.

   The Workflow Process: (new) form opens.

2. In the **Name** field and enter a unique name of up to 25 characters.

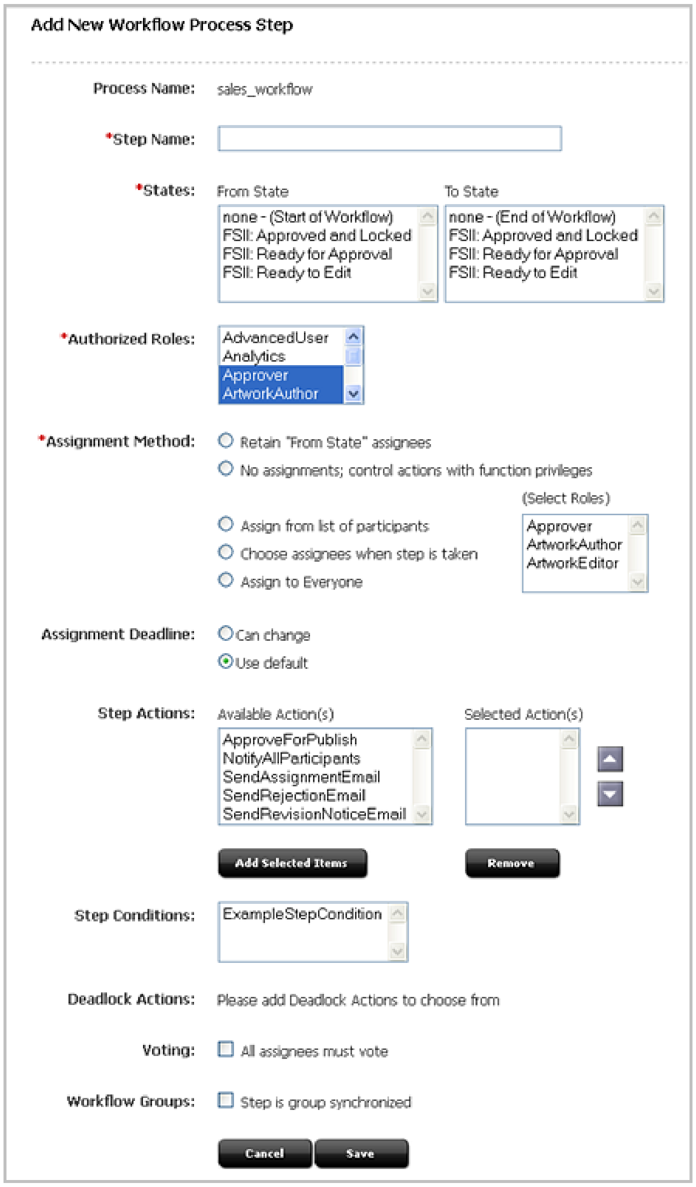
> **Note:**
>
> The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

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**.

    The Add New Workflow Process Step form opens.
11. Create the start step.

Create the Start Step for the Workflow Process

Figure 18-9 Add New Workflow Process Step Form



Add New Workflow Process Step

Process Name: sales_workflow

*Step Name:

*States: From State
none - (Start of Workflow)
FSII: Approved and Locked
FSII: Ready for Approval
FSII: Ready to Edit

To State
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
○ Choose assignees when step is taken
○ Assign to Everyone

(Select Roles)
Approver
ArtworkAuthor
ArtworkEditor

Assignment Deadline:
○ Can change
● Use default

Step Actions: Available Action(s)
ApproveForPublish
NotifyAllParticipants
SendAssignmentEmail
SendRejectionEmail
SendRevisionNoticeEmail

Selected Action(s)

Step Conditions: ExampleStepCondition

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.

> **Note:**
>
> The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

2. Configure the states as follows:
   1. In the **From State** list, select **none (start of workflow)**.
   2. 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 are assigned the asset by selecting among 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 Determining 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 called when this step is taken.
7. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
8. (Optional) Select the **Voting** check box to select that all assignees must vote to approve.
9. (Optional) Select the **Workflow Groups** check box to have the step group synchronized.
10. Click **Save**.

    The start step is added and the Steps for Workflow Process form opens.
11. Click **Save**.

    The Inspect form opens. Note that this step opens as the **Start Step** in the form.

Create the Subsequent Workflow 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.

   The Edit form opens.
2. Click **Add/Edit Steps**.

   The Steps for Workflow Process form opens.
3. Click **Add New Step**.

The Add New Workflow Process Step form opens.

4. In the **Step Name** field, enter a unique, meaningful name of up to 64 characters.

> **Note:**
>
> The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

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 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 are assigned the asset by selecting among 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 multiple participants.

     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 Determining 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 called 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 multiple participants were 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**. See Managing Deadlocks and Understanding Delegate Actions.
13. If this workflow process is 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 opens.
15. Click **Save**.

    The Inspect form opens.
16. Repeat this procedure for each step—except the end step—to create for this process.

    To create the end step, continue with (Optional) Create the End Step for the Workflow Process.

    To configure function privileges, continue to (Optional) Configure Function Privileges for the Workflow Process.

(Optional) Create the End Step for the Workflow Process

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.

   The Edit Process form opens.
2. Click **Add/Edit Steps**.

   The Steps for Workflow Process form opens.
3. Click **Add New Step**.

   The New Workflow Process Step form opens.
4. In the **Step Name** field, enter a unique, meaningful name of up to 64 characters.

   > **Note:**
   >
   > The following characters are not allowed in the **Name** field: double quote ("), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).
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 called 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 multiple participants were 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.** See Managing Deadlocks and Understanding Delegate Actions.
11. If this workflow process is 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 opens.
13. Click **Save**.

    The Inspect form opens.

If you do not have to configure function privileges, this workflow process is completed. If you do have to configure function privileges, continue to the next procedures.

(Optional) Configure Function Privileges for the Workflow Process

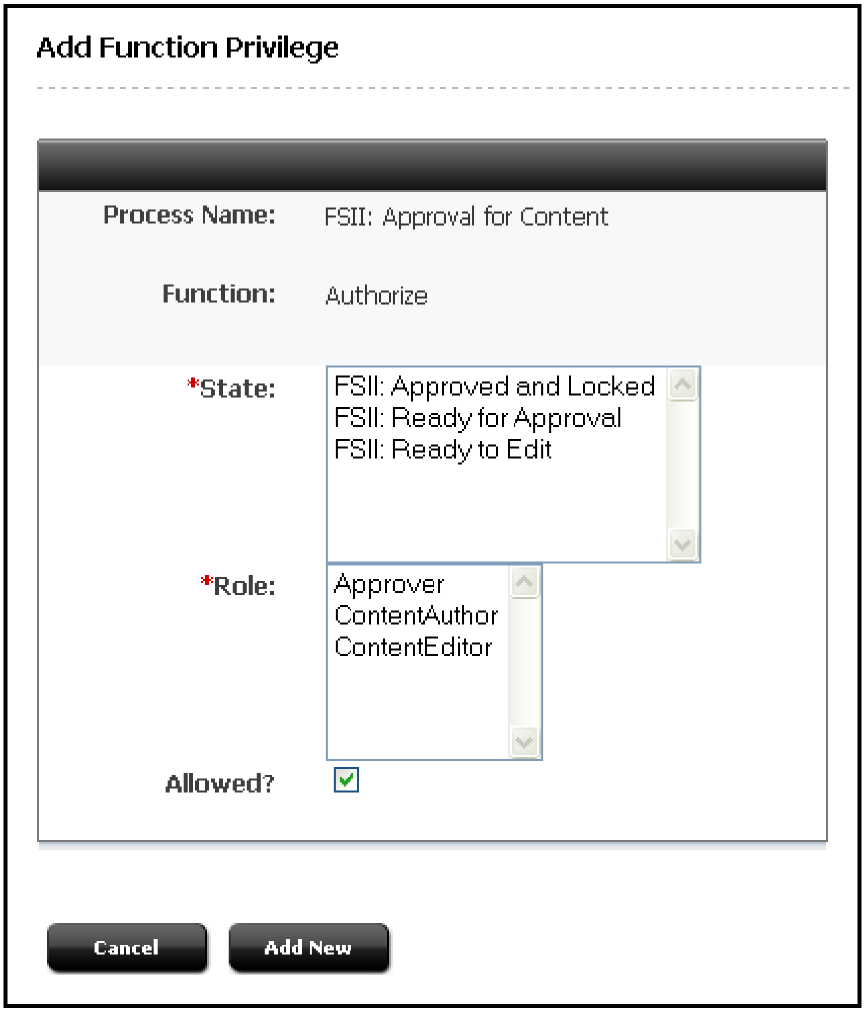
1. Find the workflow process and open its **Inspect** form.
2. In the action bar, click **Edit**.

   The Edit form opens.
3. Click **Add/Edit Function Privileges** at the bottom of the form.

   The Functions for Workflow Process form opens.
4. Scroll down to the function for which you want to set a privilege.
5. Click **New** next to the function.

   The Add Function Privilege form opens.

Figure 18-10 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? ☑

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 either 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**.

   The privilege is saved, and the Functions for Workflow Process form opens.
10. Repeat steps 3 to 8 for each function privilege that you must configure.
11. After you have configured all of your function privileges, click **Save**.

    The process is saved. The Workflow Process form opens.

The workflow process is complete.

Testing Your Workflow Process

Before you move your workflow process to the management system and implement it there, test it on your development system.

A complete methodology of testing your process includes:

- 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.

Moving Your Work

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. See Migrating a Site from One System to Another.

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, the administrator can still clear assignments from some user's assignment list.

To clear assignments for a user:

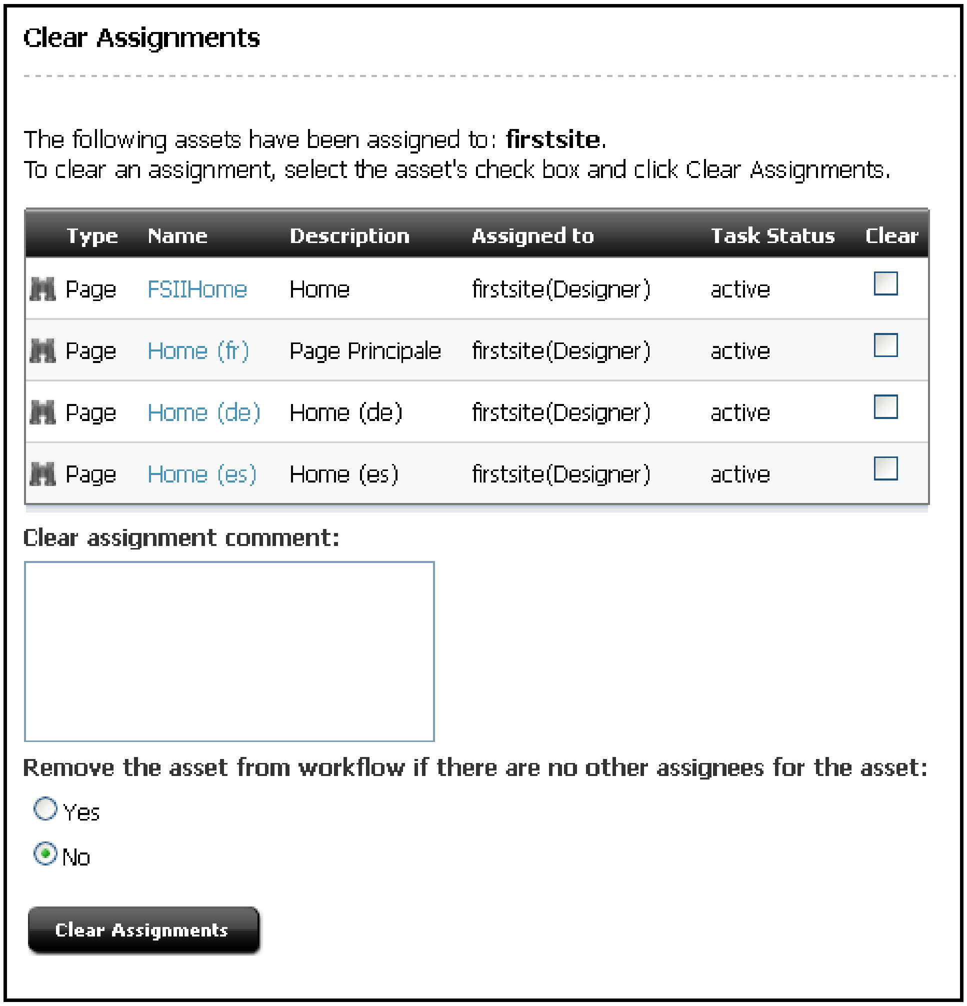
1. In the **General Admin** tree, expand the **Workflow** node, and then double-click **Clear Assignments**.

   The Search for Assignments form opens.

2. Enter a user name and click **Show Assignments**.

   The **Clear Assignments** form opens, showing a list of assets that are currently assigned to that user.

Figure 18-11   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	□
Page	Home (fr)	Page Principale	firstsite(Designer)	active	□
Page	Home (de)	Home (de)	firstsite(Designer)	active	□
Page	Home (es)	Home (es)	firstsite(Designer)	active	□

Clear assignment comment:

Remove the asset from workflow if there are no other assignees for the asset:

○ Yes

● No

3. Select the **Clear** check box next to each assignment to clear.
4. 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 opens in the **Action Taken** field in the **Workflow History** section of the asset's **Status** form.
5. Do either 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**.
6. Click **Clear Assignments**.

   The Clear Assignments Report summarizing your changes opens.

Creating and Viewing Workflow Reports

Workflow reports are used to view the status of assets in workflow. You can configure the report to show assets by type, by user assignment, by role assignment, workflow state, and deadline.

To create a workflow report, complete the following steps:

1. Click **Workflow** in the menu bar. In the displayed form, click **Create a Workflow Report**.
2. In the Create Workflow Report form, select the asset type, workflow state, users, roles, or users and roles the asset is assigned to, the assignment deadline, the process deadline, and the final report options.

   None of these options are required. If you do not select any of the options the report will display all assets in workflow.
3. Click **Report** to display the report.

The report displays all assets in workflow that meet your selected criteria. Click **Save This Workflow Report** to save the report parameters to run and view regularly. Click **Edit This Workflow Report** to change the report criteria and re-run the report.

Creating Workflow Groups

You place assets in a workflow group so you can treat all assets as one work unit. All assets within the group must be approved together before any can move to the next point in workflow.

To create a workflow group, complete the following steps:

1. Click **Workflow** in the menu bar. In the displayed form, click **Create a Workflow Group**.
2. In the Create Workflow Group form, enter a Name, Description, and a Workflow Process. You must select a workflow process from the pull-down menu before you select the participants.
3. Click **Set Participants...** to select the participants in the group. The available participants will depend on the workflow process you selected.

   After you select the participants the Create Workflow Group form re-displays.
4. Select the roles that can add and remove assets, then select the roles that can edit or delete the workflow group.
5. Select the group deadlock action.
6. Click **Save**. The workflow group is displayed in a new form.

At this point the group is created but it is empty. Click **Add to Group** to add assets to the group.

Part VII

Enabling Blog Components

Content providers to sites require access to blog components in order to create blog content. Therefore, you should enable Oracle WebCenter Sites: Blogs module on different CM sites with their start menu items.

The following topics provide information about enabling the blog components. These topics also include instructions for creating author assets:

- Enabling Blog Asset Types
- Creating a Blog Tree Tab
- Sharing Blog Assets
- Working with Blog Authors
- About Creating Workflows for and Publishing Blog Assets

19 Enabling Blog Asset Types

Content providers can access blog components and create blogs or content for the sites when you enable the blog asset type and the start menu items on the sites. They use the start menu items to create and search for assets of the types that you enable.

To enable blog asset types:

1. Log in to the Admin interface as a general administrator.
2. Select a site.
3. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, expand the node of the site for which you want to enable the blog asset types, and then 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, see Customizing the Blog Asset Form in *Developing with Oracle WebCenter Sites.*
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 Managing Access to Content Management Site Components.

20
Creating a Blog Tree Tab

You can display the hierarchy of blog categories and their assets by configuring a tree tab in the Admin interface.

To create a blog tree tab:

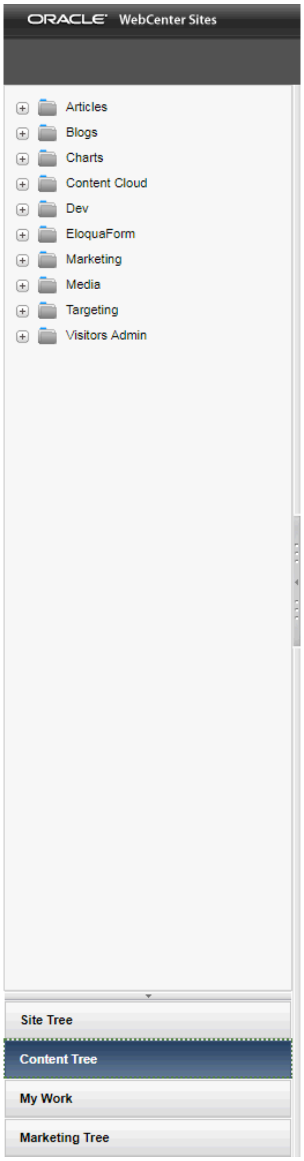
1. Log in to the Admin interface as a general administrator.
2. Select a site.
3. In the **General Admin** tree, expand the **Admin** node, and then double click the **Tree** node.
4. Click **Add New Tree Tab**.

   The Add New Tree Tab form opens.
5. In the Add New Tree Tab form, fill in the fields:
   - **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 opens when a user mouses over the tab.
   - **Site**: Select the sites that display this tab. You must enable the blog asset types on these sites. For instructions, see 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 opens on the tab and then click **Add Selected Items**.
   - (Optional) To add custom functionality to this tab, use the **Section Name** and **Element Name** fields. See Customizing Blog Components in *Developing with Oracle WebCenter Sites*.
6. Click **Save**.

   The Information page for the newly created blog opens.

   Additionally, the new blog tab opens in the tree (refresh your browser, if necessary).

Figure 20-1 Tree Showing New Blog



21 Sharing Blog Assets

You can share the blog assets created by content providers across sites. Make sure that you share all types of blog assets.

> **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 Admin interface as a general administrator.
2. Select a site.
3. Locate the asset you want to share:
   a. In the menu bar, click **Search**.
   b. Click **Find** or **Advanced Search** for the asset type.
   c. Fill in the fields, and then click **Search**.
4. In the search results list, click the asset you want to share.
5. At the top of the form in the button bar, click **Share element**.

   The Share Element page opens.
6. In the **Share to:** field, select the sites with which you want to share the blog asset you selected.
7. Click **Save Changes**.

   The blog asset you selected to share is now available on the sites with which you shared it.

For detailed information about sharing assets, see Copying vs. Sharing.

22
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 on creating blog author assets, which contain author information. This section also provides instructions on granting users access to their own blog author assets.

The following topics provide information about working with blog authors:

- Creating Author Assets
- How to Set Access Permissions to a Blog Author Asset

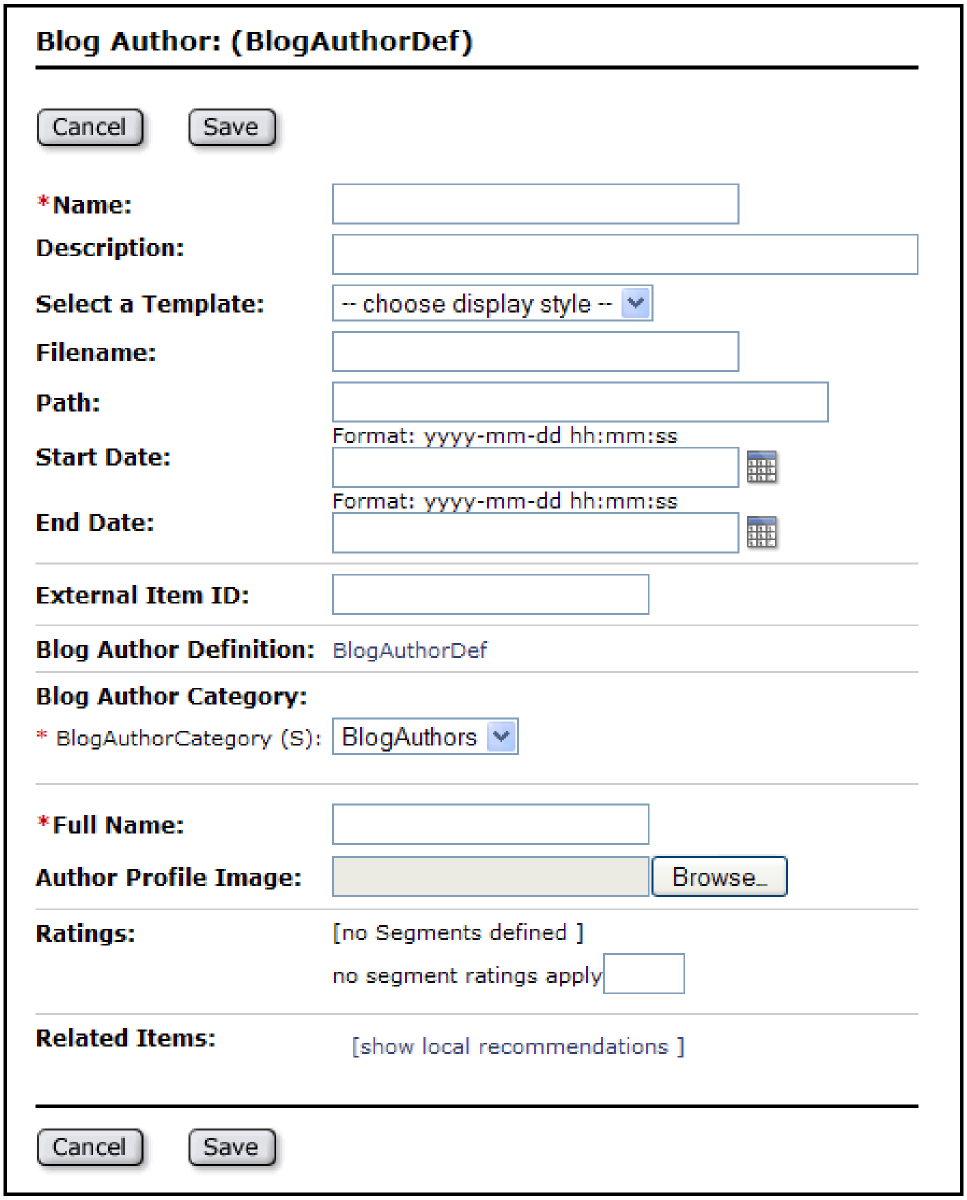
Creating Author Assets

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 Admin 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 22-1 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):	BlogAuthors
*Full Name:	
Author Profile Image:	Browse...
Ratings:	[no Segments defined] no segment ratings apply
Related Items:	[show local recommendations]

- **Name**: Enter a name for the blog author asset.

  The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), question mark (?), percent sign (%), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

  > **Note:**
  >
  > When the author creates a blog asset, the author must enter the same name into the blog asset **Author** field. Otherwise, the author asset you have created is not published to its allocated space.

- **Description**: Enter a short, descriptive summary about the author asset.
- **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 opens on the blogs that this author creates.
- **Author Profile Image**: Click **Browse**, and select an image for the content provider.

6. Click **Save**.

How to Set 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:

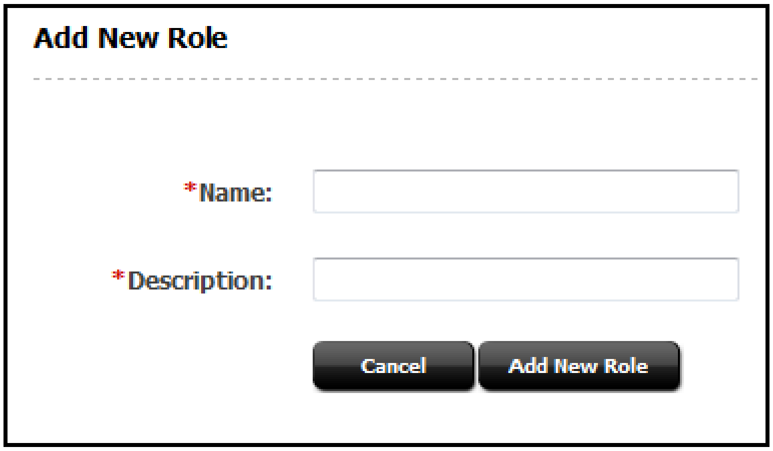
1. Creating a Unique Role for the Blog Author
2. Assigning the Role to the Blog Author
3. Setting Access Permissions to a Blog Author Asset

Creating a Unique Role for the Blog Author

1. Log in to the Admin interface as a general administrator, and select a site.
2. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **Roles** node, and then double-click **Add New**.

   The Add New Role form opens:

Figure 22-2 Add New Role Form



Add New Role

*Name:

*Description:

- **Name**: Enter a unique name for the role.

> **Note:**
>
> The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), percent (%), question mark (?), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash (\).

- **Description**: Enter a short, descriptive sentence explaining what the role does.

3. Click **Add New Role**.

   Now you must assign the role to the content provider who requires access permissions to the blog author asset.

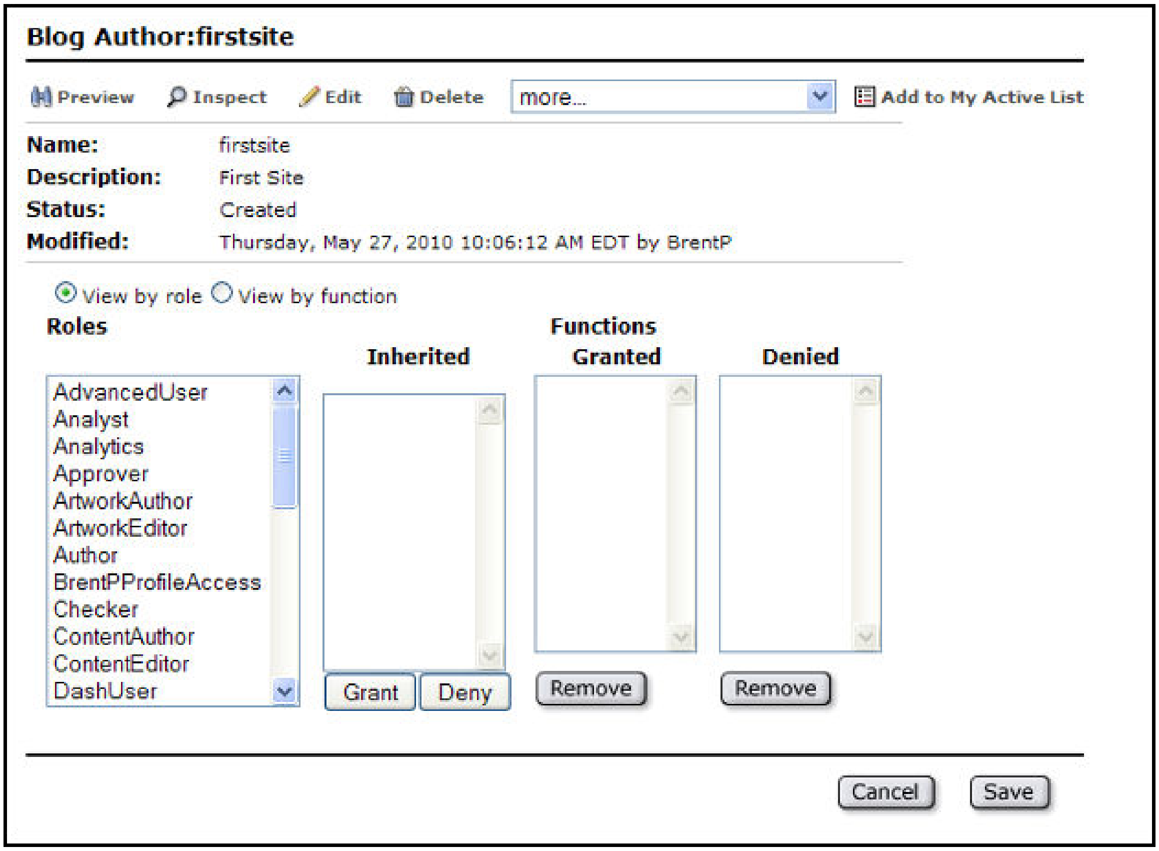
Assigning the Role to the Blog Author

1. Select the **Site** tree, expand the **Site Admin** 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 want 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 and 3.
5. Click **Save**.

Setting 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...** menu, and choose **Access Permissions**.

Figure 22-3   Access Permissions Form



Blog Author:firstsite

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

AdvancedUser
Analyst
Analytics
Approver
ArtworkAuthor
ArtworkEditor
Author
BrentPProfileAccess
Checker
ContentAuthor
ContentEditor
DashUser

4. Select the **View by role** radio button.
5. In the **Roles** list, select the role you created in Assigning the Role to the Blog Author.
6. In the **Inherited** list, select all the functions (shift-click the first and last functions), and click **Grant**.

   All the functions in the **Inherited** list move to the **Granted** list.
7. Click **Save**.

The only user who can work with this blog author asset, besides the administrator, is the blog author who is assigned the selected role.

23 About Creating Workflows for and Publishing Blog Assets

You need to include Blog assets in the workflow because they need to be approved before you publish these assets.

The following topics provide information about creating workflows for Blog assets and publishing Blog assets:

- Creating Workflows for Blog Assets
- About Publishing Blog Assets

Creating Workflows for Blog Assets

You can use a workflow to manage the work on a blog asset. Before you can publish a blog asset, you must approve it for publishing. The workflow feature routes blog assets through the steps that you specify, from creation to approval.

For example, if you want an editor to review a blog asset before you approve it for publishing, the WebCenter Sites workflow feature can route that blog asset to the appropriate user at the appropriate time. See Creating and Managing Workflow Processes.

About 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.

Part VIII
Enabling A/B Testing

The A/B Test component of WebCenter Sites enables marketers to further enrich the visitor data collected from the website. When you enable the A/B Test asset, marketers use the A/B Test feature to test design and content changes to web pages. This testing helps them determine which versions create more customer activity and interest.

> **Note:**
>
> Currently, we are making changes to the A/B testing feature. Earlier, A/B test functionality was provided through an integration with Google Analytics and to use the A/B test functionality, you had to register with Google for a Google Analytics account. This integration with Google Analytics was using Management APIs for experiment creation. You won’t be able to create experiments as these Management APIs are not available. For more information, see Content Experiments by Google Analytics.
>
> Now, we will use Oracle Maxymiser client-side integration with WebCenter Sites to experiment with design and content variations on your website pages. For information, see Working with A/B Testing in *Using Oracle WebCenter Sites*.

Part IX
Enabling Eloqua Cloud Marketing Service

WebCenter Sites integrates tightly with marketing automation tools like Eloqua to provide reusable content for campaigns and blogs. To unify the customer experience and consistency across channels, marketers can share and reuse the content from WebCenter Sites and deliver higher conversion rates.

Topic:

- Administering Eloqua Cloud Marketing Service in WebCenter Sites

24

Administering Eloqua Cloud Marketing Service in WebCenter Sites

WebCenter Sites can use Eloqua visitor profile data to target specific content to the visitor by personalizing their visit to all those pages where they navigate around on your website. By sharing both content and visitor profile data across Eloqua and WebCenter Sites, the visitor gets a consistent personalized experience that leads to higher conversions. Read further to find out how you can integrate the Eloqua Cloud Marketing Service (Eloqua) with Oracle WebCenter Sites.

Topics:

- Eloqua Integration Overview
- Accessing WebCenter Sites Content in Eloqua
- Registering the Eloqua Firehose Service
- Creating the Rendering Template for the EloquaForm Assets

Eloqua Integration Overview

Integrating Eloqua with WebCenter Sites provides marketers with the ability to converge the visitor's online experience with their marketing campaigns, as well as more effectively manage the visitor journey. The integration eliminates content duplication and streamlines content targeting across channels.

This is achieved through the following functionality enabled by the integration:

- **Embedding WebCenter Sites content into Eloqua forms and landing pages.** Drag-and-drop placement of existing WebCenter Sites content ensures consistency of the marketing message and a seamless visitor experience across channels. Specific content can be targeted at specific visitors or campaign contacts.
- **Embedding Eloqua forms into WebCenter Sites pages.** Drag-and-drop placement of Eloqua forms into WebCenter Sites pages eliminates the involvement of the site administrator or developer in the lead generation process.

Existing visitor segment and contact data can be automatically passed from Eloqua to WebCenter Sites for targeted content delivery.

Accessing WebCenter Sites Content in Eloqua

To embed WebCenter Sites content in Eloqua e-mails and landing pages, you need to configure your Eloqua instance with the WebCenter Sites content service, ensure that only the authorized Eloqua instance can access content within the target WebCenter Sites instance, and determine which WebCenter Sites asset types and associated templates are available to Eloqua marketers.

See these topics:

- Configuring Eloqua for Access to the WebCenter Sites Content Service
- Configuring OAuth Authentication
- Making WebCenter Sites Assets Available within Eloqua

Configuring Eloqua for Access to the WebCenter Sites Content Service

Before marketers can embed existing WebCenter Sites content into Eloqua e-mails and landing pages, you must configure your Eloqua instance to communicate with the WebCenter Sites content service. The example URLs below assume the default servlet path of `/cs/ContentServer` - adjust your URLs as appropriate for your environment. Also, if you have not already done so, create a new provider in Eloqua to separate the WebCenter Sites content service from other services and applications.

> **Note:**
>
> Your WebCenter Sites instance must be configured to accept HTTPS connections. Only ports 443 and 8443 are supported. All URLs, property names, and their values are case-sensitive.

To configure Eloqua for accessing the WebCenter Sites Content Service, you must do the following:

1. Log into the target Eloqua site as its administrator.
2. In the top bar, click **Settings** and select **AppCloud Developer**.
3. Click **Create New App** and fill in the form that appears as follows:

Table 24-1 Create New App Fields

Field	Value
Icon URL	https://*wcs-instance.company.com*/cs/images/eloqua/AppIcon.png
Enable URL	https://*wcs-instance.company.com*/cs/ContentServer?pagename= EloquaIntegration/Enable&installId={InstallId}&appId={AppId}&callbackUrl={CallbackUrl}
Status URL	https://*wcs-instance.company.com*/cs/ContentServer?pagename= EloquaIntegration/SitesStatus
Callback URL	https://*wcs-instance.company.com*/cs/ContentServer?pagename= EloquaIntegration/Enable

4. Click **Save App** to save your changes.
5. In the **App Details** page that appears, click **Show Secret** in the **OAuth** section, copy the client secret value displayed in the pop-up, and save it to a text file. (You need this value to configure the authentication of Eloqua requests within WebCenter Sites later in this guide.)
6. In the **Services** section click **Add Service**, and then click **Content** in the pop-up that appears.

7. Fill in the fields in the **Content Service Details** form as follows:

Table 24-2 Content Service Details Fields

Field	Value
Icon URL	`https://wcs-instance.company.com/cs/images/eloqua/ ContentServiceIcon.png`
Create URL	`https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/Instances/Add&instanceId={InstanceId}`
Configure URL	`https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/ContentSearch&instanceId={InstanceId}&installId={InstallId}`
Delete URL	`https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/Instances/Delete&instanceId={InstanceId}`
Copy URL	`https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/Instances/Copy&instanceId={InstanceId}&originalInstanceId={OriginalInstanceId}`
Layout	**Flow** Flow layout mode allows the embedded content to be dynamically resized.
Max-records	The maximum number of rendered assets passed from WebCenter Sites to Eloqua in a single call. Oracle recommends a value below 5000; however, if you experience issues with passing very large pieces of content, lower this value. For more information, consult Eloqua documentation.
Content-type	**application/json**
Landing Page Notification URL	(Optional) To make WebCenter Sites content available in Eloqua landing pages: `https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/Processor/LandingPageNotification&instanceId={InstanceId}&installId={InstallId}&executionId={ExecutionId}`
E-mail Page Notification URL	(Optional) To make WebCenter Sites content available in Eloqua e-mails: `https://wcs-instance.company.com/cs/ContentServer?pagename= EloquaIntegration/Processor/EmailNotification&instanceId={InstanceId}`

Fill in the remaining fields as appropriate for your environment, then click **Save** to commit your changes.

8. Save your changes and return to the WebCenter Sites content service properties page.

9. Click the link in the Catalog section to add the WebCenter Sites content service to your cloud service catalog. When prompted, click **Accept**.

10. Install the WebCenter Sites cloud service into your Eloqua instance.

This makes it available within the Eloqua marketer interface for the target Eloqua instance only.

a. In the Eloqua interface, click **Settings**, and then select **Setup**.
b. Click **AppCloud Catalog**.
c. Select the **Catalog** tab to display the list of available (but not installed) services.
d. Locate your target application and click **Install**.

> **Note:**
>
> To maintain security within your Eloqua instance, do not publish the WebCenter Sites content service.

Configuring OAuth Authentication

Secure HTTP calls from Eloqua to the WebCenter Sites content service are authenticated via OAuth 1.0 to ensure that only the authorized Eloqua instance can access content within the target WebCenter Sites instance.

To facilitate this authentication process, you must provide the Eloqua client secret value to the WebCenter Sites content service as follows:

1. Open the Property Management Tool and in the **Search by** section, do one of the following:
   - In the **Name** field, enter the name of the property you want to modify.
   - In the **Category** drop-down menu, select `EloquaIntegration`.
2. Click **Search**.
3. In the **Key** column, select the name of the property.
4. In the **Value** field, enter the new value.
5. Click **Save**.
6. Repeat steps 2 through 6 for all the properties you want to change.
7. Stop and restart the application server to apply the changes.

Making WebCenter Sites Assets Available within Eloqua

You need to decide which WebCenter Sites asset types and associated templates should be available to Eloqua marketers. Rendering templates must be created specifically for rendering WebCenter Sites content within Eloqua e-mails and landing pages according to the best practices listed in the next section.

Registering the Eloqua Firehose Service

The Eloqua Firehose service notifies you whenever a form that has been embedded into a WebCenter Sites page has been deleted from Eloqua. This minimizes the chance of delivering pages that are missing content.

1. Log in to the target Eloqua instance as the administrator.

2. In the top bar, click **Settings**, and then select **AppCloud Developer**.
3. In the list of services, click the WebCenter Sites content service you have created.
4. In the Services section click **Add Service**, and then select **Firehose** from the pop-up dialog.
5. In the form that appears, fill in the fields as follows:

Field	Value
Icon URL	`https://`*`wcs-instance.company.com`*`/cs/images/eloqua/AppIcon.png`
Name	WebCenter Sites Firehose
Description	WebCenter Sites Firehose Notification Service
Pattern	`Form.Deleted`
Notification URL	`https://`*`wcs-instance.company.com`*`/cs/ContentServer?pagename=EloquaIntegration/Processor/FirehoseNotification`

6. Click **Save** to store your changes.
7. Add the following line to the `protectedPages`list in the `web.xml` file on your WebCenter Sites instance:

   `EloquaIntegration/Processor/FirehoseNotification`

Once an Eloqua form has been embedded into a WebCenter Sites page, a rendered copy of that page, including the Eloqua form, is stored in the WebCenter Sites cache. If an Eloqua marketer deletes the embedded form within Eloqua, that cached page copy is no longer valid.
When notified by the Firehose service, the WebCenter Sites contributor responsible for the affected page must manually remove the deleted Eloqua form from the page and republish it; otherwise, the page is delivered to the visitor's browser incomplete.

Creating the Rendering Template for the EloquaForm Assets

To create the template that renders the `EloquaForm` proxy assets within WebCenter Sites pages, do the following:

1. Log in to the Admin interface on target WebCenter Sites site as the administrator.
2. Create a new template with the following parameters:

Field	Value
Name	`Detail`
For AssetType	`EloquaForm`
Usage	Element is used within an HTML page
Create Template Element	JSP
Element Logic	paste the contents of the `EloquaForm/Detail.jsp` element

> **Note:**
>
> If you upgrade from 12.2.1.3.0 release to Bundle patch 2, update the Element Logic of the existing EloquaForm Assets with the updated code of the EloquaForm/Detail.jsp element file. This will ensure accurate rendering of the templates in the Web mode.

3. Save the new template.

Part X
Enabling Bluekai Integration

Bluekai helps you tap into the numerous anonymous visitors who visit your site. Enabling Bluekai integration with WebCenter Sites allows you to customize your website with online interests of site visitors, including ones who are anonymous.

Topic:

- Administering Bluekai Integration

25 Administering Bluekai Integration

To configure the integration of Bluekai with WebCenter Sites, you must have an account with Bluekai. This configuration enables you to create or share campaigns in the Bluekai partner site. The configuration process involves a workflow listed in the following table:

Workflow Process	Workflow Details
Share your BlueKai campaigns	Log in to the BlueKai partner site (https://partner.bluekai.com) and share the campaigns as required to use in WebCenter Sites.
Configure BlueKai Properties	Configure the properties related to BlueKai in the Property Management Tool.

Topics:

- Working with Bluekai Partner Site
- Configuring Bluekai Integration Properties

Working with Bluekai Partner Site

To share and utilize information from the Bluekai partner site, you can create an account and set up campaigns for your site.

For information on Bluekai usage, see http://docs.oracle.com/en/cloud/saas/data-cloud/dsmkt/oracle-bluekai-integration-portal.html. In the Bluekai partner site, you can decide to partner with existing collaborators (organizations which are already registered with Bluekai) to buy or share data about site visitors.

Configuring Bluekai Integration Properties

You must configure all the properties listed in the following table to integrate Bluekai with WebCenter Sites. Configure Bluekai cookies with the `bluekai.sites.cookie` and `bluekai.segments.cookie.prefix` properties. The `bluekai.user.key` and `bluekai.private.key` properties are used for programmatic access of the Blukai site and data.

For instructions on how to access the Property Management Tool in the Admin interface and use it to verify or modify property values, see Using the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.

Property	Description
`bluekai.user.key`	This property is a user key obtained by creating an account in Bluekai. This is the Bluekai user key to connect your site with the Bluekai web services. You can use the key provided by Bluekai and set this property through the Property Management Tool.
`bluekai.private.key`	This property is a user key obtained by creating an account in Bluekai. It is a Bluekai private key to encrypt the `bluekai.user.key property`.
`bluekai.sites.cookie`	This property is a Bluekai container ID cookie that stores comma separated names of Bluekai containers or site IDs. You can configure this property in the Property Management Tool and set it as per your site name or preferences.
`bluekai.segments.cookie.prefix`	This property is a cookie prefix used in creating cookies per Bluekai container or site. If you configure this property in the Property Management Tool, these cookies store names of Bluekai segments that qualify for different site visitors.

Once you have configured these properties for your site, every time a visitor visits your site, the JavaScript APIs run and retrieve the cookies stored against the visitor. The information retrieved from the cookies helps you to customize your site with respect to the preferences of the site visitor.

Part XI

Enabling Content and Experience Cloud Integration

Oracle Content and Experience Cloud is a cloud-based content hub that lets you manage your content wherever you are, keeping all your files at your fingertips at home, in the office, at the airport, or in a hotel, halfway across the world.

With the integration of Content and Experience Cloud with WebCenter Sites, you can share, collaborate, process and manage your content easily and effectively.

Topics:

- Administering Content and Experience Cloud Integration
- Publishing Content from WebCenter Sites to Oracle Content and Experience

26 Administering Content and Experience Cloud Integration

As an administrator, you can configure and synchronize Content and Experience Cloud with WebCenter Sites. Site developers and content contributors can access content, such as images, articles, and files, that is available in the Content and Experience Cloud; and use them in WebCenter Sites.

> **Note:**
>
> - Content and Experience Cloud assets are read-only assets.
> - Data can be fetched from Content and Experience Cloud to WebCenter Sites, but Content and Experience Cloud cannot fetch data from WebCenter Sites.
> - If you modify and publish any content in Content and Experience Cloud, you can get the latest content in WebCenter Sites without any additional modifications in WebCenter Sites.
> - If you add any new content in Content and Experience Cloud, you can synchronize its references in WebCenter Sites; and use that content through the WebCenter Sites UI to build sites.

Topics:

- Getting Content and Experience Cloud Access Token
- Configuring Content and Experience Cloud with WebCenter Sites
- Configuring Content and Experience Cloud Properties
- Publishing Content and Experience Cloud Assets

Getting Content and Experience Cloud Access Token

The Content and Experience Cloud access token is used to make REST calls to Content and Experience Cloud. You can get a Content and Experience Cloud Access Token through a REST API call.

To get a Content and Experience Cloud Access Token, invoke the following REST API call on the browser with the URL of the Content and Experience Cloud instance:

```
https://<example>.oraclecloud.com/documents/integration?
IdcService=GET_ACCESSABLE_PUBLISH_TARGETS&IsJson=1
```

Enter the Content and Experience Cloud username and password with which you have configured WebCenter Sites. The returned JSON response contains the token for respective published targets. For example, `a4a899f24924c58e00f1ef209c00918b` is the Content and Experience Cloud Access Token.

```
{
    "LocalData": {....},
    "ResultSets": {
        "F:PUBLISH:TARGET:SYSTEMPUBLIC.Tokens": {
            "currentRow": 0, "fields": [.....],
            "rows": [[
                    "defaultToken", "a4a899f24924c58e00f1ef209c00918b", 
"01/01/2099"
                ]]
        },
        "F:PUBLISH:TARGET:SYSTEMSECURE.Tokens": {
            "currentRow": 0, "fields": [....],
            "rows": [[
                    "defaultToken", "da2286e5aa49206dd6e0efa89a9f732c", 
"01/01/2099"
                ]]
        },
        "PublishTargets": {.....}
}
```

Configuring Content and Experience Cloud with WebCenter Sites

You can configure Content and Experience Cloud with WebCenter Sites to enable the content contributors to access and use content that is available in the Content and Experience Cloud and use them to build sites.

To configure Content and Experience Cloud with WebCenter Sites through the Admin interface:

1. In the **General Admin** tree, expand the **Content Cloud Admin** node and the **Admin** node, and then double-click **Console**.
2. Click **Edit Configuration**.
3. In the Content Cloud Admin: Edit Configuration form, enter the **Base URL**, **Username**, **Password**, **Channel Name**, **Channel Token**, **Proxy Host**, and **Proxy Port**. See the following table for more information on these fields:

Fields	Description
Base URL	You can enter the Base URL of Content and Experience Cloud in this field. **Base URL** should be suffixed with `/content`. For example, `http://<content cloud host>/content`.
Username	You can enter the user name of the Content and Experience Cloud user in this field. **Username** should be prefixed with the domain name. For example, `domain1.bob@oracle.com`, where `domain1` is the name of the domain.
Password	You can enter the corresponding password of the Content and Experience Cloud user in this field.

Fields	Description
Channel Name	You can specify the name as WebCenter Sites. It is a read-only field where the channel name is fixed.
Channel Token	Initially this field is empty. Click **Refresh Token** to get the token for the WebCenter Sites. It is also a read-only field.
Proxy Host	The hostname of your proxy server.
Proxy Port	The port number of your proxy port.

4. (Optional) Click **Test** to verify the connection between WebCenter Sites and Content and Experience Cloud.
5. Click **Save** and then click **Synchronize Content** to start the synchronization process.

> **Note:**
>
> To schedule synchronizations between WebCenter Sites and Content and Experience Cloud, you can click **Set Sync Event** in the Content Cloud Admin: Console page. After you set the schedule for synchronization in the Set Sync Event page, click **Save** to save the sync schedule.

Configuring Content and Experience Cloud Properties

You can integrate Content and Experience Cloud with WebCenter Sites by configuring few properties through the **Console** in **Content Cloud Admin** node of the **General Admin** tree in the Admin interface.

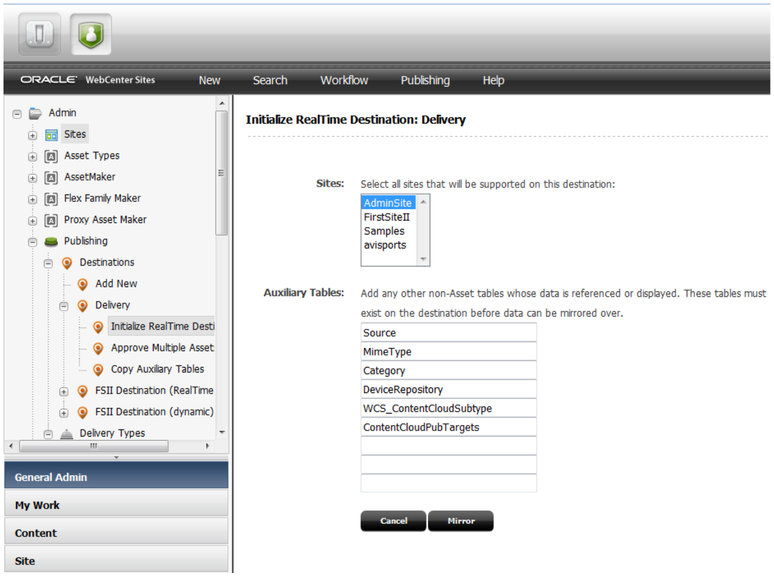
See Properties in the Content and Experience Cloud Category in the *Property Files Reference for Oracle WebCenter Sites*.

Publishing Content and Experience Cloud Assets

Just like any other WebCenter Sites assets, you can publish Content and Experience Cloud assets from WebCenter Sites.

Before you start publishing any Content and Experience Cloud assets on to the production system, ensure that the production system is also configured with Content and Experience Cloud credentials (same as source system). You can mirror the Content and Experience Cloud tables (WCS_ContentCloudSubtype and ContentCloudPubTargets) from the source system to the destination system, which may be a production system.

Figure 26-1   Publishing Content and Experience Cloud Assets



Initialize RealTime Destination: Delivery

Sites: Select all sites that will be supported on this destination:

AdminSite
FirstSiteII
Samples
avisports

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
DeviceRepository
WCS_ContentCloudSubtype
ContentCloudPubTargets

For more information, see Initializing the Destination Database.

27 Publishing Content from WebCenter Sites to Oracle Content and Experience

As an administrator, you can configure WebCenter Sites to reuse and publish content to Oracle Content and Experience. Once WebCenter Sites content is in Oracle Content and Experience, it can be delivered across other non-web channels such as Oracle Marketing Cloud, chatbots thereby fostering content reuse and consistent messaging.

To resuse WebCenter Sites content in Oracle Content and Experience:

1. You need to create an OCE mapping to include the asset types and sub types and then create a new destination and select the delivery type.
2. Approve the assets types and sub types that you have selected for publishing.
3. After publishing the assets into a zip file, upload it to Oracle Content and Experience.
4. Invoke a REST API to start importing asset types and dependencies to Oracle Content and Experience as content types and content items.

Content creators can reuse these assets in their website to deliver the content to multiple channels.

Topics:

- Configuring Assets Types
- Creating OCE Destination
- Approving Assets for Publishing
- Publishing Assets
- Downloading Published Assets
- Importing Published Assets to Oracle Content and Experience
- Limitations

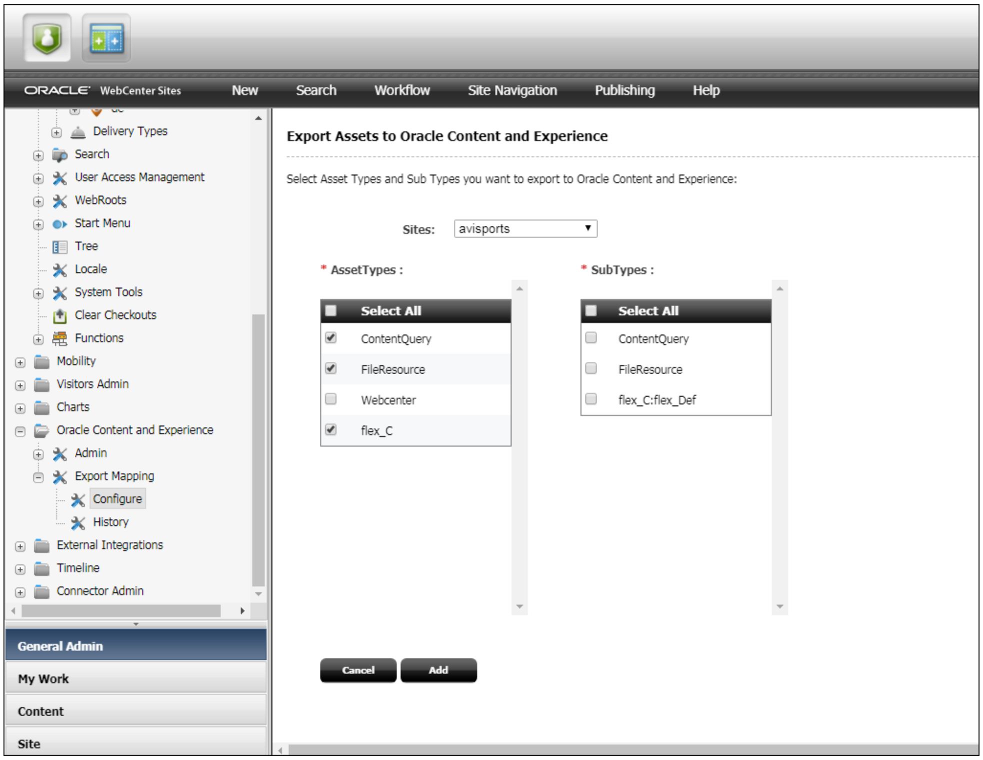
Configuring Assets Types

Create an OCE mapping to include the asset types and sub types.

To configure assets types:

1. On the source WebCenter Sites system, log in to the Admin interface.
2. In the General Admin tree, expand the **Content Cloud Admin** node, expand the **Export Mapping** node, and double-click **Configure**.
3. Click the **Add New OCE Mapping** button.
4. Select **Asset Types** and **Sub Types**, and then click **Add**.

Figure 27-1 Adding Assets and Assets Types



Export Assets to Oracle Content and Experience

Select Asset Types and Sub Types you want to export to Oracle Content and Experience:

Sites: avisports

* AssetTypes :

- □ Select All
- ☑ ContentQuery
- ☑ FileResource
- □ Webcenter
- ☑ flex_C

* SubTypes :

- □ Select All
- □ ContentQuery
- □ FileResource
- □ flex_C:flex_Def

Creating OCE Destination

You need to create a new OCE destination on WebCenter Sites and select the required delivery type.

To create a new destination:

1. In the General Admin tree, expand the **Admin** node, expand the **Publishing node**, and then expand the **Destinations** node.
2. Under the Destinations node, double-click **Add New**.
3. In the Add New Destination form, configure the fields on this form as follows:
   - **Name**: Enter a unique name for the destination.
   - **Delivery Type**: Select **Export to OCE: Export to Oracle Content and Experience**.
   - **Send Email on Failure**: If publishing fails and email notices to that effect are required, select the check box. Enter the recipient's email address in the **Email Addresses** field. (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 can 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 are allowed to approve assets.
- **Roles (Publish)**: Select the roles to which you are assigning publish privileges. All users assigned these roles can publish.

4. Click **Add New Destination**.

Approving Assets for Publishing

For publishing your assets, the assets and their dependencies need to be approved first. Each publishing destination has an option to approve multiple assets.

To approve a group of assets:

1. In the General Admin tree, expand the **Admin** node, expand the **Publishing node**, expand the **Destinations** node, and then expand the destination to approve assets for.
2. Under that destination, double-click the **Approve Multiple Assets** option.
3. In the Approve Assets form, from the Sample Queries list on the right, select a query. If the exact query that you require is not present, select the query that is most like the one wanted.

   WebCenter Sites creates a SQL query based on the items that you selected and displays it in the Query box.
4. (Optional) If necessary, edit the SQL query.
5. Click in the **Approve in Batches** of field and enter a numeric value. The default is set to 500.
6. Specify the **Approve Previously Approved Assets** option as follows:
   - If you want WebCenter Sites to ignore all previous approvals and approve again, select **Yes**. This is the option to use if previous approvals have become invalid 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.
7. Click **Approve for Publish**.

Publishing Assets

After approving the assets, you need to publish them. The process for publishing assets for OCE destination is same as for any other destination.

To publish your approved assets:

1. In the button bar, click **Publishing**.

2. In the Publish Console, select your destination, for example, *PublishToOCE* of delivery type **Export to OCE** from the drop-down list and then click **Select Destination**.

   Review the assets that are ready to be published.

3. Click **Publish**. The approved assets are added to a .zip file.

   A confirmation message opens. Click **OK** to proceed.

   On publishing the assets, a zip file having approved assets is created.

4. On 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.

6. Mouse over the session information to view summary information; click **Log** to view the session log files.

Downloading Published Assets

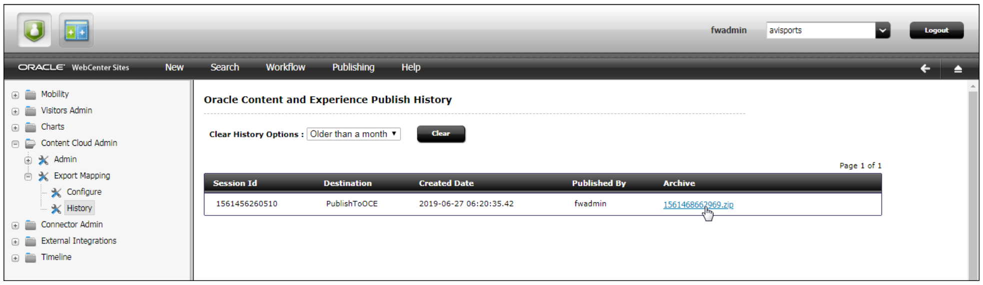
Publishing to OCE destination creates a .zip file consisting of all the assets and its type. You can download the zip file on your local system.

To download published assets:

1. In the General Admin tree, expand the **Content Cloud Admin** node, expand the **Export Mapping** node, and double-click **History**.

2. The Oracle Content and Experience Publish History page shows the history of the published sessions.

Figure 27-2 Publish History



Oracle Content and Experience Publish History

Clear History Options : Older than a month

Page 1 of 1

Session Id	Destination	Created Date	Published By	Archive
1561456260510	PublishToOCE	2019-06-27 06:20:35.42	fwadmin	1561468662969.zip

3. You can download the zip file of your destination. The zip file contains the published assets. These assets will be mapped to the content types of Oracle Content and Experience.

Importing Published Assets to Oracle Content and Experience

Importing the published assets to Oracle Content and Experience involves two steps:

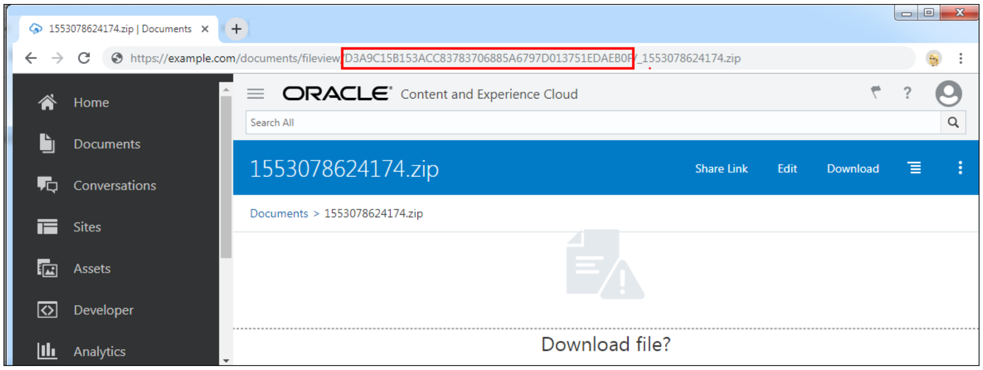
- Upload the zip file to Oracle Content and Experience.

- Call a REST API to import the published asset into an asset repository.

To import published assets:

1. Log in to Oracle Content and Experience.
2. Select **Documents** from the left bar, and then click **Upload** to upload the zip file from your local system
3. After the upload, click on the zip file. In the address bar of the browser, the URL shows the Document ID.

Figure 27-3 Document ID



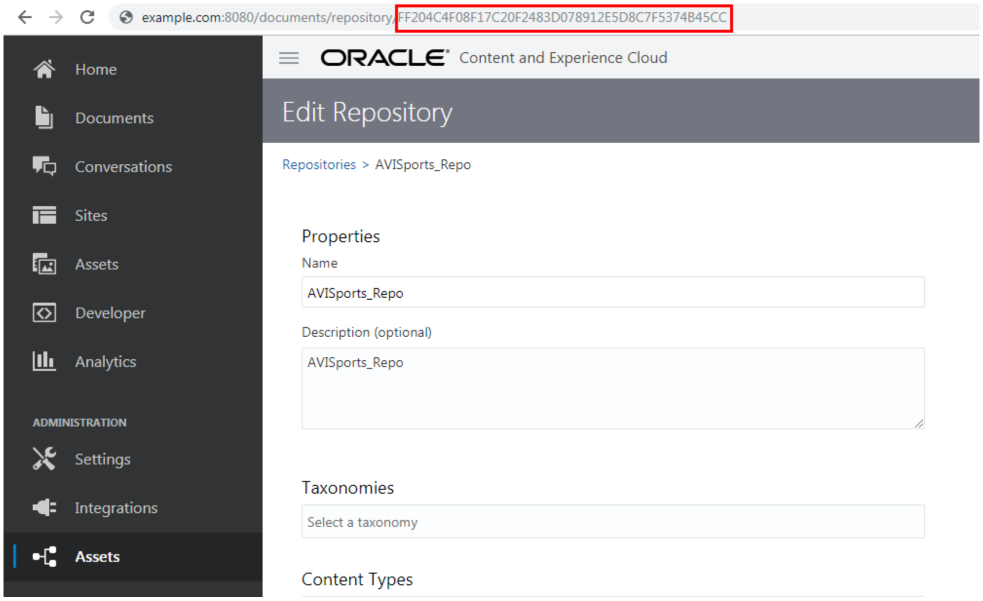
1553078624174.zip | Documents

https://example.com/documents/fileview/D3A9C15B153ACC83783706885A6797D013751EDAEB0F/_1553078624174.zip

4. You need to create a new or use an existing repository to import content from WebCenter Sites. To create a repository, click **Assets** at the bottom of the left bar and then click **Create**.

   After the new repository is created, click on your repository. In the address bar of the browser, the URL shows the Repository ID.

Figure 27-4 Repository ID



example.com:8080/documents/repository/FF204C4F08F17C20F2483D078912E5D8C7F5374B45CC

Properties

Name: AVISports_Repo

Description (optional): AVISports_Repo

Taxonomies: Select a taxonomy

Content Types

> **Note:**
>
> Note Document and Repository IDs because both are required to run the REST API.

5. Start the REST client to import the published assets from the zip file to Oracle Content and Experience as content types and content items. The POST request must have the following settings:

   a. Select the Authorization type as *No Auth*.

   b. To find the token value, enter the following URL in the address bar of your browser: `https://example.com/documents/web?IdcService=GET_OAUTH_TOKEN&IsJson=true`

   c. The POST command using the following URL:

   ```
   http://example.com:port/content/management/api/v1.1/content-templates/importjobs
   ```

   > **Note:**
   >
   > The API returns the job id when you execute a POST command so that you can track the job.

   d. Enter the following key values in the Headers:

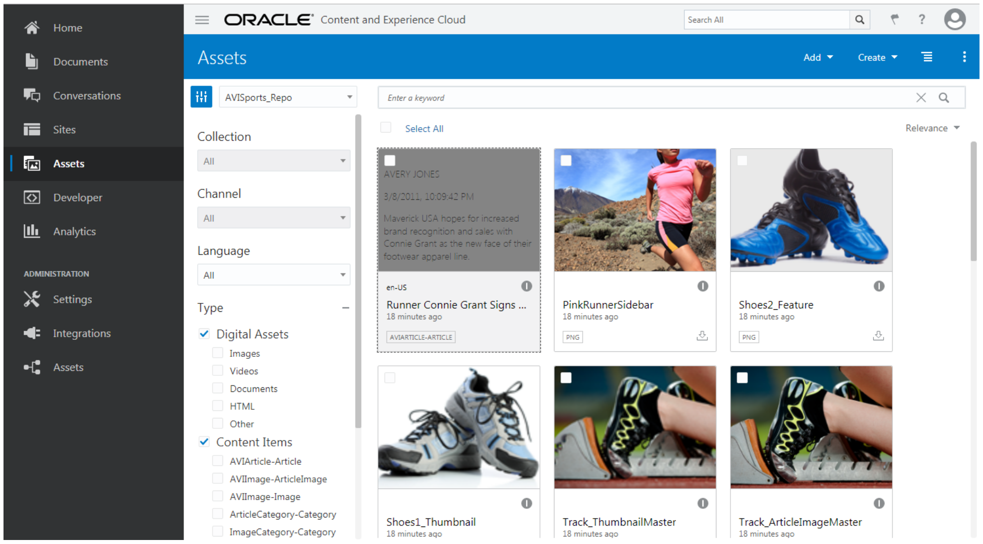
   ```
   Accept : application/json

   X-Requested-With : XMLHttpRequest

   Content-Type : application/json

   Authorization : Bearer <token value>
   ```

   e. Enter the following code in the raw section of the Body:

   ```
   {

   "exportDocId": "<DOC_ID>",

   "repositoryId": "<REPO_ID>",

   "source" : "sites"

   }
   ```

6. When the import is successful, you can log in to Oracle Content and Experience.
7. To view the imported assets, click **Assets** in the middle of the left bar and select your repository from the drop-down menu. The imported assets show up as content items. They can be filtered by their type (Content Type).

Figure 27-5 View Imported Assets



Limitations

The following are the limitations of this tool:

- Proxy assets are not published.
- Embedded assets are not published as is.
- Engage or recommendations are not published.
- All non-content assets such as Templates, CSElements, and Controllers are not published.

Part XII

Managing and Tuning Caching

WebCenter Sites uses inCache as its caching system. inCache helps you improve website performance better than the legacy method of caching to the database.

For information about the inCache framework, see the following topic:

- Using The inCache Framework

28
Using The inCache Framework

The caching system, the inCache framework, or simply inCache is built on top of Ehcache, an open source, standards-based product from Terracotta.

- Overview
- Understanding How inCache Works
- Restarting a Node
- Regenerating Pages During RealTime Publishing
- Double-Buffered Caching
- Configuring Your System for inCache Page Caching
- Tuning Options
- Working with the AssetCache Container
- Working with Asset Caching Operations
- Types of Asset Caching
- Customizing Asset Caching
- Disabling Asset Caching
- Using the Cache Management Tool
- Using inCache Features for Remote Satellite Server

Overview

inCache is a memory-based page and asset caching system that eliminates the requirement to cache 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/`.)

> **Note:**
>
> WebCenter Sites uses ehcache to cache the content at various levels. Ehcache may become out of sync if one of the nodes in the Sites cluster faces network outage. It is recommended to restart the node which faces the network outage to keep the cache in sync at various nodes.

This section contains the following topics:

- Overview of the Caching Framework
- Overview of Page Caching
- Overview of Asset Caching

Overview of the Caching Framework

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 allows for more frequent publishing sessions. Efficient page and asset invalidation ensures 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 needs 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 Understanding How inCache Works, for a basic understanding of cache containers and how they function in relation to each other.

Overview of Page Caching

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 and Satellite Server nodes as follows: `-Dcs.useEhcache=false`
2. Reconfiguring all and Satellite Server nodes to use the database and shared file system.
3. Purging 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 and Satellite Server nodes to use the same cache, and to purge the cache before its use, as recommended by best practices.

Overview of 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 the 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.

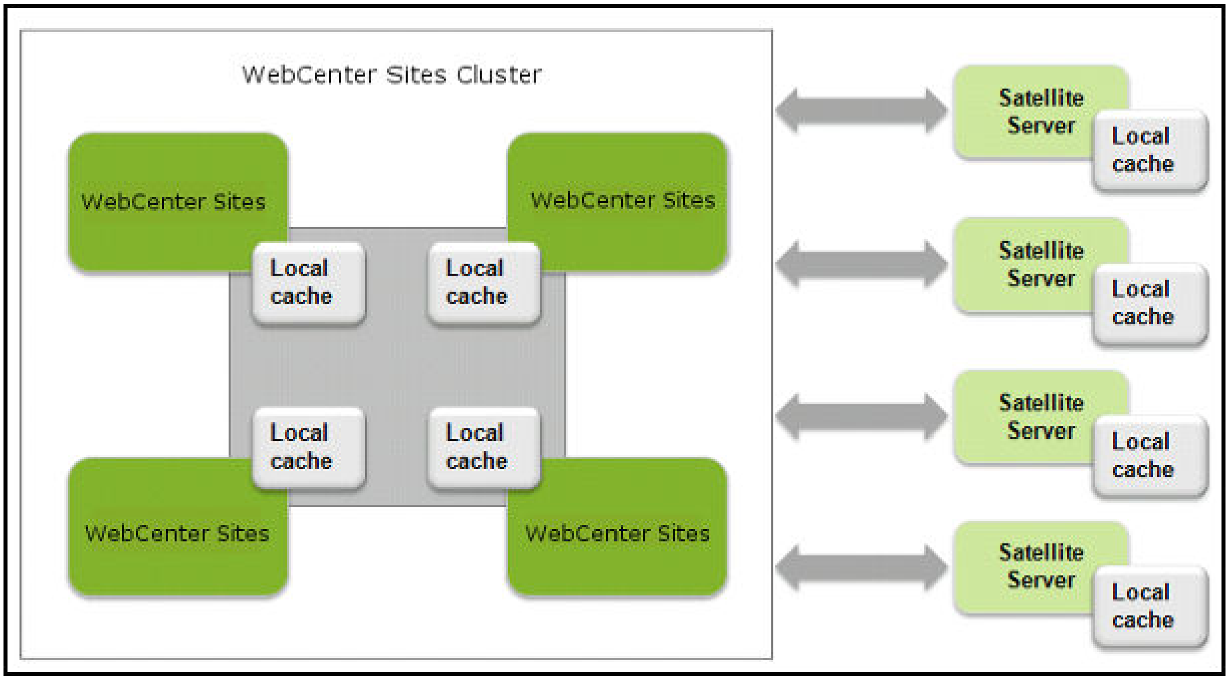
Note that while the pageByQuery cache is not used in asset caching, it serves to illustrate how a cache container works. Its counterpart AssetCache container works in a similar manner. Note that asset caching and inCache page caching work independently of each other. Each type of caching can be enabled or disabled independently of the other.

Working with the AssetCache Container describes the AssetCache container, discusses it 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.

Understanding 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 the following figure. Each WebCenter Sites node keeps its local cache current by listening for broadcasts from other WebCenter Sites nodes and communicates updates to remote Satellite Server through HTTP.

Figure 28-1 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:
  1. A WebCenter Sites user at node `A` edits asset `A`.
  2. The notifier cache on node `A` broadcasts a notice of change to all other nodes.
  3. Every WebCenter Sites node that contains asset `A` responds by invalidating the asset in its own local cache. Each node sees 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 sees its `pageByQry` cache, evaluates the page, determines the page to be invalid, flushes it, generates the new page, serves the page to the visitor's browser, and records the new page and its dependencies in the local cache (dependencies are recorded as identifiers of the assets that make up the page). From then on, the 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, though Satellite Server nodes still communicate with WebCenter Sites nodes through 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. See Working with Asset Caching Operations.

Restarting a Node

When a 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 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 were modified or deleted during a content management or publishing process. Table growth is checked by a 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 sees 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 for an update.

Regenerating Pages 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. If crawling is not implemented, pages are regenerated only when they are requested.

The crawling option requires specifying a set of URLs for the page regenerator to analyze. Typically, they are the URLs of the home page and other high-traffic pages. You can also specify the depth those pages are crawled. For example, a depth of 1 means that the specified pages and pages they link to are crawled, while 0 means that only the specified pages are 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 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. See Working with Asset Caching Operations.

Double-Buffered Caching

The double-buffered page caching method of Webcenter Sites uses the and Satellite Server caches in tandem on live web sites. Double buffering ensures that pages are always kept in cache, either on or Satellite Server, to protect 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 through http requests. Satellite Server still reads page data through 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.

Configuring Your System for inCache Page Caching

The configuration process consists of a set of required steps and optional optimization steps. For example, you 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 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 system (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, see the documentation on the Ehcache web site:

`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:
>
> - Disk persistence is turned off.
> - 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. Configuring inCache Ports. inCache relies on Ehcache's RMI-based transport mechanism for communication between cluster members. While ordinarily this requires no additional setup, it is possible to specify the exact ports for Ehcache to use. This is particularly important when there is a firewall between cluster members.

   It is recommended that you don’t use cache replication for Satellite servers to avoid network overhead. To enable cache replication, set the multicastGroupPort property in each ss-cache.xml file to the same value.

   For more information about configuring Ehcache to use specific ports, see the documentation on the Ehcache website:

   `http://ehcache.org/files/documentation/EhcacheUserGuide-1.6.pdf`

5. 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.

   > **Note:**
   >
   > In some instances, firewalls on the inCache node network prevent communication between nodes. Disable those firewalls when installing and configuring inCache for multiple nodes. Once verified the System Administrator/Network Administrator should monitor the ports during inCache testing and page propagation to Remote Satellite Server and adapt their firewalls accordingly.

   a. On each WebCenter Sites node ensure that `cs-cache.xml` and `ss-cache.xml` (in `WEB-INF/classes`) specify a `cacheManagerPeerListenerFactory`, which is 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 either:

   - 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.

6. Optional optimization. Configure the local caches as necessary, using Table 28-1 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.

7. Restart all configured WebCenter Sites and Satellite Server nodes.

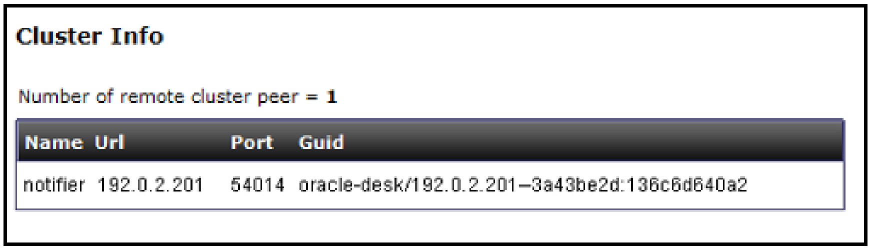
8. Verify that all active cluster members are in the caching network and recognize each other. Use the Cluster Info diagnostic tool, which lists all WebCenter Sites members that have a notifier cache. To call Cluster Info:

   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. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **Cluster Info**.

   Figure 28-2 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.

9. If you wish to stripe disks, enable page regeneration during RealTime publishing, or enable page propagation, see Tuning Options.

Table 28-1 Cache Configuration Properties

Property	Required?	Description
`name`	Y	Specifies the name of the cache. **Legal Values:** • `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.
`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. See Striping the Disk Cache. **Default value:** `1000000`
`eternal`	Y	Specifies whether cache is 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`

Table 28-1 (Cont.) Cache Configuration Properties

Property	Required?	Description
`clearOnFlush`	N	Specifies whether to clear memory as it is flushed to disk. **Default value:** `false` **Do not change the value of this property.**

Tuning Options

There are many options in the inCache framework which can be optimized to your system. This section is a collection of those options.

- Striping the Disk Cache
- Configuring for Page Regeneration During RealTime Publishing
- Setting Up Page Propagation
- Configuring for Pagelet Regeneration in the Background

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 where *X* is the number of stripes:

   ```
   -DnumOfDiskStores=X
   ```

   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 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 through 4 are the stripes of the pageByQry cache. The directories 0 through 4 in this example were placed on five separate drives by the use of symbolic links.

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 installer to use ParallelCacheRegenerator:

   Open the AdvPub.xml file in the config folder on the delivery system and verify that the PageCacheUpdater section has the code lines shown below.

> **Note:**
>
> Do not change any of the values in the PageCacheUpdater section except for numThreadsPerServer that is used to specify the number of simultaneous threads for crawling.

```
<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 are regenerated/
servlet/ContentServer</value>
     </list>
   </property>
</bean>
```

Use the Property Management tool to update `regenServers` to point to the WebCenter Sites nodes by `<address>` and `<port of server where pages are regenerated>`. You can add complete `pagename` with arguments to preview the page.

2. Restart the delivery system.
3. RealTime publish to the delivery to create the `FW_RegenCriteria` table on that system.
4. Open the `FW_RegenCriteria` table, using 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 are handled by 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
```

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 .

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.

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, no 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 following scenario, 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 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 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 covers the following topics:

- Enabling Page Propagation
- Setting Up Page Propagation on Restart

Enabling Page Propagation

Start with a system that is configured to use inCache.

To enable page propagation:

1. On all WebCenter Sites nodes:
   a. In the Property Management Tool, set `propagatecache=true`. Ensure to check **Global** to set this property for all nodes.
   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 `wcs_properties.json` file for remote satellite servers. This file is located in the config folder.

> **Note:**
>
> The config folder for Satellite Server is placed in a user-defined location during installation. For clustered environments, the recommended location is under the sites shared directory (<`WCSITES_SHARED DIR`>). For non-clustered environments, he recommended location is outside of the home directory (<`WCSITES_PRODUCT HOME`>). If you cannot locate the config folder, contact your installation engineers.

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 follows:
   - 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 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 exists.
   - The response on Satellite Servers proceeds as on WebCenter Sites nodes, but over Java RMI.

Setting Up Page Propagation on Restart

If a node enabled for page propagation is restarted, its local caches must be re-initialized for the node to recognize page propagations.

To reinitialize a local cache:

1. 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.

2. Restart a node. The propagations it missed during its period of inactivity are not reproduced on the node, even though its local caches are reinitialized.

Configuring for Pagelet Regeneration in the Background

Remote Satellite Server can be configured to serve invalidated pagelets while they are being regenerated by a background process.

To enable the serving of invalidated pagelets:

- Add `serveStale=true` to the remote Satellite Server's `wcs_properties.json` file (located under *`WCSites_Shared_Dir`* for clustered environments and under *`WCSites_Product_Home`* for nonclustered environments).

  If a pagelet is then invalidated, it is regenerated in these 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 are 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 the `logging-config.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."
  ```

  After the process, the message reads:

  ```
  "Background process for request <cache key> has completed successfully,
  data in cache is updated."
  ```

Working with the 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. See Types of Asset Caching and 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 28-2. 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 through Oracle WebCenter 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.

The following table shows how asset caching works independently of page caching, but still shares various cache containers.

Table 28-2 Cache Containers Used by Asset and Page Caching

Asset Caching Function	Page Caching Function	`AssetCache` Container	`pageByQry` Container	`dependencyRepository` 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

Working with Asset Caching Operations

This section covers the following topics:

- Working with Asset Loading
- Comparing Asset Caching with inCache Page Caching
- Flushing AssetCache

Working with Asset Loading

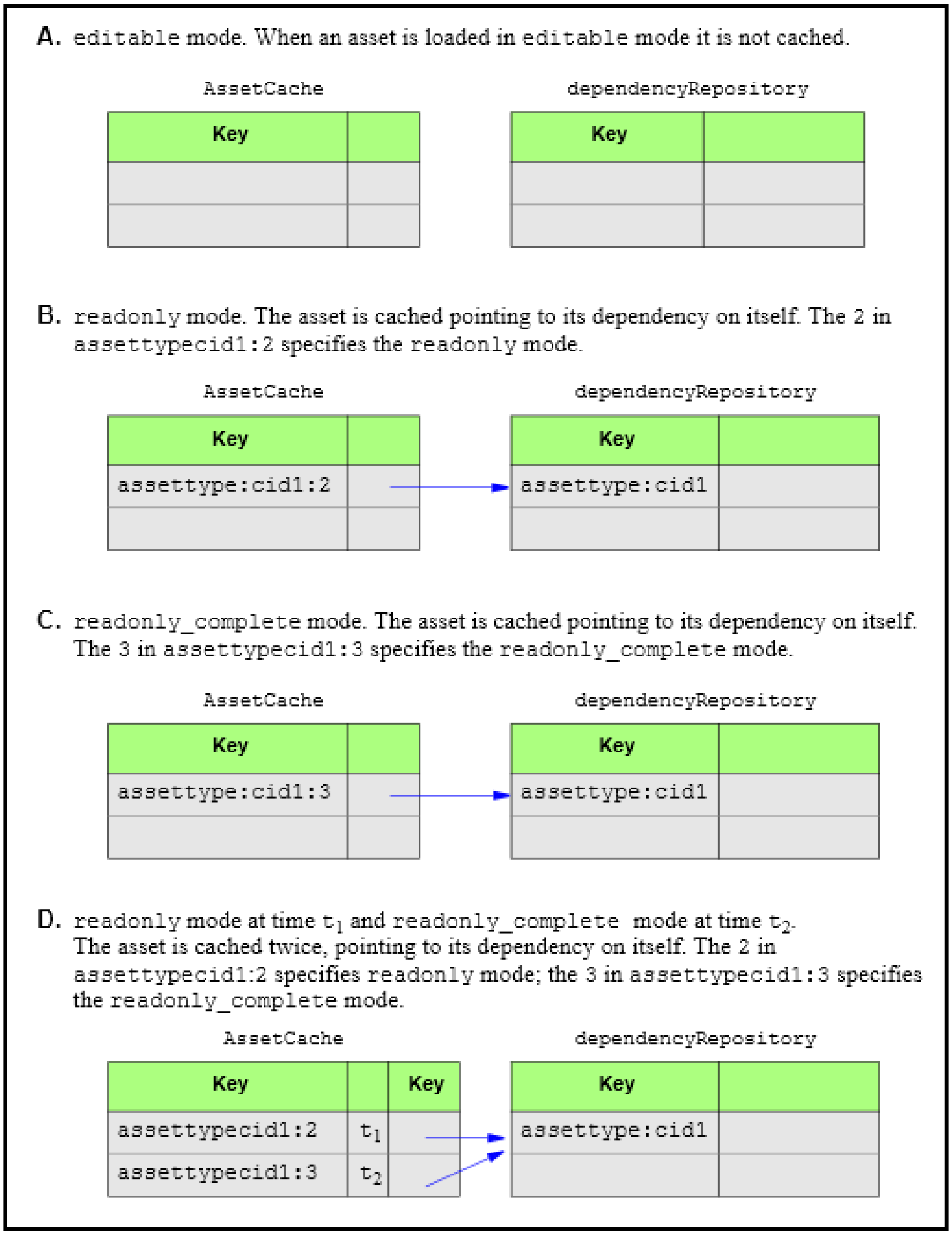
An asset's data can be read in several ways, by using APIs or asset:load tags.

- 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 28-3. 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 readonly_complete.

The following modes determine whether an asset is cached and how its key is named in the AssetCache and dependencyRepository.

Figure 28-3 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 assettypecid1:2 specifies the readonly mode.

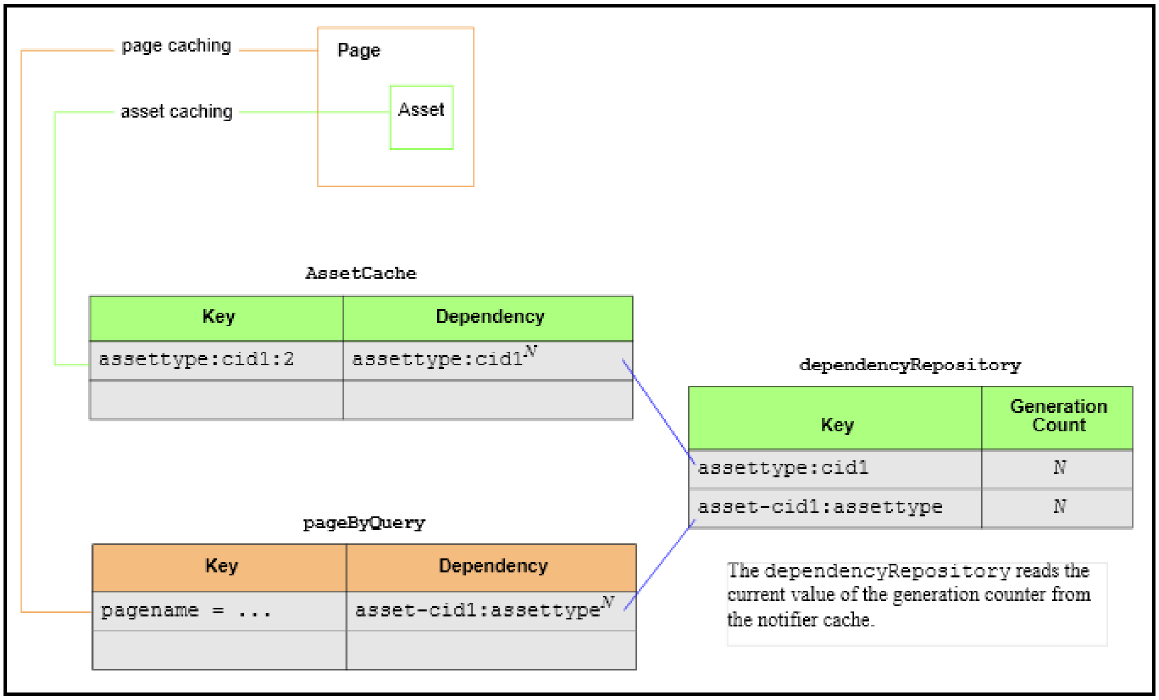
C. readonly_complete mode. The asset is cached pointing to its dependency on itself. The 3 in assettypecid1:3 specifies the readonly_complete mode.

D. readonly mode at time t_1 and readonly_complete mode at time t_2. The asset is cached twice, pointing to its dependency on itself. The 2 in assettypecid1:2 specifies readonly mode; the 3 in assettypecid1:3 specifies the readonly_complete mode.

Comparing Asset Caching with inCache Page Caching

Keys are named differently for cached assets and pages, and for their dependencies, as shown in the following figure. Understanding the differences helps to interpret caching information shown in the Cache Management tool (available under the **System Tools** node, in the **Admin** node of the **General Admin** tree).

Figure 28-4 Dependence Naming Conventions in Asset Caching and Page Caching



AssetCache

Key	Dependency
assettype:cid1:2	assettype:cid1^N

dependencyRepository

Key	Generation Count
assettype:cid1	N
asset-cid1:assettype	N

pageByQuery

Key	Dependency
pagename = ...	asset-cid1:assettype^N

The dependencyRepository reads the current value of the generation counter from the notifier cache.

For example, this preceding figure 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.

The key of a cached page begins with `pagename`. Mode does not apply to pages. If a page is cached, it is cached only one time.

See Working with Asset Loading and About Using the Cache Management Tool.

Flushing AssetCache

Assets can be flushed from the AssetCache if they are saved through 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`

Types of Asset Caching

When asset caching is enabled on a WebCenter Sites node, the `cs-cache.xml` file on that node contains one or more 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 AssetCache*Asset_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** node of the **General Admin** tree) shows dialogs showing the contents of `AssetCache` containers, corresponding entries in the dependency Repository, and other types of information. Searches are also supported.

See Customizing Asset Caching and About Using the Cache Management Tool.

Customizing Asset Caching

These steps show how to configure type-specific asset caches.

To configure asset caching:

1. When default asset caching is enabled on a node, you will see the following entry in the `cs-cache.xml` file on nodes, which specifies that assets of all types are cached in a generic container named `AssetCache` (parameters, other than `name`, are defined in Table 28-2):

   ```
   <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 are cached. Assets having their own type-specific container are cached exclusively in their own container, which can be individually tuned by use of the parameters shown in Table 28-2.
   > - You can replace the generic AssetCache container with one or more type-specific containers. Only assets of the named types are 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="AssetCache`*`Asset_type_name`*`"` (only one asset type is supported for each custom `<cache>` element). Configure the cache by using the parameters in Table 28-2.

   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 does not cache assets of type Content_C.

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. See Types of Asset Caching.

Using the Cache Management Tool

When asset caching is configured, the Cache Management tool shows dialogs showing the contents of AssetCache containers and corresponding entries in the dependencyRepository, among other types of information. Searches are also supported. See About Using the Cache Management Tool.

Using 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.

Page propagation

The page propagation option enables all WebCenter Sites nodes and Satellite Server nodes to host the same pages without each node regenerating pages. Nodes receive generated and regenerated pages into local caches from the nodes the pages were (re)generated and cached from. Page caching triggers the propagation. For more information about page propagation, see Setting Up Page Propagation.

Background page regeneration

Remote Satellite Server can be configured to serve invalidated pagelets while they are being regenerated by a background process. See Configuring for Pagelet Regeneration in the Background.

Part XIII

Monitoring and Maintaining WebCenter Sites

To monitor and maintain the content management sites in WebCenter Sites, familiarize yourself with the processes of troubleshooting sites, replicating sites, and tracking various versions of assets. Also become familiar with configuring easy to read and manage site URLs and third-party search engines.

Topics:

- Troubleshooting Using System Tools
- Replicating Content Management Sites
- Managing Revision Tracking
- Configuring Vanity URLs
- Configuring the Lucene Search Engine
- Using the Site Capture Application
- Administering WebCenter Sites and Its Components in Enterprise Manager

29

Troubleshooting Using System Tools

A wide range of diagnostic tools is available for you on the **System Tools** node under the **Admin** node of the **General Admin** tree. With these diagnostic tools, you can configure log4j loggers, access various types of system information, manage caches, and test the performance of the shared file system.
Topics:

- Using the Configure Log ODL Tool
- Using the System Information Tool
- About Using the Cache Management Tool
- Working with Cache Management - CAS Cache
- Working with Cache Management - Resultset Cache
- Working with Cache Management - Sites Cache
- Working with Cache Management - Co-Res Cache
- Using the Log Viewer
- Performance Test
- Working with Slots Assets
- Working with the URL Node
- Working with the Purge Voided Assets Node

Using the Configure Log ODL Tool

By default WebCenter Sites is set up to use the Oracle Diagnostic Logging (ODL) system. The Configure Log ODL tool enables you to view current logger levels, change logger levels, and add new loggers directly from the Admin 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 manually adding new loggers to the `logging-config.xml` file.

The following topics provide information about using the Configure Log ODL tool in the Admin interface:

- Viewing Loggers
- Changing Logger Levels
- Adding New Loggers

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 `logging-config.xml` file, in the `<sites-shared>/config` directory. The file is read only on system startup. If, during runtime, you make changes to the `logging-config.xml` file and save them, your changes are not displayed in the **Log ODL Configuration** form. They are 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 Adding New Loggers.

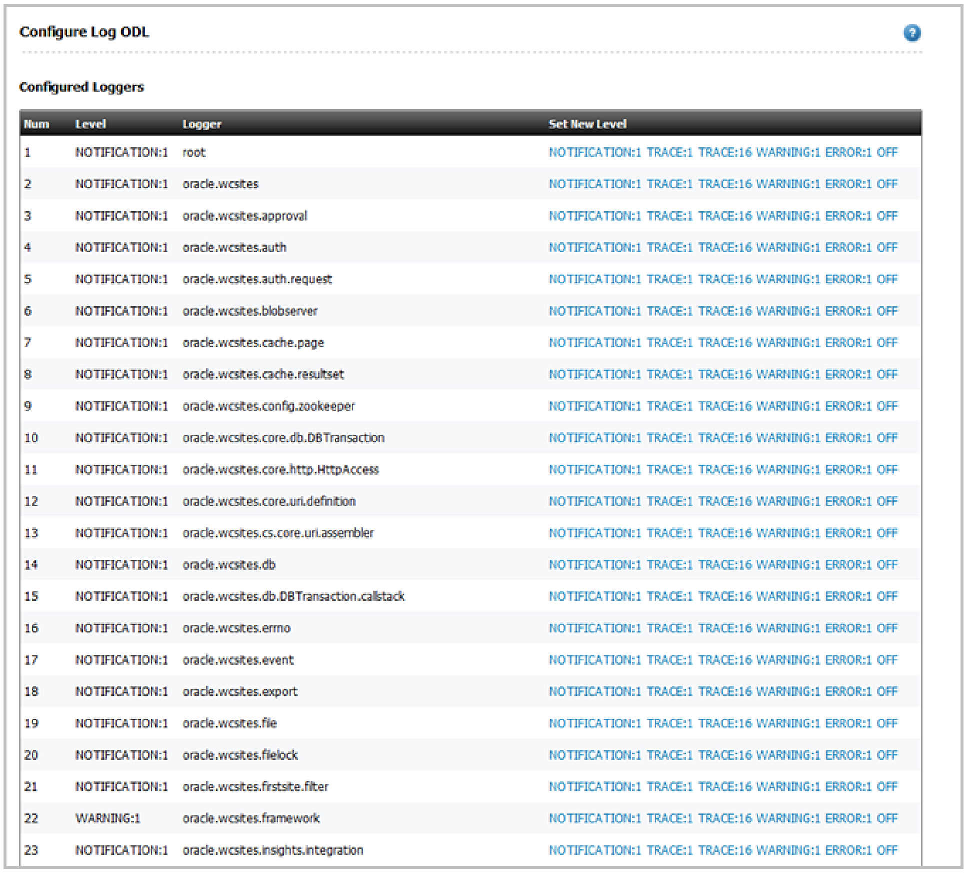
The collapsed view of the list shows the root logger and parent loggers, all of which are explicitly assigned verbosity levels. If you require a more granular view, you can expand the list to also show loggers that inherit levels from the root logger or parent loggers.

To view loggers:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Configure Log ODL**.

   The list of loggers with explicitly assigned levels is shown in the following figure.

Figure 29-1 Log ODL Configuration Form



Configure Log ODL

Configured Loggers

Num	Level	Logger	Set New Level
1	NOTIFICATION:1	root	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
2	NOTIFICATION:1	oracle.wcsites	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
3	NOTIFICATION:1	oracle.wcsites.approval	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
4	NOTIFICATION:1	oracle.wcsites.auth	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
5	NOTIFICATION:1	oracle.wcsites.auth.request	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
6	NOTIFICATION:1	oracle.wcsites.blobserver	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
7	NOTIFICATION:1	oracle.wcsites.cache.page	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
8	NOTIFICATION:1	oracle.wcsites.cache.resultset	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
9	NOTIFICATION:1	oracle.wcsites.config.zookeeper	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
10	NOTIFICATION:1	oracle.wcsites.core.db.DBTransaction	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
11	NOTIFICATION:1	oracle.wcsites.core.http.HttpAccess	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
12	NOTIFICATION:1	oracle.wcsites.core.uri.definition	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
13	NOTIFICATION:1	oracle.wcsites.cs.core.uri.assembler	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
14	NOTIFICATION:1	oracle.wcsites.db	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
15	NOTIFICATION:1	oracle.wcsites.db.DBTransaction.callstack	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
16	NOTIFICATION:1	oracle.wcsites.ermo	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
17	NOTIFICATION:1	oracle.wcsites.event	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
18	NOTIFICATION:1	oracle.wcsites.export	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
19	NOTIFICATION:1	oracle.wcsites.file	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
20	NOTIFICATION:1	oracle.wcsites.filelock	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
21	NOTIFICATION:1	oracle.wcsites.firstsite.filter	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
22	WARNING:1	oracle.wcsites.framework	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 OFF
23	NOTIFICATION:1	oracle.wcsites.insights.integration	NOTIFICATION:1 TRACE:1 TRACE:16 WARNING:1 ERROR:1 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.

3. To view only loggers with explicitly assigned levels again, click **Show Configured Loggers** at the bottom of the form.

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 `logging-config.xml` 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 change the logger value manually in the `logging-config.xml` file, in the `<sites-shared>/config` 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:16 and you want to prevent one of its child loggers from inheriting that level, you can assign the child logger a different level.

> **Note:**
>
> Log4j log levels set by users cannot be migrated to ODL logs. Users must set these levels manually after the upgrade.

The following logger levels are available, listed in order of decreasing verbosity:

`ERROR:1, WARNING:1, NOTIFICATION:1, TRACE:16, TRACE:1, OFF`

OFF indicates that a logger does not write any statements to the log. For more information about logger levels, see the Oracle Diagnostic Logging (ODL) documentation.

To change a logger level:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Configure Log ODL**.

   The list of loggers with explicitly assigned levels opens. 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 to change and select a new logger level in the **Set New Level** column.

   The new level opens in the Level column.

   When the logger is executed, you can track the impact of the new level in the log. See Viewing the Log.

3. If you want the new logger level to remain in effect after WebCenter Sites is restarted, you must manually change the logger level in the logging-config.xml file. For instructions, see Changing Logger Levels in *Property Files Reference for Oracle WebCenter Sites*.

Adding New Loggers

If you wish to add new loggers to the ODL 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 through the Admin 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 manually add the logger properties to the `logging-config.xml` file in the `<sites-shared>/config` directory on your application server.

To add a new logger:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Configure Log ODL**.

   The list of loggers with explicitly assigned levels is shown.
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 Viewing the Log).
4. If you want the added logger to remain in effect after WebCenter Sites is restarted, add the logger to the `logging-config.xml` file. For instructions, see Adding Loggers in *Property Files Reference for Oracle WebCenter Sites*.

Using the System Information Tool

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 covers the following topics:

- Viewing WebCenter Sites Information
- Viewing Database Information
- Viewing 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)**.

Viewing 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:

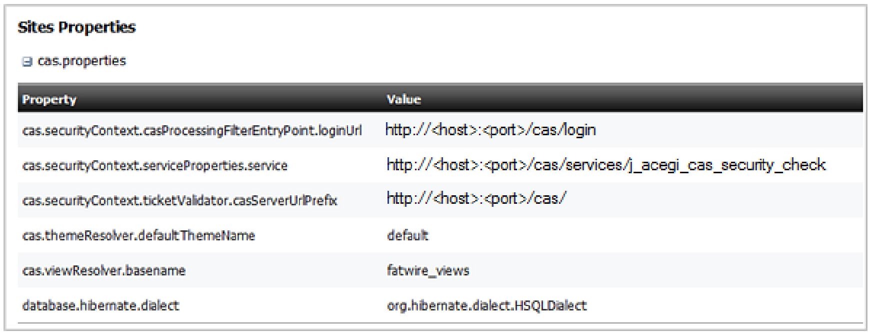
1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, then expand the **System Information** node, and then double-click **Sites Info**.

   The WebCenter Sites Information form is shown.

   This form shows memory information and additional information that you can select to download or view.
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, and WebCenter Sites session variables and system variables, the properties and values are shown in the following figure.

Figure 29-2 Properties and Values for cas.properties



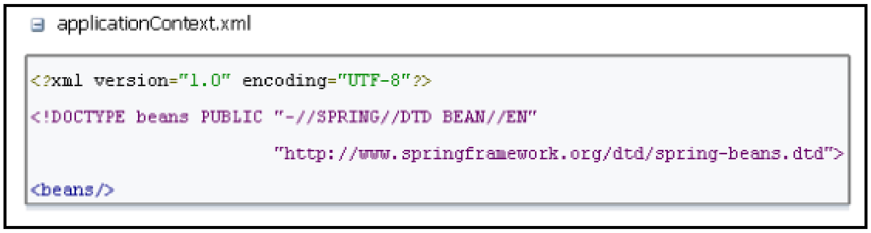
Sites Properties

⊟ cas.properties

Property	Value
cas.securityContext.casProcessingFilterEntryPoint.loginUrl	http://<host>:<port>/cas/login
cas.securityContext.serviceProperties.service	http://<host>:<port>/cas/services/j_acegi_cas_security_check
cas.securityContext.ticketValidator.casServerUrlPrefix	http://<host>:<port>/cas/
cas.themeResolver.defaultThemeName	default
cas.viewResolver.basename	fatwire_views
database.hibernate.dialect	org.hibernate.dialect.HSQLDialect

For `xml` files, the code is shown in the following figure.

Figure 29-3 XML File Code Display

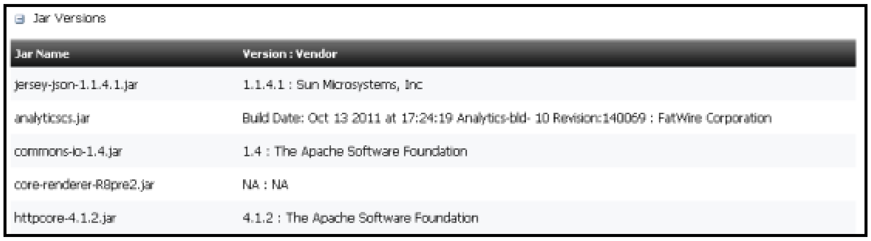


⊟ applicationContext.xml

```
<?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 shown in the following figure.

Figure 29-4 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:140069 : 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

`NA : NA` indicates that a `jar` file does not provide its version and vendor.

Viewing 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 Access Control List (ACL) assigned to the table is assigned to your user. See Access Control Lists (ACLs) and Types of Database Tables in *Developing with Oracle WebCenter Sites*.

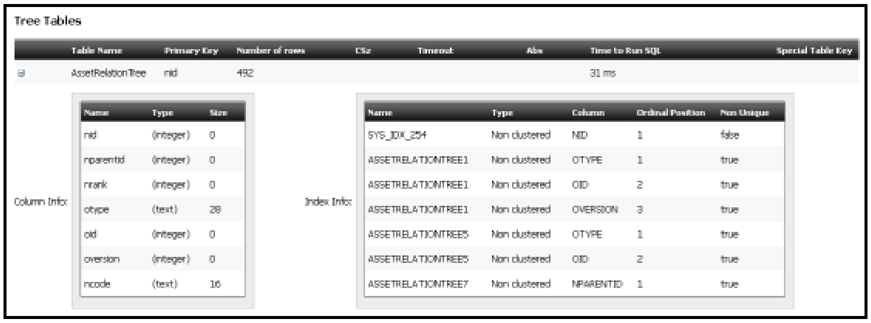
To download or view database information:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **System Information**, and then double-click **DB Info**.

   The Database Information form is displayed with basic information about your database and the tables that you can select to download or view.
2. In the System Tables, Tree Tables, Sites Object Tables, and Other Sites Tables sections, select the tables 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 opens:
      - 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 one time, it extends across multiple rows, as illustrated in the following figure.

Figure 29-5 Tree Tables Information



Tree Tables

	Table Name	Primary Key	Number of rows	CSz	Timeout	Abs	Time to Run SQL	Special Table Key
⊟	AssetRelationTree	nid	492				31 ms	

Column Info:

Name	Type	Size
nid	(integer)	0
nparentid	(integer)	0
nrank	(integer)	0
otype	(text)	28
oid	(integer)	0
oversion	(integer)	0
ncode	(text)	16

Index 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	OVERSION	3	true
ASSETRELATIONTREE5	Non clustered	OTYPE	1	true
ASSETRELATIONTREE5	Non clustered	OID	2	true
ASSETRELATIONTREE7	Non clustered	NPARENTID	1	true

Viewing 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 could result in resource starvation or thread hangs.

To download or view thread information:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **System Information** node, and then double-click **Thread Info**.

   The Thread Information form is shown.

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 to search for.

4. 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 are 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:
>
> - 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 JDK.
>
> If a compatible `jstack` is not found, the following message is shown: "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 opens:

   - 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

About Using the Cache Management Tool

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 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.

> **Note:**
>
> The Cache Management tool provides caching information only for 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
> ```

Working with Cache Management - CAS Cache

This section shows you cache management for CAS Cache.

This section includes the following topics:

- Overview of the CAS Cache Summary Form
- Working with Cluster Information

Overview of the CAS Cache Summary Form

The CAS Cache Summary 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 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.

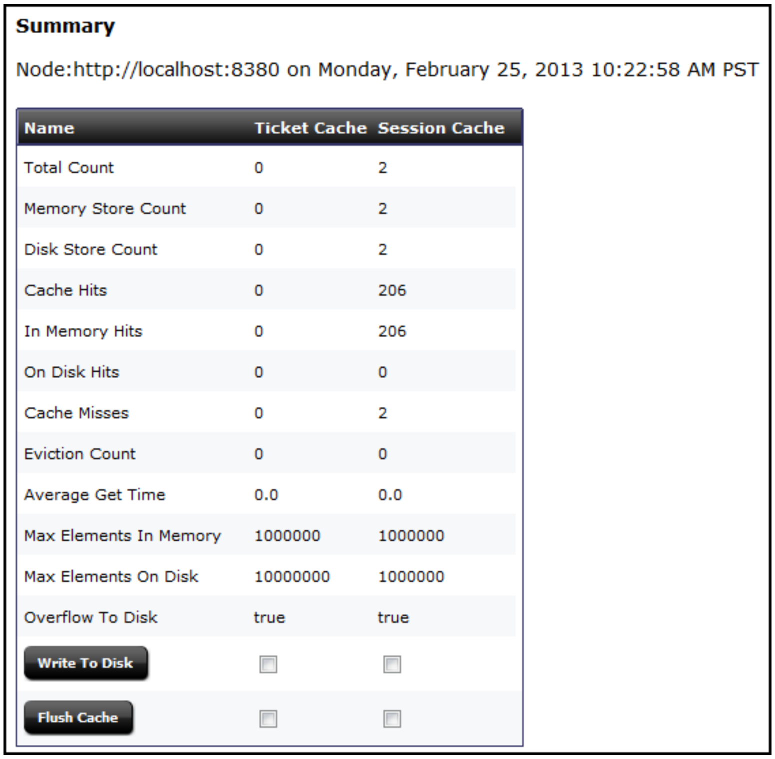
> **Note:**
>
> If legacy page caching is enabled, cached page data is stored in the `SystemPageCache` table and information about the stored data is not displayed in the Summary window.

To view and work with summary cache information:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **CAS Cache** node, and then double-click **Summary**.

   The Summary page is shown in the following figure.

Figure 29-6   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	☐	☐
Flush Cache	☐	☐

2. Select the check box 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 check box 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.

Working with Cluster Information

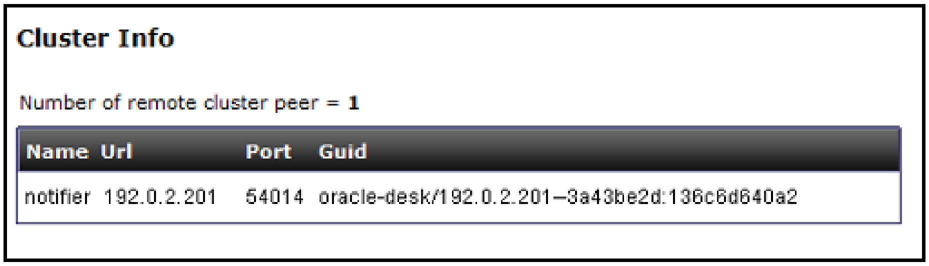
If Sigar is configured, this form lists the name, URL, port, and GUID of each additional WebCenter Sites cluster member.

To view and work with cluster information:

1. In the **General Admin** tree, expand the **Admin**, and then expand the **System Tools** node.
2. Under the **System Tools** node, expand the **Cache Management** node, expand the **CAS Cache** node, and then double-click **Cluster Info**.

The Cluster Info page is shown in the following figure.

Figure 29-7 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

Working with 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, WebCenter Sites stores resultsets within the inCache framework, instead of hash tables. Both caching frameworks use Java memory. You can switch between frameworks by using the Property Management Tool in the Admin interface to set the `rsCacheOverInCache` property (in `wcs_properties.json`) to either `true` or `false`.

When resultset caching over inCache is enabled, the **System Tools** node shows 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 Working with Resultset Caching and Queries in the *Developing with Oracle WebCenter Sites*.

This section covers the following topics:

- Enabling and Configuring Resultset Caching over inCache
- Summary View
- Viewing Cluster Information

Enabling and Configuring Resultset Caching over inCache

If resultset caching over inCache is not enabled, 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 Admin interface, access the Property Management Tool, find the `rsCacheOverInCache` property, and set its value to `true`. (For instructions about using the Property Management Tool to modify properties, see Properties in the Cache Category in *Property Files Reference for Oracle WebCenter Sites*.
2. Configure resultset caching:

In the Property Management Tool, 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 is calculated for cached resultsets.

Summary View

The summary view of resultset cache shows 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. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Resultset Cache** node, and then double-click **Summary**.

   The Resultset Cache Over InCache form is shown.

2. Click a buttons, 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 is read from the database and reloaded into cache.

   The following figure shows how each column value is calculated.

Table 29-1 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: `(Count / MaxCount) x 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.

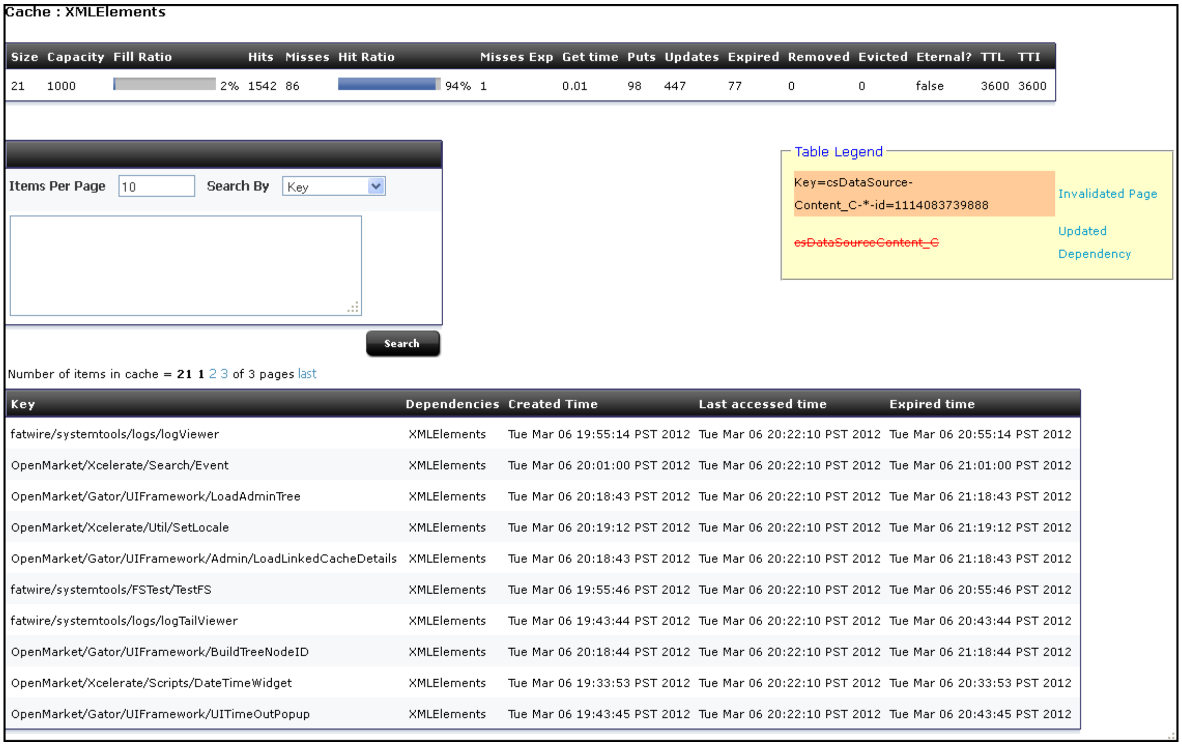
Table 29-1 (Cont.) Descriptions of Columns in the Summary Table

Column Name	Description
Hit ratio	Percentage of hits: `[Hits / (Hits + Misses)] x 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.

3. 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 (in the following figure.).

Figure 29-8 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: 10

Search By: Key

Table Legend

Key=csDataSource-Content_C-*-id=1114083739888	Invalidated Page
~~csDataSourceContent_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

To refresh your form, click **Search**.

The following table describes the columns in the Cache Detail.

Table 29-2 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.

4. To view the resultset element for a key, click the key (query).

   A ResultSet form is shown.

   Columns in the table that opens are dynamic and depend on your site's data model and how the query is set up.

Viewing Cluster Information

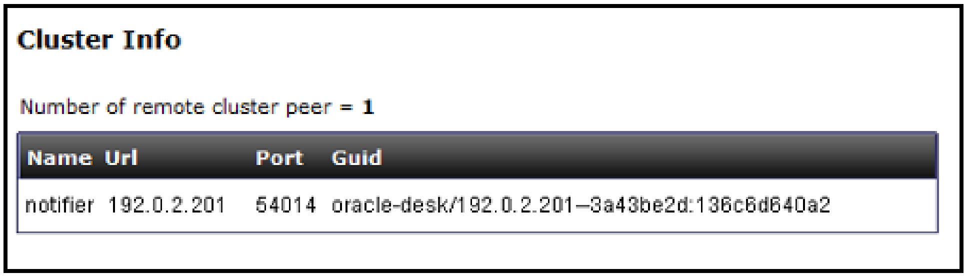
If Sigar is configured, the **Cluster Info** form lists the name, URL, port, and GUID of each additional cluster member.

To view and work with cluster information:

- In the General Admin tree, expand the **Admin** node, expand **System Tools**, select **Cache Management**, expand **Resultset Cache**, and then double-click **Cluster Info**.

  The Cluster Info form is shown (in the following figure.).

Figure 29-9   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

Working with Cache Management - Sites Cache

The following topics provide information about cache management for Sites Cache:

- Overview of the Sites Cache Summary Form
- Working with Page Cache
- Working with the Dependency Cache
- Working with the Asset Cache
- Working with the URL Cache
- Viewing Cluster Information

Overview of the Sites Cache Summary Form

The Sites Cache Summary 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

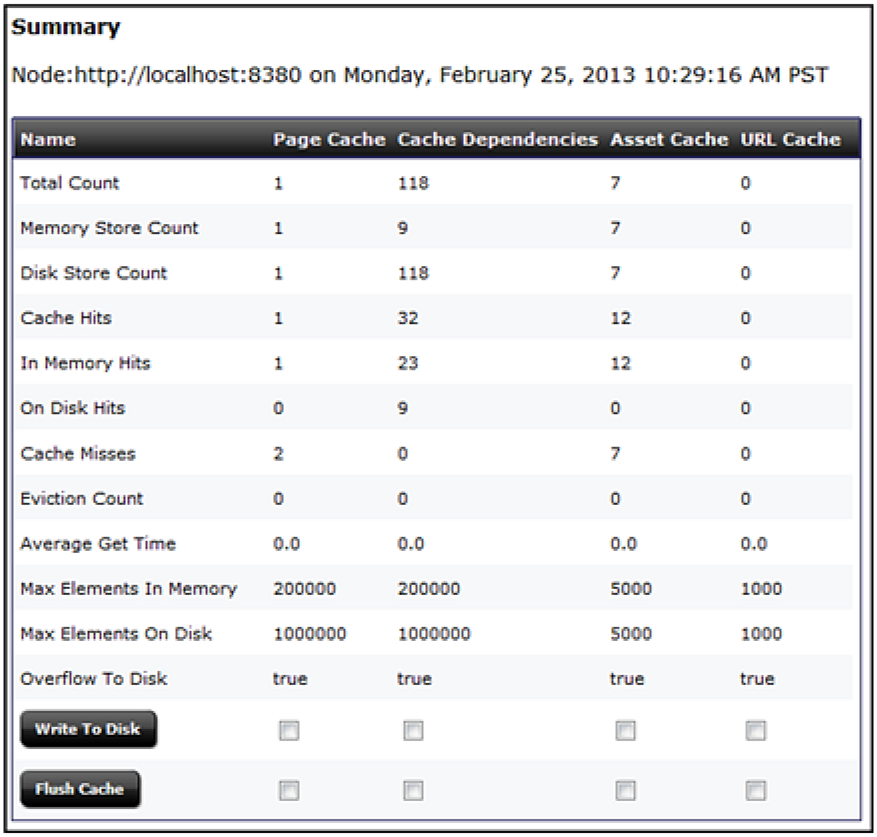
> **Note:**
>
> If legacy page caching is enabled, cached page data is stored in the SystemPageCache table and information about the stored data is not displayed in the Summary window. This window also shows the **Clear Page Cache** button, which allows you to flush the stored data in System Page Cache 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. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **Summary**.

   The Summary form is shown (in the following figure).

Figure 29-10 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	□	□	□	□
Flush Cache	□	□	□	□

2. Select the check box 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 check box 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.

Working with Page Cache

This form lists statistics for each cached page, such as visit 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:

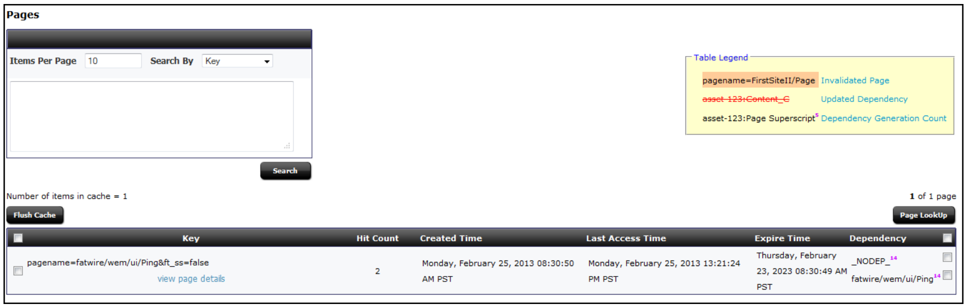
- Searching the cache for specific pages – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing pages from the cache.
- Viewing html information for pages.
- Looking up page dependencies.

The view and work with items in the Page Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **Page Cache**.

   The Pages form is shown (in the following figure).

Figure 29-11 Sites Pages Cache



Pages

Items Per Page 10 Search By Key

Table Legend

pagename=FirstSiteII/Page	Invalidated Page
~~asset-123:Content_C~~	Updated Dependency
asset-123:Page Superscript^{5}	Dependency Generation Count

Number of items in cache = 1

1 of 1 page

□	Key	Hit Count	Created Time	Last Access Time	Expire Time	Dependency	□
□	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 opens, making it easier to find items in the cache.
4. Underneath each item in the **Key** column, a **view page details** link opens. Click **view page details** to open a new window that shows the page's HTML information.
5. To flush selected assets from the cache, select the check box on the left side of each asset. Select the check box 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, select the check box on the right side of each item. Select the check box 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 check boxes 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.

Working with the Dependency Cache

This form lists the contents of the dependency cache and the generation count for each dependency. In addition, it provides the following options:

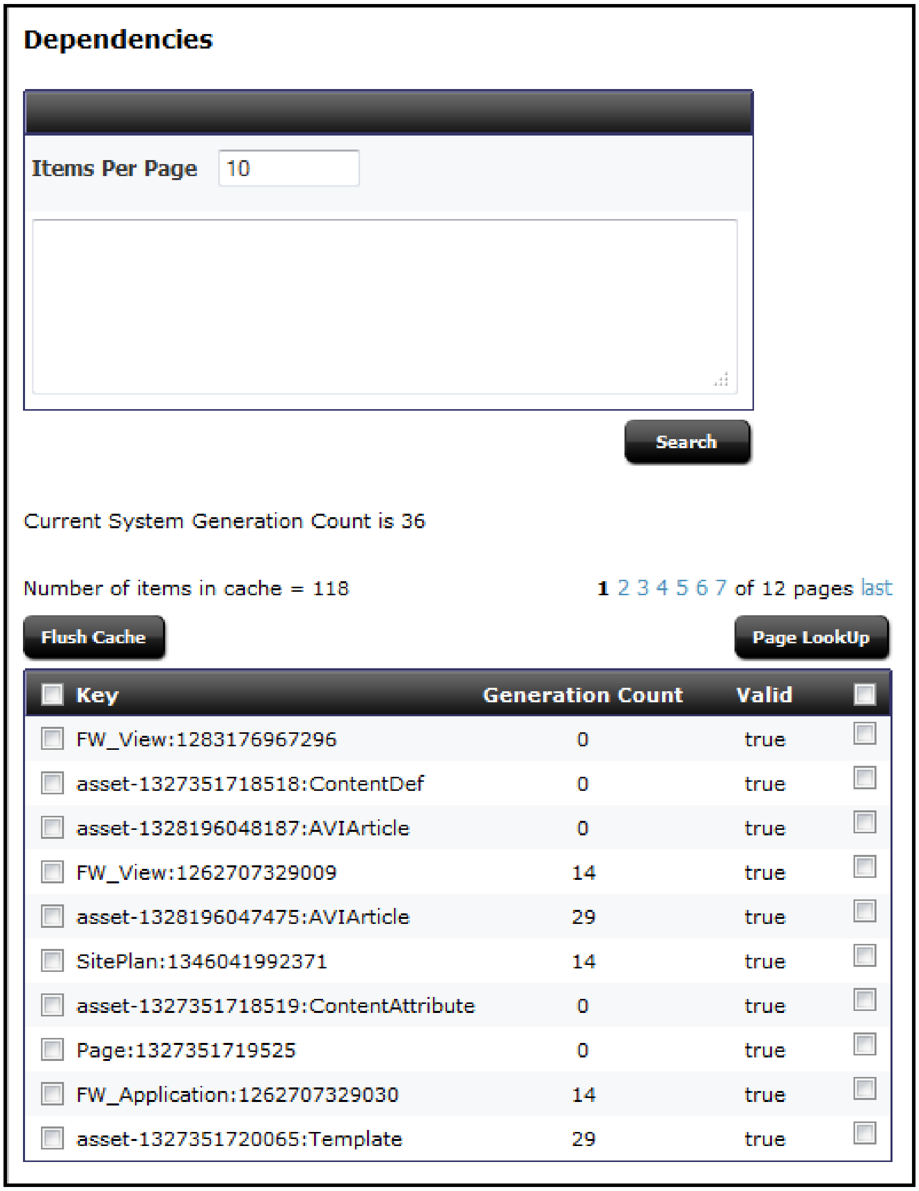
- Searching the cache for specific dependencies – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing dependencies from the cache
- Looking up page dependencies

The view and work with items in the Dependency Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and double-click **Dependency**.

   The Dependencies form is shown (Figure 29-12).

Figure 29-12 Sites Cache Dependencies



Dependencies

Items Per Page: 10

Current System Generation Count is 36

Number of items in cache = 118

1 2 3 4 5 6 7 of 12 pages last

☐ Key	Generation Count	Valid	☐
☐ FW_View:1283176967296	0	true	☐
☐ asset-1327351718518:ContentDef	0	true	☐
☐ asset-1328196048187:AVIArticle	0	true	☐
☐ FW_View:1262707329009	14	true	☐
☐ asset-1328196047475:AVIArticle	29	true	☐
☐ SitePlan:1346041992371	14	true	☐
☐ asset-1327351718519:ContentAttribute	0	true	☐
☐ Page:1327351719525	0	true	☐
☐ FW_Application:1262707329030	14	true	☐
☐ asset-1327351720065:Template	29	true	☐

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 opens, making it easier to find items in the cache.

3. To flush selected assets from the cache, select the check box on the left side of each asset. Select the check box 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, select the check box on the right side of each item. Select the check box in the header to mark all displayed dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

> **Note:**
>
> After 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.

Working with the Asset Cache

This form lists statistics for each cached asset, such as visit 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:

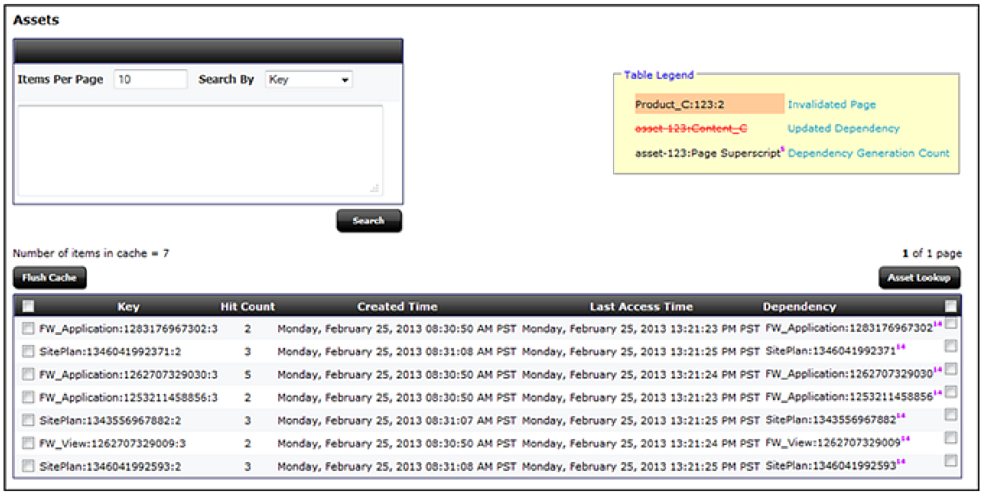
- Searching the cache for specific assets. Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing assets from the cache
- Looking up asset dependencies

The view and work with items in the Asset Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **Asset Cache**.

   The Assets form is shown (in the following figure).

Figure 29-13 Asset Cache



Assets

Items Per Page: 10
Search By: Key

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

□	Key	Hit Count	Created Time	Last Access Time	Dependency	□
□	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}	□
□	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}	□
□	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}	□
□	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}	□
□	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}	□
□	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}	□
□	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 opens, making it easier to find items in the cache.
4. To flush selected assets from the cache, select the check box on the left side of each asset. Select the check box 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, select the dependency check box on the right side of each asset. Select the dependency check box in the header to mark all the 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.

Working with the URL Cache

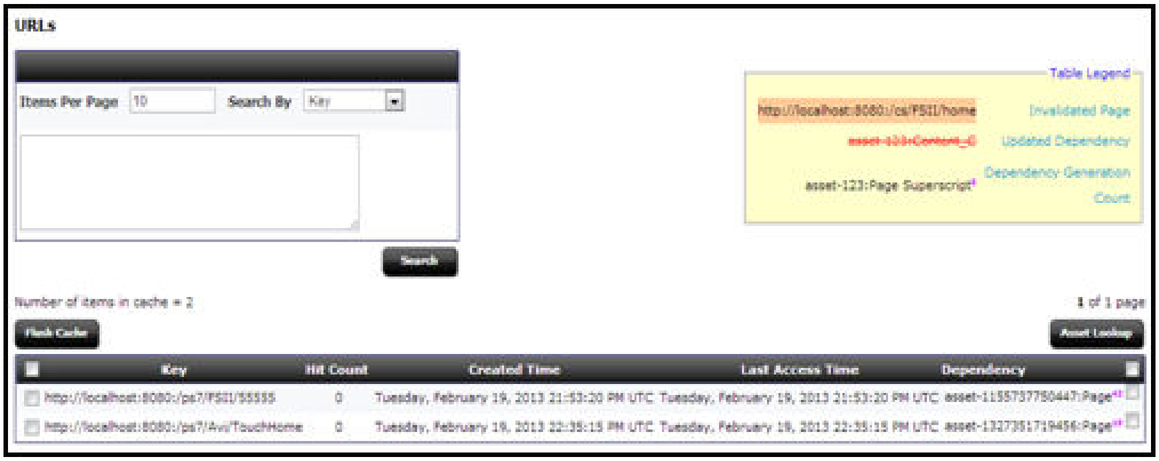
All the Vanity URLs are cached after the resolution is done in a web page in inCache. The cache can be looked up and managed in the **Cache Management** node under **System Tools**.

The view and work with items in the URL Cache:

1. In the General Admin tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **URL Cache**.

The URLs form is shown (in the following figure).

Figure 29-14   URLs form, URL Cache



URLs

Items Per Page 10   Search By Key

Table Legend

http://localhost:8080:/cs/FSII/home	Invalidated Page
~~asset-123:Content_C~~	Updated Dependency
asset-123:Page Superscript$^{#}$	Dependency Generation Count

Number of items in cache = 2

1 of 1 page

□	Key	Hit Count	Created Time	Last Access Time	Dependency	□
□	http://localhost:8080:/ps7/FSII/55555	0	Tuesday, February 19, 2013 21:53:20 PM UTC	Tuesday, February 19, 2013 21:53:20 PM UTC	asset-1155737750447:Page^{43}	□
□	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^{43}	□

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 opens, making it easier to find items in the cache.
4. To flush selected URLs from the cache, select the check box on the left side of each URL. Select the check box in the header to mark all displayed URLs of the assets. Click **Flush Cache** to flush the memory and stores of the URLs.
5. To view the URL for a specific dependency within the cache, select the dependency check box on the right side of each URL. Select the check box in the header to mark all displayed URLs of the assets. Click **Asset Lookup** to show the dependencies.

Viewing Cluster Information

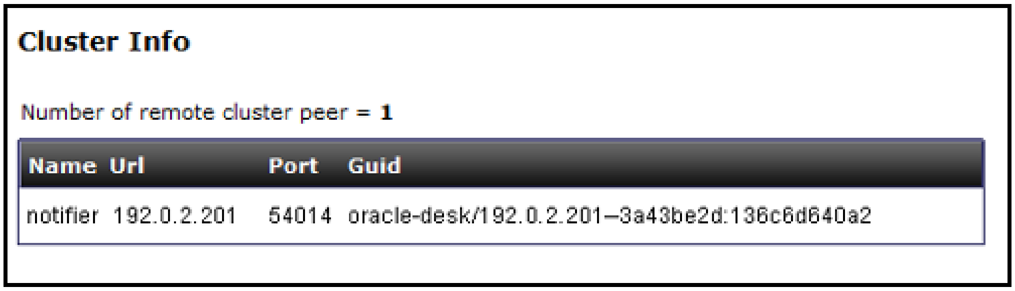
If Sigar is configured, The **Cluster Info** form lists the name, URL, port, and GUID of each additional cluster member.

To view and work with cluster information:

- In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Sites Cache** node, and then double-click **Cluster Info**.

  The Cluster Info form is shown (in the following figure).

Figure 29-15   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

Working with Cache Management - Co-Res Cache

The following topics provide information about cache management for the Co-Res Cache:

- Overview of the Co-Res SS Cache Summary Form
- Working with the Page Cache
- Working with the Dependency Cache
- Working with the URL Cache
- Viewing Cluster Information

Overview of the Co-Res SS Cache Summary Form

The cache summary form for the Co-Res Satellite Server 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.

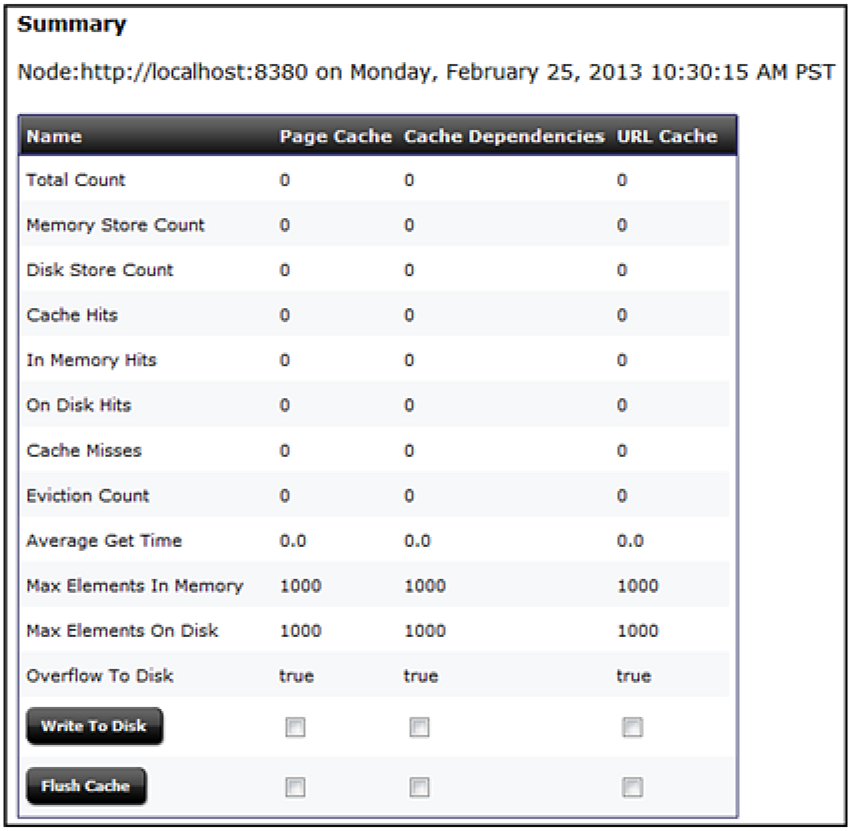
> **Note:**
>
> If legacy page caching is enabled, cached page data is stored in the SystemPageCache table and information about the stored data is not displayed in the Summary window.

To view and work with summary cache information:

1. In the General Admin tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Co-Res SS Cache** node, and then double-click **Summary**.

   The Summary page is shown (in the following figure).

Figure 29-16 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	□	□	□
Flush Cache	□	□	□

2. Select the check box 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 check box for the **PageCache**, **Cache Dependencies**, or **URL Cache** and click **Flush Cache** to flush the memory and disk stores for the selected caches.

Working with the Page Cache

This form lists statistics for each cached page, such as visit 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:

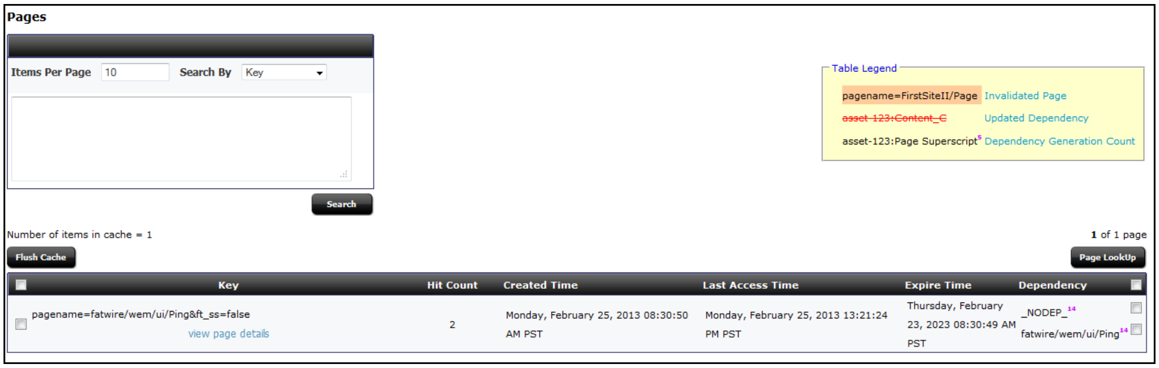
- Searching the cache for specific pages – Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing pages from the cache.
- Viewing html information for pages.
- Looking up page dependencies.

The view and work with items in the Page Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Co-Res SS Cache** node, and then double-click **Page Cache**.

   The Pages form is shown (in the following figure).

Figure 29-17   Co-Res SS Pages Cache



Pages

Items Per Page 10   Search By Key

Table Legend

pagename=FirstSiteII/Page	Invalidated Page
~~asset-123:Content_C~~	Updated Dependency
asset-123:Page Superscript[5]	Dependency Generation Count

Number of items in cache = 1

1 of 1 page

□	Key	Hit Count	Created Time	Last Access Time	Expire Time	Dependency	□
□	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 opens, making it easier to find items in the cache.
4. Underneath each item in the **Key** column, a **view page details** link opens. Click **view page details** to open a new window that shows the page's HTML information.
5. To flush selected assets from the cache, select the check box on the left side of each asset. Select the check box 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, select the check box on the right side of each item. Select the check box in the header to mark all. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note that each item in cache can have multiple check boxes on the right side, reflecting multiple dependencies.

> **Note:**
>
> After 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.

Working with the Dependency Cache

This form lists the contents of the dependency cache and the generation count for each dependency. In addition, it provides the following options:

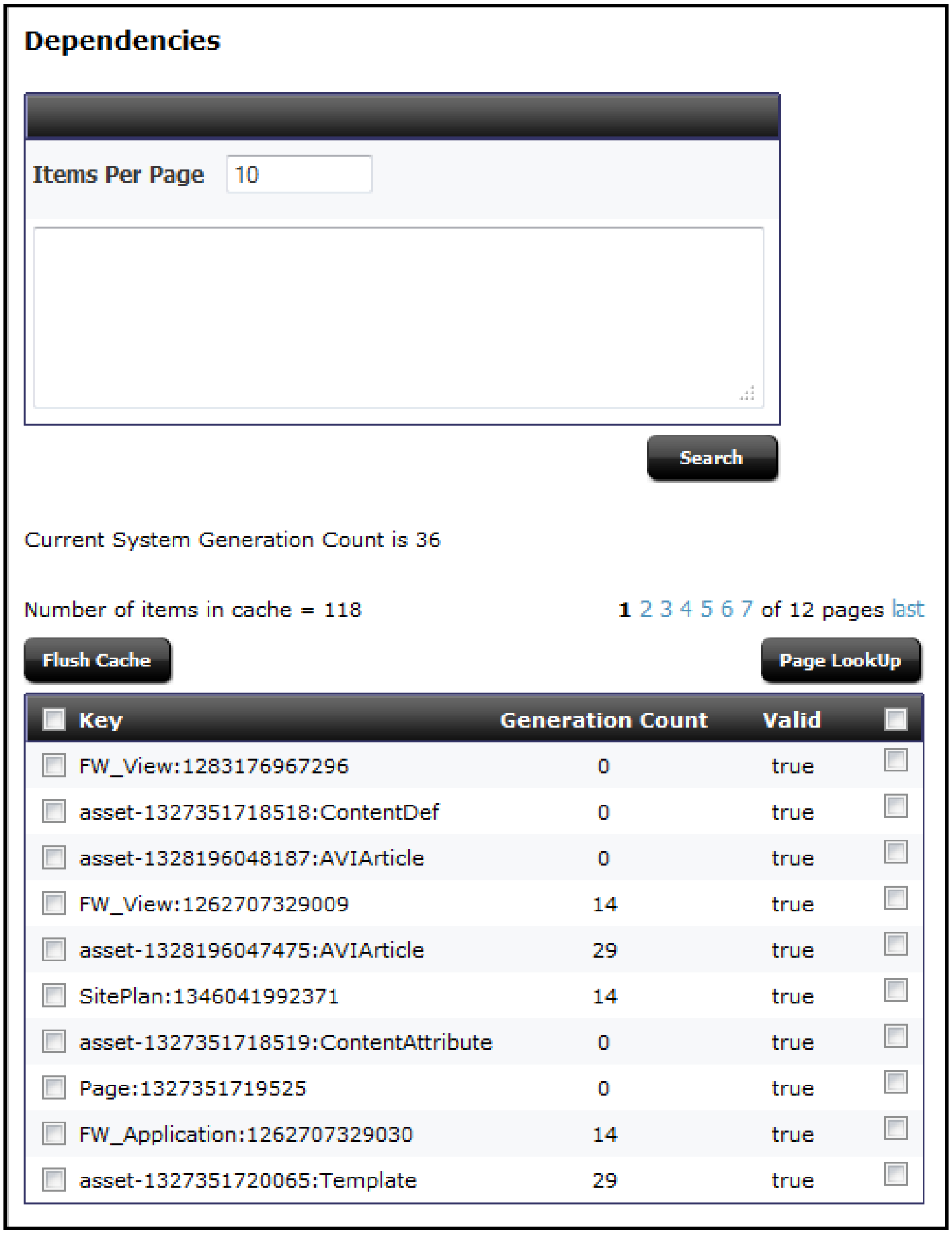
- Searching the cache for specific dependencies. Searches are CPU intensive and should be used with caution on a delivery system.
- Flushing dependencies from the cache.
- Looking up page dependencies.

The view and work with items in the Dependency Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Co-Res SS Cache** node, and then double-click **Dependency**.

   The Dependencies form is shown (in the following figure).

Figure 29-18 Co-Res SS Cache Dependencies



Dependencies

Items Per Page 10

Current System Generation Count is 36

Number of items in cache = 118

□	Key	Generation Count	Valid	□
□	FW_View:1283176967296	0	true	□
□	asset-1327351718518:ContentDef	0	true	□
□	asset-1328196048187:AVIArticle	0	true	□
□	FW_View:1262707329009	14	true	□
□	asset-1328196047475:AVIArticle	29	true	□
□	SitePlan:1346041992371	14	true	□
□	asset-1327351718519:ContentAttribute	0	true	□
□	Page:1327351719525	0	true	□
□	FW_Application:1262707329030	14	true	□
□	asset-1327351720065:Template	29	true	□

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 opens, making it easier to find items in the cache.

3. To flush selected assets from the cache, select the check box on the left side of each asset. Select the check box 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, select the check box on the right side of each item. Select the check box in the header to mark all displayed dependencies. Click **Page Lookup** to show the pages for the dependency or dependencies.

Note:

After 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.

Working with the URL Cache

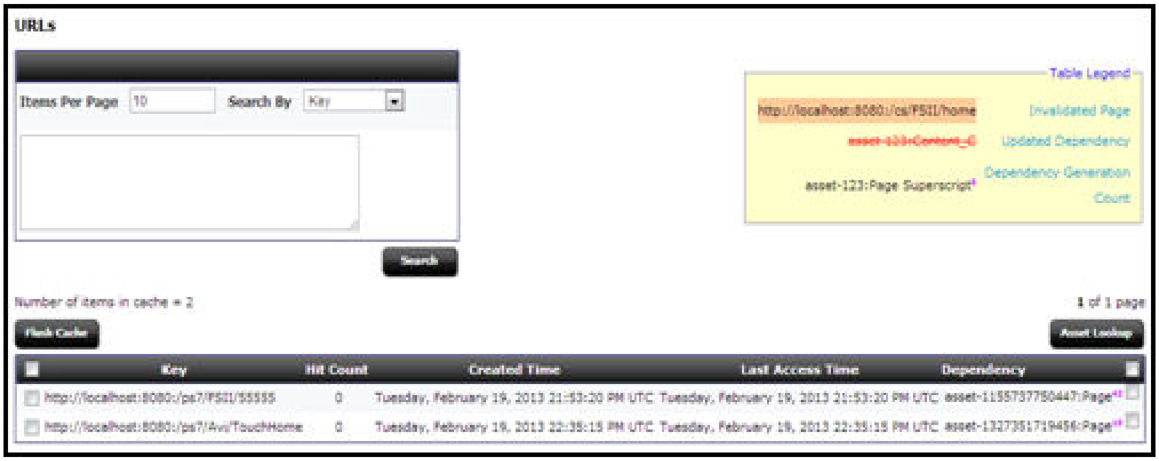
All the Vanity URLs are cached after the resolution is done in a web page in inCache. The cache can be looked up and managed in the **Cache Management** node under **System Tools**.

The view and work with items in the URL Cache:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Co-Res SS Cache** node, and then double-click **URL Cache**.

   The URLs form is shown (in the following figure).

Figure 29-19 URLs form, URL Cache



URLs

Items Per Page: 10 Search By: Key

Number of items in cache = 2

1 of 1 page

□	Key	Hit Count	Created Time	Last Access Time	Dependency	□
□	http://localhost:8080:/ps7/FSII/55555	0	Tuesday, February 19, 2013 21:53:20 PM UTC	Tuesday, February 19, 2013 21:53:20 PM UTC	asset-1155737750447:Page	□
□	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	□

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, 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 opens, making it easier to find items in the cache.
4. To flush selected URLs from the cache, select the check box on the left side of each URL. Select the check box in the header to mark all displayed URLs of the assets. Click **Flush Cache** to flush the memory and stores of the URLs.
5. To view the URLs for a specific dependency within the cache, select the dependency check box on the right side of each URL. Select the check box in the header to mark all displayed URLs of the assets. Click **Asset Lookup** to show the dependencies.

Viewing Cluster Information

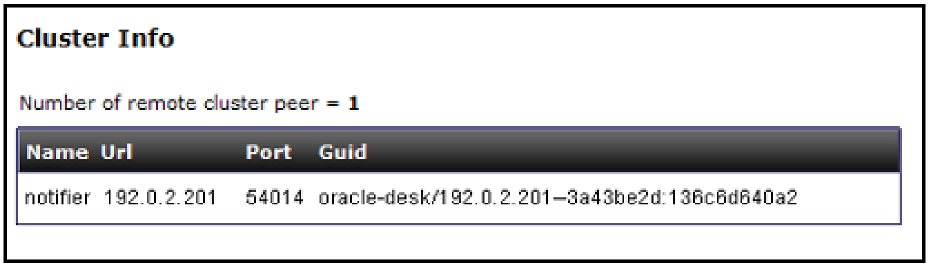
If Sigar is configured, the **Cluster Info** form lists the name, URL, port, and GUID of each additional cluster member.

To view and work with cluster information:

- In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Cache Management** node, expand the **Co-Res SS Cache** node, and then double-click **Cluster Info**.

  The Cluster Info form is shown (in the following figure).

Figure 29-20 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

Using the Log Viewer

The Log Viewer tool enables you to view, tail, download, and search the contents of the WebCenter Sites log specified in the `logging-config.xml` file directly from the WebCenter Sites Admin 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.

The following topics provide information about using the Log Viewer tool:

- Viewing the Log
- Tailing the Log
- Downloading the Log
- Searching the Log

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. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Log Viewer**.

   The Log Viewer form shows the most recent 100 lines in the log.

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 operates only on log statements that were written with the timestamping pattern specified in the `logging-config.xml` file.

   b. Click **View Logs**.

      The Log Viewer form is shown.

      The form shows the statements written at and after the specified date and time.

Tailing the Log

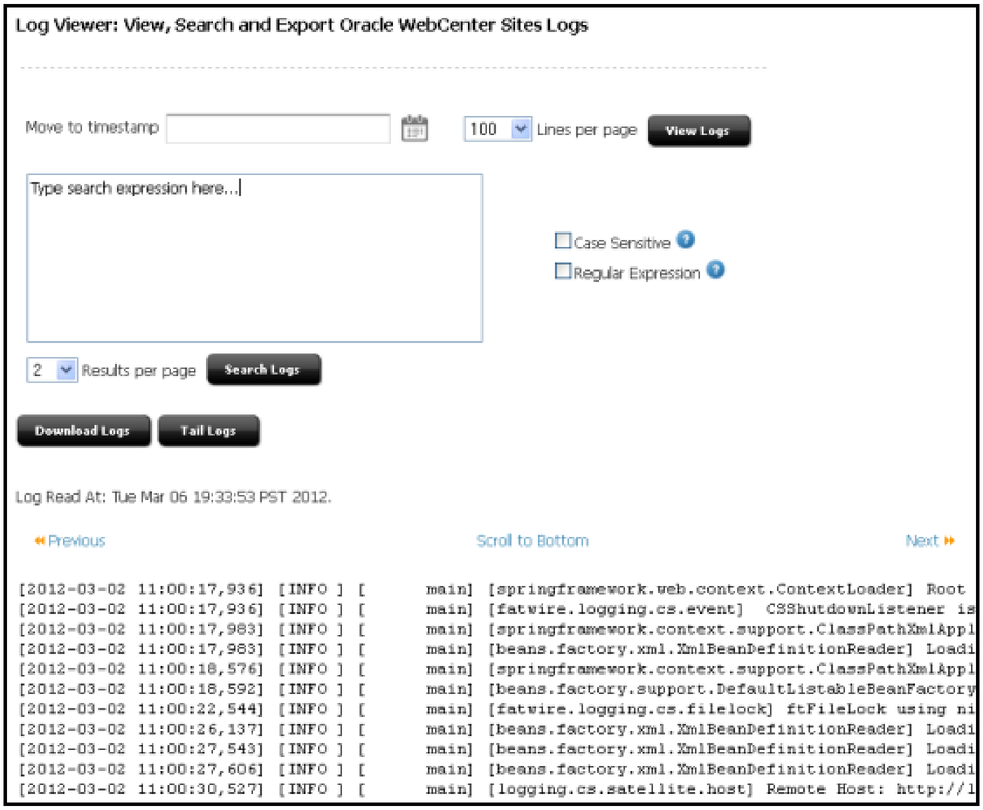
You can tail the WebCenter Sites log in a separate window. This option is particularly useful when you want to keep track of the statements written to the log while executing an action that causes an exception.

To tail the contents of the log:

1. In the General Admin tree, expand the **Admin** node, expand **System Tools**, and then double-click **Log Viewer**.

   The Log Viewer form is shown (in the following figure).

Figure 29-21   Log Viewer Form



Log Viewer: View, Search and Export Oracle WebCenter Sites Logs

100 Lines per page

2 Results per page

Log Read At: Tue Mar 06 19:33:53 PST 2012.

```
[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://l
```

2. Click **Tail Logs**.

   The most recent 100 lines in the log are shown in a new window and are refreshed every five seconds.

Downloading the Log

You can download a compressed file containing the entire contents of the log.

To download the contents of the log:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Log Viewer**.

   The Log Viewer form is shown (Figure 29-21).

2. Click **Download Logs**.
3. Open or save the compressed file containing the log. If the log is spread over several files, each file is included.

Searching the Log

You can search the contents of the WebCenter Sites log for specific information.

To search the contents of the log:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **Log Viewer**.

The Log Viewer form is shown (Figure 29-21).

2. 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 opens when 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. When 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 to find in the log.
5. In the **Results per page** field, select the number of results to list on each page.
6. Click **Search Logs**.

   The Log Viewer form shows the lines in the log that match your 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.

Performance Test

You can determine performance of File System and Lucene Indexing by tools provided in WebCenter Sites under System Tools in Admin tab.

The performance test tools in WebCenter Sites include:

- File System Test
- Lucene Indexing Test

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`) to compare their performance with provided benchmarks.

Table 29-3 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`
sync	Shared synchronization directory for all WebCenter Sites cluster members. Sample path: `\cs76\Shared\clustersync`

A test suite is made of several tests 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 that runs a total of 16 tests (4 tests are performed on each of the directories listed in the table above).
- Medium Test that runs a total of 64 tests (16 tests are performed on each of the directories listed in the table above).
- Extensive Test – runs a total of 384 tests (96 tests are performed on each of the directories listed in the table above).

To test the file system:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Performance Test** node, and then double-click **File System Test**.

   The File System Test form is shown.

2. Select the type of test you want to perform and click **Start Test**.

   > **Note:**
   >
   > - If the test takes too long or affects performance too much, 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. 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 through 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 the test was performed on.
- **Path** specifies the path to the directory a file system test was performed on.
- **Create** specifies the average number of microseconds a create operation was performed.
- **Write** specifies the average number of microseconds a write operation was performed.
- **Read** specifies the average number of microseconds a read operation was performed.
- **Delete** specifies the average number of microseconds 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).

5. To view the results of a single test, either hover over a point on the graph or navigate to the row in the test results table.

   The following information is shown in the table for each test:

   - **Test** lists a serial number for each test.
   - **Threads** lists the number of threads created during the test.
   - **Files** lists the number of files created per thread.
   - **Size** lists the number of bytes per file.
   - **Reads** lists the number of times each file was read by a test.
   - **Lock** lists if each file was locked (random access lock operation).
   - **Attributes** lists whether each file's attributes were read during the test.
   - **Mode** lists the Java I/O random access mode for accessing the files in the thread.
   - **FileSystem** lists the directory on which the test was performed.
   - **Create** lists the minimum, maximum, and average number of microseconds in which the test performed a create operation.
   - **Write** lists the minimum, maximum, and average number of microseconds in which the test performed a write operation.
   - **Read** lists the minimum, maximum, and average number of microseconds in which the test performed a read operation.
   - **Delete** lists the minimum, maximum, and average number of microseconds in which the test performed a delete operation.
   - **Total Test Time** lists the total number of microseconds in which a test was completed.

   In the test results table, expand a row to view detailed information about each thread that was created during a specific test. The following information is shown in the thread detail table:

- **Thread** lists a serial number for each thread.
- **Create** lists the total number of microseconds the thread created an empty file.
- **Write** lists the total number of microseconds the thread finished writing to the file.
- **Read** lists the total number of microseconds the thread read the file it created.
- **Delete** lists the total number of microseconds the thread deleted the file.

> **Note:**
>
> - Each point on a graph and row in a table shows information for a single test.
> - To view the details of a single test, when you hover over a point in the graph, use a browser other than Internet Explorer.

Lucene Indexing Test

You can use the Lucene Indexing Test tool to test the performance of Lucene indexing, either on shared file system, that is (`<cs_install_dir>/Shared/lucene`) or on NIO (Non-blocking I/O), that is `sharefs://Shared/lucene`. The results of these tests are compared against benchmark results.

A test suite that is composed of a number of tests is performed on the Lucene index files on shared file systems or NIO. Each of these tests creates indexing for 20 assets, re-indexing for 20 assets, and deletes indexing for 20 assets. The following test suites are available:

- Short test suite that runs a total of 16 tests.
- Medium test suite that runs a total of 64 tests.
- Extensive test suite that runs a total of 384 tests.

To test Lucene indexing:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, expand the **Performance Test** node, and then double-click **Lucene Indexing Test**.
2. In the Lucene Indexing Test form, select the type of test you want to perform and click **Start Test**.

> **Note:**
>
> - If the test takes too long or affects performance, you can click **Stop Test**.
> - The results of the Lucene Indexing Test can be compared with the provided benchmark only when the test is completed. These benchmarks are provided by running standard tests on an Oracle Linux 7u1 (Intel 64-bit, 16GB RAM) and 12.1.0.2 Oracle Database with no additional traffic during the test.

3. Once the test is complete, review the overall results in the Performance Summary table.

   The following information is shown in the Performance Summary:

   - Numbers displayed in bold specify benchmarks. The benchmarks were established by internal testing.
   - **FileSystem** specifies the directory on which the test was performed.
   - **Path** specifies the path to the directory on which a Lucene Indexing Test was performed.
   - **Index** specifies Min/Max/Average of Index operation for 20 asserts in microseconds for all the tests performed.
   - **Reindex** specifies Min/Max/Average of Re-index operation for 20 asserts in microseconds for all the tests performed.
   - **Delete** specifies Min/Max/Average of delete operation for 20 asserts in microseconds for all the tests 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).

5. 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.

   The following information is shown in the graph

   - **Thread** specifies the number of threads created. In Lucene Indexing Test is Single threaded.
   - **Index** specifies the time taken in 1000 μS to create indexing for 20 assets.
   - **Reindex** specifies the time taken in 1000 μS to re-index 20 assets.
   - **Delete** specifies the time taken in 1000 μS to delete indexing for 20 assets.

   The following information is shown in the table for each test:

   - **Test** specifies a serial number for each test.
   - **Thread** specifies the number of threads created. In Lucene Indexing Test, a test is single threaded.
   - **Index** specifies the time taken in μS to create indexing for 20 assets.
   - **Reindex** specifies the time taken in μS to re-index 20 assets.
   - **Delete** specifies the time taken in μS to delete indexing for 20 assets.

- **Total Test Time** specifies the total number of microseconds in which a test was completed.

> **Note:**
>
> - Each point on the graph and row in the table shows information about a single test.
> - In the test results table, the values in each column index, reindex, and delete represents time taken for indexing, re-indexing, and deleting indexes for 20 assets in micro seconds.

Working with Slots Assets

Slots assets are used to store presentation changes made when editing a website in context. Slots assets are managed transparently, they are not available in the user interfaces. The **Slots** node in System Tools where you 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. However, it is not removed from the database so it will also not show up in any searches and similar inquiries. Voided assets can be physically removed in a purge. See Working with the Purge Voided Assets Node.

To void Assets or delete Assets:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click the **Slots** node.

   The Slot Assets Utility form opens.
2. The **Slot Assets Utility** form shows the slot assets. Click a table heading 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.
3. Select one or more assets by selecting the check boxes.
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 opens. The list of remaining assets redisplay.

When 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 about how to purge voided assets, see Working with the Purge Voided Assets Node.

Working with the URL Node

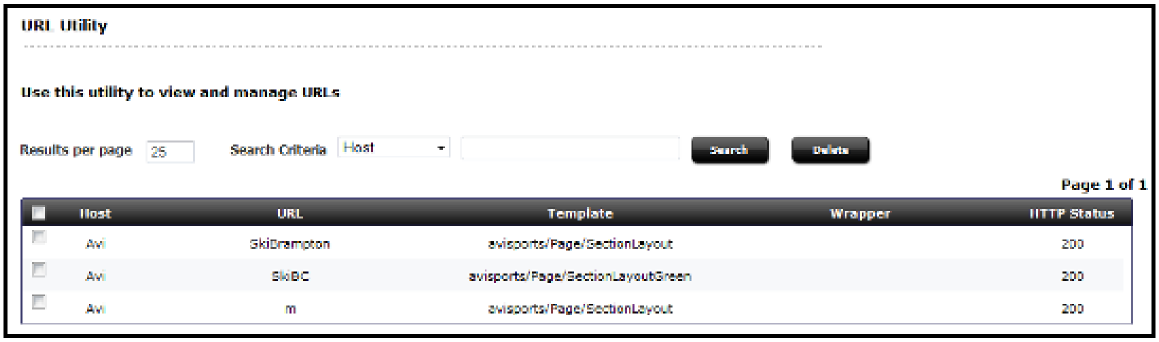
The URL node allows you to search, view, and delete custom URLs, also known as vanity URLs. These URLs are domains used to redirect to a certain section of your website, which may or may not be accessible to the rest of the hosted website. Often these vanity URLs are used to prevent a long URL that has coded strings within it, making it easier for the web user to view your page.

To view and manage vanity URLs:

1. In the **General Admin** tree, expand the **Admin** node, expand the **System Tools** node, and then double-click **URL**.

   The URL Utility form is shown (in the following figure).

Figure 29-22   URL Utility Form



URL Utility

Use this utility to view and manage URLs

Results per page 25   Search Criteria Host ▾ [ ]

Page 1 of 1

□	Host	URL	Template	Wrapper	HTTP Status
□	Avi	SkiDrampton	avisports/Page/SectionLayout		200
□	Avi	SkiBC	avisports/Page/SectionLayoutGreen		200
□	Avi	m	avisports/Page/SectionLayout		200

   All vanity URLs, including auto-generated URLs, are shown.

2. You can limit the list by entering the search criteria for **Host**, **URL**, **Template**, **Wrapper**, or **HTTP Status**, then click **Search**.
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, select the check box and click **Delete**.

Working with the Purge Voided Assets Node

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**.
2. Select the asset type, or all asset types, to purge the 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.

30 Replicating Content Management Sites

In WebCenter Sites, you can replicate sites with Site Launcher. You can replicate a Content Management (CM) site and share or copy its components to the new site. You can also modify and deploy the new site.

Topics:

- Site Launcher Overview
- Preparing for Replication
- Enabling Site Replication
- Completing Post-Replication Tasks

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 requirements of 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 requires.

The rest of this chapter provides the steps that you must follow to successfully replicate sites for use in any content management model—1:1, 1:n, or x:n, described in 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.

Preparing for Replication

Before attempting to replicate a site, you must 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.

- Replication Requirements for the Source Site
- Planning the New Site

Replication Requirements for the Source Site

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 must 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, ensure the source site resides on the management system. (If you must first create a replicable source site, follow the steps in Replicating Content Management Sites.)

Planning the New Site

The following topics outline your site replication options and the system requirements.

- Copying vs. Sharing
- Prefixes for Asset Names
- Planning Users
- Replication Schedule

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 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 if its components are shared by default. Other components are shared or copied at your discretion. Users are neither copied nor shared.

Table 30-1 Components and Replication

Source Site Component	Replication method
asset subtypes	Shared
asset types	Shared
assets	Either copied or shared (admin decides)
associations	Shared
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 are 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:callelement` tags) or copied templates (constructed with a template name using `Variables.site` prepended to the name in the `render:satellitepage` and `callelement` 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 must designate the source site accordingly.

  If you decide to copy template assets, ensure 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 exception is Template assets. A Template asset maintains a SiteCatalog entry for each site to which it is shared. When a template is shared, the template's list of 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. 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 could 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 shown 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 is set in the replicate sites. When users in the new site call 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 and 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 on the management system. For information about site modeling, see Content Management Models.

Prefixes for Asset Names

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 is 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.

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 are different from those on the source site, you must create user accounts. In the process, consider the users' functions on the new site, the roles they will require, their access to start menu items, workflows, tree tabs, and so on.

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.

Replication Schedule

Because normal operations affect site replication, it is best to run Site Launcher when the source site is not being actively edited. Make sure all assets in the source site are checked in to the database before replication. Otherwise, if revision tracking is enabled and an asset to be shared is checked out, the sharing of the asset will fail.

Enabling Site Replication

This section shows you how to enable and use Site Launcher. Once you have ensured that source sites and planned sites meet the requirements (see Preparing for Replication), you can configure and replicate the source site.

- Configuring the Source Site for Site Launcher
- Replicating the Source Site

Configuring the Source Site for Site Launcher

To configure Site Launcher:

When you complete the steps in this section, follow up immediately with the steps in Replicating the Source Site. Immediate follow-up ensures that the condition of the source site has not changed.

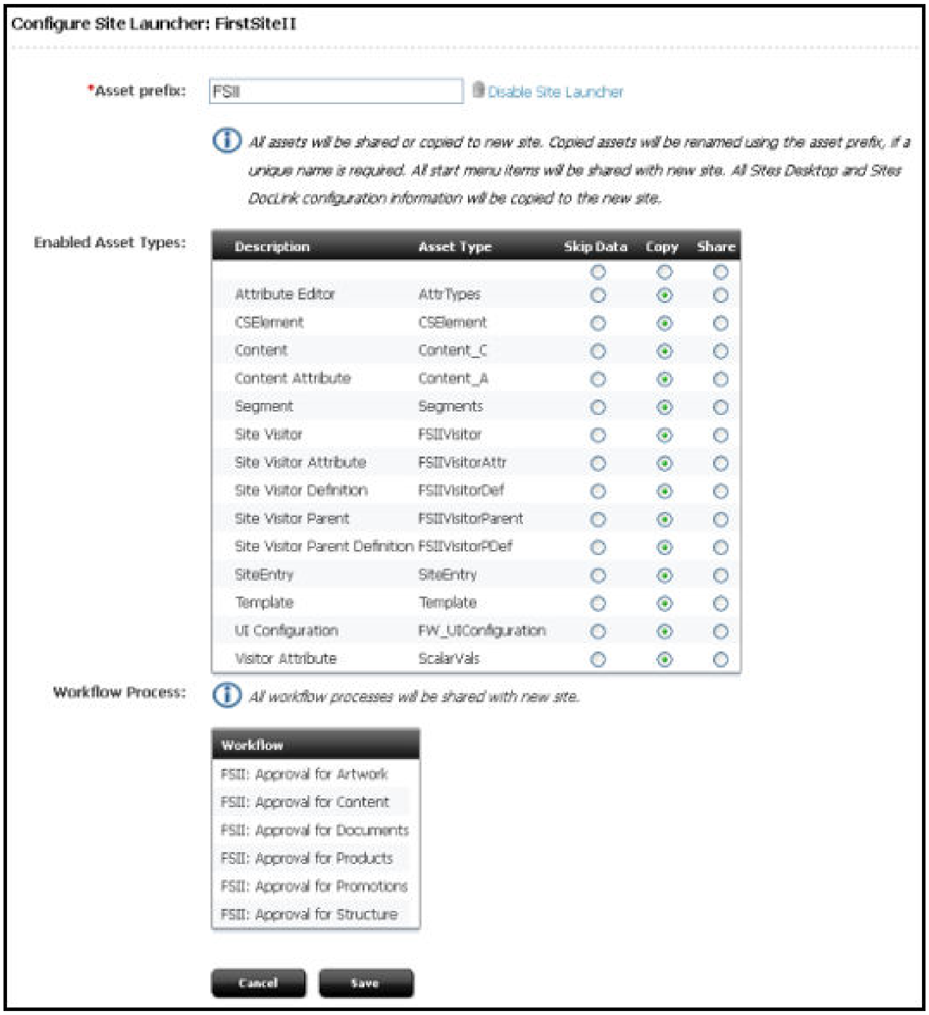
1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click the site you want to replicate. This is your source site.

   The Inspect form opens.

2. Click **Configure Site Launcher for this site**.

   The Configure Site Launcher form opens.

Figure 30-1 Configure Site Launcher Form



Configure Site Launcher: FirstSiteII

*Asset prefix: FSII

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	○	●	○
CSElement	CSElement	○	●	○
Content	Content_C	○	●	○
Content Attribute	Content_A	○	●	○
Segment	Segments	○	●	○
Site Visitor	FSIIVisitor	○	●	○
Site Visitor Attribute	FSIIVisitorAttr	○	●	○
Site Visitor Definition	FSIIVisitorDef	○	●	○
Site Visitor Parent	FSIIVisitorParent	○	●	○
Site Visitor Parent Definition	FSIIVisitorPDef	○	●	○
SiteEntry	SiteEntry	○	●	○
Template	Template	○	●	○
UI Configuration	FW_UIConfiguration	○	●	○
Visitor Attribute	ScalarVals	○	●	○

Workflow Process:

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

3. 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).
4. 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.
5. Click **Save**.

Replicating the Source Site

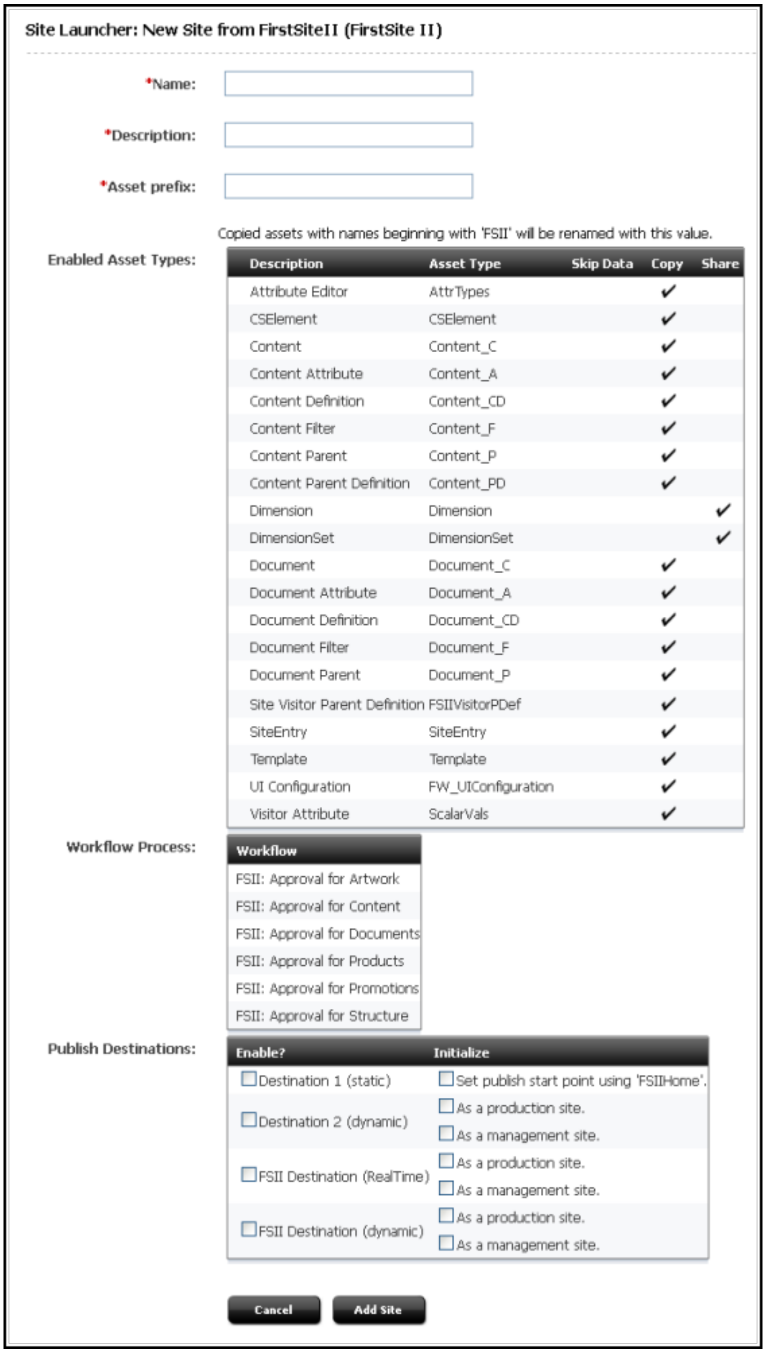
When you have enabled the source site for replication, you can use Site Launcher to create as many copies of the source site as you wish.

To replicate the source site:

1. Ensure you have completed the steps in Configuring the Source Site for Site Launcher.
2. At the bottom of the **Inspect** form, click **Launch Copy**.

   The Site Launcher form opens.

Figure 30-2 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
□Destination 1 (static)	□Set publish start point using 'FSIIHome'.
□Destination 2 (dynamic)	□As a production site. □As a management site.
□FSII Destination (RealTime)	□As a production site. □As a management site.
□FSII Destination (dynamic)	□As a production site. □As a management site.

3. In the **Name** field, enter the name for the new site. Note the following conventions when naming the site:
   - The name must be between 1 and 255 characters.
   - The following characters are not allowed: single-quote ('), double quote ("), semicolon (;), colon (:), question mark (?), ampersand (&), less-than sign (<), and greater-than sign (>). Additionally, the name cannot end with a backslash.
   - The name cannot contain spaces.
4. In the **Description** field, enter the description for the new site.

5. In the Publish Destinations area:

   a. Initialize the destinations by selecting the check boxes, as necessary.

   b. Select the publish destinations you would like to be available for the new site.

6. Click **Add Site** to copy the site.

   After the new site is created, a summary is shown 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.

7. At this point, you must add users and otherwise ensure that the new site is properly configured. For instructions and guidelines, see the following section.

Completing Post-Replication Tasks

When the new site is established, you must 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 Granting Users Access to a Site (Assigning Roles to Users).

   - To establish new users, follow the guidelines in Roadmap for Administering WebCenter Sites.

2. If workflow processes exist, ensure 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.

31

Managing Revision Tracking

You can decide which asset types you would like to track and how many revisions you would like to store. You can track various versions of assets created for a site with the revision tracking functionality of WebCenter Sites. When you make various versions of assets available for content contributors, they can compare them and choose what they need.
Topics:

- Revision Tracking Overview
- Enabling Revision Tracking
- Disabling Revision Tracking
- Unlocking Revisions

Revision Tracking 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 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 tracked, WebCenter Sites provides check in, check out, and other revision tracking features on the New and Edit forms for assets of those types.

This section covers the following topics:

- Tracker Tables and Storage Directories
- RTInfo Table
- Revision Tracking and the Two Asset Models
- Implicit vs. Explicit Checkin and Explicit Checkout
- Revision Tracking and Non-Asset Tables
- How Many Versions?

Tracker Tables and Storage Directories

When you enable revision tracking for an asset type, 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 requires 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, creates a 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 Explorer. Do not attempt to modify the information in any of the tracker tables with a database tool.

RTInfo Table

While the tracker tables are kept hidden, the RTInfo table is visible through Oracle WebCenter Sites Explorer. This table holds information about which tables are being revision tracked. It has the following columns.

Table 31-1 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.

Table 31-1 (Cont.) RTInfo Table

Column	Description
`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 `shared/rtStorage/AssetType`.
`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.

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.

Implicit vs. Explicit Checkin and Explicit Checkout

When revision tracking is on for an asset type, provides both implicit (or automatic) and explicit (or manual) checkout, and explicit check in 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 check in 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 they 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 returns to the asset. Once an implicitly checked out asset is navigated away from, it must be explicitly checked in when 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 in.

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`.

When revision tracking is enabled for a non-asset table, you can use either the revision tracking features accessible from the menus in Oracle WebCenter 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 have to provide additional support outside of the Oracle WebCenter 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 more information, see the *Tag Reference for Oracle WebCenter Sites* and the *Java API Reference for Oracle WebCenter Sites*.

> **Note:**
>
> If you have to delete a non-asset table from the WebCenter Sites database and that table is being revision tracked, untrack the table before deleting it.

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 requirement 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.

Enabling Revision Tracking

You can enable revision tracking for assets and for non-asset tables, as the following topics describe:

> **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 be corrupt.
>
> It is not recommended to enable Revision Tracking on the target destination.

- Enabling Revision Tracking for Assets
- Enabling Revision Tracking for Non-Asset Tables
- Editing Revision Tracking Settings

Enabling Revision Tracking for Assets

To enable revision tracking for assets:

1. In the **General Admin** tree, expand the **Asset Types** node, and 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 opens.
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. After 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 require the ability to roll back to previous versions, you can set this field to 1.
4. Click **Enable Revision Tracking**.

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 **General Admin** tree, expand the **Admin** node, expand **User Access Management**, and then double-click **Revision Tracking**.
2. In the form that opens, 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 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**.

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 covers the following topics:

- Changing the Root Storage Directory
- Changing the Number of Revisions

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.

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.

1. To increase the number of revisions to be stored for an asset type, simply increase the value in the **Revisions to Keep** field.
2. 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 opens to increase the value.

Although you could have to decrease the number of versions while you are testing configuration settings on a development system or while you are fine-tuning the management system, it is best if you do not decrease the number of versions stored by Revision Tracking on a 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 stored, 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 to modify, **Revision Tracking,** and double-click **Set Revisions**. Use the form to decrease the value.

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, not deleted. The versions stored in it are orphaned, not deleted. 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 orphaned, not deleted.

If you decide to re-enable revision tracking for that asset type, the old versions and text files are ignored and 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 **General Admin** tree, expand the **Admin** node, expand the **Asset Types** node, and then the asset type for which you want to disable revision tracking.
2. Under the selected asset type, expand the **Revision Tracking** node and double-click **Untrack**.

   WebCenter Sites displays a warning message.
3. Click **Untrack**.

Unlocking Revisions

Occasionally, user operations can 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.

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 **General Admin** tree, expand the **Admin** node, and then double-click **Clear Checkouts**.

   The Search for Checkouts form is displayed.
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**.

32 Configuring Vanity URLs

Vanity URLs are short, readable, and easily managed URLs. WebCenter Sites provides you the facility to create vanity URLs that often include the name or other descriptive information about the content on a web page to which the vanity URL points.
Vanity URLs do not replace the existing URLs in the product. They work along with the existing URLs. Users can use existing or vanity URLs to access the content. Content Contributors can auto-generate or create a vanity URL in the Contributor interface.

An example of a "normal" WebCenter Sites URL could be:

```
http://www.example.com/<context>/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), and 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.

Other important information about managing vanity URLs are provided in the following sections:

- Generating Vanity URLs
- Using Vanity URL with System Tools
- Using mod_rewrite with Vanity URLs
- Customizing Vanity URLs

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 Working with Vanity URLs in *Using Oracle WebCenter Sites*). The other method generates a vanity URL from a pattern, and is described in this section.

Topics:

- Configuring Auto-Generated URLs
- Working with WebRoots
- Working with URL Patterns
- Disabling Redirection to Vanity URLs

Configuring Auto-Generated URLs

Vanity URLs based on patterns are automatically generated for individual assets. After 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 for each site that shares 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.

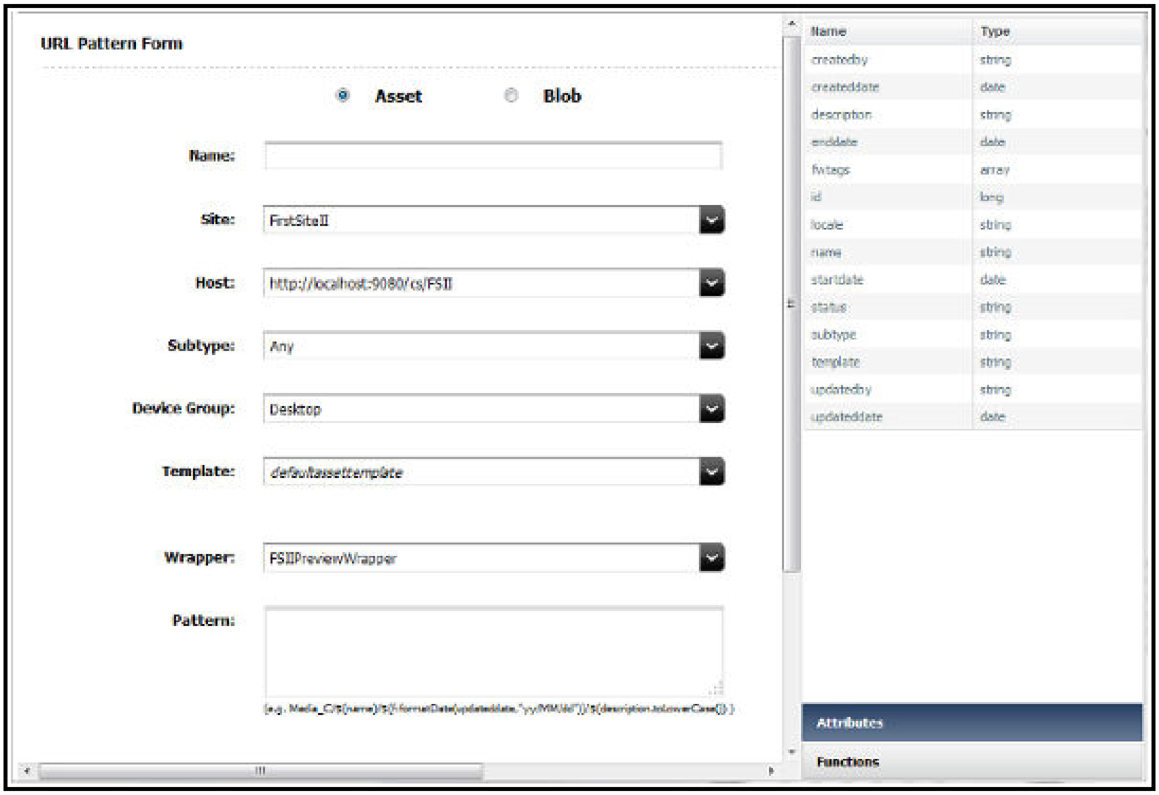
> **Note:**
>
> Changes to patterns are only processed when an asset is created or saved. If you modify a pattern or create a new one it only applies to assets created after the pattern is saved.

To configure an auto-generated vanity URL:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Asset Types** node, 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 then double-click **Add New**.

   The URL Pattern Form opens.

Figure 32-1 URL Pattern Form - Asset



URL Pattern Form

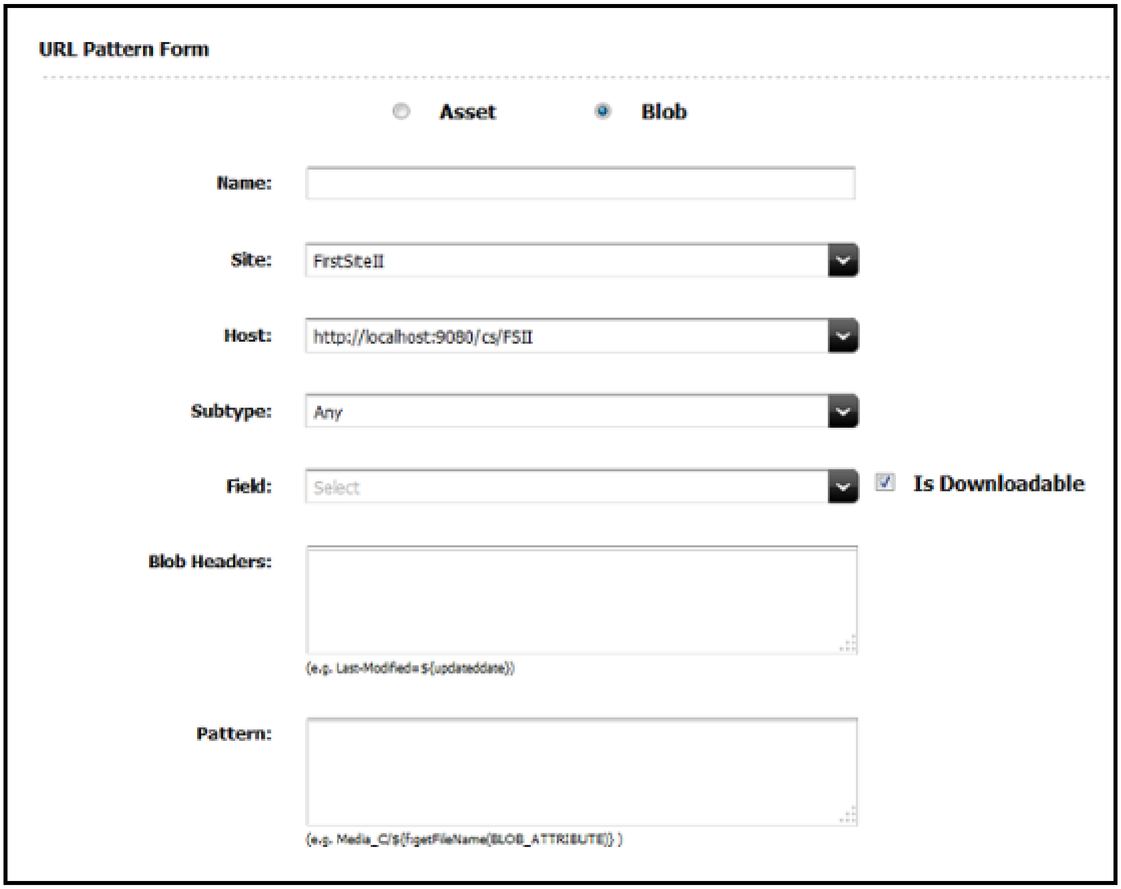
● Asset ○ Blob

Name:	
Site:	FirstSiteII
Host:	http://localhost:9080/cs/FSII
Subtype:	Any
Device Group:	Desktop
Template:	*defaultassettemplate*
Wrapper:	FSIIPreviewWrapper
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

Figure 32-2 URL Pattern Form - Blob (Detail)



URL Pattern Form

○ Asset ● Blob

Name:

Site: FirstSiteII

Host: http://localhost:9080/cs/FSII

Subtype: Any

Field: Select ☑ Is Downloadable

Blob Headers:
(e.g. Last-Modified=${updateddate})

Pattern:
(e.g. Media_C/${fgetFileName(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 can be changed at any time.

> **Note:**
>
> The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), percent (%), ampersand (&), question mark (?), less-than sign (<), greater-than sign (>), and backslash (\).

3. In the **Pattern** field, enter the rule that defines the vanity URL. Lists of valid attributes and functions which can be used are displayed on the right side. See 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 is applied for, and only lists sites for which the asset type is enabled. The selection of the site also narrows 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 could result in a broken page.

6. The **Host** is the WebRoot to which the vanity URL is applied. Only WebRoots that are defined for this site or "any" are displayed. The pattern does not work without a WebRoot.
7. The **Subtype** is the subtype of the asset type to which the vanity URL is applied. The selection of **Subtype** further restricts the available items for **Template** and **Wrapper**.

   > **Note:**
   >
   > The **Subtype** field for blobs do not show `Any` as an option for flex assets. The **Subtype** field for blobs do not appear for basic assets.

8. The **Field** is the blob to be rendered by this URL. This field is present only when a blob is selected. There may be multiple items available if the asset contains multiple blob entries.

   Selecting the **Is Downloadable** checkbox has 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 defined device groups. This field is present only when an asset is selected. Device groups are formatted into lists depending on the extension and regardless of their current state within the site. For more information on device groups, see About Device Groups and Suffixes in *Developing with Oracle WebCenter Sites*.
10. The **Template** is the template for the URL and all templates valid for the asset type, subtype and site are displayed regardless of the existence of a mobile variation. All templates that are valid are 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 non-default device group.
11. The **Wrapper** provides a list of all wrapper elements that are valid for the selected **Template**. If none exist then this field is disabled.
12. At this point click **Save** to save the pattern and continue working, or evaluate the URL pattern.

    The Evaluate URL Pattern section of the form shows an example of what a pattern generated vanity URL will look like. It is recommended to evaluate the URL pattern before saving to validate your pattern.

    > **Note:**
    >
    > If there is no asset of the given type then you cannot use this feature.

13. The **Asset** list allows you to select an asset currently defined for the pattern type. The "look ahead" feature also works to enter an asset's name. After an asset is selected, WebCenter Sites evaluates the pattern and shows how this asset's URL will appear. This is just an example of the URL's final appearance. The URL is not valid until the pattern is saved and the asset is edited and saved.

Working with WebRoots

WebRoots form integral parts of vanity URLs. WebRoots are required to distinguish between vanity URLs which have the same base URL, but are used for different environments, for example production, management, administrative, and so on.

The WebRoot is used to control how the vanity URL is interpreted. WebRoots are like assets. After it is 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 required 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. Using VirtualRoot requires setting an environment identifier (`sites.environment`) in the `wcs_properties.json` file 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 can have reason to have both absolute and relative roots defined at the same time.

The following table lists the advantages and disadvantages of these types of WebRoots.

Table 32-1 Advantages and Disadvantages of WebRoot Types

WebRoot Type	Advantages	Disadvantages
Absolute WebRoot	• Full URL opens to the Content Contributor	• A unique URL is required for every environment. • Virtual roots require setting a WebCenter Sites property. • Difficult to test vanity URLs before publishing.
Relative WebRoot	• Single WebRoot works across all Systems • Vanity URLs can be easily tested before publishing	• Content Contributors do not see the full URL, only the Path. • May require additional steps in rewriting the URL.
Combination	• Full URL opens to the Content Contributor • Single WebRoot works across all Systems • Vanity URLs can be easily tested before publishing	• More URLs are present and must be stored/published. • Content contributors will see both relative and absolute WebRoot URLs. Since there are more URLs it can be confusing to the Content Contributor.

Different Types of WebRoots Required For a URL: Examples

The following examples use the same vanity URL used in the introduction — `http://www.example.com/Running/Home` — and 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`

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 the Property Management Tool, set the `sites.environment` property to `management`. The site is `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 is `avisports`.

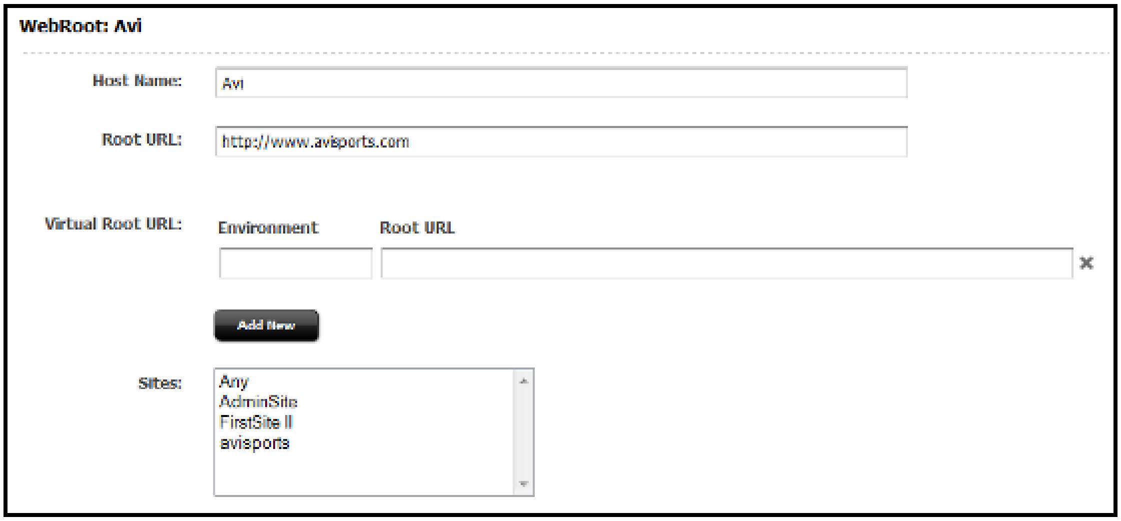
Creating a New WebRoot

To create a new WebRoot:

1. From the **General Admin** tree, expand the **Admin** node, expand the **WebRoots** node, and then double-click **Add New**.

   The WebRoot edit form opens, as shown in the following figure, but with all fields blank.

Figure 32-3 WebRoot Edit Form



WebRoot: Avi

Host Name: Avi

Root URL: http://www.avisports.com

Virtual Root URL: Environment | Root URL

Sites:
Any
AdminSite
FirstSite II
avisports

2. Enter a name in the **Host Name** field.

This name is the unique identifier for the WebRoot instance. It can be used in render tags to get a specific vanity URL of an asset. This value cannot be changed after the WebRoot is saved.

> **Note:**
>
> The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), greater-than sign (>), percent (%), and question mark (?). Additionally, the name cannot end with a backslash (\).

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. 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. See the examples of Root URLs in the following table:

Table 32-2 Examples of Root URLs

Absolute Root URL...			Relative Root URL...	
Using Only the Hostname	**Using Both the Hostname and Prefix for the Path**	**Using the Hostname for a Mobile Site**	**Using**	**Using**
Production URL: http://www.example.com Complete vanity URL: http://www.example.com/Running/Home	Production URL: http://www.example.com/Prefix Complete vanity URL: http://www.example.com/Prefix/Running/	Production URL: http://www.example.mobi Complete vanity URL: http://www.example.mobi/Running/Home	Production URL: /<context>/example Complete vanity URL: http://<Current Host Name>/<context>/example/Running/Home	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 `sites.environment=<name>` in the `wcs_properties.json` file. Set this property from the Property Management Tool in the Admin interface. For instructions, see the Properties in the Core Category in *Property Files Reference for Oracle WebCenter Sites*.
   - **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. The following table provides fields of the virtual URL that you need to populate:

Table 32-3 Fields of Virtual URLs

Management URL	Using Both the Hostname and Prefix for the Path	Using the Hostname for a Mobile Site
If, for example, management URL is http://management.example.com/Running/Home, populate the fields as follows: • Environment:`management` • Root URL:`management.example.com` • `sites.environment=management` in `wcs_properties.json` (use the Property Management Tool to set this property).	If, for example, staging URL is http://staging.example.com/Running/Home, populate the fields as follows: • Environment:`staging` • Root URL:`management.example.com` • `sites.environment=staging` in `wcs_properties.json` (use the Property Management Tool to set this property).	If, for example, development (including port) URL is http://dev.example.com:8080/Running/Home, populate the fields as follows: • Environment:`dev` • Root URL:`dev.example.com:8080` • `sites.environment=dev` in `wcs_properties.json` (use the Property Management Tool to set this property).

> **Note:**
>
> Virtual Root URL is not used in display in the interface. 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 `wcs_properties.json`.

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`, because these URLs are unique to this site.

6. After all information is added, click the **Save** icon.

   The WebRoot is created.

After the WebRoot is created for the first time, it is available as an asset type in the **Admin** tab under the **Asset Type** node.

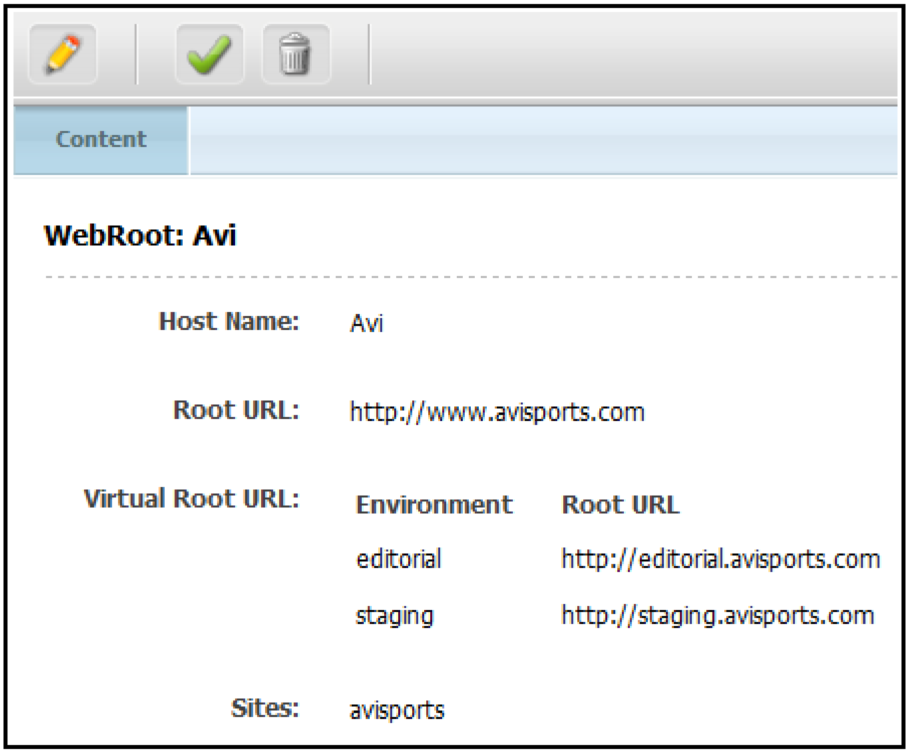
Viewing a WebRoot

To view a WebRoot:

- From the **General Admin** tree, expand the **Admin** node, expand the **WebRoots** node, and then double-click a listed WebRoot.

  The WebRoot Content form opens, showing the saved information for that WebRoot.

Figure 32-4 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

Editing an Existing WebRoot

To edit an existing WebRoot:

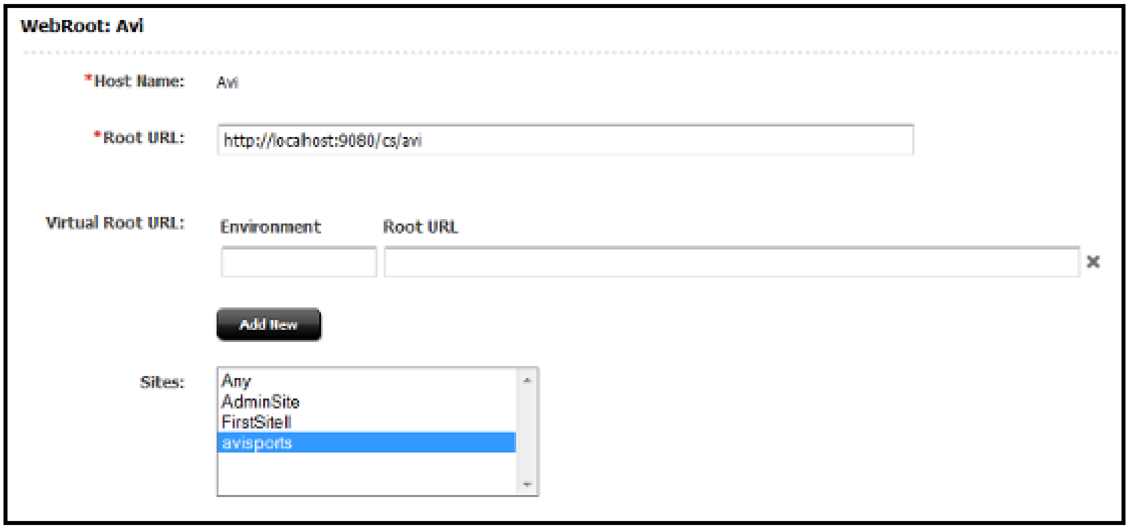
1. From the **General Admin** tree, expand the **Admin** node, expand the **WebRoots** node, and then double-click a listed WebRoot.

   The WebRoot Content form opens.

2. Click the Edit icon to edit the WebRoot.

   The WebRoot Edit form opens.

Figure 32-5 WebRoot Edit Form



WebRoot: Avi

*Host Name: Avi

*Root URL: http://localhost:9080/cs/avi

Virtual Root URL: Environment Root URL

Sites: Any
AdminSite
FirstSiteII
avisports

3. The stored information opens 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 `sites.environment=<name>` in the `wcs_properties.json` file. Set this property from the Property Management Tool in the Admin interface. For instructions, see the Properties in the Core Category in *Property Files Reference for Oracle WebCenter Sites*.
   - **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.

4. After all information is added, click the **Save** icon. The WebRoot is updated.

Approving and Publishing a WebRoot

WebRoot can be approved for publishing in the same way other assets are.

1. Click the green checkmark icon when previewing the WebRoot.
2. The administrator must approve and publish the WebRoot separately. WebRoots are not set as blocking assets.

Working with URL Patterns

Automatic patterns are the heart of vanity URLs. However, 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 that are present will differ based on the asset type selected, but all asset types share some common variables. The following table lists the most common variables.

Table 32-4 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

Click a variable to place it after the pattern field. It does not make a difference where the cursor is located, 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 have 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 have to be formatted; a function `formatDate` is provided for this.

Other types, such as url, may not be usable or may require a function to generate 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. For example, on the string variable "name," `name.toLowerCase()` converts the name to lower case.

This section covers the following topics:

- Using Functions
- Using Patterns
- Deleting Patterns

Using Functions

Functions are useful in converting and formatting data to include in the vanity URL. There are predefined functions in that are useful in creating vanity URLs.

1. Click a function to show it after the pattern field.

   It does not make a difference where the cursor is located, the new pattern is always the last item. The following table provides a list of predefined functions:

Predefined Functions	Description
`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 it displays as "2/28/2013". The string used 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

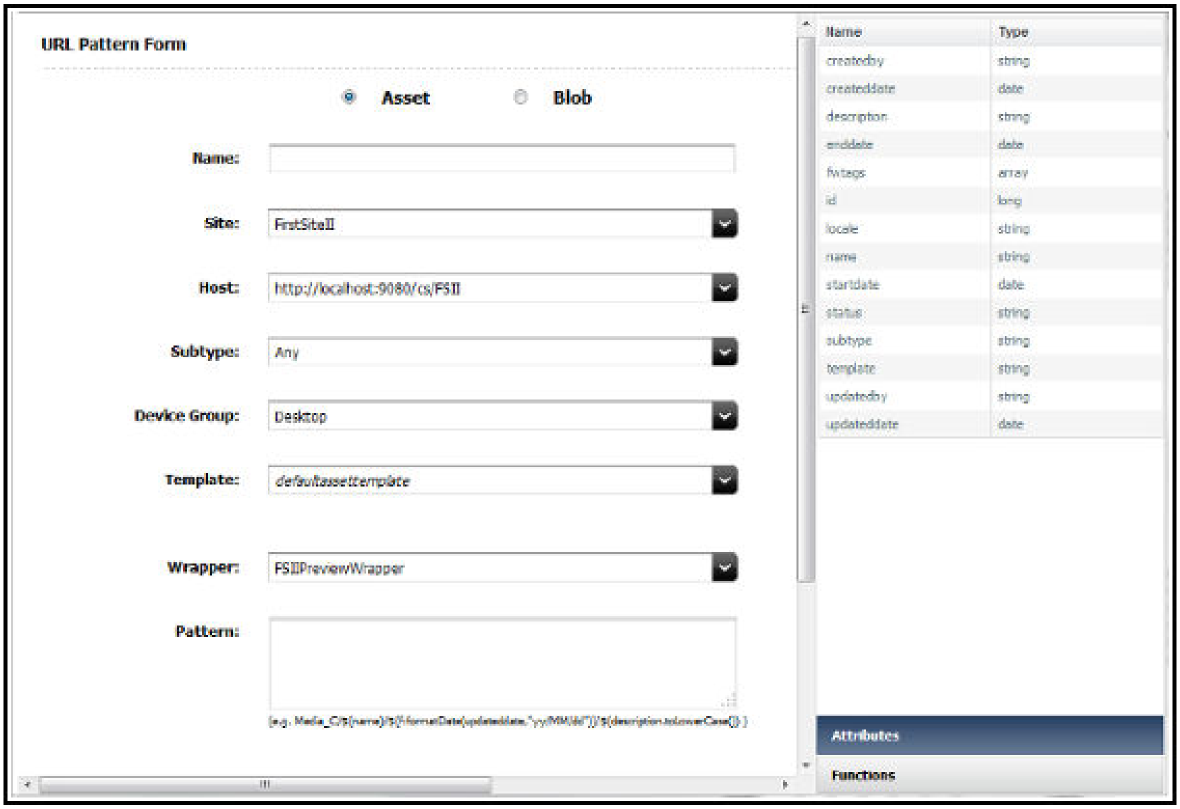
```
${f:spaceToUnderscore("Page 001")}
```

2. 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:`.

Using Patterns

The examples are based off the avisports Article asset type. The asset name is "Baker Likely to Stay With Stars."

Figure 32-6 URL Pattern Form - Asset



URL Pattern Form

● Asset ○ Blob

Name:	
Site:	FirstSiteII
Host:	http://localhost:9080/cs/FSII
Subtype:	Any
Device Group:	Desktop
Template:	defaultassettemplate
Wrapper:	FSIIPreviewWrapper
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

> **Note:**
>
> The first part of the URL is always the WebRoot. This is the local system name and context root in the example. The next part, Avi, is the filter prefix. Only the unique path is provided in these examples.

Given the asset "Baker Likely to Stay With Stars," there are several options available to generate a URL. Each example shows the function and the resulting URL.

Steps	Function	Resultant URL
Start with a simple name pattern.	`/Article/${name}`	`/Article/Baker+Likely+to+Stay+With+Stars`
Convert the name to lower case.	`/Article/${name.toLowerCase()}`	`/Article/baker+likely+to+stay+with+stars`
Remove the spaces from this URL	`/Article/${f:spaceToUnderscore(name)}`	`/Article/Baker_Likely_to_Stay_With_Stars`
Use multiple variables	`/Article/${id}/${createdby}`	`/Article/1328196049037/fwadmin`
Use a function within a function	`/Article/${(f:formatDate(createddate,"yy/MM/dd")}`	`/Article/12-04-27`

Deleting Patterns

Patterns can be deleted using the navigation tree.

To delete a pattern:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Asset Types** node, and then double-click the asset type containing the pattern.

   The selected asset type expands.

2. Double-click **URL Pattern**.

   The currently saved patterns are displayed and can be inspected, edited, or deleted as any other asset.

About Publishing URL Patterns

Patterns are automatically published with the asset type. There is no special approval or publishing required for URL patterns.

Disabling Redirection to Vanity URLs

There are some instances where the developer will want vanity URLs to be temporarily disabled. For example, a developer might want a testing instance that does not redirect to the vanity URL. You can disable redirection to vanity URLs in WebCenter Sites by setting a JVM parameter.

Set the `-Dsites.disablevanityurl` option as a JVM parameter in Java options on the application server.

For example,
`JAVA_OPTIONS=$JAVA_OPTIONS -Dsites.disablevanityurl="true"`

Using Vanity URL with System Tools

Often administrators like to view all the vanity URLs defined across the system in one place, for example, to resolve conflicts. Using the System Tools node, an administrator can view all presently defined vanity URLs and resolve URL conflicts as the following sections describe:

- Viewing All Vanity URLs
- Resolving Vanity URL Conflicts

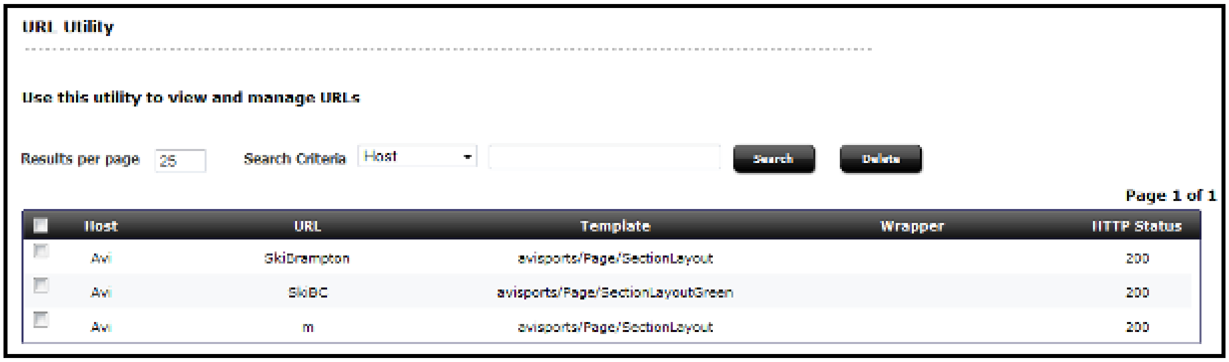
Viewing All Vanity URLs

To view all vanity URLs:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Asset Types** node, and then expand the **System Tools** node.
2. Double-click **URL**.

   The URL Utility Form opens.

Figure 32-7   URL Utility Form



URL Utility

Use this utility to view and manage URLs

Results per page: 25   Search Criteria: Host

Page 1 of 1

☐	Host	URL	Template	Wrapper	HTTP Status
☐	Avi	SkiDrampton	avisports/Page/SectionLayout		200
☐	Avi	SkiBC	avisports/Page/SectionLayoutGreen		200
☐	Avi	m	avisports/Page/SectionLayout		200

Because every URL defined (for all sites) are present on this list, it is 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 inspect if it is a problem or confirm where a conflict is originating. However, if the URL is based upon a pattern then it is automatically regenerated when the asset is next saved.

Resolving Vanity URL Conflicts

Vanity URLs are unique in that there cannot be multiple vanity URLs with the same path (even across WebCenter Sites). For example, 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.

Whichever page is saved first will succeed and create a vanity URL, but, the second page will fail to create a vanity URL.

1. When this occurs, use the System Tool to locate what is blocking the creation of the Vanity URL and, if required, remove the existing URL to allow the new one to be created.
2. This issue can be mitigated by ensuring that the WebRoot is unique to a single site.

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.

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 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 it must have the context root and a unique prefix to ensure that the proper URLs are treated as vanity URLs.

> **Note:**
>
> "FSII" and "Avi" are the parameters pre-configured for URL rewrite filter when sample sites are installed. Values are set in the site.prefix property.

To enable the rewriter filter:

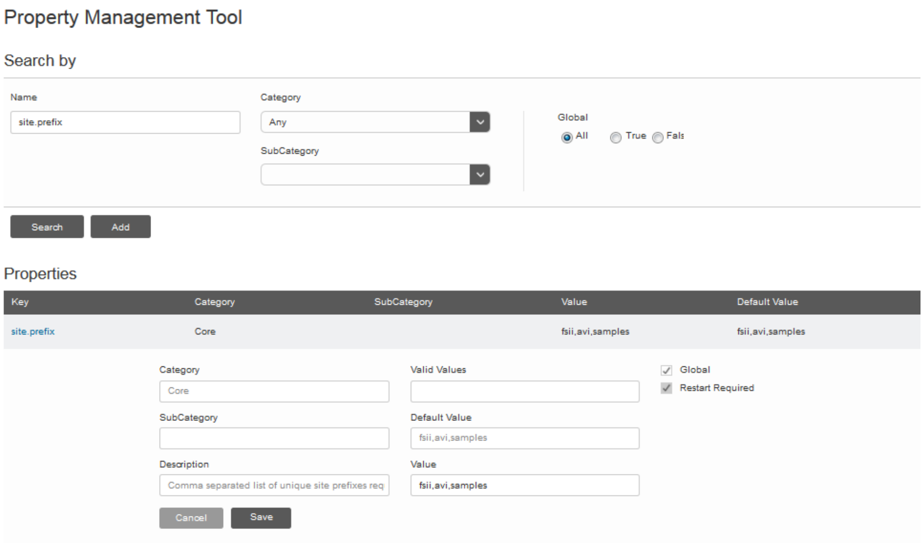
1. In the **General Admin** tree, expand the **Admin** node, expand the **WebRoots** node, and then double-click **Add New**.

   The WebRoot edit form opens.

2. Create a WebRoot.

   See Creating a New WebRoot.

3. In the Admin interface, open the Property Management Tool, and search for the `site.prefix` property. (For instructions on accessing the Property Management Tool, see the Properties in the Core Category in *Property Files Reference for Oracle WebCenter Sites*.)



Property Management Tool

Search by

Name: site.prefix

Category: Any

SubCategory:

Global: ● All ○ True ○ Fals

Properties

Key	Category	SubCategory	Value	Default Value
site.prefix	Core		fsii,avi,samples	fsii,avi,samples

Category: Core

Valid Values:

☑ Global

☑ Restart Required

SubCategory:

Default Value: fsii,avi,samples

Description: Comma separated list of unique site prefixes req

Value: fsii,avi,samples

4. Add the newly created WebRoot to the comma separated value of `site.prefix` property.
5. Restart the server.

For example:

Assume a relative WebRoot 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
```

Because 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. Required parameters have been removed from vanity URLs, though sometimes additional data must be passed as part of the URL. In such cases the existing parameters are concatenated onto the existing URL. For instance, passing in `time=true` the URL above sent to sites would look like:
>
> ```
> http://<Sites Host>:<Sites Port>/<Context root>/Avi/Running/Home?
> time=true
> ```

Using Vanity URLs 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, complete these items:

1. WebRoot must be set for the Mobile domain names. The Root URL of the WebRoot should have mobile domain names.
2. Choose the right host names when creating vanity URLs for mobile device groups. This ensures that the links are correctly generated for the mobile domain names.
3. All other steps including Web Server URL Rewrite are the same as mentioned in Working with WebRoots, Generating Vanity URLs, and Using Vanity URL with System Tools.

Using the Web Server with Vanity URLs

If there is a requirement to access vanity URLs in a similar way to the filter and you want to have all the control and power that URL rewriting entails, you can use a web server.

When using a web server, all rewritten vanity URLs must be split into two sections before sending them to WebCenter Sites. One is the WebRoot, the other is the path. For example, use the WebRoot `http://www.example.com` and a pattern that generates `/Running/Home` for the URL `http://www.example.com/Running/Home`. To pass this URL 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 sees the WebRoot (full server name, port and prefix if absolute WebRoot, or only the prefix if relative WebRoot). The lookuppage sees the remainder of the PATH after the prefix. Using the above URL as an example, the URL passed to WebCenter Sites would have 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. Required parameters have been removed from vanity URLs, though sometimes additional data must be passed as part of the URL. In such cases the existing parameters are concatenated onto the existing URL. For instance, passing in `time=true` the URL above sent to sites would look like:
>
> ```
> http://<Sites Host>:<Sites Port>/<Context root>/Avi/Running/Home?
> time=true
> ```

Using mod_rewrite with Vanity URLs

Every web server and site are unique, so the following should be viewed as guidelines 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 directly transform the URL for the WebCenter Sites filter, or it can be used in conjunction with the URL rewriter filter to perform the same action in a way that will allow for much simpler `mod_rewrite` rules. Examples of both methods are provided. To determine which is right for you, decide if the extra step (and hence cost) of having the WebCenter Sites filter break apart the vanity URLs is worth the simplified rewrite rules.

All subsequent steps are based off manually compiled Apache 2.4.x. If using a version shipped with an operating system, the changes required to enable and configure `mod_rewrite` may differ, but the rules and concepts will not.

Topics:

- Setting Up mod_rewrite
- About Converting with mod_rewrite
- Using the WebCenter Sites Filter with mod_rewrite
- Using the Rewriter Filter with mod_rewrite
- Using Static Content with mod_rewrite

Setting Up mod_rewrite

To set up mod_rewrite:

1. Enable `mod_rewrite` by editing `httpd.conf` and uncommenting the following line:

   ```
   LoadModule rewrite_module modules/mod_rewrite.so
   ```

2. Enable the Rewrite engine by adding this line to `httpd.conf`:

   ```
   RewriteEngine On
   ```

3. For debugging, only enable logging on mod_rewrite (this is very high cost operation so only use it when debugging):

   ```
   LogLevel alert rewrite:trace6
   ```

4. Save the changes to `httpd.conf`; `mod_rewrite` is now enabled (without any rules). Ensure 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

To provide some examples of `mod_rewrite` rules, first define what the environment looks like. Below is a sample based on the vanity URLs used in this guide. You should use the same elements changing out the values for your environment to start.

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 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`

The following table provides the URL structure:

URL	WebRoot	Path	Query Psarameters
`http://www.example.com/Page/Surfing`	`http://www.example.com`	`/Page/Surfing`	
`http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars`	`http://www.example.com`	`/Article/Baker_Likely_to_Stay_With_Stars`	
`http://www.example.com/Home`	`http://www.example.com`	`/Home`	
`http://www.example.com/Home?time=true`	`http://www.example.com`	`/Home`	`time=true`
`http://dev.example.com:8080/Running/Home`	`/Avi`	`/Running/Home`	

There are two filters. The first is the vanity URL filter, the second is the rewrite filter. When using the rewrite filter it passes the data to the vanity URL filter transparently.

When using the 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 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`

About 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:**
>
> 1. `mod_rewrite` is an optional component that you have previously enabled. If you create a `mod_rewrite` rule without actually enabling the `mod_rewrite` component, 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.

This example uses the first URL for production:

```
http://www.example.com/Page/Surfing
```

Create this rule for it:

```
RewriteRule ^(.*)$ /<context>/Sites?lookuphost=http://
www.example.com&lookuppage=$1 [L]
```

This rule has four component parts which are described in the following table:

`RewriteRule –`	**`^(.*)$`**	**`/<context>/Sites?lookuphost=http://www.example.com&lookuppage=$1`**	**`[L]`**
It is a statement that the following entry is a `mod_rewrite` rule).	It is 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.	It is the context root, filter, and path for vanity URL, which breaks down further as follows: • `/<context>/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).	It informs `mod_rewrite` that this is the end of the rules. All `mod_rewrite` rules must end with this.

Using the WebCenter Sites Filter with mod_rewrite

The following rule handles `/Page/Surfing`, but it also handles any other page sent in, extracting everything after the host to the end of the line.

```
RewriteRule ^(.*)$ /<context>/Sites?lookuphost=http://www.example.com&
lookuppage=$1 [L]
```

So the rule also handles:

```
http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars
```

and

```
http://www.example.com/Home
```

However, this rule does not handle the following URl because it has a query parameter being passed in along with the host and path:

```
http://www.example.com/Home?time=true
```

To handle the query parameter, do the following:

1. Use a `mod_rewrite` built-in variable "`%{QUERY_STRING}`" to update the `/<context>/Sites? lookuphost=http://www.example.com& lookuppage=$1` rule:

   to:

   ```
   /<context>/Sites? lookuphost=http://www.example.com& lookuppage=$1&%
   {QUERY_STRING}
   ```

   If no Query parameters are passed in the URL then `%{QUERY_STRING}` is empty. Thus it will work for the three previous URLs, and also for this URL.
2. Next, look at the Development URL:

   ```
   http://dev.example.com:8080/Running/Home
   ```

   This must be passed in to the vanity URL filter as:

   ```
   lookuphost=/<context>/Avi&/<context>/Sites?lookuppage=/Running/Home
   ```

   Because `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 the following rule:

   ```
   RewriteRule ^(.*)$ /<context>/Sites?lookuppage=$1&lookuphost=/
   <context>/Avi [L]
   ```

   In this case because development utilizes a relative WebRoot, we have to adjust the name; as the rest of the URL is unchanged the same rule still applies.

Using the Rewriter Filter with mod_rewrite

When using the rewrite filter, URLs do not have 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 before the introduction of vanity URLs, so it is likely that modify existing rules can be easily modified to support vanity URLs.

The following example uses `http://www.example.com/Page/Surfing` as the URL, which expects `/<context>/AVISPORTS/Page/Surfing`:

1. To handle the above URL, create a new `mod_rewrite` rule:

   ```
   RewriteRule ^(.*)$ /<context>/AVISPORTS$1 [L]
   ```

   This rule simply takes what is given and prefixes `/<context>/AVISPORTS` to it.

   The same rule works for:

   `http://www.example.com/Article/Baker_Likely_to_Stay_With_Stars` and `http://www.example.com/Home`

2. To handle `http://www.example.com/Home?time=true`, add the `%{QUERY_STRING}` parameter to the rule, so it becomes:

   ```
   RewriteRule ^(.*)$ /<context>/AVISPORTS$1?%{QUERY_STRING}  [L]
   http://www.example.com/Home?time=true
   ```

Using Static Content with mod_rewrite

An issue 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. Often the rewrite rules will result in making them inaccessible.

1. Use the following example and assume that the CSS files are kept in a directory named `AVISPORTS_STATIC`:

   ```
   http://www.example.com/Page/Surfing
   ```

2. To handle this, a second `mod_rewrite` rule is required. This rule precedes the former rule which ensures that calls to this directory are not rewritten.

   This can be written as:

   ```
   RewriteRule ^.*/AVISPORTS_STATIC/(.*)$ /<context>/<site-prefix>/$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 About Converting with mod_rewrite) sends the remaining URLs to the vanity URL filter.

   ```
   RewriteRule ^.*/AVISPORTS_STATIC/(.*)$ /<context>/<site-prefix>/$1 [L,PT]
   RewriteRule ^(.*)$ /<context>/Sites?lookuphost=http://
   www.example.com&lookuppage=$1&%[QUERY STRING} [L]
   ```

Customizing Vanity URLs

WebCenter Sites allows you to customize auto-generated vanity URLs, so that you can dynamically set them.

The system checks for the presence of custom elements in pre-defined locations, before an asset is saved. The presence of the following custom elements, according to precedence, is checked:

- `CustomElements/VanityUrl/sitename/assettype/ModifyWebrefs`
- `CustomElements/VanityUrl/assettype/ModifyWebrefs`
- `CustomElements/VanityUrl/sitename/ModifyWebrefs`

- `CustomElements/VanityUrl/ModifyWebrefs`

Where `sitename` is the name of the site where the user is logged in and `assettype` is the type of the asset that is being saved.

If a custom element is found, it will be executed.

> **Note:**
>
> Only one element according to precedence is executed.

Table 32-5 describes the two objects that are available in the element scope: assetData and webrefs.

Table 32-5 Description of assetData and webrefs

Object	Description
assetData	An instance of AssetData containing current data of the asset.
webrefs	List of instance of WebReference. It contains list of auto-generated webreferences for the current asset. It also acts as a return object. All the webreferences given by this object will be assigned to the asset.

Creating a Customized Vanity URL

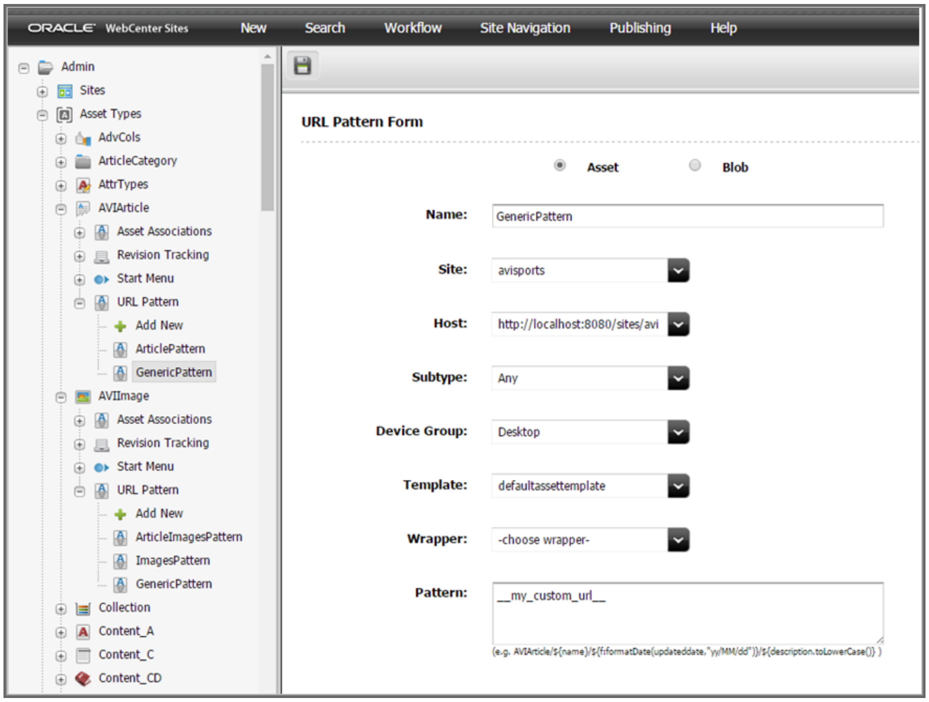
The following example explains how to create a customized vanity URL.

Assume that for assettype AVIArticle in avisports site, you would like to add a vanity URL with category and assetid for AVIArticle assets. You can create this customized URL by the following steps:

1. In the Admin UI, navigate to the **General Admin** tree. Under the **Admin** node, expand **Asset Types**, and then expand the asset type for which you want to create a customized vanity URL. In this case AVIArticle, as shown in the following figure.
2. Create an URL pattern as shown in the following figure. For detailed steps, see Generating Vanity URLs.

   An attempt will be made to create a static URL for the asset.

Figure 32-8   Create an URL Pattern



URL Pattern Form

● Asset ○ Blob

Name: GenericPattern

Site: avisports

Host: http://localhost:8080/sites/avi

Subtype: Any

Device Group: Desktop

Template: defaultassettemplate

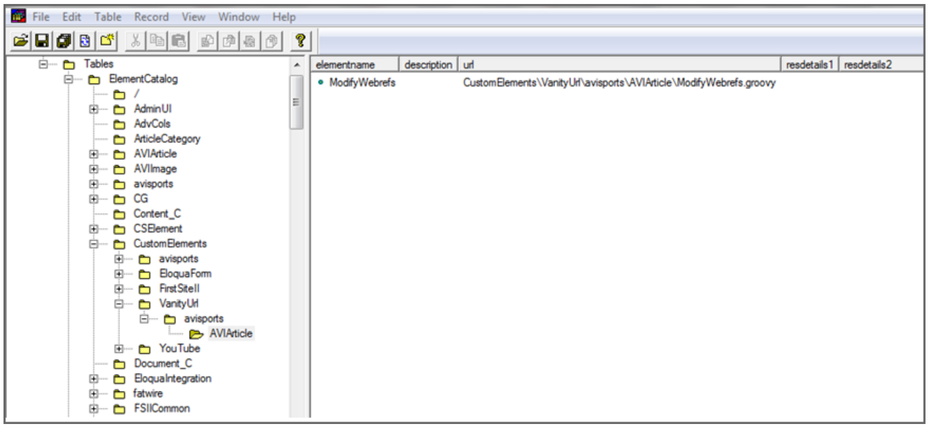
Wrapper: -choose wrapper-

Pattern: __my_custom_url__

(e.g. AVIArticle/${name}/${f:formatDate(updateddate,"yy/MM/dd")}/${description.toLowerCase()})

3. Create a new element ‘`CustomElements/VanityUrl/avisports/AVIArticle/ModifyWebrefs`’. For this example, we have created a groovy element using Sites Explorer as shown in the following figure.

Figure 32-9   Create a Groovy Element Using Sites Explorer



elementname	description	url	resdetails1	resdetails2
ModifyWebrefs		CustomElements\VanityUrl\avisports\AVIArticle\ModifyWebrefs.groovy		

4. Add the following lines to the element and save it.

```
/*
* This element is invoked during asset save. It lets users to edit
the webreferences for the asset. Following objects are available in
ics scope when this element is called
* assetData - updated values of the asset that is being saved
* webrefs   - list of auto generated webreferences for this asset
(input/output)
*
```

```
* Webreferences available in ics will be associated with current
asset being saved. Any changes to assetData object are ignored.
*/

import com.fatwire.assetapi.data.*
import java.util.*

String token = "__my_custom_url__"

// get the asset data and form the vanity url based on asset data
AssetData aData = AssetData.class.cast(ics.GetObj("assetData"))
String category = aData.getAttributeData("category").getData()[0]
String id = String.valueOf(aData.getAssetId().getId())
String url = category+"/"+id

// get the auto generated webreferences and iterate over the list
List webrefs = List.class.cast(ics.GetObj("webrefs"))
Iterator it = webrefs.iterator()
while(it.hasNext())
{
    WebReferenceImpl webref =
WebReferenceImpl.class.cast(it.next())
    // replace existing token with asset data
    if(webref.getAssetUrl().equals(token)) {
        webref.setAssetUrl(url)
    }
}
```

This code reads the AssetData of the current asset and auto generated webreferences from ics object. It reads the category and asset id from AssetData and dynamically creates a new URL. It then loops over all the webreferences to locate one with the static token and then replaces it with the newly formed URL.

> **Note:**
>
> Changes to assetData object does not affect the asset. You can add, remove or update webreferences in webrefs object. All the webreferences given by the webrefs object will be assigned to current asset.

5. Edit and save an Article asset of site avisports to see the custom vanity URL assigned to the asset.

33

Configuring the Lucene Search Engine

Lucene third-party search engine is integrated with WebCenter Sites. Lucene empowers the search feature in the Contributor interface and enables the Public Site Search API to support search capabilities on websites.
Topics:

- Overview of WebCenter Sites Search Functions
- Setting Up Search Indices
- Disabling the Lucene Search Engine
- Maintaining Search Indexes
- Writing Code that Queries the Search Index

Overview of WebCenter Sites Search Functions

When you install WebCenter Sites, the search feature in the WebCenter Sites database and Lucene are available. The Lucene engine is set up as WebCenter Sites is installed, allowing content contributors, website visitors, and third party applications 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.

Topics:

- Indexing for Search Functions
- Using WebCenter Sites Search Functions

Indexing for Search Functions

searches are run against a Lucene search index, not database. A search index is built by an automated process called indexing, which collects, and stores asset data in a format that can be quickly retrieved during a search.

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 are indexed and therefore, searchable. Asset types you omitted from indexing are not indexed and are not searchable.

When the index is built, the Lucene search engine checks every thirty seconds 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. By default, index data is stored in the `<cs_shared_dir>/lucene` directory (where `<cs_shared_dir>` is the shared file system directory).

Using 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 are 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. An asset's user-defined attribute values are stored together in one table cell. Attribute names are omitted. 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 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. See Using Public Site Search in *Developing with Oracle WebCenter Sites.*

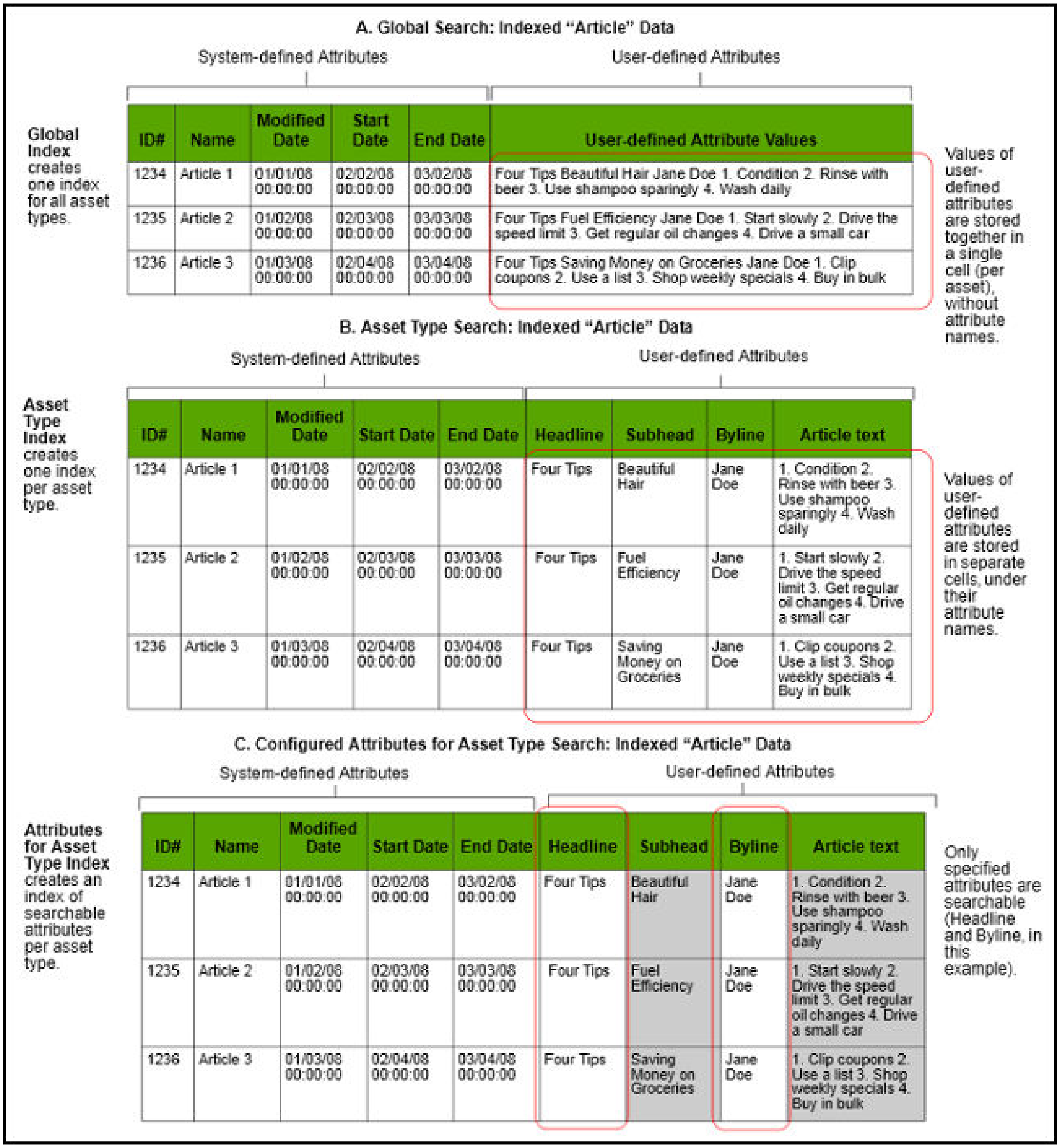
Asset type search has a level of specificity to its search results that Global search does not. An asset type search can return search results more quickly than a Global search. By limiting searches to only the relevant attributes, Asset Type search can eliminate the necessity for a search to run across 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.

The following figure and table 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 stores indexed data.

Figure 33-1 Search Function Tables and Differences



A. Global Search: Indexed "Article" Data

System-defined Attributes | User-defined Attributes

Global Index creates one index for all asset types.

ID#	Name	Modified Date	Start Date	End Date	User-defined Attribute Values
1234	Article 1	01/01/08 00:00:00	02/02/08 00:00:00	03/02/08 00:00:00	Four Tips Beautiful Hair Jane Doe 1. Condition 2. Rinse with beer 3. Use shampoo sparingly 4. Wash daily
1235	Article 2	01/02/08 00:00:00	02/03/08 00:00:00	03/03/08 00:00:00	Four Tips Fuel Efficiency Jane Doe 1. Start slowly 2. Drive the speed limit 3. Get regular oil changes 4. Drive a small car
1236	Article 3	01/03/08 00:00:00	02/04/08 00:00:00	03/04/08 00:00:00	Four Tips Saving Money on Groceries Jane Doe 1. Clip coupons 2. Use a list 3. Shop weekly specials 4. Buy in bulk

Values of user-defined attributes are stored together in a single cell (per asset), without attribute names.

B. Asset Type Search: Indexed "Article" Data

System-defined Attributes | User-defined Attributes

Asset Type Index creates one index per asset type.

ID#	Name	Modified Date	Start Date	End Date	Headline	Subhead	Byline	Article text
1234	Article 1	01/01/08 00:00:00	02/02/08 00:00:00	03/02/08 00:00:00	Four Tips	Beautiful Hair	Jane Doe	1. Condition 2. Rinse with beer 3. Use shampoo sparingly 4. Wash daily
1235	Article 2	01/02/08 00:00:00	02/03/08 00:00:00	03/03/08 00:00:00	Four Tips	Fuel Efficiency	Jane Doe	1. Start slowly 2. Drive the speed limit 3. Get regular oil changes 4. Drive a small car
1236	Article 3	01/03/08 00:00:00	02/04/08 00:00:00	03/04/08 00:00:00	Four Tips	Saving Money on Groceries	Jane Doe	1. Clip coupons 2. Use a list 3. Shop weekly specials 4. Buy in bulk

Values of user-defined attributes are stored in separate cells, under their attribute names.

C. Configured Attributes for Asset Type Search: Indexed "Article" Data

System-defined Attributes | User-defined Attributes

Attributes for Asset Type Index creates an index of searchable attributes per asset type.

ID#	Name	Modified Date	Start Date	End Date	Headline	Subhead	Byline	Article text
1234	Article 1	01/01/08 00:00:00	02/02/08 00:00:00	03/02/08 00:00:00	Four Tips	Beautiful Hair	Jane Doe	1. Condition 2. Rinse with beer 3. Use shampoo sparingly 4. Wash daily
1235	Article 2	01/02/08 00:00:00	02/03/08 00:00:00	03/03/08 00:00:00	Four Tips	Fuel Efficiency	Jane Doe	1. Start slowly 2. Drive the speed limit 3. Get regular oil changes 4. Drive a small car
1236	Article 3	01/03/08 00:00:00	02/04/08 00:00:00	03/04/08 00:00:00	Four Tips	Saving Money on Groceries	Jane Doe	1. Clip coupons 2. Use a list 3. Shop weekly specials 4. Buy in bulk

Only specified attributes are searchable (Headline and Byline, in this example).

Besides the way they index data, the searches enable different functions, as summarized in the following table.

Table 33-1 Search Functions

Global Index	Asset Type Index
Enables searches across all selected asset types.	Enables searches per asset type (multiple asset types can be enabled).

Table 33-1 (Cont.) Search Functions

Global Index	Asset Type Index
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.

See About Search API in *Went through the requirements, scope and sprint plan for Sites CECS integration.*.

Setting Up Search Indices

The major steps to set up Lucene are:

1. Before you can configure Lucene, you must enable it on your system.
2. This indicates to Lucene which assets it should index. You can add asset types to the global index and the asset type index.

   After you have selected which assets to index and enabled binary file indexing, 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 selected asset types, while an asset type index creates one index per asset type.

   After the global index has been created and users conduct a search, assets are returned by the search feature in the Contributor interface or on the live site. After the asset type index is created, and users conduct a search, items are returned by the search feature on the live site.
3. After you have created an index, you can select specific attributes to search on. After an index for these specified attributes is created and populated, and you conduct a search, the specified attributes are returned by the search feature on the live site.
4. 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.

Use the information from the following topics to set up the Lucene engine:

- Enabling the Lucene Search Engine
- Adding Asset Types to the Search Index
- Configuring Attributes for Asset Type Index
- Indexing Binary Files

Enabling the Lucene Search Engine

This section explains how to enable the Lucene engine.

To start the Lucene Engine:

1. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then double-click **Start/Stop Search Engine** Indices.
2. Click **Start Search Engine**.

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.

After 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 also have to delete index data. See Deleting Index Data.

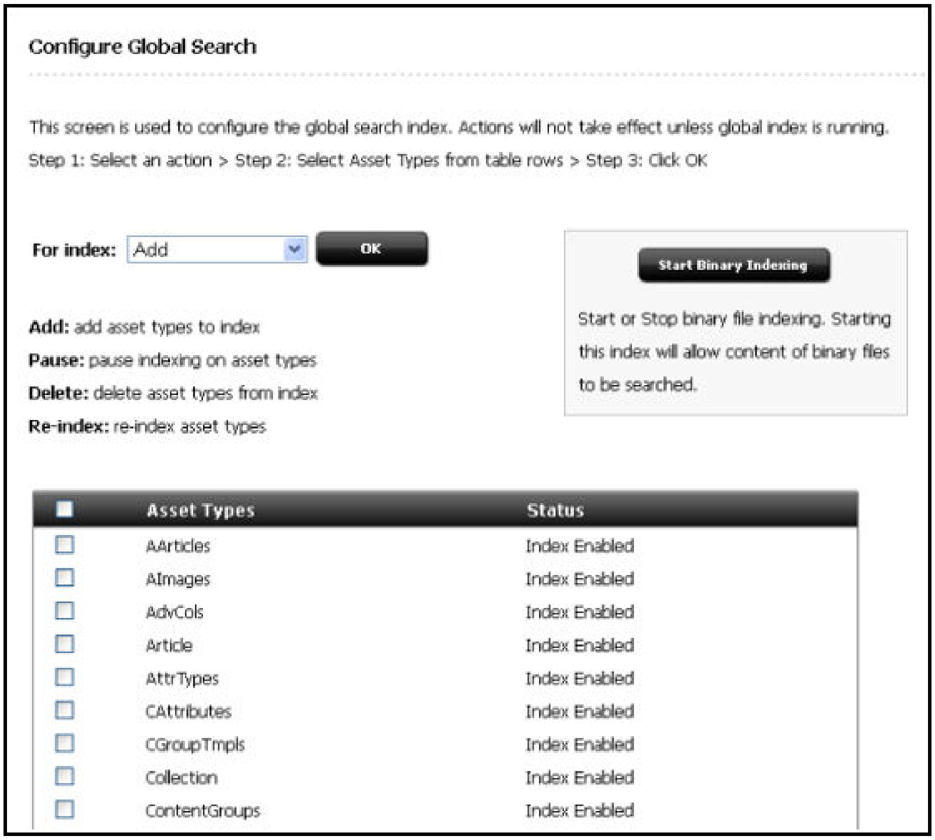
Adding Asset Types to the Search Index

This section explains how to add asset types to the Global Search Index and the Asset Type index. After each initial index has been created, Lucene checks for changes every five seconds. By default, index data is stored in the `<cs_shared_dir>/lucene` directory (where `<cs_shared_dir>` is the WebCenter Sites shared file system directory). After the data is added, it is maintained until indexing is stopped entirely or paused for a selected asset type. Assets of the selected types are not returned by the search feature until Lucene has indexed them.

To add new asset types to the search index:

1. Enable the Lucene engine:
   a. In the **General Admin** tree, expand the **Admin** node, expand the **Search** node, and then 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:** list, select **Add**. WebCenter Sites displays a list of asset types that are not currently being indexed.

Figure 33-2 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

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

□	Asset Types	Status
□	AArticles	Index Enabled
□	AImages	Index Enabled
□	AdvCols	Index Enabled
□	Article	Index Enabled
□	AttrTypes	Index Enabled
□	CAttributes	Index Enabled
□	CGroupTmpls	Index Enabled
□	Collection	Index Enabled
□	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:** list, select **Add**.

   WebCenter Sites displays a list of asset types that are not currently being indexed.

   c. In the list, select the asset types you want to index.

   d. Click **OK**.

   In the confirmation pop-up dialog that opens, 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 wanted (click **Start Binary Indexing**). For more information about binary file indexing, see Indexing Binary Files.

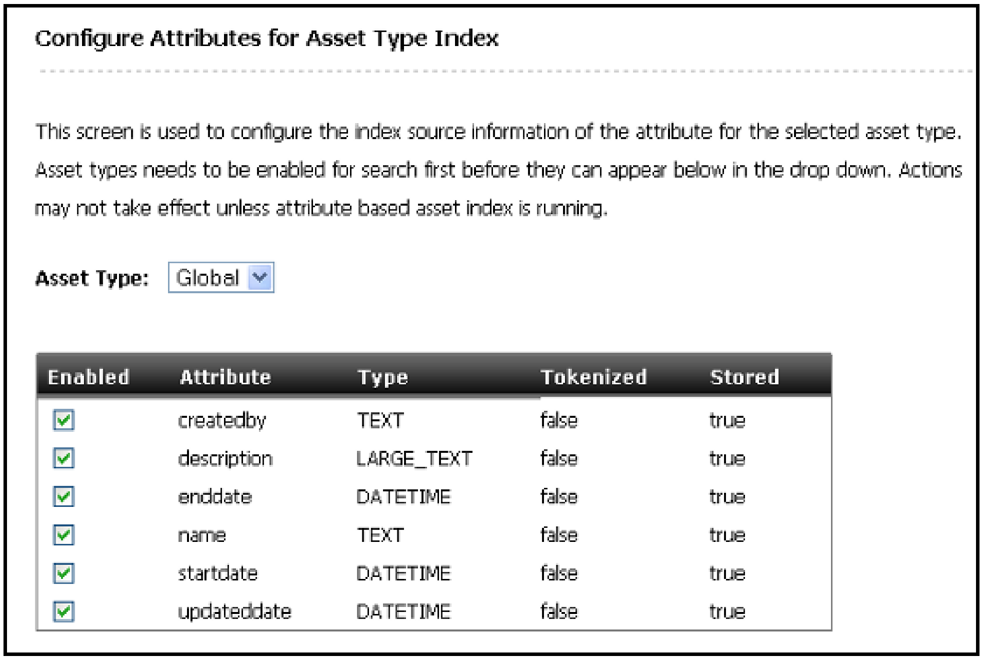
Configuring Attributes for Asset Type Index

You can configure indexing on specific attributes for specific asset types. The selected asset type must be enabled for indexing first before you can select any specific attributes for the asset type. After you have enabled Lucene and conducted a search on the live site, assets with indexed attribute data matching the search terms are returned.

To configure attributes for a selected asset type:

1. If you have not done so, add the asset type you want to configure to the asset type index. For instructions, see Adding Asset Types to the Search Index.
2. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then double-click **Configure Attributes for Asset Type Index.**
3. In the **Asset Type:** list, select the asset type you want to configure. WebCenter Sites displays a list of attributes for the selected asset type.

Figure 33-3 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: Global

Enabled	Attribute	Type	Tokenized	Stored
☑	createdby	TEXT	false	true
☑	description	LARGE_TEXT	false	true
☑	enddate	DATETIME	false	true
☑	name	TEXT	false	true
☑	startdate	DATETIME	false	true
☑	updateddate	DATETIME	false	true

4. The following information also opens:
   - **Enabled**: Indicates if the specified attribute is enabled for indexing for that specific asset type. To disable an attribute, deselect its check box.
   - **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 possibly 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. After you have made your changes to the selected asset types, click **Save**.

Indexing Binary Files

Binary files are files of type other than text, such as Word and PDF documents. You can 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 covers the following topics:

- Enabling Indexing of Binary Files
- Disabling Indexing of Binary Files

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 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 done so, enable the Lucene engine:
   a. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then double-click **Start/Stop Search Engine** Indices.
   b. Click **Start Search Engine**.

      The button name changes to **Stop Search Engine**.
2. In the **General Admin** tree, expand the **Admin** node, and then expand **Search**.
3. Complete either:
   - 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.

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 done so, enable the Lucene engine:
   a. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then double-click **Start/Stop Search Engine** Indices.
   b. Click **Start Search Engine**.

      The button name changes to **Stop Search Engine**.
2. In the **General Admin** tree, expand the **Admin** node, and then expand **Search.**
3. Complete either:
   - 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.

Disabling the Lucene Search Engine

You can stop the Lucene engine to improve performance. After the engine is stopped, you are no longer able to add or delete assets, or pause indexing. You are also no longer able to re-index assets.

To stop indexing:

1. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then 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 returns assets included during the last time the index was built. If you wish to delete the index data (and therefore remove search capability), see Deleting Index Data.

> **Note:**
>
> If you stop Global search indexing, the search index used for the Contributor interface search function and the public site is not updated. Thus, search results will not be accurate.
>
> - To remove search capability from Contributor, you also have to delete all assets from the Global index. See Deleting Index Data.
> - If you add or remove assets while indexing is stopped, you must rebuild the index to create an accurate search index when you restart indexing. See Re-indexing.

Maintaining Search Indexes

After you have set up Lucene, you may have to perform tasks such as temporarily suspending indexing to perform bulk operations on assets, re-indexing, deleting index data, or writing code to specifically query the search engine.

This section covers the following topics:

- Pausing and Resuming Indexing
- Re-indexing
- Deleting Index Data

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. 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 then have to index all assets of the type that you added or deleted, using the re-indexing function.

- Pausing Global and Asset Type Indexing
- Resuming Global and Asset Type Indexing

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. 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 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 do not reflect changes made to the database after indexing was paused.

To pause indexing:

1. If you have not done so, enable the Lucene engine:
   a. In the **General Admin** tree, expand the **Admin** node, expand **Search**, and then double-click **Start/Stop Search Engine** Indices.
   b. Click **Start Search Engine**.

      The button name changes to **Stop Search Engine**.
2. In the **General Admin** tree, expand the **Admin** node, and then expand **Search.**
3. Complete either:
   - 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:** list, select **Pause**.

   WebCenter Sites displays the list of asset types for which you can pause indexing.

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 list, stop here. Either indexing is paused for all asset types, or no asset types have yet been selected for indexing.

6. Click the **OK** button next to the list of operation selections.

7. In the confirmation pop-up dialog that opens, 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.

Resuming Global and Asset Type Indexing

After pausing or disabling indexing, you must re-index to ensure that all your asset data is in the index.

To resume indexing:

1. Restart indexing on the paused asset types. See 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.

3. If you deleted assets while indexing was paused, the regular indexing process detects which assets were deleted and removes 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.

   See Deleting Index Data.

Re-indexing

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 are returned only after the Lucene search engine has indexed them.

To re-index assets:

1. If you have not done so, enable the Lucene engine:

   a. In the **General Admin** tree, expand the **Admin** node, expand the **Search** node, and then double-click **Start/Stop Search Engine** Indices.

   b. Click **Start Search Engine**.

2. In the **General Admin** tree, expand the **Admin** node, and then expand the **Search** node.
3. Complete either:
   - 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:** list, select **Re-index**.

   displays the asset types currently selected for indexing.

   > **Note:**
   >
   > If no asset types are displayed when you select **Re-index** from the list, stop here. No asset types are in the indexing queue or indexing has been paused for all asset types in the queue.

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 opens, click **OK**.

   Indexing begins.

   The status of the selected asset types changes to Enabled.

   Updated search results for assets of the selected types are returned only after the Lucene search engine has indexed them.

Deleting Index Data

If you no longer have to perform searches on assets of a particular type, search results can 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 could 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.

When you delete index data, WebCenter Sites first pauses indexing of the assets of the selected asset types, then deletes the index data of those assets. After you delete data from the index, index data is no longer available for assets of the selected types. Search results no longer return data from assets of the selected types.

To delete data from the index:

1. In the **General Admin** tree, expand the **Admin** node, and then expand the **Search** node.
2. Complete either:
   - 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:** list, select **Delete**.

The asset types currently being indexed are displayed.

4. In the list, select the asset types whose index data you want to delete.
5. Click **OK**.
6. In the confirmation dialog box that opens, 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 are no longer be returned by the search feature in the Contributor interface or the public site.

   To make the assets searchable again, you must add the asset types back to indexing. For instructions, see Adding Asset Types to the Search Index.

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 code is used 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.

```
import COM.FutureTense.CS.Factory;
import COM.FutureTense.Interfaces.ICS;
import com.fatwire.cs.core.search.data.ResultRow;
import com.fatwire.cs.core.search.engine.*;
import com.fatwire.cs.core.search.query.Operation;
import com.fatwire.cs.core.search.query.QueryExpression;
import com.fatwire.cs.core.search.source.IndexSourceConfig;
import com.fatwire.cs.core.search.source.IndexSourceMetadata;
import com.fatwire.search.engine.SearchEngineConfigImpl;
import com.fatwire.search.query.QueryExpressionImpl;
import com.fatwire.search.source.IndexSourceConfigImpl;
import com.fatwire.search.source.SearchIndexFields;

import java.util.Collections;

public class SearchTest {
    public static void main(String[] args) {
        SearchTest searchTest = new SearchTest();
        String assetType = "Content_C";
        int maxResults = 100;
        try {
            searchTest.testSelect(assetType, maxResults);
        } catch (Exception e) {
            //
        }
    }

    public void testSelect(String assetType, int maxResults) throws Exception {
        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)
                sourceMd.getProperty(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);
    }
}
```

34

Using the Site Capture Application

Through the Site Capture application in WebCenter Sites, you can download sites in different modes and store them in different directories. You can manage and monitor Site Capture , as well as create and modify the crawlers that are run to download sites. Topics:

- Site Capture Model
- Logging in to the Site Capture Application
- Using the Default Crawlers
- Setting Up a Site Capture Operation
- Enabling Publishing-Triggered Site Capture
- Managing Statically Captured Sites
- About Managing Archived Sites
- Notes and Tips for Managing Crawlers and Captured Data
- General Directory Structure
- Custom Folders

Site Capture Model

You can initiate a crawl session manually from the Site Capture interface, or you can trigger one by the completing a WebCenter Sites RealTime publishing session. In each scenario, the crawler downloads the website to disk in either a static or archive mode, depending on how you choose to run the crawler.

Topics:

- Capture Modes
- Crawlers

Capture Modes

When you download a site 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.

Table 34-1 Static Capture Mode and Archive Mode

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.
You can initiate static crawl sessions manually from the application interface or after a publishing session. However, you can manage downloaded sites from the Site Capture file system only.	Like static sessions, you can manually initiate archive crawl sessions from the Site Capture interface or after a publishing session. However, because the zip files are referenced by pointers in the Site Capture database, you can manage them from the Site Capture interface. You can download the files, preview the archived sites, and set capture schedules.

For any capture mode, logs are generated after the crawl session to provide such information as crawled URLs, HTTP status, and network conditions. In static capture, you must obtain the logs from the file system. In archive capture, you can download them 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.

Crawlers

Starting any type of site capture process requires you to define a crawler in the Site Capture interface. To help you get started quickly, Site Capture comes with two sample crawlers, Sample and FirstSiteII. This guide assumes the crawlers were installed during the Site Capture installation process and uses the Sample crawler primarily.

To create your own crawler, you must name the crawler (typically, after the target site), and upload a text file named `CrawlerConfigurator.groovy`, which controls the site capture process. You must code the `groovy` file with methods in the `BaseConfigurator` class that specify at least the starting URIs and link extraction logic for the crawler. Although the `groovy` file controls the site capture process, the capture mode is set outside the file.

To use a crawler for publishing-triggered site capture, you must take an additional step: you must name the crawler and specify its capture mode on the publishing destination definition on the source system that is integrated with Site Capture, as described in the Configuring Site Capture with the Configurator in *Installing and Configuring Oracle WebCenter Sites*. (On every publishing destination definition, you can specify one or more crawlers, but only a single capture mode.) Information about the successful start of crawler sessions is stored in the Site Capture file system and in the log files (`futuretense.txt`, by default) of the source and target systems.

The exercises in this chapter cover both types of crawler scenarios: manual and publishing-triggered.

Logging in to the Site Capture Application

You access the Site Capture application by logging in to WebCenter Sites.

1. Access WebCenter Sites at the following URL:

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

   In the previous example, `<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 is deployed on the server.

2. Log in as a general administrator. **Login credentials are case-sensitive.** In this guide, we use the admin credentials.

3. Click **Login**.

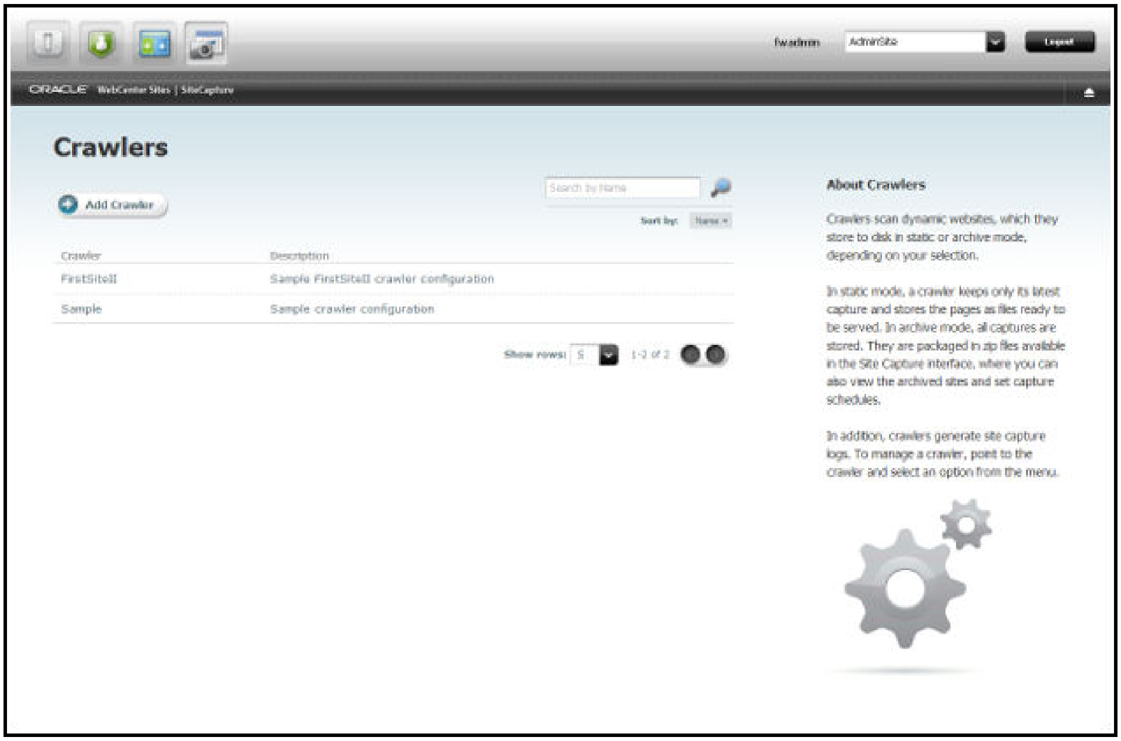
4. If you are logging in for the first time, the following dialog opens:

   Select the AdminSite (to which the Site Capture application is assigned by default) and select the Site Capture icon.

   The Crawlers page opens.

5. If the default crawlers were installed with Site Capture, they are listed under the names **Sample** and **FirstSiteII**.

Figure 34-1 Crawlers Page



Crawler	Description
FirstSiteII	Sample FirstSiteII crawler configuration
Sample	Sample crawler configuration

About Crawlers

Crawlers scan dynamic websites, which they store to disk in static or archive mode, depending on your selection.

In static mode, a crawler keeps only its latest capture and stores the pages as files ready to be served. In archive mode, all captures are stored. They are packaged in zip files available in the Site Capture interface, where you can also view the archived sites and set capture schedules.

In addition, crawlers generate site capture logs. To manage a crawler, point to the crawler and select an option from the menu.

6. Your next step depends on your requirements:

   - To learn more about the default crawlers, continue to 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 Setting Up a Site Capture Operation.
- To learn about crawler configuration code, see Coding the Crawler Configuration File in *Developing with Oracle WebCenter Sites*.

Using the Default Crawlers

You should have installed the default crawlers, Sample and FirstSiteII, with the Site Capture application, and they are displayed in its interface. To define your own crawlers, see Defining a Crawler.

Topics:

- Sample Crawler
- FirstSiteII Crawler
- Running a Default Crawler

Sample Crawler

You can use the Sample crawler 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 can 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.

For more information about these methods, see Crawler Customization Methods in *Developing with Oracle WebCenter Sites*.

FirstSiteII Crawler

The FirstSiteII crawler is used to download the 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. See Interfaces in *Developing with Oracle WebCenter Sites*.

Running a Default Crawler

In this section, you will run either the Sample crawler or the FirstSiteII crawler. Using the FirstSiteII crawler requires you to publish the WebCenter Sites FirstSiteII sample site.

1. On the Crawlers page, point to a default crawler – **Sample** or **FirstSiteII** – and select **Edit Configuration**.

> **Note:**
>
> If the default crawlers are not listed, skip to Setting Up a Site Capture Operation to define your own crawler.

2. Set the starting URI for the crawler by editing the crawler configuration file. For instructions, skip to step 1 in Defining a Crawler, and continue with the rest of the steps to run the crawler and manage its captured data.

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.

Topics:

- Creating a Starter Crawler Configuration File
- Defining a Crawler
- Editing the Crawler Configuration File
- Starting a Crawl

Creating a Starter Crawler Configuration File

Before you can create a crawler, you must have a configuration file that controls the site capture process for the crawler. The fastest way to create a useful file is to copy sample code and recode, as necessary.

1. Copy the Sample configuration file to your local computer in either of these 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 computer, and save the file as `CrawlerConfigurator.groovy`.
   - Go to the Site Capture host computer and copy the `CrawlerConfigurator.groovy` file from `<SC_INSTALL_DIR>/fw-site-capture/crawler/Sample/app/` to your local computer.

> **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 computer.
>
> 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. See Managing Statically Captured Sites.

2. Your sample `groovy` file specifies a sample starting URI, which you will reset when you create the crawler in the next step. (In addition to the starting URI, you can set crawl depth and similar parameters, call 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 do the latter, continue to the next step Defining a Crawler.

Defining a Crawler

To define a crawler:

1. Go to the Crawlers page and click **Add Crawler**.
2. On the Add Crawlers page:
   a. Name the crawler after the site to be crawled.

   > **Note:**
   >
   > - After you save a crawler, it cannot be renamed.
   > - This guide assumes that every custom crawler is named after the target site and is not 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 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 computer. You can edit the file directly in the Site Capture interface.

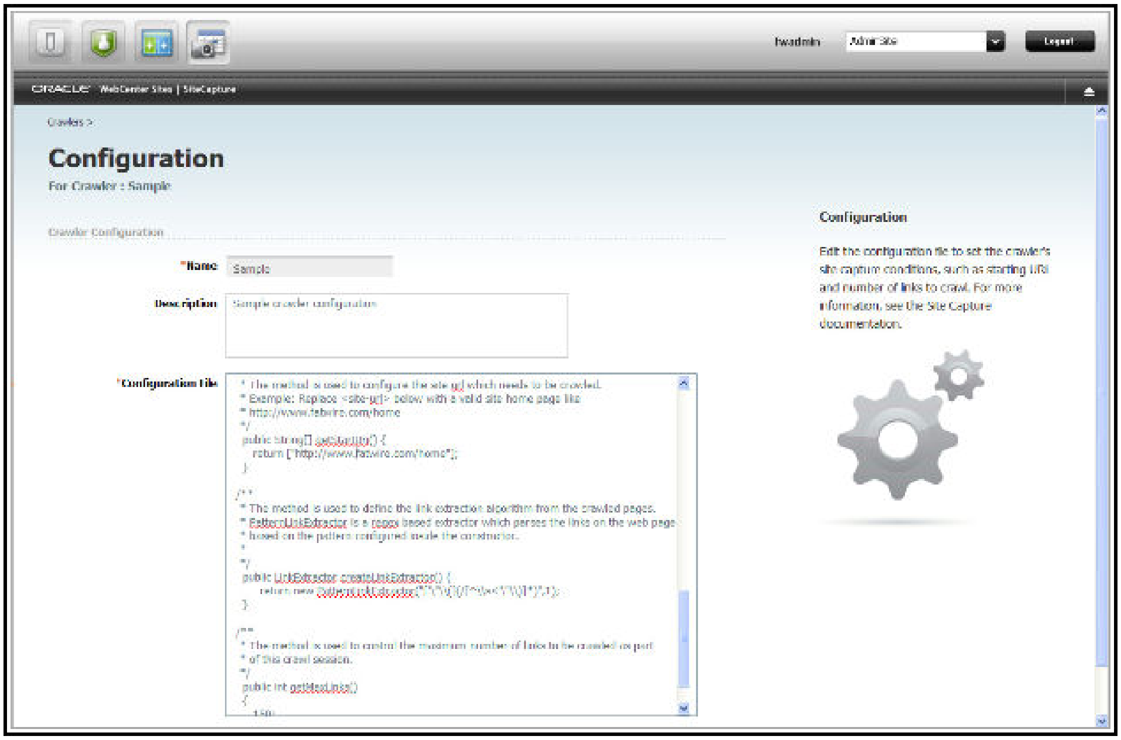
3. Continue to Editing the Crawler Configuration File.

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 starting URI.

1. On the Crawlers page, point to the crawler you just defined and select **Edit Configuration**.

Figure 34-2 Configuration Page



*Name: Sample

Description: Sample crawler configuration

*Configuration File:

```
 * The method is used to configure the site url which needs to be crawled.
 * Example: Replace <site-url> below with a valid site home page like
 * http://www.fatwire.com/home
 */
 public String[] getStartUri() {
   return ["http://www.fatwire.com/home"];
 }

/**
 * The method is used to define the link extraction algorithm from the crawled pages.
 * PatternLinkExtractor is a regex based extractor which parses the links on the web page
 * based on the pattern configured inside the constructor.
 *
 */
 public LinkExtractor createLinkExtractor() {
     return new PatternLinkExtractor("['\"\\(](/[^\\s<'\"\\)]*)",1);
 }

/**
 * The method is used to control the maximum number of links to be crawled as part
 * of this crawl session.
 */
 public int getMaxLinks()
 {
   150;
```

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.

The Configuration File field displays the crawler's `CrawlerConfigurator.groovy` file, located in `<SC_INSTALL_DIR>/fw-site-capture/crawler/<crawler name>/app`.

2. Set the starting URI for the crawler in the following method:

```
public String[] getStartUri() {
  return ["http://www.example.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.example.com/product","http://www.example.com/support"];  }
> ```
>
> Your configuration file includes the `createLinkExtractor` method, which calls the logic for extracting the links to be crawled. The links are extracted from the markup that is downloaded during the crawl session. For additional information about this method and the extraction logic, see createLinkExtractor in *Developing with Oracle WebCenter Sites*.
>
> 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 must stop the application server. You can stop archive captures from the Site Capture interface.
>
> See the Coding the Crawler Configuration File in *Developing with Oracle WebCenter Sites*.

3. Click **Save**.
4. Continue to Starting a Crawl.

Starting a Crawl

You can start a crawl in several ways. If you used a crawler in one mode, you can rerun it in a different mode:

- Run the Crawler Manually in Static Mode
- Run the Crawler Manually in Archive Mode
- Schedule the Crawler for Archive Capture
- About Publishing a Site in RealTime Mode

Run the Crawler Manually in Static Mode

To 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 menu.

   The Crawlers page displays the following message when capture begins:

   "**Success.** `Static capture started by crawler <crawlerName>.`"

2. At this point, the Site Capture interface does not display any other information about the crawler or its process, nor does 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 called 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`.

See Managing Statically Captured Sites.

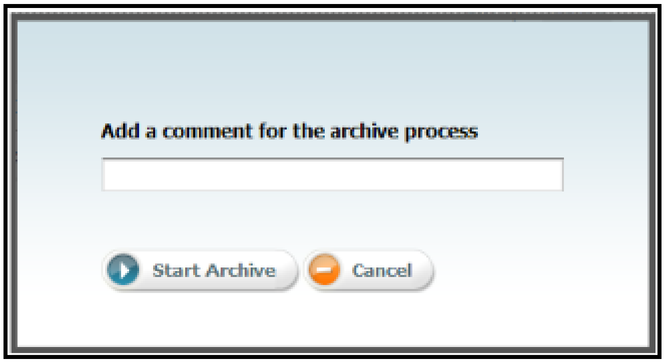
Run the Crawler Manually in Archive Mode

To run the crawler manually in archive mode:

1. On the Crawlers page, point to the crawler that you created and select **Start Archive**.

   A comment dialog opens.
2. In the dialog, add a comment about the upcoming job:

Figure 34-3 Comment Dialog



Add a comment for the archive process

> **Note:**
>
> You cannot add a comment after a crawler starts running.
>
> If you choose to add a comment in the dialog above, it opens 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 opens, where you can manage the archive process in several way. To follow this exercise, click **Refresh** (next to **Job State**) until **Finished** opens.

   - **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 crawlers session. Any captured resources are archived and the job state is changed from **Running** to **Finished** (click **Refresh** to see the change).
   - **Preview** is displayed when the job state is **Finished**. Clicking **Preview** displays the archived site.
   - **Cancel** redirects you to the Jobs page. The crawler, if running, continues to run.

4. When the archive crawl ends, results are made available in the Site Capture interface. For example:

   - The crawler report opens 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.
   - Click **Preview** on the Job Details page to render the archived site. Next to the site is the Archive ID table with archive management options, which are displayed when you point to an archive.

     > **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 About Managing Archived Sites.

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 that they do not overlap.

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.
3. Click **Save** and add another schedule if necessary.

About Publishing 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 call one or more crawlers to capture the newly published site. For instructions, see Enabling Publishing-Triggered Site Capture.

About Managing Captured Data

For more information about accessing various data associated with static and archive captures, see Managing Statically Captured Sites.

Notes and Tips for Managing Crawlers and Captured Data presents a collection of notes and tips to bear in mind when managing crawlers and captured data.

Enabling Publishing-Triggered Site Capture

An administrative user can configure as many publishing destination definitions for Site Capture as necessary and can call as many crawlers as necessary. The following are the main steps for enabling publishing-triggered site capture:

- About Integrating the Site Capture Application with Oracle WebCenter Sites
- Configuring a RealTime Publishing Destination Definition for Site Capture
- Matching Crawlers
- Managing Site Capture

About Integrating the Site Capture Application with Oracle WebCenter Sites

You can enable site capture after a RealTime publishing session only if the Site Capture application is first integrated with the source and target systems used in the publishing process. If Site Capture is not integrated, see Integrating Site Capture with the WebCenter Sites Publishing Process in *Installing and Configuring Oracle WebCenter Sites* for integration instructions, then continue with the steps below.

Configuring a RealTime Publishing Destination Definition for Site Capture

When configuring a publishing destination definition, you name the crawlers that are called after the publishing session. You also specify the capture mode.

1. Go to the WebCenter Sites source system that is integrated with the Site Capture application.
   a. Create a RealTime publishing destination definition pointing to the WebCenter Sites target system that is integrated with Site Capture. See Adding a New RealTime Destination Definition.
   b. In the More Arguments section of the publishing destination definition, name the crawlers to be called after the publishing session, and set the capture mode by using the following parameters:
      - `CRAWLERCONFIG`: Specify the name of each crawler. If you use multiple crawlers, separate their names with a semicolon (;).

        Examples:

        For a single crawler:`CRAWLERCONFIG=`crawler1

        For multiple crawlers:`CRAWLERCONFIG=`crawler1`;`crawler2`;`crawler3

        > **Note:**
        >
        > The crawlers 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 starts when the publishing session ends.
> - You can set both crawler parameters 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 run in that mode. To run some crawlers in a different mode, configure another publishing destination definition.

2. Continue to the next section.

Matching Crawlers

Crawlers named in the publishing destination definition must exist in the Site Capture interface. Do the following:

1. Verify that crawler names in the destination definition and Site Capture interface are identical. The names are case-sensitive.
2. Ensure that a valid starting URI for the target site is set in each crawler configuration file. For information about navigating to the crawler configuration file, see Editing the Crawler Configuration File. For more information about writing configuration code, see the Coding the Crawler Configuration File in *Developing with Oracle WebCenter Sites*.

Managing Site Capture

To manage site capture:

1. When you have enabled publishing-triggered site capture, you are ready to publish the target site. When publishing ends, site capture begins. The 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 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 session, nor does it make the captured site available to you.
     - To determine that the crawlers were called, 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 startup 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 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 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 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 About Managing Archived Sites.

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 configuration file, captures, and logs for the crawler as shown in Figure 34-4, while describes the `<crawlerName>` folder and its contents.

> **Note:**
>
> To access static captures and logs, you must 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 34-4 Site Capture Custom Folders: <crawlerName>

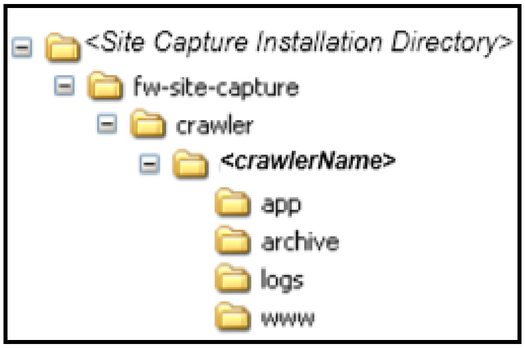


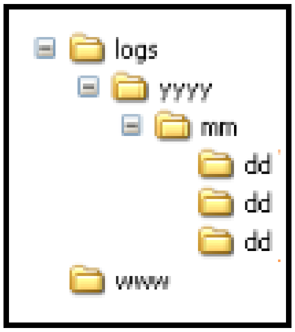
Table 34-2 <crawlerName> Folder and its Contents

Folder	Description
`/fw-site-capture/crawler/<crawlerName>`	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, both crawlers are listed in the Site Capture interface, and have identically named folders in the Site Capture file system. **Note:** In addition to the subfolders (described below), the `<crawlerName>` folder contains an `inventory.db` file, which lists statically crawled URLs. The file is created when the crawler takes its first static capture. Do not delete or modify `inventory.db`. It is used by the Site Capture system.
`/fw-site-capture/crawler/<crawlerName>/app`	Contains the `CrawlerConfiguration.groovy` file. Its code controls the crawl process. The `/app` folder is created when the crawler is created and saved.
`/fw-site-capture/crawler/<crawlerName>/archive`	The `/archive` folder is used strictly for archive capture. This folder contains a hierarchy of `yyyy/mm/dd` subfolders. The `/dd` subfolder stores all of the archive captures in time-stamped zip files. 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. **Note:** Archive captures are accessible from the Site Capture interface. Each zip file contains a URL log named `__inventory.db`. Do not delete or modify `__inventory.db`. It is used by the Site Capture system.
`/fw-site-capture/crawler/<crawlerName>/www`	Contains the latest *statically* captured site only (when the same crawler is rerun in static mode, it overwrites the previous capture). The site is stored as `html`, `css`, and other files that are readily served. The `/www` folder is created when the crawler first runs in static mode. **Note:** Static captures are accessible from the Site Capture file system only.

Table 34-2 (Cont.) <crawlerName> Folder and its Contents

Folder	Description
`/fw-site-capture/crawler/<crawlerName>/logs/yyyy/mm/dd`	Contains log files with information about crawled URLs. Log files are stored in the `/dd` subfolders and named as shown in Figure 34-5. • The `audit.log` file lists the crawled URLs with data such as timestamps, crawl depth, HTTP status, and download time. • The `links.txt` file lists the crawled URLs. • The `report.txt` 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 34-6). **Note:** If the crawler captured in both static mode and archive mode, the `/dd` subfolders contain logs for static captures and archive captures. The `/logs` folder is also used to store a transient file named `lock`. The file is created at the start of the static capture process to prevent the crawler from being called for additional static captures. The `lock` file is deleted when the crawl session ends.

Figure 34-5 Log Files Location



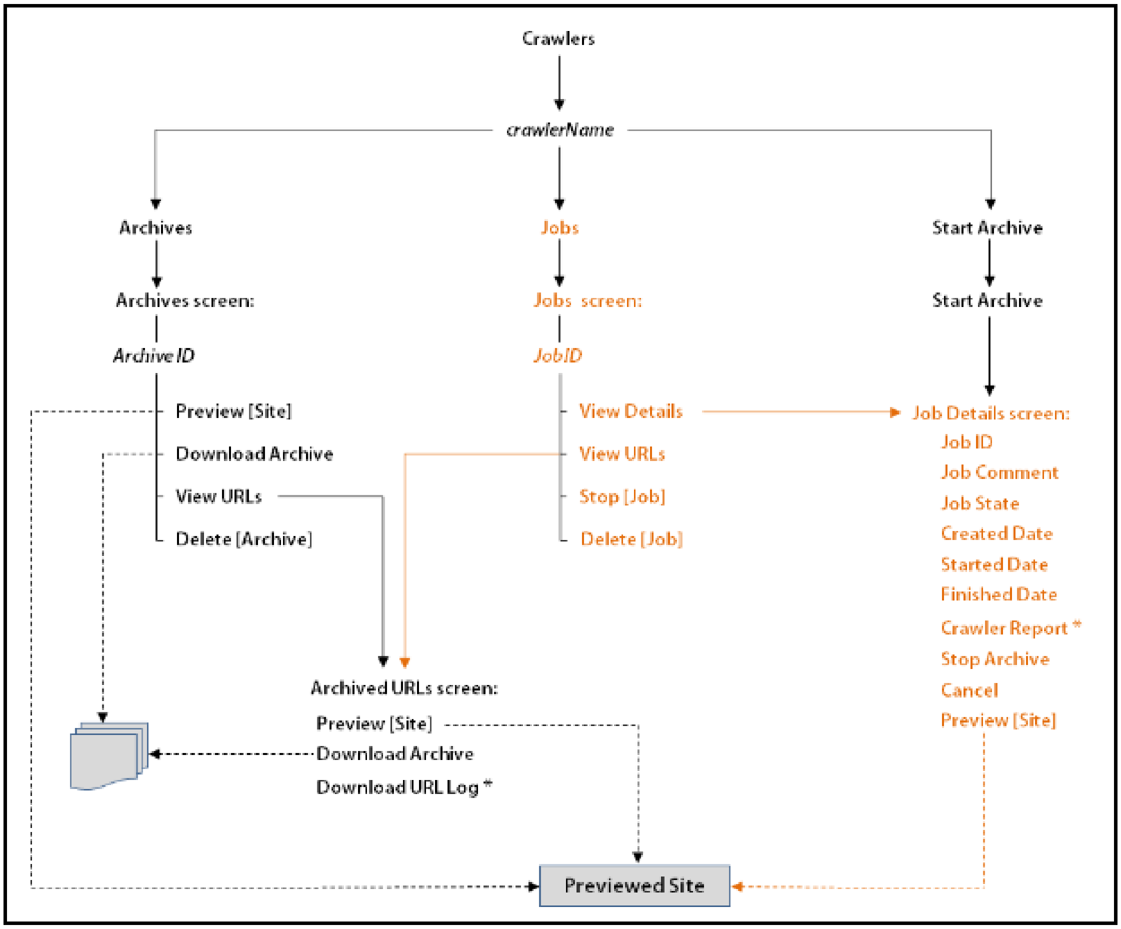
The folders under `logs/yyyy/mm/` contain the following logs:

- `<yyyy-mm-dd-hh-mm-ss>-audit.log`
- `<yyyy-mm-dd-hh-mm-ss>-links.txt`
- `<yyyy-mm-dd-hh-mm-ss>-report.txt`

About Managing Archived Sites

You can manage archived sites from different forms of the Site Capture interface. The following figure shows some pathways to various information: archives, jobs, site preview, crawler report, and URL log:

Figure 34-6 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 **Archives** path, **Jobs** path, or **Start Archive** path for the crawler. 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 (*).

Notes and Tips for Managing Crawlers and Captured Data

The following topics summarize notes and tips for managing crawlers and captured data:

- Tips for Creating and Editing Crawlers
- Notes for Deleting a Crawler
- Notes for Scheduling a Crawler
- About Monitoring a Static Crawl

- About Stopping a Crawl
- About Downloading Archives
- Notes About Previewing Sites
- Tips for Configuring Publishing Destination Definitions
- About Accessing Log Files

Tips for Creating and Editing Crawlers

When creating crawlers and editing their configuration code, consider the following information:

- Crawler names are case-sensitive.
- Every crawler configuration file is named `CrawlerConfigurator.groovy`. You must not change this name.
- You can configure a crawler to start at one or more seed URIs on a given site and to crawl one or more paths. You can use additional Java methods to set parameters such as crawl depth, call post-crawl commands, specify session timeout, and more. You can implement interfaces to define logic for extracting links, rewriting URLs, and sending email after a crawl session. See Coding the Crawler Configuration File in *Developing with Oracle WebCenter Sites*.
- 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, you cannot use it to run a second static capture process.
- While a crawler is running an archive capture process, you can use it to run a second archive capture process. The second process is marked as Scheduled and starts after the first process ends.

Notes for Deleting a Crawler

If you must delete a crawler (which includes all of its captured information), do so from the Site Capture interface, not the file system. Deleting a crawler 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 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.

Notes for Scheduling a Crawler

Only archive crawls can be scheduled.

- When setting a crawler schedule, consider the publishing schedule for the site and avoid overlapping the two.
- You can create multiple schedules for a single crawler – for example, one schedule to call the crawler periodically, and another schedule to call the crawler at one particular and unique time.
- When creating multiple schedules, ensure they do not overlap.

About Monitoring a Static Crawl

To determine whether a static crawler session is in progress or completed, look for the crawler 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 called for an additional static capture. The lock file is deleted when the crawler session ends.

About 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 session.

- If you must terminate an archive crawl, use the Site Capture interface. (Select **Stop Archive** on the Job Details form.)
- If you must terminate a static crawl, you must stop the application server.

About 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.

You can obtain archive size from the crawler report, on the Job Details form. Paths to the Job Details form are shown in Figure 34-6. See getPostExecutionCommand in *Developing with Oracle WebCenter Sites*.

Notes About 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 groovy file). Although the external domains can be browsed, they are not archived.

Tips for 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 one capture mode only. All crawlers run in that mode. To run some crawlers in a different mode, configure another publishing destination definition.

About Accessing Log Files

- For statically captured sites, log files are available in the Site Capture file system only:
  - 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 34-6).

- 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 34-6).

General Directory Structure

The Site Capture file system provides the framework in which Site Capture organizes custom crawlers and their captured content. The 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 following figure shows the folders most frequently accessed in Site Capture 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 the following table and Custom Folders.

Figure 34-7 Site Capture File System

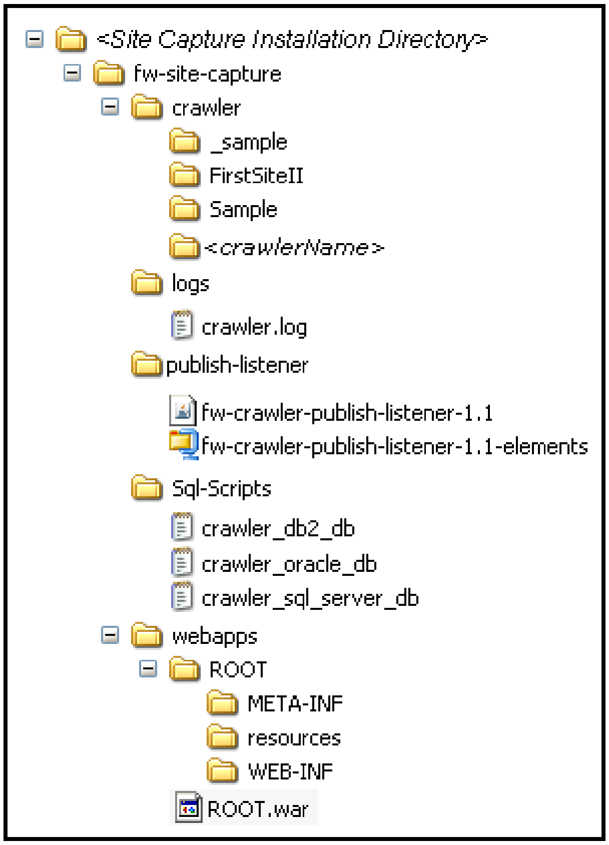


Table 34-3 Site Capture Frequently Accessed Folders

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.

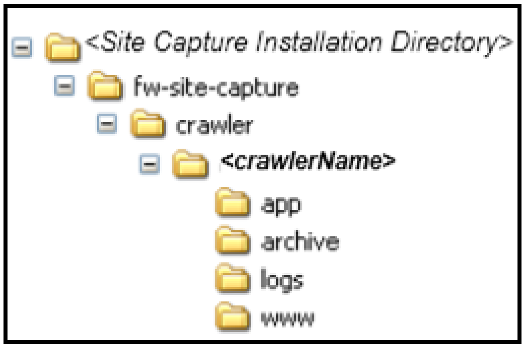
Table 34-3 (Cont.) Site Capture Frequently Accessed Folders

Folder	Description
/fw-site-capture/crawler/Sample	Represents a crawler named "Sample." This folder is created only if the "Sample" crawler is 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 called 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 required for installing Site Capture for publishing-triggered crawls: • fw-crawler-publish-listener-1.1-elements.zip • fw-crawler-publish-listener-1.1.jar
/fw-site-capture/Sql-Scripts	Contains the following scripts, which create database tables that are required by Site Capture to store its data: • 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: • sitecapture.properties file, where you can specify information for the WebCenter Sites application on which Site Capture is running. The information includes the WebCenter Sites host computer name (or IP address) and port number. • root-context.xml file, where you can configure the Site Capture database.

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 configuration file, captures, and logs for the crawler, as summarized in the following figure.

Figure 34-8 Site Capture Custom Folders: <crawlerName>



```
<Site Capture Installation Directory>
    fw-site-capture
        crawler
            <crawlerName>
                app
                archive
                logs
                www
```

35

Administering WebCenter Sites and Its Components in Enterprise Manager

You can monitor performance metrics, audit system performance, and view system messages for Oracle WebCenter Sites and Visitor Services with the Enterprise Manager.
Topics:

- Starting WebCenter Sites and Its Components in the WebLogic Server Console
- Starting WebCenter Sites and Its Components in Enterprise Manager
- Viewing the Home Pages of WebCenter Sites and Its Components
- Managing Log Files for WebCenter Sites and Its Components
- Monitoring the Performance Metrics of WebCenter Sites and Its Components

Starting WebCenter Sites and Its Components in the WebLogic Server Console

The Oracle WebLogic Server Administration Console is a browser-based user interface that you use to manage a WebLogic Server domain. The Administration Server hosts the Administration Console, which is a Web application accessible from any supported Web browser with network access to the Administration Server.

To navigate to component functions in WebLogic Server:

1. Enter the WebLogic Server URL, which includes the name of the host and the port number assigned to WebLogic Server during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/console
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.
2. Enter the system administrator user name and password and click **Login**.
3. From the navigation pane, expand the farm and then WebLogic Domain, and click **Sites** or the **Visitor Services** component.

   The domain page is displayed for your selection.

   In the WebLogic Server Administration Console, you can manage the environment including servers, clusters, virtual hosts, and work managers. See Getting Started Managing Oracle Fusion Middleware in *Administering Oracle Fusion Middleware*.

Starting WebCenter Sites and Its Components in Enterprise Manager

Oracle Enterprise Manager Fusion Middleware Control is a browser-based management application that is deployed when you install Oracle WebCenter. From Oracle Enterprise Manager, you can monitor and administer the following:

- Oracle WebCenter Sites
- Oracle WebCenter Sites: Visitor Services

To navigate to Oracle WebCenter Sites Enterprise Management pages in Oracle Enterprise Manager:

1. Enter the Oracle Enterprise Manager Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Oracle Enterprise Manager during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/em
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.

2. Enter the system administrator user name and password, and click **Login**.
3. From the **Target Navigation** pane, expand the farm, then **WebCenter**, and then **Sites**.
4. Choose **Sites** or **Visitor Services** , and then select the server node (for example, Visitor Services).

   The home page for the component you started is displayed.

5. Click **WebCenter Sites: *<Component Name>*** in the left pane to display the ***<Component>*** menu.

   The following table lists some of the ***<Component>*** server menu options on this page.

Table 35-1 WebCenter Sites: <Component> Menu Options

Menu Option	Description
Home	Use this option to navigate to the WebCenter Sites or Visitor Services home page.
Monitoring	Use this option to monitor key performance metrics for the WebCenter Sites or Visitor Services server.
Control	Use this option to start or stop the WebCenter Sites or Visitor Services application deployed on the server.
Logs	Use this option to view the logs or to control the log level.
System MBean Browser	The System MBean Browser lists all the configured, runtime and application defined MBeans. Use this option to access the attributes of the MBeans and to perform operations on them.
Target Sitemap	Use this option to view menu options for the WebCenter Sites or Visitor Services server.

Table 35-1 (Cont.) WebCenter Sites: <Component> Menu Options

Menu Option	Description
Target Information	Use this option to view general information about the WebCenter Sites or Visitor Services server.

Viewing the Home Pages of WebCenter Sites and Its Components

You can view the status of Oracle WebCenter Sites or Visitor Services server using Oracle Enterprise Manager or Oracle WebLogic Server. The home pages for Oracle WebCenter Sites and Visitor Services in Oracle Enterprise Manager display the status of your system. The home pages display the state, version, and uptime of the Oracle WebCenter Sites and Visitor Services systems.

To view the status of your Oracle WebCenter Sites or Visitor Services system installation:

1. Enter the Oracle Enterprise Manager Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Oracle Enterprise Manager during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/em
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.
2. Enter the system administrator user name and password, and click **Login**.

   The component home page displays.
3. On the General Information section on the home page, check the status of Oracle WebCenter Sites or its component systems (state, version, and uptime).
4. On the Performance Summary section on the lower pane, view the component Performance metrics. See Monitoring the Performance Metrics of WebCenter Sites and Its Components for more information about Performance metrics.

You can navigate to the home page from any other page by clicking **Home** in the component menu options in the right pane.

> **Note:**
>
> For more information about viewing the status of your Oracle WebCenter Sites or WebCenter Sites: Visitor Services system using the Oracle WebLogic Server administration console, see the Oracle WebLogic Server documentation.

Managing Log Files for WebCenter Sites and Its Components

You can enable logging of configuration messages, debug messages, and trace messages and configure the log files using Oracle Enterprise Manager.

This section includes the following topics:

- Configuring Log File Settings for WebCenter Sites and Its Components
- Viewing Log Files for WebCenter Sites and Its Components

Configuring Log File Settings for WebCenter Sites and Its Components

By default, errors, warnings, and major application life cycle events in Enterprise Manager are logged on the application server. As a system administrator, you can progressively increase the log level to enable logging of configuration messages, debug messages, and trace messages. You can use Oracle Enterprise Manager to view and configure log messages for your WebCenter Sites system. For more information about managing log files, see Managing Log Files and Diagnostic Data in *Administering Oracle Fusion Middleware*.

The `{weblogic_server_name}` is the name of the WebLogic Server, for example `VisitorServices01 or Sites01`.

Out of the box, the log files are named `sites.log` and `cas.log`, and are created in the application server log directory.

You can configure the log files for WebCenter Sites using Oracle Enterprise Manager.

You can configure the names and locations of log files, the size of the log files, the level of information written to the log files, the format, and the Locale encoding, as described in Configuring Settings for Log Files in *Administering Oracle Fusion Middleware.*

Viewing Log Files for WebCenter Sites and Its Components

You can view log files for Oracle WebCenter Sites or Visitor Services from the Oracle Enterprise Manager.

To view the log files using the Oracle Enterprise Manager:

1. Enter the Oracle Enterprise Manager Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Oracle Enterprise Manager during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/em
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.
2. Enter the system administrator user name and password, and click **Login**.

3. Click the component server in the left pane to display the component menu options.
4. To view the log files, choose **Logs**, and then **View Log Messages** from the component menu. The Log Messages page displays.
5. In this page, you can search for messages by specified date range, type, or containing certain keywords. You can also select a message from the list and view its details.

Monitoring the Performance Metrics of WebCenter Sites and Its Components

For information about monitoring the Oracle WebCenter Sites or Visitor Services system, including an overview of monitoring and viewing the status of your system, see these topics:

- Viewing the Status of WebCenter Sites and Its Components
- Viewing the Performance Metrics of WebCenter Sites and Its Components

> **Note:**
>
> You can set the `CURSOR_SHARING` initialization parameter to `FORCE` to enhance the performance of WebCenter Sites. If you set the `CURSOR_SHARING` initialization parameter to `FORCE`, the database replaces literals with system-generated bind variables.

Viewing the Status of WebCenter Sites and Its Components

You can view the status of Oracle WebCenter Sites or Visitor Services using Oracle Enterprise Manager or Oracle WebLogic Server. The component home page in Oracle Enterprise Manager displays the status of your component system. It displays the state, version, and uptime of the component system.

To view the status of your component system installation:

1. Enter the Oracle Enterprise Manager Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Oracle Enterprise Manager during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/em
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.
2. Enter the system administrator user name and password, and click **Login**.
3. On the ***<Component>*** - General Information section on the home page, check the status of your component system (state, version, and uptime).

You can navigate to the home page from any other page by clicking **Home** in the component menu options in the right pane.

> **Note:**
>
> For more information about viewing the status of your component system using the Oracle WebLogic Server administration console, see the Oracle WebLogic Server documentation.

Viewing the Performance Metrics of WebCenter Sites and Its Components

To view the performance metrics of your Oracle WebCenter Sites or Visitor Services system installation:

1. Enter the Oracle Enterprise Manager Fusion Middleware Control URL, which includes the name of the host and the port number assigned to Oracle Enterprise Manager during the installation. The following shows the format of the URL.

   ```
   http://hostname.domain:port/em
   ```

   Port is the IP port number of the Administration Server running on Oracle WebLogic Server. By default, the port number is 7001.
2. Enter the system administrator user name and password, and click **Login**.
3. Click component server in the right pane to display the component menu options.
4. To view performance metrics, choose **Monitoring**, and then **Performance Summary** from the component menu. The Performance Summary page displays. By default, the Performance Summary page displays the performance metrics for the past 15 minutes.
5. Optionally, click **Slider** in the top right side of the page to view the time slider.

   Use the time slider to select the time period for which to display the metrics.
6. Optionally, click the **Enter Time** icon next to **Slider** in the top right side of the page.

   The Enter Time dialog box displays.
7. Enter the start and end time for which to display the metrics, then click **OK**.
8. Click **Show Metrics Palette** in the right pane. The Metric Palette displays.
9. In the Metric Palette, select the metrics that you want to display in the Performance Summary page. To view metrics in a table format, select **Table View**.

For more information about Oracle WebCenter Sites or Visitor Services metrics, see these sections:

- WebCenter Sites Performance Metrics
- Visitor Services Performance Metrics

WebCenter Sites Performance Metrics

This section describes metrics to monitor the performance of WebCenter Sites. It includes the following:

- BlobServer Servlet Details, ContentServer Servlet Details, and Satellite Servlet Details
- Cache Details
- JDBC Info
- ThreadPool Info
- OS Detail
- Number of Errors
- NIO Details
- Memory Details Metrics
- Connections Metrics
- ServerState Metrics

BlobServer Servlet Details, ContentServer Servlet Details, and Satellite Servlet Details

Metric Name	Description
Execution Time Average	Displays the average amount of time (in milliseconds) it took to execute all invocations of the servlet since it was most recently deployed.
Execution Time High	Displays the average amount of time (in milliseconds) the single longest invocation of the servlet has executed since it was most recently deployed.
Execution Time Total	Displays the average amount of time (in milliseconds) all invocations of the servlet have executed since it was most recently deployed.
Invocation Total Count	Displays the total number of times the servlet has been invoked since WebLogic Server started.

Cache Details

Metric Name	Description
CacheHits	Provides the number of times a requested item was found in the cache.
CacheHitsPercentage	Provides the percentage of cache accesses that found a requested item in the cache.
CacheMisses	Provides the number of cache accesses that did not find a requested element in the cache.
CacheMissPercentage	Provides the percentage of cache accesses that did not find a requested element in the cache.
MemoryStoreObjectCount	Provides the number of objects in the MemoryStore.
DiskStoreObjectCount	Provides the number of objects in the DiskStore.

JDBC Info

Metric Name	Description
ActiveConnectionsAverageCount	Average number of active connections in this instance of the data source. Active connections are connections in use by an application.
ActiveConnectionsCurrentCount	The number of connections currently in use by applications.
ActiveConnectionsHighCount	Highest number of active database connections in this instance of the data source since the data source was instantiated. Active connections are connections in use by an application.
ConnectionDelayTime	The average amount of time, in milliseconds, that it takes to create a physical connection to the database. The value is calculated as summary of all times to connect divided by the total number of connections.
ConnectionsTotalCount	The cumulative total number of database connections created in this data source since the data source was deployed.
MaxCapacity	Maximum number of physical database connections that this connection pool can contain. Different JDBC Drivers and database servers may limit the number of possible physical connections.
NumAvailable	The number of database connections that are currently idle and available to be used by applications in this instance of the data source.
PrepStmtCacheAccessCount	The cumulative, running count of the number of times that the statement cache was accessed.
PrepStmtCacheAddCount	The cumulative, running count of the number of statements added to the statement cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements added to the caches for all connections in the connection pool.
PrepStmtCacheCurrentSize	The number of prepared and callable statements currently cached in the statement cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements in the caches for all connections in the connection pool.
PrepStmtCacheDeleteCount	The cumulative, running count of statements discarded from the cache. Each connection in the connection pool has its own cache of statements. This number is the sum of the number of statements that were discarded from the caches for all connections in the connection pool.
PrepStmtCacheHitCount	The cumulative, running count of the number of times that statements from the cache were used.
PrepStmtCacheMissCount	The number of times that a statement request could not be satisfied with a statement from the cache.
ReserveRequestCount	The cumulative, running count of requests for a connection from this data source.

ThreadPool Info

This section describes the metrics related to the thread pool:

Metric Name	Description
completedRequestCount	The total number of requests handled by WebCenter Sites or its component application.
executeThreadIdleCount	Number of executing threads that are currently idle. It also indicates the additional unused capacity.
executeThreadTotalCount	Total number of execute threads that are current active in the system.
pendingUserRequestCount	Number of requests in the queue waiting to be processed. Ideally the number of requests in the queue should be a low.
queueLength	The number of pending requests in the priority queue. This is the total of internal system requests and user requests.
stuckThreadCount	Number of threads that are stuck. Weblogic considers a thread as stuck if no work is done in over 600 seconds. Stuck threads are an indication that there is trouble in the system.
throughput	The system throughput.

OS Detail

This section describes the metrics related to the operating system:

Metric Name	Description
commitedVirtualMemory Size	The size of the virtual memory in megabytes.
freePhysicalMemorySize	The size of the unused physical memory in megabytes.
freeSwapSpaceSize	The size of memory available for swap space in megabytes.
maxFileDescriptorCount	The maximum number of file descriptors in the system.
openFileDescriptorCount	The number of open file descriptors in the system.
processCpuLoad	The CPU time utilized by the Java process hosting WebCenter Sites or its components.
systemCpuLoad	The total CPU utilization across all processes in the system.
systemLoadAverage	The number of tasks waiting to be processed, or stalled waiting on a CPU to become available. This is a system wide metric and should be below three in a normal system.
totalPhysicalMemorySize	The physical memory size in bytes.
totalSwapSpaceSize	The swap memory space in bytes.

Number of Errors

This section describes the metrics related to Number of errors:

Metric Name	Description
Number of errors	The number of exceptions caught by the application during runtime.

NIO Details

This section describes the metrics related to the NIO details:

Metric Name	Description
count	Non-blocking I/O (NIO) object counts.
Memory Used	The memory used in the system in megabytes.
Total Capacity	The total memory in the system in megabytes.

Connections Metrics

This section describes the metrics related to the connections:

Metric Name	Description
Active Threads	The number of active threads in the system.
Daemon Threads	The number of daemon threads in the system. Daemon threads are threads that run in the background.
Peak Threads Count	The maximum number of threads in the system.

Memory Details Metrics

This section describes the metrics related to memory details:

Metric Name	Description
Available Processors	The number of available processors in the system.
Free Memory(bytes)	The free memory in the system displayed in bytes.
Total Memory(bytes)	The total memory in the system in bytes.
Maximum Memory(bytes)	The maximum memory in the system in bytes.

ServerState Metrics

The following table lists the Visitor Services ServerState metrics:

Metric Name	Description
State	The status of the selected component server. To select a server, click **List Metric Rows** icon and select the component server.

Visitor Services Performance Metrics

This section describes performance metrics for monitoring Visitor Services. It includes the following:

- ThreadPool Info
- OS Detail
- Number of Errors
- NIO Details
- Memory Details Metrics
- Connections Metrics

- ServerState Metrics

ThreadPool Info

This section describes the metrics related to the thread pool:

Metric Name	Description
completedRequestCount	The total number of requests handled by WebCenter Sites or its component application.
executeThreadIdleCount	Number of executing threads that are currently idle. It also indicates the additional unused capacity.
executeThreadTotalCount	Total number of execute threads that are current active in the system.
pendingUserRequestCount	Number of requests in the queue waiting to be processed. Ideally the number of requests in the queue should be a low.
queueLength	The number of pending requests in the priority queue. This is the total of internal system requests and user requests.
stuckThreadCount	Number of threads that are stuck. Weblogic considers a thread as stuck if no work is done in over 600 seconds. Stuck threads are an indication that there is trouble in the system.
throughput	The system throughput.

OS Detail

This section describes the metrics related to the operating system:

Metric Name	Description
commitedVirtualMemory Size	The size of the virtual memory in megabytes.
freePhysicalMemorySize	The size of the unused physical memory in megabytes.
freeSwapSpaceSize	The size of memory available for swap space in megabytes.
maxFileDescriptorCount	The maximum number of file descriptors in the system.
openFileDescriptorCount	The number of open file descriptors in the system.
processCpuLoad	The CPU time utilized by the Java process hosting WebCenter Sites or its components.
systemCpuLoad	The total CPU utilization across all processes in the system.
systemLoadAverage	The number of tasks waiting to be processed, or stalled waiting on a CPU to become available. This is a system wide metric and should be below three in a normal system.
totalPhysicalMemorySize	The physical memory size in bytes.
totalSwapSpaceSize	The swap memory space in bytes.

Number of Errors

This section describes the metrics related to Number of errors:

Metric Name	Description
Number of errors	The number of exceptions caught by the application during runtime.

NIO Details

This section describes the metrics related to the NIO details:

Metric Name	Description
count	Non-blocking I/O (NIO) object counts.
Memory Used	The memory used in the system in megabytes.
Total Capacity	The total memory in the system in megabytes.

Connections Metrics

This section describes the metrics related to the connections:

Metric Name	Description
Active Threads	The number of active threads in the system.
Daemon Threads	The number of daemon threads in the system. Daemon threads are threads that run in the background.
Peak Threads Count	The maximum number of threads in the system.

Memory Details Metrics

This section describes the metrics related to memory details:

Metric Name	Description
Available Processors	The number of available processors in the system.
Free Memory(bytes)	The free memory in the system displayed in bytes.
Total Memory(bytes)	The total memory in the system in bytes.
Maximum Memory(bytes)	The maximum memory in the system in bytes.

ServerState Metrics

The following table lists the Visitor Services ServerState metrics:

Metric Name	Description
State	The status of the selected component server. To select a server, click **List Metric Rows** icon and select the component server.

Part XIV

Integrating with Oracle WebCenter Content

You can configure and integrate WebCenter Content (WCC to synchronize content between Oracle WebCenter Content and Oracle WebCenter Sites.

- Understanding WebCenter Content Integration
- Setting Up Content Synchronization
- Working With WCC Connector Configuration Files

36

Understanding WebCenter Content Integration

To integrate WebCenter Content with WebCenter Sites, become familiar with the prerequisites of WCC integration. Also learn how you can configure WCC and manage synchronized content.
This chapter also describes enabling two WCC Integration parts: the SitesIntegration component and the **Connector Admin** tab.

The Oracle WebCenter Content Integration feature, also called WCC Integration, is a GUI-controlled connector. It synchronizes 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 is created directly in WebCenter Sites.

Before you can start configuring your system for content synchronization, you must enable the SitesIntegration component and the **Connector Admin** tab. Each step is described in its own section.

The following topics provide information about WebCenter Content integration:

- Prerequisites for Administering WCC Integration
- Basics of WebCenter Content Integration
- WebCenter Content Data Model
- Asset Type Mapping and Connector Rules
- WebCenter Content Integration and Its Parts
- Enabling the SitesIntegration Component
- Enabling Access to the **Connector Admin** Node

Prerequisites for Administering WCC Integration

WCC Integration is for administrators with a working knowledge of WebCenter Content and WebCenter Sites, especially their data models. You must have a clear understanding of content items, renditions, conversions, and metadata defined in WebCenter Content. You must also have a clear understanding 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 you have to create the asset types, you must collaborate with developers.

For more information about WebCenter Content, see Getting Started with Oracle WebCenter Content in *Using Oracle WebCenter Content*.

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. You can create renditions with 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. For more information about content transformation, see Viewing Alternate Renditions in *Using Oracle WebCenter Content*.

Before you enable and set up WCC Integration, review the concepts that are fundamental to 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 configure to guide the connector when it polls WebCenter Content. The concepts are discussed in WebCenter Content Data Model and Asset Type Mapping and Connector Rules.

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:

  - You can convert native documents with the core transformation engine (available by default in WebCenter Content) into web formats for easier viewing on the web, such as 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.
  - Digital Asset Manager can convert native image files and videos 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.
  To allow the Dynamic Converter and the Sites Integration components to operate correctly together, set the default character set in the JVM to `UTF-8`. This allows the Sites Integration component to correctly interpret the HTML generated by Dynamic Converter. Apply the JVM argument `-Dfile.encoding=UTF-8` to the JVM that hosts the WebCenter Content.
- 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).

You can trigger import and synchronization manually, or you can schedule them 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. You can connect any number of WebCenter Sites clients to a single WebCenter Content instance.

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 covers the following topics:

- Asset Type Mapping
- Connector Rules

Asset Type Mapping

You can create and reuse target asset types. 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.
  - Content storage attributes to hold the imported files (native files, renditions, conversions, or any combination of them).
  - A template storage attribute to enable asset previewing, assuming asset previewing is a requirement (which is typically the case).

If you create a template attribute, you must give special attention to template code for previewing dynamically converted HTML files and their referenced images.

- The rest of the requirements are specific to your requirements.

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 **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 `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 been synchronized to WebCenter Sites and, therefore, to either create an 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. This is done only one time, on a new system, before connector rules are created and the first synchronization session is started.

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.

Template Attributes

Typically, asset types in 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:

```
<img src="${wcmUrl('dcresource','20673.jpg')}"
width="360" height="270" alt="2007_06240117.JPG" />
```

Because does not recognize a format such as `${wcmUrl('dcresource','20673.jpg')}`, you must convert the links 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 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 the following table:

Table 36-1 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 must update the sample `WccConversionPreview.jsp` template with the attribute names used in the target asset type.

Administrator-Defined Specifications

You always have the option to add attributes for any metadata that you must synchronize to a target asset type.

You can enable target asset types on as many content management sites as necessary. During a synchronization session, content items are synchronized to the target asset types on all sites that you specify in the connector rules.

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 use on which content management sites 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 specifies a single target asset type only, 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 want to import a dynamically converted HTML file and its referenced images into a single asset type. You 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 want 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 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 rule is 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 36-2 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 36-2 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	`surfing.jpg` and `surfing.png` are imported into this asset type	`skiing.png` is imported into this asset type
If rule 2 is executed first	`surfing.jpg` is imported into this asset type	`surfing.png` and `skiing.png` are imported into this asset type

The connector uses rules to synchronize content items to WebCenter Sites as follows:

- When the connector runs, it examines new and updated content items, but only if they are (or 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. After its search is complete, the connector turns to WebCenter Sites (regardless of whether it finds matching renditions).
- On WebCenter Sites, the connector operates on the database:

The connector calls 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 attributes for the asset.

After 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

After a content item is synchronized to WebCenter Sites, it is 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 WebCenter Content Integration and Its Parts.

WebCenter Content Integration and Its Parts

The WCC Integration feature consists of several parts, which you must 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. You must enable it to support the WCC connector.

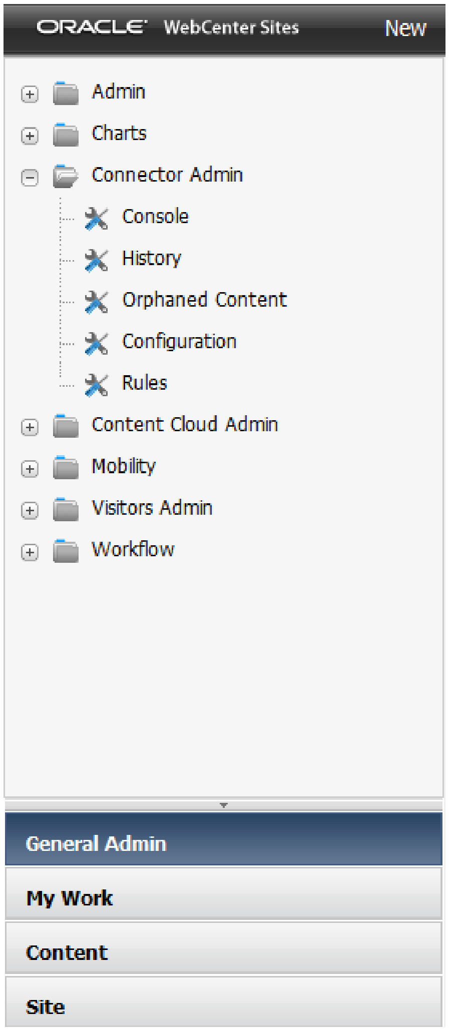
  Enabling the SitesIntegration component adds the following features to WebCenter Content:

  - **Sync to Sites** option. This option opens on content check-in, 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 rules you create.
  - WCS Queue Support Information. This graphical interface lists token history for each of the WebCenter Sites clients. You can use token information to roll back to previous updates.

  See Enabling the SitesIntegration Component.

- **WCC Connector**: This connector is installed on WebCenter Sites to communicate with WebCenter Content. You can install it during the WebCenter Sites installation or upgrade process, or at a later time (the former method is preferred). The connector polls WebCenter Content after it is configured to do so from the **Connector Admin** tab in the WebCenter Sites Admin interface.
- **Connector Admin**: This is a graphical interface used to control the connector. When WCC Integration is installed, the **Connector Admin** node is added to the navigation tree of the WebCenter Sites Admin interface, as shown in the following figure.

Figure 36-1 WCC Connector Admin Node



Use the **Connector Admin** node 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.

See Enabling Access to the **Connector Admin** Node.

Enabling the SitesIntegration Component

The SitesIntegration component is installed during the full Oracle WebCenter Content installation or upgrade process. After 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 check-in, update, and search forms.
- The WCS Queue Support Information page is added to provide a listing of WebCenter Sites clients and their token histories. After synchronization begins, the same page logs the synchronized content items and their update histories.

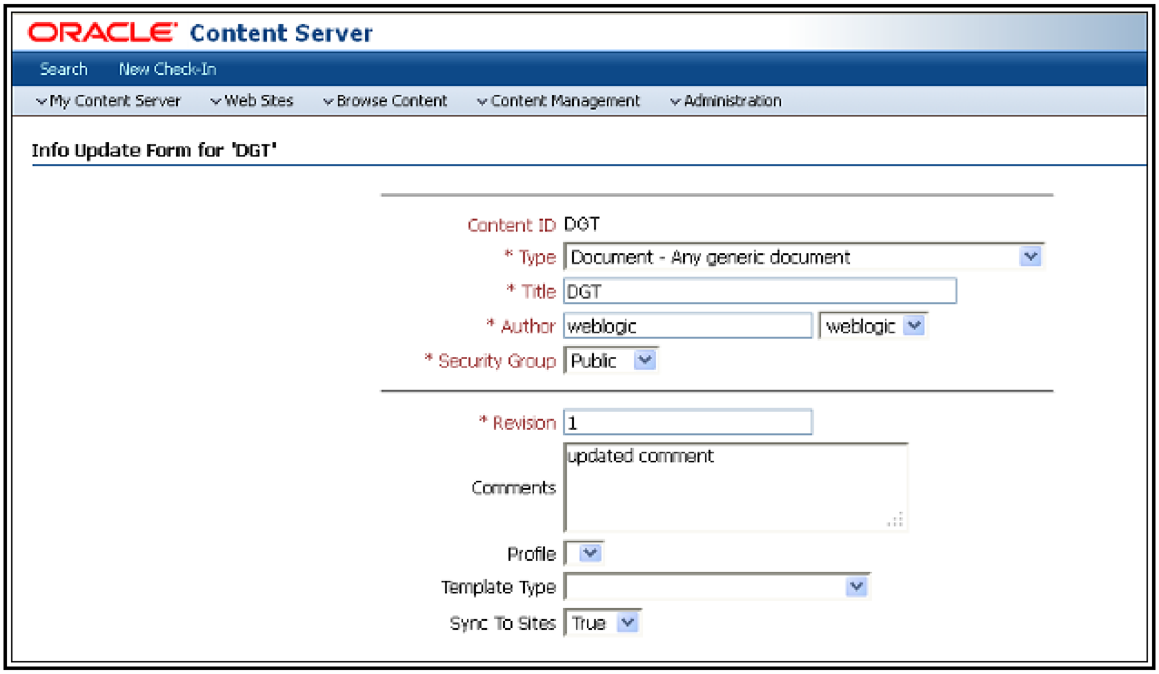
To enable the SitesIntegration component

1. On WebCenter Content, open the Component Manager page:
   1. Click **Administration** on the portal navigation bar for your Content instance.
   2. Click **Admin Server**.
2. On the Component Manager page:
   1. Scroll down to the Integration section and select the **SitesIntegration** check box.
   2. Click **Update**.

> **Note:**
>
> For more information about enabling components, see Configuring the WebCenter Content Domain in *Installing and Configuring Oracle WebCenter Content* .

3. Restart WebCenter Content.
4. Open any content check-in, update, or search form to verify that the **Sync To Sites** check box opens in the form, as shown in the following figure.

Figure 36-2 Sync To Sites Option on a Content Item Update Form

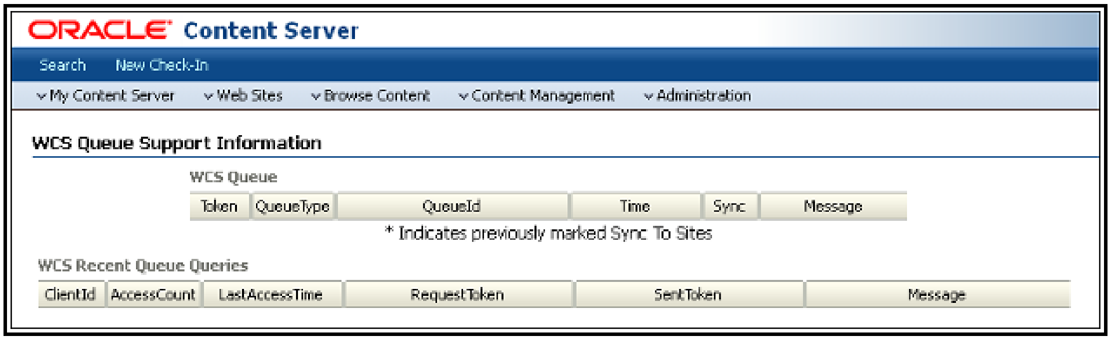


Info Update Form for 'DGT'

Content ID DGT
* Type Document - Any generic document
* Title DGT
* Author weblogic weblogic
* Security Group Public
* Revision 1
Comments updated comment
Profile
Template Type
Sync To Sites True

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 (in the following figure).

Figure 36-3 WCS Queue Support Information Page on WebCenter Content Server



WCS Queue Support Information

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 Enabling Access to the **Connector Admin** Node.

Enabling Access to the **Connector Admin** Node

You must assign the ConnectorAdmin role to anyone who must use WCC Integration. The only exception is noted below.

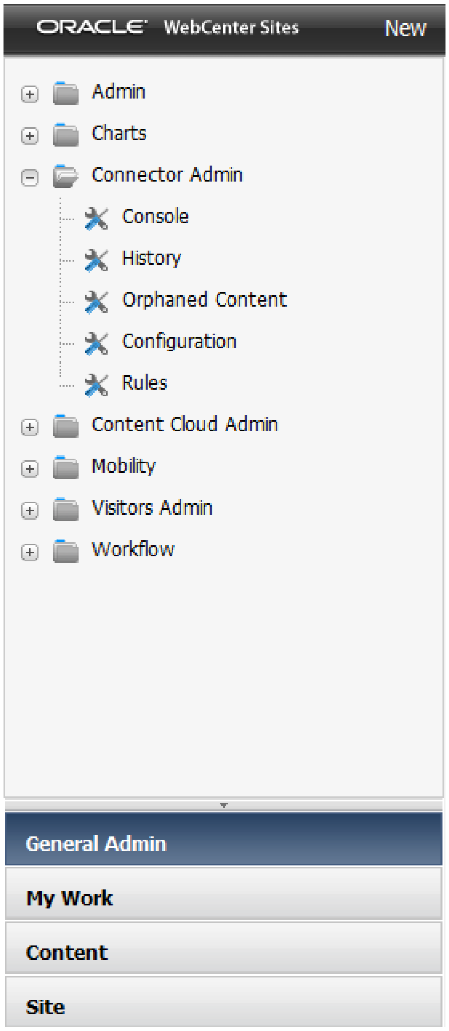
> **Note:**
>
> The `ConnectorAdmin` role is automatically assigned to one user only, and that is the general administrator whose credentials were used during the WebCenter Sites installation process. All other users, including general administrators, must have the role assigned to them manually.

The `ConnectorAdmin` role grants access to the **Connector Admin** node, 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 node:

1. Open the Admin interface on the content management site you intend to work with, and look for the **Connector Admin** node (shown in the following figure).

Figure 36-4 Connector Admin Tab



2. If the **Connector Admin** node is not displayed, assign yourself (and other intended users) the ConnectorAdmin role. See 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.

37 Setting Up Content Synchronization

You need to configure your system to import content from WebCenter Content into WebCenter Sites and synchronize it.
Each of your steps is described in its own section. Additionally, this chapter describes rollback procedures and information about the synchronization token.

> **Note:**
>
> Before starting the steps in this chapter:
>
> - Read Understanding WebCenter Content Integration for information about the WebCenter Content data model and the asset types and connector rules you must configure.
> - Ensure that all procedures in Enabling the SitesIntegration Component and Enabling Access to the **Connector Admin** Node have been completed.
> - Read this chapter to gain an understanding of the steps you must follow. 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.

Topics:

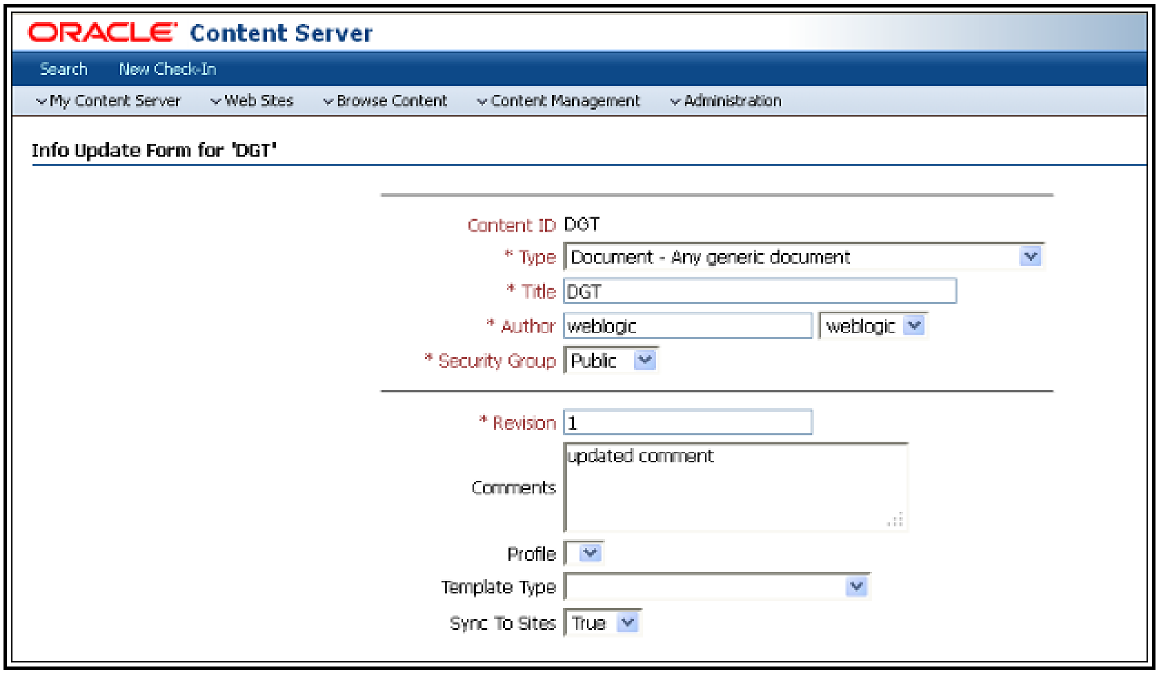
- Enabling Content Items for Synchronization
- Determining Which Renditions to Synchronize
- Selecting or Creating Target Asset Types
- Connecting to WebCenter Content
- Creating an Active Rule
- Prioritizing Active Rules and Managing Rules
- Manually Running the Connector
- Monitoring the Connector
- Viewing Connector History
- Verifying Imported Content
- Modifying Your Setup
- Scheduling the Connector
- Managing Synchronized Assets
- Rolling Back the Synchronization Point

- About the Generated Token Value

Enabling Content Items for Synchronization

Content items are typically enabled for synchronization by WebCenter Content users. They set the **Sync to Sites** option for each item to `True`, as shown in the following figure. Valid content items are documents and digital assets.

Figure 37-1   Sync To Sites Option on the WebCenter Content Server Update Form



Info Update Form for 'DGT'

Content ID DGT
* Type Document - Any generic document
* Title DGT
* Author weblogic weblogic
* Security Group Public
* Revision 1
Comments updated comment
Profile
Template Type
Sync To Sites True

Synchronization-enabled content items are available to the WCC connector for processing. However, you can import them 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.

Determining Which Renditions to Synchronize

Content items enabled for synchronization in Enabling Content Items for Synchronization have renditions. Default rendition types are `primary` and `web`:

- A `primary` rendition is the native (original) file that is checked in to WebCenter Content (for example, a Microsoft Word document, before any renditions are 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 Connecting to WebCenter Content).

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 Asset Type Mapping . For example, they must belong to the flex family.
  - Your asset types are configured to accept the content items that are enabled in Enabling Content Items for Synchronization.
- 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 `wcs_properties.json` file. After the parameter is reset and synchronization has started, do not edit this value again.

  For information about `dDocName`, see Content Identifier Attribute. For information about the `wcc.keyfield` parameter, see 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.

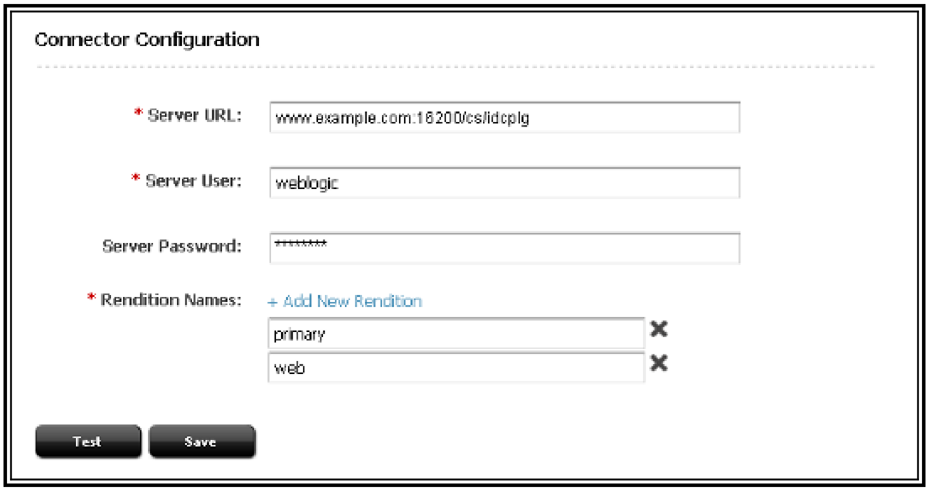
Connecting to WebCenter Content

In this section, you will connect WebCenter Sites to WebCenter Content and specify which renditions (determined in 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** node, select **Configuration** to open the Connector Configuration form.

Figure 37-2 Connector Configuration Form with Example Values



Connector Configuration

* Server URL:	www.example.com:16200/cs/idcplg
* Server User:	weblogic
Server Password:	*********
* Rendition Names:	primary web

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 about determining the URL path of a WebCenter Content instance, see Accessing Oracle WebCenter Content in *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, then you must define the user name as that of a local user with administrative privileges (by default, **sysadmin**).

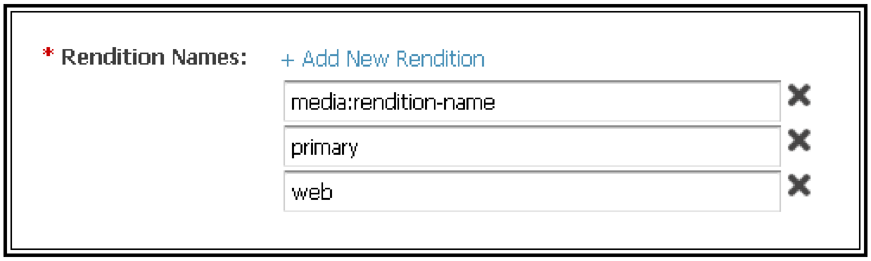
   c. In the **Server Password** field, provide a password for the administrative user only if you specified an HTTP or HTTPS connection in the **Server URL** field.

   > **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.

   To add a rendition, click **Add New Rendition** and enter its name. If you add a video rendition, use a colon-separated format such as `media:`*`rendition-name`*. For example, `media:flash8`.

Figure 37-3 Example Video Rendition



media:rendition-name
primary
web

   The rendition types `primary` and `web` are provided by default. You can keep or delete them, depending on your requirements.

3. Click **Test** to test your entries. If your entries are valid, a confirmation message opens.

4. Click **Save** to save the entries. A confirmation message opens when the configuration has been saved successfully.

Creating an Active Rule

An active rule is a set of instructions enabled for the connector to follow when it runs. The connector uses active rules to do the following:

1. Sample the content items that are enabled in 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 Selecting or Creating Target Asset Types and named in your active rules.
3. Synchronize the renditions that you specified in Connecting to WebCenter Content and in your active rules.

> **Note:**
>
> Before creating rules, read Connector Rules for planning and configuration guidelines.
>
> You can enable rules when you create them (in this step), or in a later step (in Prioritizing Active Rules and Managing Rules).

To Create a Rule:

1. From the **Connector Admin** node, select **Rules**.
2. In the Connector Rules form, click **Add New Rule**.
3. The rules wizard opens. 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.

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. You can enable or disable the rule at any time from the current wizard or the Connector Rules form.

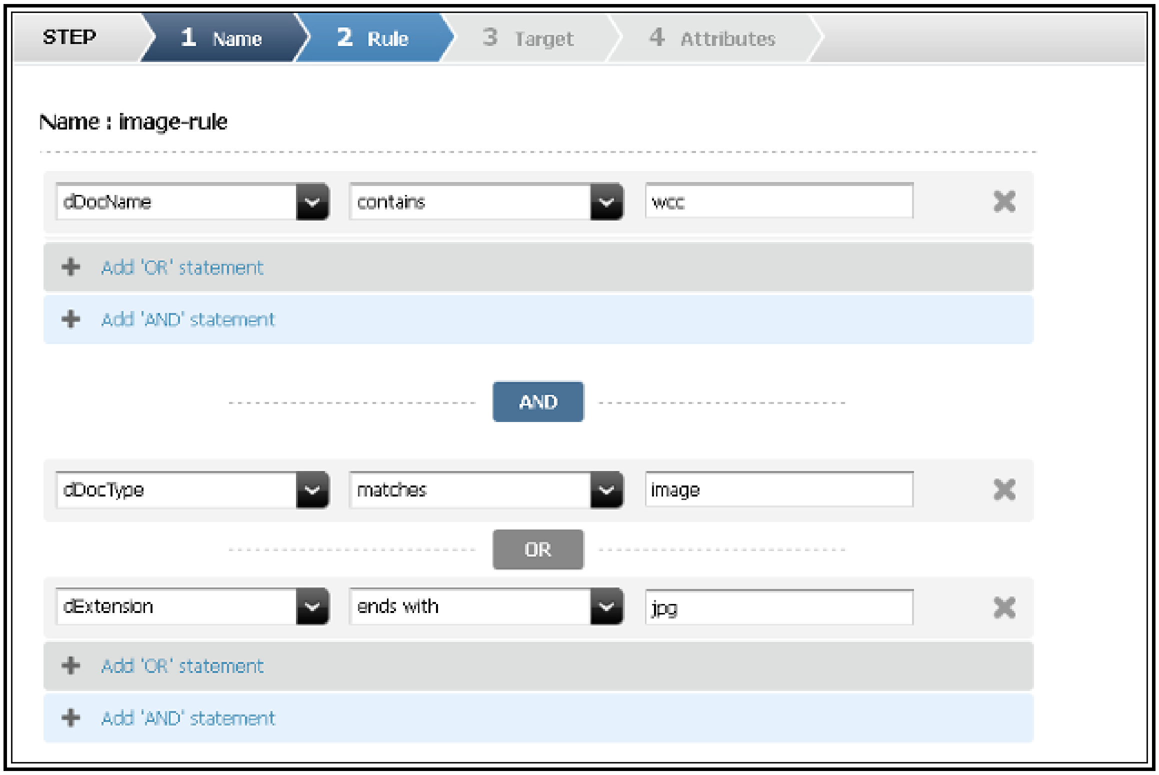
4. Continue to Rules Wizard: Step 2 Rule.

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.

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 multiple metadata names and values by adding `AND`/`OR` statements.

Figure 37-4 Step 2 Rule Page with Example Values



Name : image-rule

dDocName | contains | wcc

dDocType | matches | image

dExtension | ends with | jpg

To specify which metadata the connector must sample:

1. In the left-hand list, select a metadata name.

   The list displays all metadata names defined in WebCenter Content. For descriptions of the metadata, see Metadata Fields in the *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 list, select an operator:

   - **matches**: The connector searches 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 recognizes `image`, but not `images`.

- **contains**: The connector performs 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 recognizes `abc`, `abcde`, and any other `dDocName` that contains `abc`.
- **starts with**: The connector searches for metadata values that begin with the string you provide in the right-hand field.
- **ends with**: The connector searches 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 wanted.
5. Add as many `AND` statements as wanted.
6. Continue to Rules Wizard: Step 3 Target.

Rules Wizard: Step 3 Target

Use this step to select the target asset type, content management sites where the asset type is enabled, and any other information for which you are prompted.

> **Note:**
>
> The content of this step depends on the asset type.

To specify the target asset type and the content management site where it is enabled:

1. Select the target asset type from the list.

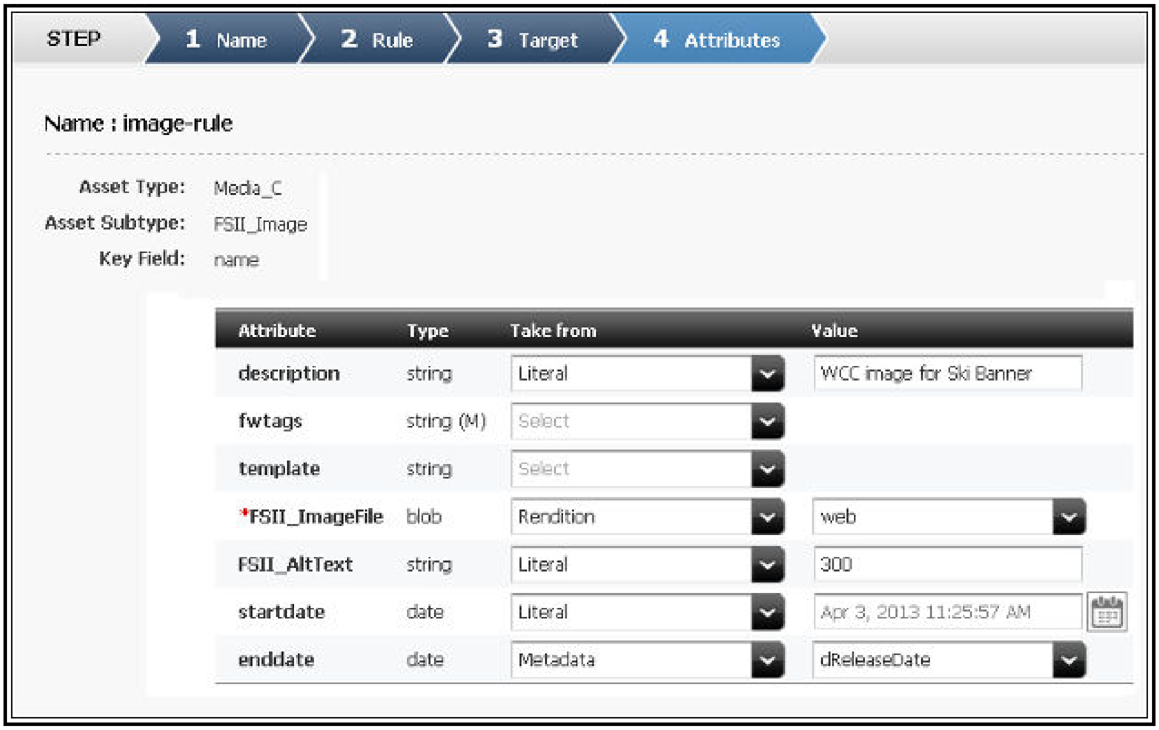
   Depending on the asset type you select, different options are displayed on the form.
2. Select the asset subtype.
3. Enable the parents from the list.
4. The Sites list names the content management sites on which your selected asset type is enabled. Select one or more sites. The connector imports content items and metadata into the target asset type on all of your selected sites.
5. Provide any other information for which you are prompted.
6. Continue to Rules Wizard: Step 4 Attributes.

Rules Wizard: Step 4 Attributes

Use this step to map the metadata of the synchronization-enabled content items to the attributes of your target asset type.

The Step 4. Attributes page of the Rules Wizard lists the attribute names and their data types in the asset type you selected in step 3 Target. 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 37-5 Step 4 Attributes Page with Example Values



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 list. Options depend on the attribute type. The options are:

- **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 displays 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 is 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 Selecting or Creating Target Asset Types).
    - For attributes of type `date` (for example, attributes named `startdate` or `enddate`), select **Literal** to pick a date using the date-picker.
  - If you select **Metadata** from the list in the **Take From** column, select the metadata name from the **Value** column.

> **Note:**
>
> When you select **Metadata**, the list in the **Value** column lists metadata fields whose data type matches the data type of the attribute you want to target. For example, if your attribute is of type `date`, the **Value** column lists metadata fields of type `date` only. For detailed descriptions of metadata types, see Meta Services (Core Content Server) in *Services Reference for Oracle WebCenter Content*.

- **Rendition or Conversion**: These options apply to attributes of type `blob` only.
  - If you select **Rendition** from the list in the **Take From** column, select the rendition type from the **Value** column (for example, `web`, `primary`, or other rendition type).

    > **Note:**
    >
    > When you select **Rendition**, the list in the **Value** column names only the rendition types that you specified in the Connector Configuration form, in Connecting to WebCenter Content.

  - If you select **Conversion** from the list in the **Take From** column, select the 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 37-1 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. You can enable or disable it at any time from the Connector Rules form.

2. Create additional rules, if necessary.

Prioritizing Active Rules and Managing Rules

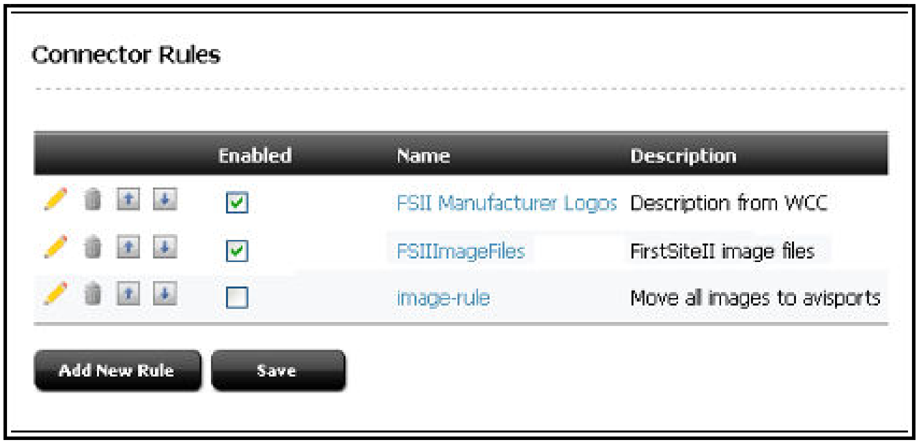
An important aspect of managing rules is to enable them for the connector and prioritize them in order of execution. The order can determine how the connector distributes content items among the target asset types in a synchronization session. For more information about prioritization, see Connector Rules.

You can complete other operations, such as editing, deleting, and adding rules in the same form you use to enable and prioritize rules.

To prioritize and otherwise manage rules:

1. From the **Connector Admin** node, select **Rules** to open the Connector Rules form.

Figure 37-6 Connector Rules Form with Example Rules



Connector Rules

Enabled	Name	Description
☑	FSII Manufacturer Logos	Description from WCC
☑	FSIIImageFiles	FirstSiteII image files
□	image-rule	Move all images to avisports

2. 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 want 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 is synchronized according to that one rule. All other rules for synchronizing that content item are ignored.
   c. Click **Save**.
3. 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 opens. For information about configuring rules, see 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 is synchronized according to that one rule. All other rules for processing that content item are ignored.

- Select or deselect 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 are 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 opens.
- The **Description** column provides the rule description. This description is defined when you create a rule.
- Click **Add New Rule** to add a new rule. The rules wizard opens.
- Click **Save** to save any changes, such as enabling or disabling a rule.

4. After you have optimized, enabled, and prioritized your active rules, you should start the process of manually running the connector.

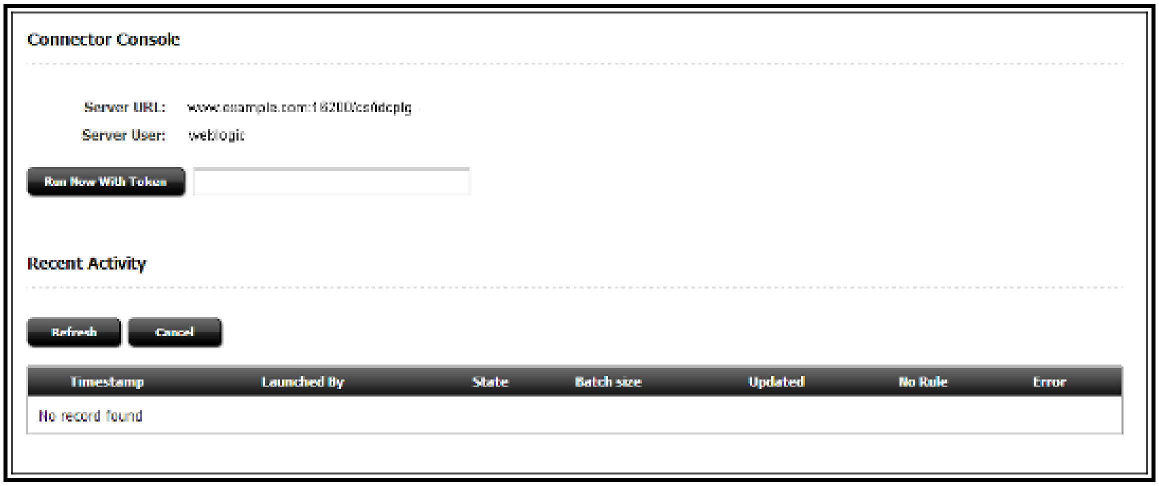
Manually Running the Connector

In this step, you will manually run the connector to test the rules you configured and prioritized in Creating an Active Rule and Prioritizing Active Rules and Managing Rules.

To run the connector:

1. From the **Connector Admin** node, select **Console** to open the Connector Console.

Figure 37-7 Connector Console with Example Server Information



Connector Console

Server URL: www.example.com:16200/cs/idcplg

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 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 Rolling Back the Synchronization Point.

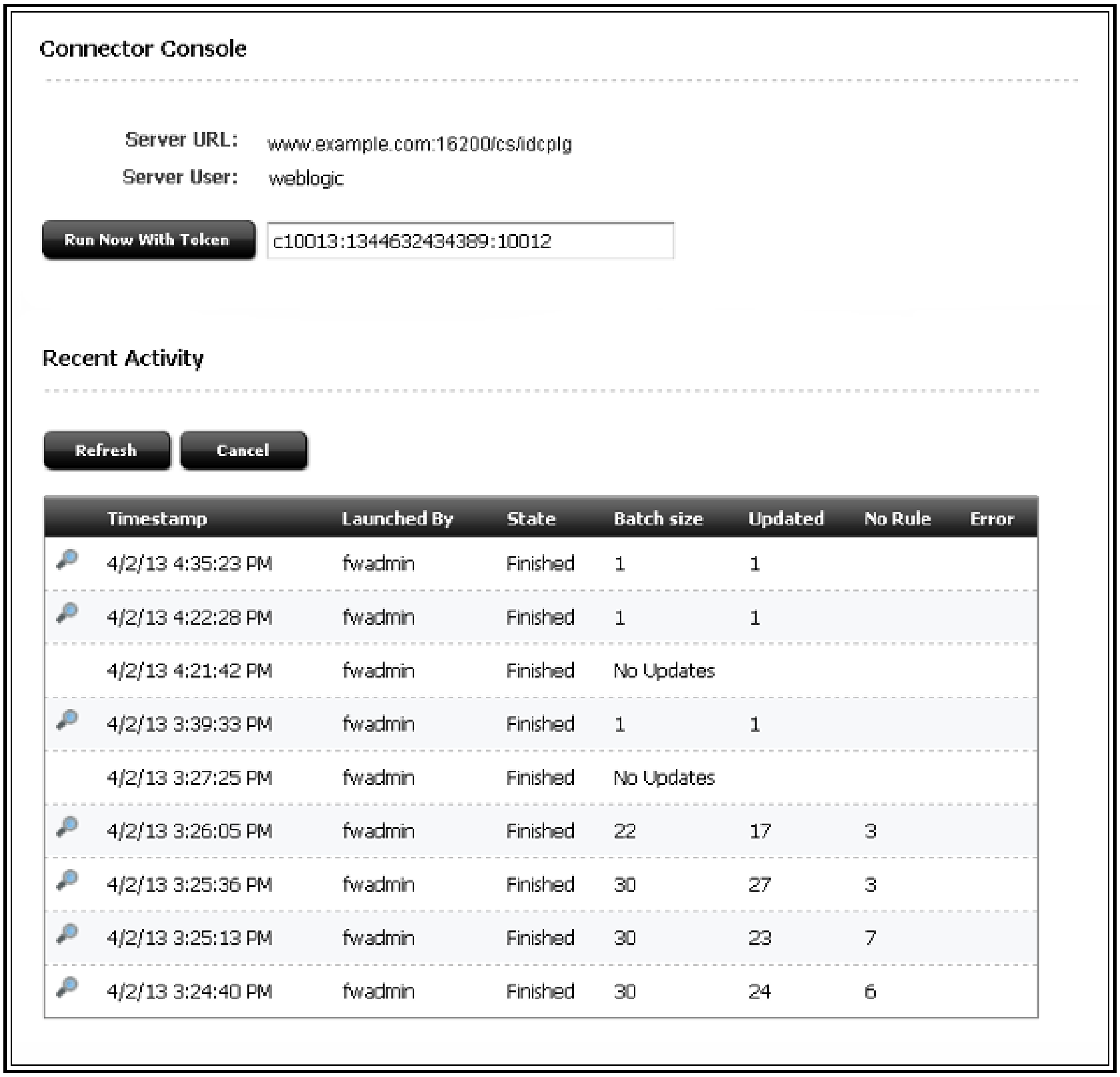
   > **Note:**
   >
   > You can schedule the connector to run on a timed basis, as described in Scheduling the Connector.

4. While the connector runs, you can monitor its activity from the same form you are now using.

Monitoring the Connector

When the connector runs, the Connector Console (populated in the previous step) displays the Recent Activity table, shown in the following figure.

Figure 37-8 Connector Console Form with Filled Recent Activity Table



Connector Console

Server URL: www.example.com:16200/cs/idcplg

Server User: weblogic

c10013:1344632434389:10012

Recent Activity

	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. You must manually refresh the table 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 User Configurable Parameters.

1. 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:

   - Click **Refresh** to update the table.
   - Click **Cancel** to cancel the currently running connector. Note that processing of the next batch (if any) is cancelled. The current batch is processed to completion.
   - Click the **Inspect this item** icon (magnifying glass) to view connector history for the selected batch. See 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. See 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 Working With WCC Connector Configuration Files.
- The **Updated** column lists the number of items that are updated in WebCenter Sites during batch processing. The number includes assets that are 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 history (for all or selected batches). Continue to Viewing Connector History.

   If you prefer to inspect the synchronized assets to ensure they are stored in the intended asset types, skip to Verifying Imported Content.

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 covers the following topics:

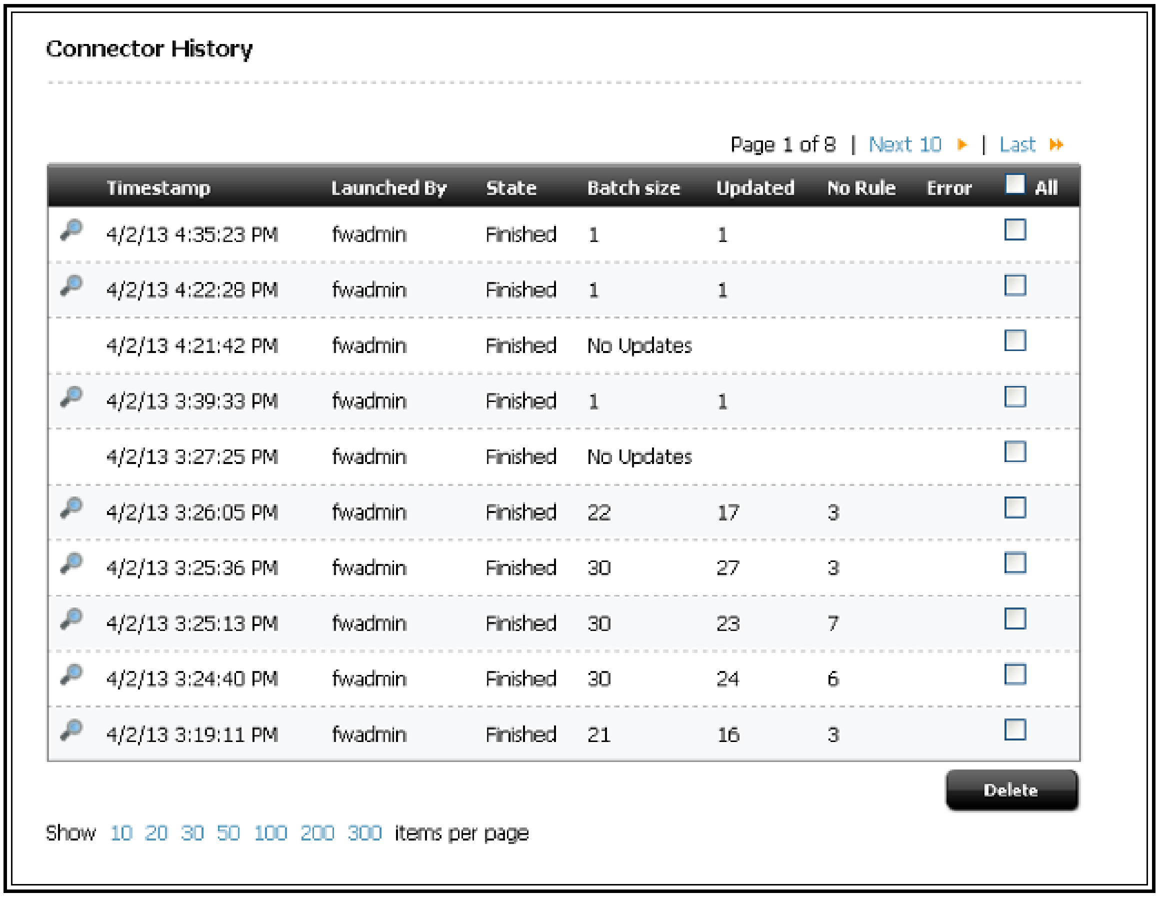
- Viewing the Entire Connector History
- Viewing Connector History for a Single Batch

Viewing the Entire Connector History

To view connector history for all batches:

1. From the **Connector Admin** node, select **History** to open the Connector History table.

Figure 37-9 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			□
🔍	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		□
🔍	4/2/13 3:19:11 PM	fwadmin	Finished	21	16	3		□

Show 10 20 30 50 100 200 300 items per page

2. 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. See 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. See 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 Working With WCC Connector Configuration Files.
   - The **Updated** column lists the number of items that are updated in WebCenter Sites during batch processing. The number includes assets that are 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 **Delete** to delete all activity listings.
- Select a batch (a row) and click **Delete** 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 the figure in Viewing Connector History for a Single Batch.

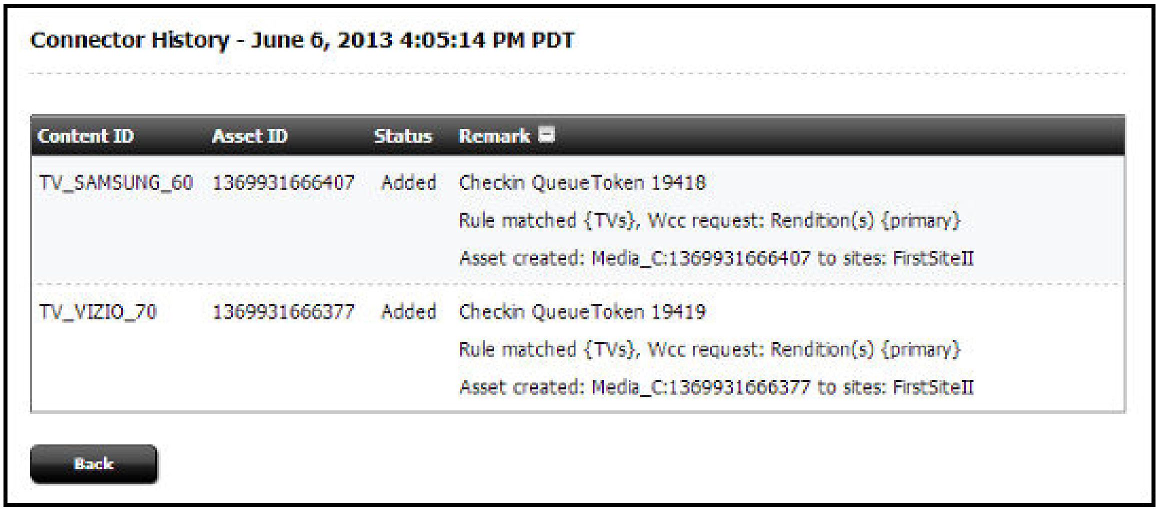
Viewing Connector History for a Single Batch

To view connector history for a single batch:

1. From the **Connector Admin** node, select **History** to open the Connector History table.
2. Locate the batch (a row), and click its **Inspect** icon (magnifying glass).

   The Connector History table lists all items that were processed in the selected batch.
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.
   - Click the **Back** button to go back to the previous table (which displays the history for all batches).
4. The following figure shows an example of possible messages in the **Remark** column. A more comprehensive list is available below the figure.

Figure 37-10   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

Some 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 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 Managing Synchronized Assets).

- `Checkin QueueToken {0}`: Value `{0}` is the token ID. This message indicates a queue token check-in 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 available renditions for the item on WebCenter Content.
- `Requested Renditions not available`: Indicates that the requested rendition is not available. See `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.

Verifying Imported Content

At this point, you are likely to have enough information about how well content synchronization meets your requirements. To confirm this, inspect 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 are specified in the attribute mapping part of the rules are met. To locate the content items and the assets, see the Connector History table for Content IDs and their counterpart Asset IDs.

After verification is complete, do either:

- If adjustments are necessary, continue to Modifying Your Setup.
- If the results meet your requirements, continue to the following sections:
  – Scheduling the Connector
  – Managing Synchronized Assets

Modifying Your Setup

If the results of your synchronization sessions do not meet your requirements, you have several ways to adjust system configuration. Before taking action, it is best to review your options:

- You can return to some 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 `wcs_properties.json` file to reconfigure various parameters, such as maximum batch size. See Working With 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 are 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. After the assets are deleted, you can reset the token. See Rolling Back the Synchronization Point.

After you have successfully modified your setup, continue to the following sections:

- Scheduling the Connector
- Managing Synchronized Assets

Scheduling the Connector

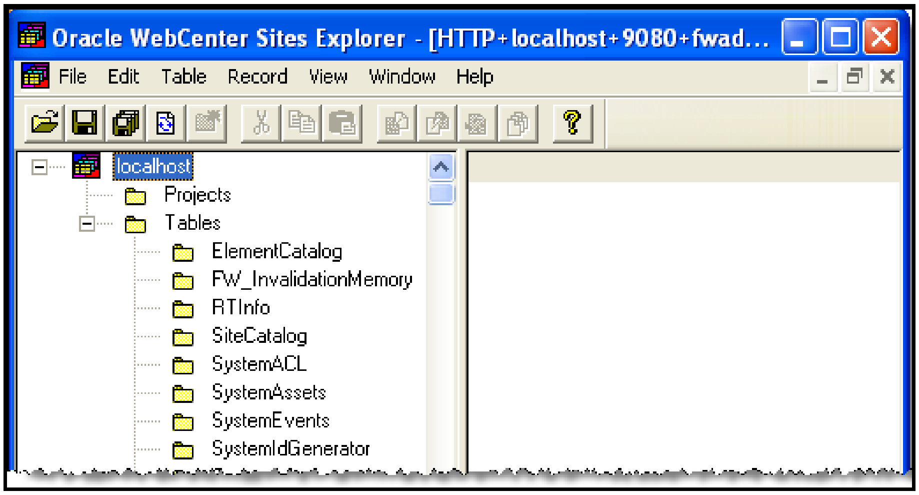
After you have configured and verified content synchronization, you can schedule the connector to run on a timed basis. Use the Explorer utility to add a timed entry to the `SystemEvents` table.

> **Note:**
>
> Oracle WebCenter Sites Explorer is a Windows-only utility.

To schedule the connector:

1. Launch the Oracle WebCenter Sites Explorer executable file (`ContentServerExplorer.exe`) located in the `SitesExplorer` directory of the distribution file.
2. Log in to Oracle WebCenter Sites Explorer. (See Oracle WebCenter Sites Explorer in *Developing with Oracle WebCenter Sites.)*
3. In Oracle WebCenter Sites Explorer, do the following:
   a. Expand the **Tables** node.

Figure 37-11 Explorer with Expanded Tables Node



   b. Select the **SystemEvents** table (see the preceding figure).
   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 the following table. Below the table is an example of a time pattern.

Table 37-2 Fields of a Time Pattern in the SystemEvents Table

Field	Legal Values	All Legal Values	Single Value Format	Inclusive Range Format	List Format
`h` (hours)	0–23	`*`	`h`	`h-h`	`h,h`
`m` (minutes)	0–59	`*`	`m`	`m-m`	`m,m`
`s` (seconds)	0–59	`*`	`s`	`s-s`	`s,s`
`W` (day of the week)	0–6, where 0=Sunday	`*`	`W`	`W-W`	`W,W`
`D` (day of the month)	0–31	`*`	`D`	`D-D`	`D,D`
`M` (month)	1–12	`*`	`M`	`M-M`	`M,M`

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 is triggered in all months.

Specify days with one or both of 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 one field only, set the other field to `*`.

4. Click **Save**.
5. To help content contributors identify and manage synchronized assets start the process of managing synchronized assets.

Managing Synchronized Assets

After 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 run on a regular basis. Administrators and content contributors who publish also have to know how the assets are affected by synchronization sessions.

The following topics provide information about synchronized assets:

- Mapped Attributes
- Orphaned Content

Mapped Attributes

If content contributors modify assets by changing the values of attributes that are explicitly mapped in the attributes mapping part of the connector rules, those values are updated when the content items are updated and resynchronized. All other attributes preserve their values.

> **Note:**
>
> If the value of a mapped attribute is changed, it is overwritten when its content item is updated and resynchronized. Keeping changes to mapped attributes requires copying the assets.

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 you cannot delete when its counterpart content item is either deleted, expired, or disabled from synchronization (by having its **Sync To Sites** option set to `False`). You cannot delete the orphaned asset because dependencies were created on that asset since the last synchronization session.

> **Note:**
>
> If a synchronized asset is shared across content management sites, it is deleted, unless it has dependents.

To determine orphaned assets:

1. From the Admin interface, in the **General Admin** tree, open the **Connector Admin** node.
2. Select **Orphaned Content** to open the Connector Orphaned Content table.

   The Connector Orphaned Content table shows 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 `update_number` for the token, from which you can derive the date and time when the content items were processed by the connector. You can obtain date and time from the WCS Queue Support Information table, described in Rolling Back the Synchronization Point.
   - 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.

Rolling Back the Synchronization Point

WebCenter Sites and WebCenter Content are synchronized through a generated token, which is rarely reset after the content synchronization process is established.

> **Caution:**
>
> Reset tokens during system setup only, when synchronization-related objects (such as rules) are configured, tested, and re-configured. After your system is configured for actual, uninterrupted operation, do not modify token values.

During system setup, however, you may have re-configured rules or asset types and therefore must 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 include content items whose **Sync to Sites** option is (or was) enabled. If updates were made, WebCenter Content generates a new `update_number` and returns the updated token to WebCenter Sites.

You can roll back a token value to its original value (the original synchronization point), or to a particular update, as described next.

> **Caution:**
>
> Do not reset the token for an established content synchronization process.

- 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 re-examines and resynchronizes every content item whose **Sync to Sites** option is enabled or was enabled at one time.

> **Note:**
>
> The connector re-examines previously enabled content items (which are no longer marked as **Sync to Sites**) in case their counterpart assets must be deleted from WebCenter Sites.

- Replacing the `update_number` for the token with a previously generated `update_number` rolls back the point of synchronization to the previous update. The next time the connector runs, it polls content item versions at the new synchronization point. The connector re-examines and resynchronizes every item whose **Sync to Sites** option is enabled or was enabled at one time. The connector re-examines previously enabled content items (which are no longer marked as **Sync to Sites**) in case their counterpart assets must 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 more information about token value, see About the Generated Token Value.

This section includes the following topics:

- Rolling Back to the Original Starting Point
- Rolling Back to a Particular Update

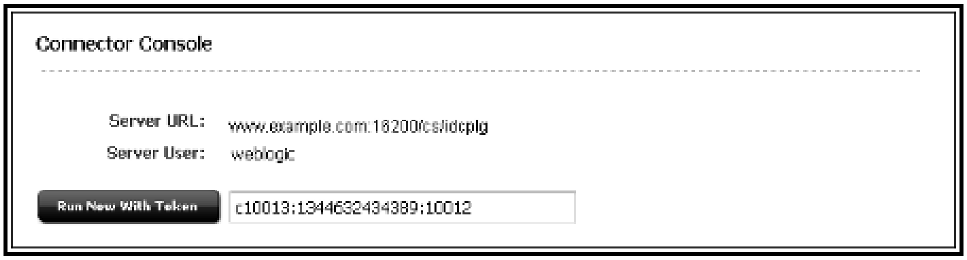
Rolling Back to the Original Starting Point

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. In the Admin interface, select the **General Admin** tree, and then click the **Connector Admin** node.
2. Select **Console**.
3. In the Connector Console form, delete the token value (next to the **Run with Token** button).

Figure 37-12 Connector Console Form



Connector Console

Server URL: www.example.com:16200/cs/idcplg
Server User: weblogic

c10013:1344632434389:10012

Rolling Back to a Particular Update

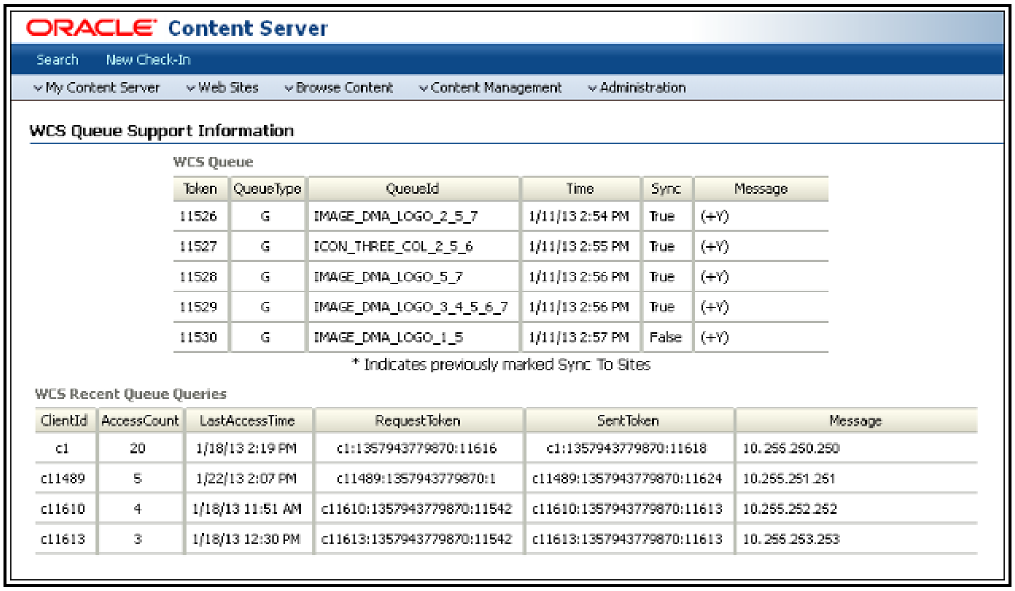
Resetting the `update_number` for the token to a previously generated value rolls back the current point of synchronization to that particular update.

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 37-13 WCS Queue Support Information Table in WebCenter Content Server



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 `update_number` for the item.
2. Reset the token value in :
   a. From the Admin interface, open the **Connector Admin** tab.
   b. Select **Console** to open the Connector Console form and replace the `update_number` (last set of digits) for the token, with the new `update_number`.

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 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 gets 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 Content ID and `update_number` for the item are logged in the WCS Queue table. That information in 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 always remains 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.

38

Working With WCC Connector Configuration Files

During the WCC Integration process, you need to use some configuration files. The wcs_properties.json contains user configurable parameters that you can edit to optimize conditions for the WCC connector and to troubleshoot.

- About WCC Connector Configuration Files
- User Configurable Parameters

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 require editing is the wcs_properties.json file, which is used to optimize conditions for the WCC connector and to troubleshoot.

> **Note:**
>
> The following configuration files must not be edited. They are used internally for data storage:
>
> - wcc-matchers.ini: Contains the rule statements that are created using 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.

User Configurable Parameters

The wcs_properties.json configuration file contains several system parameters that are user configurable. These parameters are not configurable in the WebCenter Sites Admin interface.

The wcs_properties.json 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:

```
#wcs_properties.json
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 you process a large amount of content in batches, this parameter specifies the maximum size of the batch.

  When the wcs_properties.json configuration file is created, the default batch size is set to 50. You can change this value, but you cannot leave the parameter blank and you must provide 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 you can set the connector to either automatically run again after the batch is processed (true), or the connector can wait for the next manual or next timed run (false).

  When the wcs_properties.json configuration file is created, this parameter is set to a default value of true. You can change this value, but you cannot leave the parameter blank and you must provide 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 either forces the temporary files to be cleaned up (true), or leaves files on disk for analysis (false).

  When the wcs_properties.json configuration file is created, this parameter is set to a default value of true. You can change this value, but you cannot leave the parameter blank and you must provide a valid value. Valid values are true or false.

- wcc.keyfield=[name]

  This parameter specifies the name of the asset type attribute that receives 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 uncommon to change this value.

  You can change this one time, on a new system, before rules are created, and before content synchronization begins. After synchronization starts, you should not edit this value. Otherwise, assets can be duplicated. An example situation where the default value could be changed is when the name field is in use. For more information about dDocName, see 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 `wcs_properties.json` configuration file is created, this parameter is set to a default value of `true`. Valid values are `true` or `false`.

Part XV
Reference

Reference information is available on dynamic tables of database, system defaults, and site management using an LDAP server.

Topics:

- System Defaults
- WebCenter Sites Database Tables
- Users, Sites, and Roles in LDAP-Integrated Sites Systems
- Alternate Publishing Methods

39
System Defaults

If you see the system defaults data, you can get more information about the more routinely used default data, such as ACLs, roles, asset types, and tree tabs.
Topics:

- ACLs
- System ACLs and Their Permissions
- ACLs of Default Users
- Required ACLs for Custom Users
- System Roles
- System Asset Types
- Default Tree Nodes

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 the default system ACLs.

This section covers the following topics:

- Permissions
- Accessing ACLs

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.

The following table lists all the permissions that can be specified in an ACL.

Table 39-1 Permissions Supported by WebCenter Sites

Permissions	Bit Mask (see note 1)	Action
Read	1	Read data from a table.
Write	2	Write information to a table. (see note 2)
Create	4	Create a table.
Delete	8	Delete information from a table.

Table 39-1 (Cont.) Permissions Supported by WebCenter Sites

Permissions	Bit Mask (see note 1)	Action
Retrieve	16	Retrieve the contents of a URL column, also known as an upload field. For more information about URL columns, see Indirect Data Storage with the WebCenter Sites URL Field in the *Developing with Oracle WebCenter Sites*.
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.

Note:

- 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.
- To add a row to a table, the user must have Create permission.

Accessing ACLs

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 **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, and then double-click **ACLs**.
2. In the drop-down menu, 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.

See System ACLs and Their Permissions.

System ACLs and Their Permissions

The following table lists the system ACLs and their permissions. Each system ACL exists 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 39-2 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
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

The following table describes the functions of each ACL and how each ACL is used by Oracle WebCenter Sites and the WebCenter Sites content applications.

Table 39-3 System ACLs and Their Descriptions

ACL Name	Description
Browser	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. 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. 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.
ContentEditor	Used in a sample WebCenter Sites site. This ACL is assigned to the tables that support the sample site.

Table 39-3 (Cont.) System ACLs and Their Descriptions

ACL Name	Description
ElementEditor	Allows users to write data to the ElementCatalog and SystemSQL tables. Site designers and anyone who creates templates, CSElement, and SiteEntry assets require this ACL.
ElementReader	Allows users to read data in the ElementCatalog and SystemSQL tables. WebCenter Sites users require this ACL so they can inspect the templates assigned to their assets.
PageEditor	Allows users to create page entries in the SiteCatalog table. Site designers and anyone who creates a template, CSElement, or SiteEntry asset requires this ACL.
PageReader	Allows users to read page entries from the SiteCatalog table. WebCenter Sites users require this ACL so they can inspect the templates assigned to their assets.
RemoteClient	Grants users the ability to log in to the WebCenter Sites management system through a remote client.
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 asset types (which causes tables to be created) require this ACL. Administrators or anyone else who will use the Initialize Mirror Destination feature also requires this ACL.
UserEditor	Allows users to manage user accounts. Administrators require 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 require 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. • Any Engage user who requires to create Recommendation assets requires 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). See Understanding DefaultReader, secure.CatalogManager, and secure.TreeManager.
VisitorAdmin	Grants users the ability to create visitor attributes, history attributes, and history types. Any Engage user who requires to create assets of those types requires 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.
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.

Table 39-3 (Cont.) System ACLs and Their Descriptions

ACL Name	Description
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 require this ACL. Also, because the **Admin** tab has both administrative and site design functions, site designers also require this ACL.
xceleditor	Grants users the ability to log in to the WebCenter Sites content applications. The login request code verifies whether a user has the ACL. All users of the management system requires this ACL.
xcelpublish	Grants users the ability to view the Publish Console.

ACLs of Default Users

The following table describes which ACLs are assigned to the default users.

Table 39-4 Default Users and Their ACLs

User Name	Browser	Description
fwadmin	Browser ElementEditor PageEditor PageReader RemoteClient TableEditor UserEditor UserReader Visitor VisitorAdmin xceladmin xceleditor xcelpublish wsadmin wseditor wsuser	Basic administrator user that WebCenter Sites creates so that you can begin configuring your WebCenter Sites content applications. Do not delete this user unless another user with identical ACLs exists.
WebCenter Sites (the installation's user account)	Browser ContentEditor ElementEditor ElementReader PageEditor PageReader SiteGod TableEditor UserEditor UserReader	User account that the installation program creates during the installation of the products. The name of this account is whatever the installers chose for it.

Table 39-4   (Cont.) Default Users and Their ACLs

User Name	Browser	Description
DefaultReader	Browser Visitor	Browser is the ACL that WebCenter Sites assigns to non-authenticated site visitors on the delivery system. The Visitor ACL is also automatically assigned.

Required ACLs for Custom Users

The following table lists the ACLs that would be required by users based on common user descriptions.

Table 39-5   System ACLs Required by Users

User	Required ACLs
All users	Browser, Element Reader, PageReader, UserReader, xceleditor
Workflow Administrator Site Administrator	xceladmin
General Administrator	xceladmin, TableEditor, UserEditor, VisitorAdmin (for)
Site Designer	xceladmin, ElementEditor, PageEditor, TableEditor, Visitor (for), Visitor Admin (for)
Users	Visitor
Users of Web Mode in the interface	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor (for), xceleditor

System Roles

The following table describes the system roles in , and the access the role has.

Table 39-6   System Roles

Role	Description
GeneralAdmin	Default system role for global administrators. Required for users who require access to the **Admin** tab (and all other possible functions) in the tree. **Note**: A user with the GeneralAdmin role must also have the xceladmin ACL to use any of the functions in the **Admin** tab.
SiteAdmin	Default system role for site administrators. Required for users who are administrators of selected sites and therefore require access to the **Site Admin** tab (which shows a subset of the functions in the **Admin** tab). Assign the SiteAdmin role to users who will manage, but not create, other site users. **Note**: A site user with the SiteAdmin role must also have the xceladmin ACL to use any of the functions on the **Site Admin** tab.

Table 39-6 (Cont.) System Roles

Role	Description
WorkflowAdmin	Default system role for workflow administrators. Required for users who require access to the **Workflow** tab in the tree. **Note**: A user with the WorkflowAdmin role must also have the xceladmin ACL to use any of the functions on the **Workflow** tab.
AdvancedUser	Grants users access to the Admin interface.
SitesUser	Grants users access to the interface.

System Asset Types

The following table lists the default asset types. Unlike custom asset types, system asset types cannot be deleted.

Table 39-7 System Asset Types

Asset Type	Description
Attribute Editor	An attribute editor specifies how data is entered for a flex attribute when that attribute is shown 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 for to use when it shows an attribute in the interface—not when it shows 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 to call from multiple templates (a banner perhaps). You also use CSElements to provide the queries that are required to create DynamicList recommendations in .
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.
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 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 .
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 .

Table 39-7 (Cont.) System Asset Types

Asset Type	Description
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 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 .
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 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 .
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 .
SiteEntry	Represents a 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 page.
Template	Stores code (XML or JSP and Java) that renders other assets into 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.
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 .

Default Tree Nodes

The following table lists the default nodes in the tree in the WebCenter Sites Admin interface. These nodes are critical to WebCenter Sites. All features which stem from WebCenter Sites can be accessed through these nodes; they are automatically created upon installation.

Table 39-8 Default Tree Nodes in the WebCenter Sites Admin interface

Tab	Description
Admin	Shows 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.
Bookmarks	Holds a list of all bookmarked pages. Frequently-used pages can be marked and accessed from this tab to make finding them easier.
Connector Admin	Shows the administrative functions for administering WCC Connector.
History	Shows 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.
Mobility	Shows the administrative arrangement of mobile devices and their groupings. The administration of the mobile devices is done from here.
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 have to create users or sites.
Dev	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.
Site Navigation	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.

40
WebCenter Sites Database Tables

The dynamic tables of the WebCenter Sites database can potentially grow to unmanageable sizes. As an admin, you can purge the inactive data from such tables that is not required to efficiently manage the WebCenter Sites database.
Topics:

- Cache Management Tables
- Approval System Tables
- Publishing System Tables
- Workflow Tables
- Basic Asset Tables
- Flex Asset Tables
- Visitor Tables (Engage)

Cache Management Tables

delivers the CacheManager, a page caching utility that manages both the page cache and the Satellite Server caches.

Because cached pages have 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 and the dependencies that exist between the pages and pagelets stored in the cache. It stores this information in the SystemPageCache and SystemItemCache tables.

Every system uses a CacheManager, which means that these tables grow dynamically on any system—development, management, or delivery. The following table describes how cache-tracking tables grow.

Table 40-1 Cache Management Tables

Table	Number of Rows
SystemPageCache	One row for every cached page. When a page expires, the row is removed.
SystemItemCache	One row for each asset, pagelet, or other item that is referenced by a cached page in the SystemPageCache 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.

Approval System Tables

The 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 following table describes how approval system tables grow.

Table 40-2 Approval System Tables

Table	Number of Rows
ApprovedAssets	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.
ApprovedAssetDeps	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.

Publishing System Tables

The 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 management system increases, so does the number of rows in these tables. The following table describes how publishing system tables grow.

Table 40-3 Publishing System Tables

Table	Number of Rows
PubKey	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.
PublishedAssets	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 1 rows (one for each asset) for that page.

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. The following table describes how workflow tables grow.

Table 40-4 Workflow Tables

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.

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.

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.

The following table describes how flex asset tables grow.

Table 40-5 Flex Asset Tables

Table	Number of Rows
FlexAssetType (For example: AVIArticle)	One row for every asset of this type.

Table 40-5 (Cont.) Flex Asset Tables

Table	Number of Rows
FlexAssetType_AMap (for example: AVIArticle_AMap)	One row per attribute value—whether the attribute value is inherited or directly assigned—for each of the assets of that type.
FlexAssetType_Group (for example: ArticleCategory_Group)	One row per ancestor relationship between flex parent asset and flex asset—includes rows for grandparent, great-grandparent, and so on, relationships.
FlexAssetType_Mungo (for example: AVIArticle_Mungo)	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.
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.

Visitor Tables (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. The following table describes how visitor tables grow.

Table 40-6 Visitor Tables in Engage

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 one row per visit. Memory-centric visitor tracking creates one row per visitor, 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.

Table 40-6 (Cont.) Visitor Tables in Engage

Table	Number of Rows
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.
VMVISTORSCALARBLOB	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 covers the following topics:

- Managing the Attribute Tables
- Managing the Session Objects (Scratch) Table
- Deleting Unnecessary .class Files

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 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 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 calls 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 APPEVENT tag (which functions like `kron`). For information about this tag, see the *Tag Reference for Oracle WebCenter Sites Reference* and About Adding to the System Tables in the *Developing with Oracle WebCenter Sites*.

Note that the value that you pass to the `STARTDATE` parameter must be in epoch time. You can use the `DATE.CONVERT` tag to obtain an epoch value for the date 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"
   [HOUR="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 *Tag Reference for Oracle WebCenter Sites Reference*.

Managing the Session Objects (Scratch) Table

The scratch table can grow quickly because at least five session objects are stored for each visitor in each session. Each object has a timestamp; you should purge old objects regularly, based on their timestamps.

To delete old session objects from the scratch table:

- Use the following XML object method or its JSP equivalent to delete.

  ```
  <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.

  The value that you pass to the STARTDATE parameter must be in epoch time. You can use the DATE.CONVERT tag to obtain an epoch value for the date 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 *Tag Reference for Oracle WebCenter Sites*.

  > **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).

Deleting Unnecessary .class Files

is a Java-based application and it generates Java `.class` files each time an event 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 called by .

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, 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 Management Tool to examine the `vis.genclasspath` property and note the directory name designated by that parameter. This is the directory where 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.

regenerates any `.class` files that it requires.

41

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

Depending upon the type of LDAP server, the operation, and whether you are connected as a user with read-only or read-write permissions; certain operations that you perform through the LDAP server can result in errors. As you manage users, ACLs, sites, and roles from the Admin interface of an LDAP-integrated WebCenter Sites system, familiarize yourself with the operations that affect both the WebCenter Sites database and the LDAP server to troubleshoot the errors, if any.
Topics:

- LDAP-Integrated WebCenter Sites Systems
- User Management Operations
- Site and Role Management Operations

LDAP-Integrated WebCenter Sites Systems

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.

Topics:

- LDAP Schema
- LDAP Connectivity for Site and Role Management
- LDAP Users and Their Permissions to LDAP Servers
- LDAP-Integrated Operations

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.

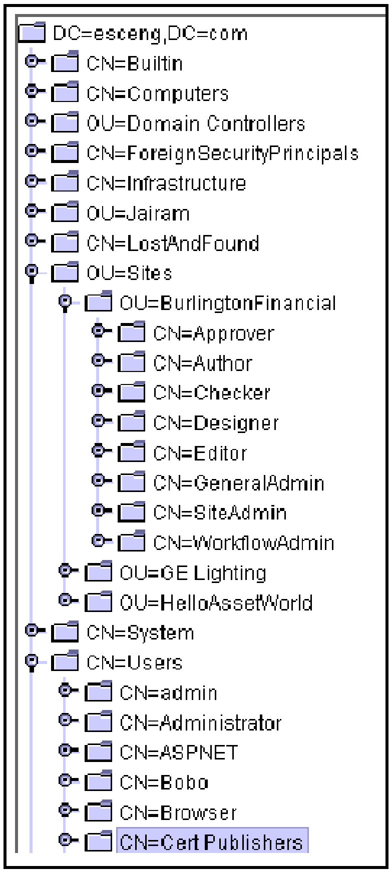
- Flat-schema LDAP supports operations on sites and roles in WebCenter Sites and levels the sites and roles so that no hierarchy exists among them.
- Hierarchical-schema LDAP supports operations on sites and roles in WebCenter Sites. This schema requires LDAP users to define a site organizational unit in the LDAP server in which roles must be subordinated to their relevant sites (for an example, see the following figure).

The following table 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 LDAP Connectivity for Site and Role Management.

Table 41-1 LDAP Integration Scenarios

Create/Modify/Delete Operations on...	Flat-Schema LDAP	Hierarchical-Schema LDAP
ACLs, Users, User Profiles, User Attributes	Table 41-2	Table 41-2
Sites and Roles	Table 41-3	Table 41-4

Figure 41-1 LDAP Hierarchies



```
DC=esceng,DC=com
  CN=Builtin
  CN=Computers
  OU=Domain Controllers
  CN=ForeignSecurityPrincipals
  CN=Infrastructure
  OU=Jairam
  CN=LostAndFound
  OU=Sites
    OU=BurlingtonFinancial
      CN=Approver
      CN=Author
      CN=Checker
      CN=Designer
      CN=Editor
      CN=GeneralAdmin
      CN=SiteAdmin
      CN=WorkflowAdmin
    OU=GE Lighting
    OU=HelloAssetWorld
  CN=System
  CN=Users
    CN=admin
    CN=Administrator
    CN=ASPNET
    CN=Bobo
    CN=Browser
    CN=Cert Publishers
```

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. Use the Property Management Tool in the Admin interface to set the values of two properties in the `wcs_properties.json` 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 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 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.LDAPSchemaUs
    erManager
    xcelerate.rolemanagerclass=com.openmarket.xcelerate.roles.RoleManager
    ```

  - For flat-schema LDAP, both properties must specify LDAP. For example:

    ```
    xcelerate.usermanagerclass=com.openmarket.xcelerate.user.FlatLDAPSche
    maUserManager
    xcelerate.rolemanagerclass=com.openmarket.xcelerate.roles.FlatLDAPSch
    emaRoleManager
    ```

- 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 database.

LDAP Users and Their Permissions to LDAP Servers

In an LDAP-integrated system, the administrator of the system may or may not be an administrative user of the LDAP server, depending on the value of the `jndi.connectAsUser` property in the `wcs_properties.json` file, categorized under User. The value determines how the administrator is connected to the LDAP server, and therefore defines the LDAP user:

- If jndi.connectAsUser is set to `true`, then defines the LDAP user to be the same one that is logged in to and connects that user to the LDAP server.
- If jndi.connectAsUser is set to `false`, then defines the LDAP user to be the one that is specified in the `jndi.login` property (in the `wcs_properties.json` file, categorized under User) 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:**
>
> All LDAP properties are prefixed by `jndi`. For more information about LDAP-related properties, see User Properties in the *Property Files Reference for Oracle WebCenter Sites*.

LDAP-Integrated Operations

For the administrator to successfully perform an operation (such as creating an ACL in both the database and the LDAP server), it is critical for to be properly integrated with the LDAP server. Barring integration issues, the outcome of an operation depends on the following factors:

- The LDAP user, as defined by the jndi.connectAsUser property (see the previous section).
- The LDAP user's permissions, as defined in the LDAP server.

  If the administrator is connected to the LDAP server as a user without administrative rights, his operations (such as deleting an ACL from the database) cannot be written to the LDAP server. The system responds by either writing the operations to the 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.
- 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 administrator is connected with write permissions.
- 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 41-2, Table 41-3 and Table 41-4 summarize the results of operations that can be performed by a administrator who is connected to the LDAP server as a user with read-only and read/write permissions.

User Management Operations

Table 41-2 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 an LDAP option:

- LDAP-2 server with read-only permissions for the LDAP user (defined in LDAP Connectivity for Site and Role Management).
- 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 41-2.

> **Note:**
>
> The term "user management" in this appendix includes the management of ACLs, user accounts, user profiles, and user attributes.

Table 41-2 System Response to User Management Operations in WebCenter Sites

Operation in WebCenter Sites	Result	System Response with LDAP-2: User has Read-Only Permissions (see note 1)	System Response with LDAP-2: User has Write Permissions (see note 2)	System Response with Other LDAP Server: Write not Supported (see note 3)
LDAP Server Response for Creating an ACL	*ACL is created in:*	Returns error (see note 4)	LDAP + WebCenter Sites database	Unpredictable
LDAP Server Response for Editing an ACL	*ACL is modified in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Deleting an ACL	*ACL is deleted in:*	Returns error	LDAP + WebCenter Sites database	Unpredictable
LDAP Server Response for Assigning ACLs to Tables	*ACL is assigned in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries)	*ACL is assigned in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Creating a User (Granting ACLs for Access Privileges)	*User is created in:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Editing a User	*User is modified in:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Deleting a User	*User is deleted from:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Creating/Editing a User Profile	*User profile is created/ edited in:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Deleting a User Profile	*User profile is deleted from:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Creating User Attributes	*User profile is created in:*	Returns error	LDAP	Unpredictable

Table 41-2 (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 (see note 1)	System Response with LDAP-2: User has Write Permissions (see note 2)	System Response with Other LDAP Server: Write not Supported (see note 3)
LDAP Server Response for Editing/Deleting User Attributes	*User profile is deleted from:*	Returns error	LDAP	Unpredictable

Note:

- 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.
- 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.
- 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.
- "Returns error" means that the operation is not performed (the system returns an error).

The following topics provide reference information about user management options:

- LDAP Server Response for Creating an ACL
- LDAP Server Response for Editing an ACL
- LDAP Server Response for Deleting an ACL
- LDAP Server Response for Assigning ACLs to Tables
- LDAP Server Response for Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries)
- LDAP Server Response for Creating a User (Granting ACLs for Access Privileges)
- LDAP Server Response for Editing a User
- LDAP Server Response for Deleting a User
- LDAP Server Response for Creating/Editing a User Profile
- LDAP Server Response for Deleting a User Profile
- LDAP Server Response for Creating User Attributes
- LDAP Server Response for Editing/Deleting User Attributes

LDAP Server Response for Creating an ACL

User Management Operation:

The 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 database.
2. LDAP-2 Server with Write Permissions—The system creates the ACL in both the LDAP server and the 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 .

LDAP Server Response for Editing an ACL

User Management 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).

LDAP Server Response for Deleting an ACL

User Management 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.

LDAP Server Response for Assigning ACLs to Tables

User Management 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).

LDAP Server Response for Assigning ACLs to WebCenter Sites Pages (Site Catalog Page Entries)

User Management 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).

LDAP Server Response for Creating a User (Granting ACLs for Access Privileges)

User Management Operation:

You can attempt to create a user by using the **User Access Management** node under the **Admin** node of the **General Admin** tree.

> **Note:**
>
> WebCenter Sites shows 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.

LDAP Server Response for Editing a User

User Management Operation:

You can attempt to modify the ACLs that are assigned to a user or to change the user's password by using the **User Access Management** node under the **Admin** node of the **General Admin** tree.

> **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 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.

LDAP Server Response for Deleting a User

User Management Operation:

You can attempt to delete a user by using the **User Access Management** node under the **Admin** node of the **General Admin** tree.

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.

LDAP Server Response for Creating/Editing a User Profile

User Management Operation:

The WebCenter Sites administrator attempts to create or modify a user profile by using the **User Profiles** node under **Admin** node of the **General Admin** tree.

System Response:

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot update the user's profile or Locale Preference. The LDAP administrator must manually update the user profile in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system updates the user's profile or locale preference in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually update the user profile in the LDAP server.

LDAP Server Response for Deleting a User Profile

User Management Operation:

You can attempt to delete a user profile by using the **User Profiles** node under the **Admin** node of the **General Admin** tree.

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.

LDAP Server Response for Creating User Attributes

User Management Operation:

The WebCenter Sites administrator attempts to create attributes for the user from the **User Access Management** node under the **Admin** node of the **General Admin** tree.

> **Note:**
>
> User attributes are stored in only one place—either in the database or in LDAP.

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.

LDAP Server Response for Editing/Deleting User Attributes

User Management Operation:

You can attempt to delete a user attributes by using the **User Access Management** node under the **Admin** node of the **General Admin** tree.

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.

Site and Role Management Operations

The following topics provide information about site and role management operations with flat- and hierarchical-schema LDAP servers.

- Operations with Flat-Schema LDAP Servers
- Operations with Hierarchical-Schema LDAP Servers
- LDAP Server Response for Creating a Site
- LDAP Server Response for Editing a Site
- LDAP Server Response for Deleting a Site
- LDAP Server Response for Granting Users Access to Sites
- LDAP Server Response for Removing Users' Access to Sites
- LDAP Server Response for Creating a Role
- LDAP Server Response for Editing a Role
- LDAP Server Response for Deleting a Role

Operations with Flat-Schema LDAP Servers

The following table 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 an LDAP option:

- An LDAP-2 server with read-only permissions for the LDAP user (defined in LDAP Connectivity for Site and Role Management).
- 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 in the sections that are noted in the left-hand column of the following table .

Table 41-3 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 (see note 1)	System Response with Flat-Schema LDAP-2: User has Write Permissions (see note 2)	System Response with Flat-Schema LDAP Server (Other Than LDAP-2): Write Not Supported (see note 3)
LDAP Server Response for Creating a Site	*Site is created in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Editing a Site	*Site is edited in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Deleting a Site	*Site is deleted from:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Creating a Role	*Role is created in:*	Returns error (see note 4)	LDAP + WebCenter Sites database (see note 5)	Unpredictable
LDAP Server Response for Editing a Role	*Role is edited in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Deleting a Role	*Role is deleted from:*	Returns error	LDAP + WebCenter Sites database	Unpredictable
LDAP Server Response for Granting Users Access to Sites	*Access is granted in:*	Returns error	LDAP	Unpredictable
LDAP Server Response for Removing Users' Access to Sites	*Access is removed from:*	Returns error	LDAP	Unpredictable

Note:

- 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).
- 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).
- 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.
- "Returns error" means that the operation is not performed (the system returns an error).
- 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 site exists, the role is created only in the WebCenter Sites database.

Operations with Hierarchical-Schema LDAP Servers

The following table 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 an LDAP option:

- An LDAP-2 server with read-only permissions for the LDAP user (defined)
- 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 in the sections that are noted in the left-hand column of the following table .

Table 41-4 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 (see note 1)	System Response with Hierarchical-Schema LDAP-2: User has Write Permissions (see note 2)	System Response with Hierarchical-Schema LDAP Server (Other Than LDAP-2): Write Not Supported (see note 3)
LDAP Server Response for Creating a Site	*Site is created in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database

Table 41-4 (Cont.) 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 (see note 1)	System Response with Hierarchical-Schema LDAP-2: User has Write Permissions (see note 2)	System Response with Hierarchical-Schema LDAP Server (Other Than LDAP-2): Write Not Supported (see note 3)
LDAP Server Response for Editing a Site	*Site is edited in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Deleting a Site	*Site is deleted in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Creating a Role	*Role is created in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Editing a Role	*Role is edited in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Deleting a Role	*Role is deleted in:*	WebCenter Sites database	WebCenter Sites database	WebCenter Sites database
LDAP Server Response for Granting Users Access to Sites	*Access is granted in:*	Returns error (see note 4)	LDAP	Unpredictable
LDAP Server Response for Removing Users' Access to Sites	*Access is removed from:*	neither the WebCenter Sites database nor the LDAP server	neither the WebCenter Sites database nor the LDAP server	neither the WebCenter Sites database nor the LDAP server

Note:

- 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).
- 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).
- 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.
- "Returns error" means that the operation is not performed (the system returns an error).

LDAP Server Response for Creating a Site

Site Management Operation:

You can attempt to create a site by using the **Sites** node under the **Admin** node of the **General Admin** tree.

> **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 are not 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).

LDAP Server Response for Editing a Site

Site Management Operation:

You can attempt to edit the description of a site by using the **Site Edit** option on the **Admin** node.

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).

LDAP Server Response for Deleting a Site

Site Management Operation:

To delete the description of the site, do the following:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click the site you want to edit.
2. In the Site form, click **Edit** at the top of the form.

> **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 are not 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).

LDAP Server Response for Granting Users Access to Sites

Site Management Operation:

The WebCenter Sites administrator attempts to grant users access to a site by using the **Users** node of the site, under the **Sites** node of the **Admin** node, in the **General Admin** tree.

> **Note:**
>
> WebCenter Sites shows 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.

LDAP Server Response for Removing Users' Access to Sites

Site Management Operation:

The WebCenter Sites administrator attempts to remove user access to sites by using the **Users** node of the site, under the **Sites** node of the **Admin** node, in the **General Admin** tree.

> **Note:**
>
> WebCenter Sites shows 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.

LDAP Server Response for Creating a Role

Role Management Operation:

To create a role for a site,

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **Roles** node, and then double-click **Add New**.
2. In the Add New Role form, enter the **Name** and **Description** of the Role, and then click **Add New Role**.

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.

LDAP Server Response for Editing a Role

Role Management Operation:

To modify the description of a role for a site, do the following:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **Roles** node, and then double-click the role you want to modify.
2. In the Role form, click **Edit**.

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).

LDAP Server Response for Deleting a Role

Role Management Operation:

To delete a role for a site, do the following:

1. In the **General Admin** tree, expand the **Admin** node, expand the **User Access Management** node, expand the **Roles** node, and then double-click the role you want to modify.
2. In the Role form, click **Delete**, and then in the Delete Role form, click **Delete Role**.

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.

42
Alternate Publishing Methods

In addition to Realtime publishing, you can use On Demand publishing where the delivery system is a database or an external system running an application other than WebCenter Sites, Export to Disk publishing where the delivery system is a web server, Mirror to Server publishing where the delivery system is a WebCenter Sites system, and Exporting Assets to XML where the delivery system is a database or an external system running an application other than WebCenter Sites.

Topics:

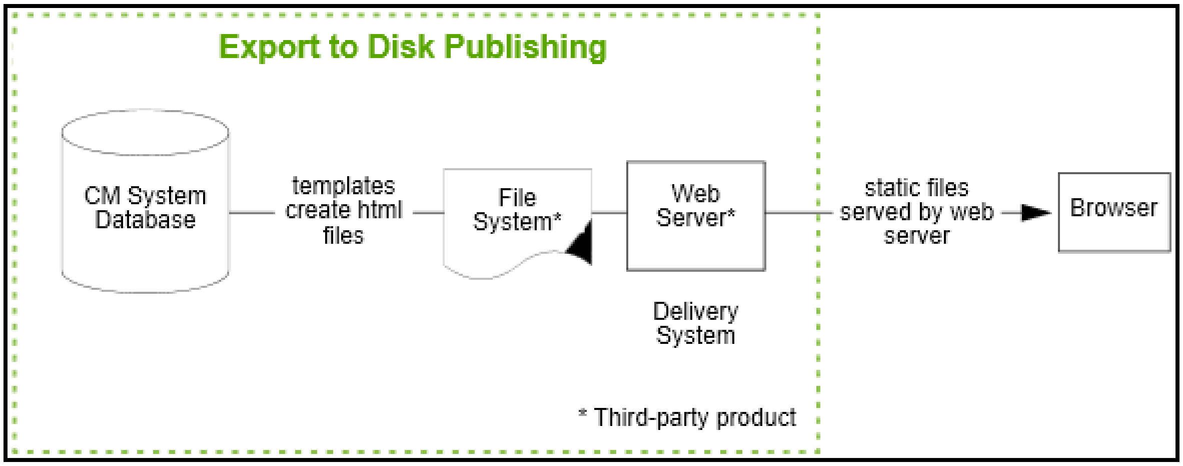
- About Alternate Publishing Methods
- Using On-Demand Publishing
- Unapproving Selected Assets from a Publishing Session
- Learning Export to Disk Publishing Terminology
- Using Approvals and Export to Disk Publishing
- Working with Export to Disk Publishing
- Frequently Asked Questions About the Export to Disk Publishing Process
- Investigating Publishing Behavior and Functionality
- Understanding the Approval and Publish-Time Templates: What Happens When They Differ?
- Working with Sample Templates
- Rendering Export to Disk Pages
- Understanding Export to Disk Publishing
- Working with the Path Naming Convention
- Configuring Your System for Export to Disk Publishing
- Understanding Mirror to Server Publishing
- Pre-Configuring Mirror to Server Publishing
- Configuring Your System for Mirror to Server Publishing
- Retrieving Logs From Delivery WebCenter Sites Systems
- Exporting Assets to XML Publishing Method
- Configuring Your System for Export Assets to XML
- Migrating a Site from One System to Another
- Approving Multiple Assets
- Creating Destinations
- Editing Destinations
- Deleting Destinations
- Creating Export Starting Points

About Alternate Publishing Methods

This section describes the publishing methods that WebCenter Sites supports. The following three figures summarizes the publishing methods.

For **Export to Disk publishing**, the delivery system is a web server. 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.

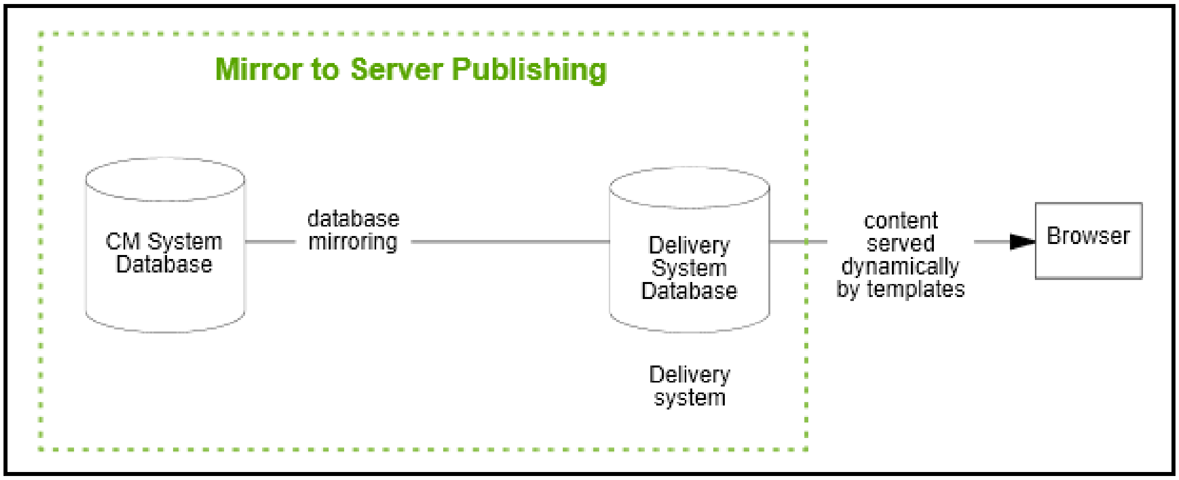
Figure 42-1 Export to Disk Publishing


Export to Disk Publishing
CM System Database
templates create html files
File System*
Web Server*
Delivery System
static files served by web server
Browser
* Third-party product

For **Mirror to Server** publishing, the delivery system is a WebCenter Sites system. 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 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 mirrored assets. 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.

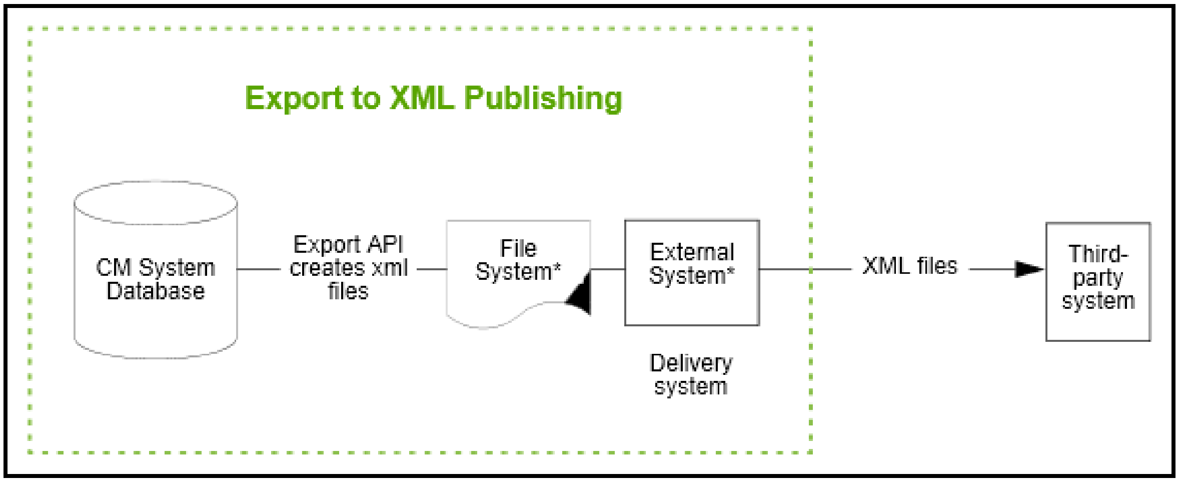
Figure 42-2 Mirror to Server Publishing



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.

Figure 42-3 Export to XML Publishing



Using On-Demand Publishing

When you publish, you can immediately and selectively publish any of the assets on the approved assets list. This is known as publishing assets on-demand.

As with scheduled RealTime publishing, an asset is published with all its dependent assets. Therefore, the dependent assets are also placed in the On-Demand Queue.

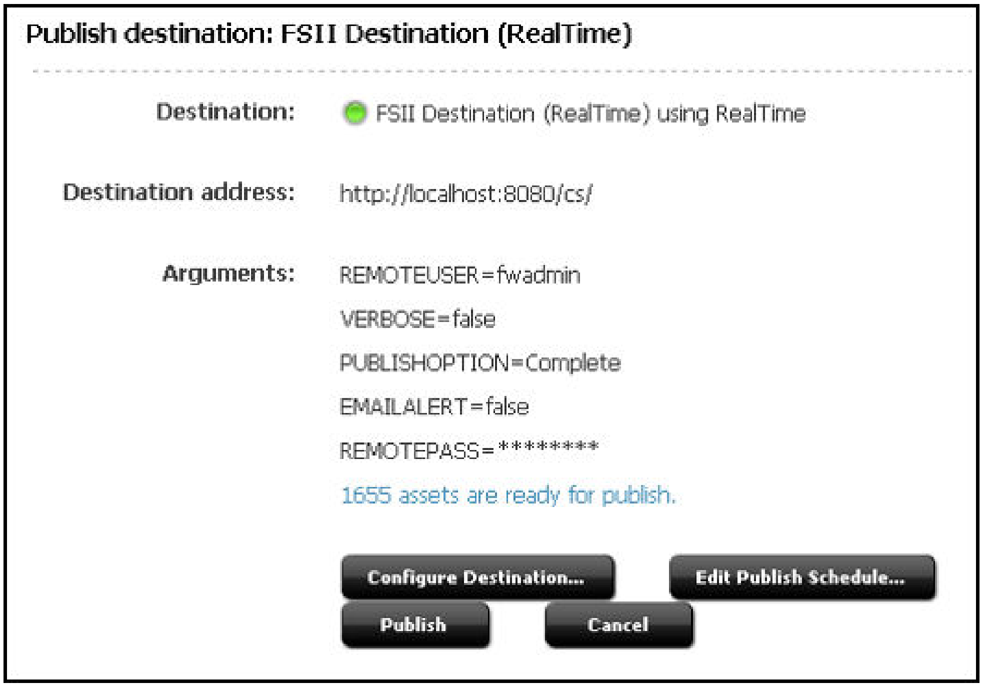
The list of approved assets is searchable, making it easier to find the assets you have to publish. After the selected assets are published, they are removed from the approved assets list for the next publishing session.

> **Note:**
>
> On Demand publishing can be either complete or delayed.

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 42-4   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.

4. (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.

5. Add assets to the On-Demand Queue.

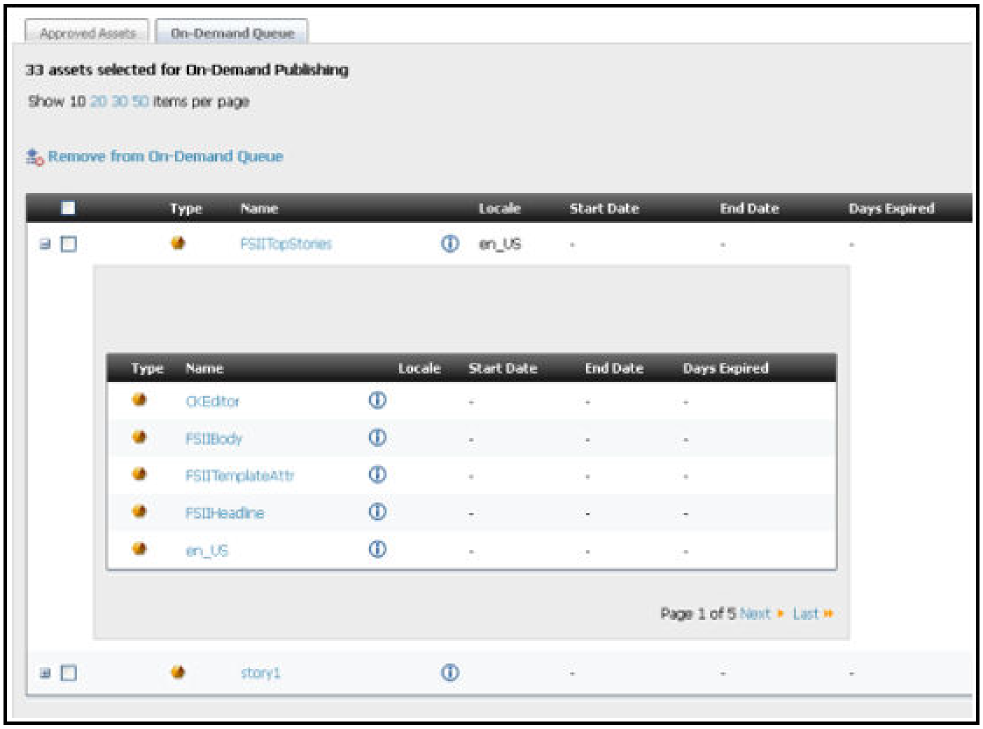
   > **Note:**
   >
   > On-Demand supports immediate publication. Selections that you place into the On-Demand queue remain there only throughout the current WebCenter Sites session. When you log out (or the system logs you out), the On-Demand queue is cleared.

   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 opens while the dependencies are calculated. 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. Complete either:
   - 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 are removed from the On-Demand Queue and placed back in the **Approved Assets** list. (A dialog box prompts you to confirm your action.)

Figure 42-5 On-Demand Tab



33 assets selected for On-Demand Publishing

Show 10 20 30 50 items per page

Remove from On-Demand Queue

□	Type	Name		Locale	Start Date	End Date	Days Expired
□	●	FSIITopStories	ⓘ	en_US	-	-	-

Type	Name		Locale	Start Date	End Date	Days Expired
●	CKEditor	ⓘ		-	-	-
●	FSIIBody	ⓘ		-	-	-
●	FSIITemplateAttr	ⓘ		-	-	-
●	FSIIHeadline	ⓘ		-	-	-
●	en_US	ⓘ		-	-	-

Page 1 of 5 Next ▸ Last »

□	Type	Name		Locale	Start Date	End Date	Days Expired
□	●	story1	ⓘ		-	-	-

7. Click **OK**.

   The Publish Console opens showing the publishing session in process.

8. For instructions on observing the publishing session, see Using RealTime Publishing.

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**.

   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.

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.

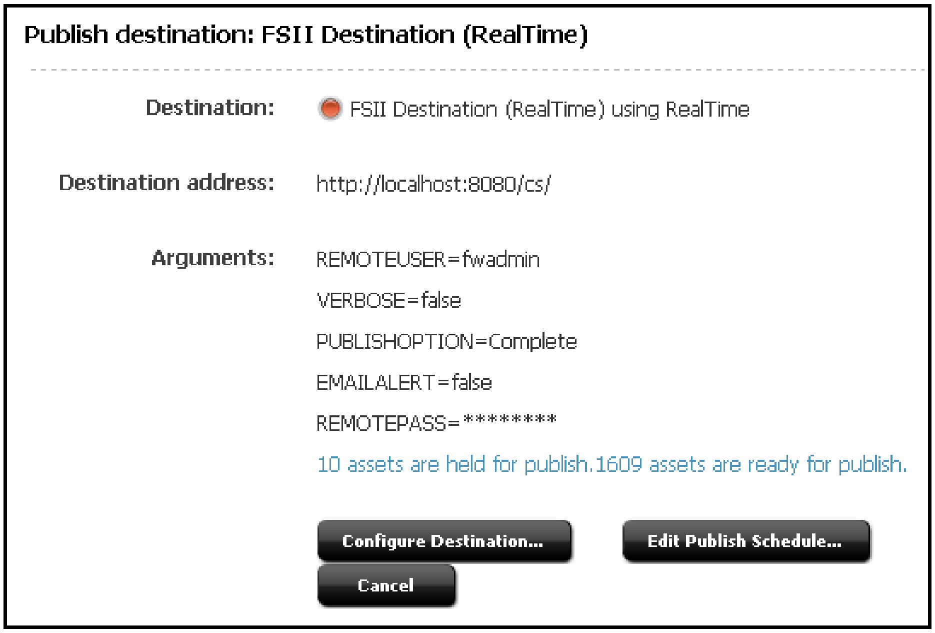
5. On either the **Approved Assets** tab or the **Search Results** tab, select assets to unapprove. Click **Unapprove**.

   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. 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 42-6 Publish Destination



Publish destination: FSII Destination (RealTime)

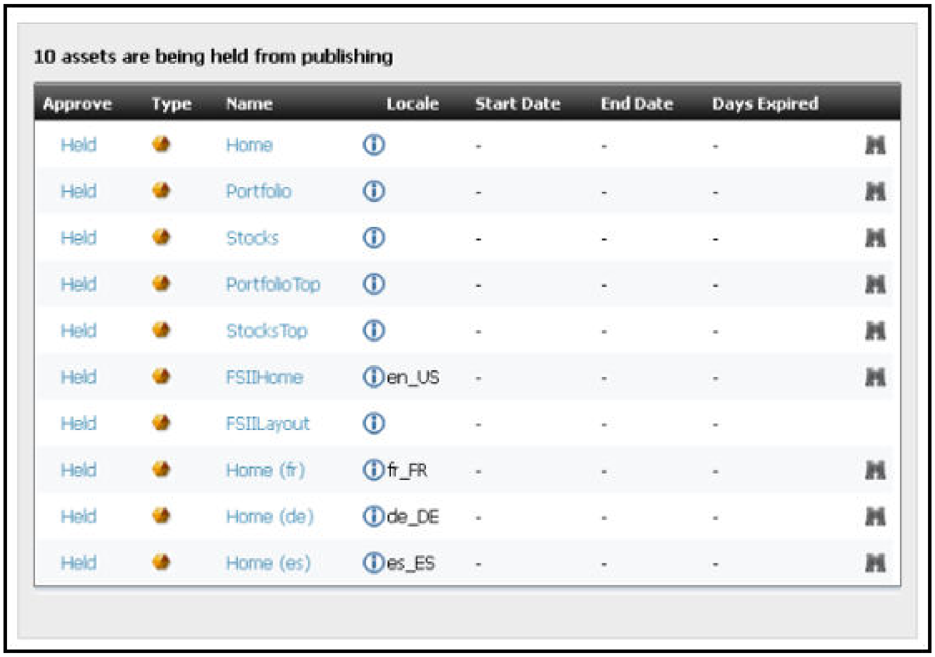
Destination: FSII Destination (RealTime) using RealTime

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.

The held queue opens.

Figure 42-7 Assets in Held Queue



10 assets are being held from publishing

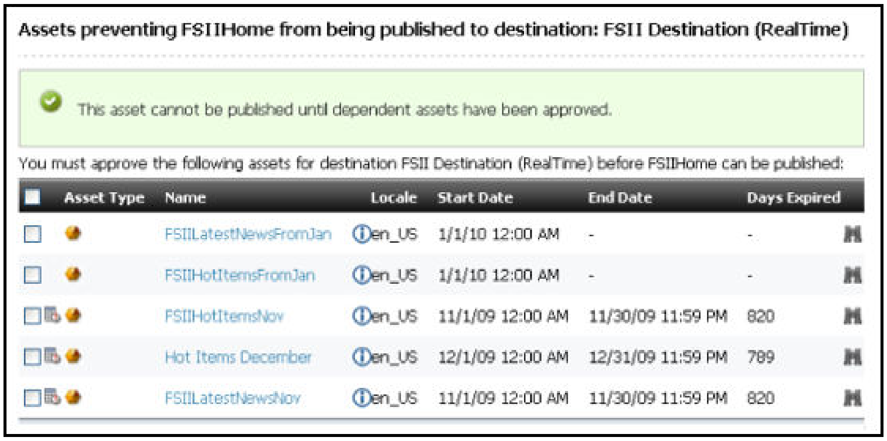
Approve	Type	Name	Locale	Start Date	End Date	Days Expired	
Held	●	Home	ⓘ	-	-	-	
Held	●	Portfolio	ⓘ	-	-	-	
Held	●	Stocks	ⓘ	-	-	-	
Held	●	PortfolioTop	ⓘ	-	-	-	
Held	●	StocksTop	ⓘ	-	-	-	
Held	●	FSIIHome	ⓘen_US	-	-	-	
Held	●	FSIILayout	ⓘ	-	-	-	
Held	●	Home (fr)	ⓘfr_FR	-	-	-	
Held	●	Home (de)	ⓘde_DE	-	-	-	
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 opens stating that the asset is approved.
- If the asset has dependencies, a list of those dependent assets is shown.

Figure 42-8 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:

☐	Asset Type	Name	Locale	Start Date	End Date	Days Expired	
☐	●	FSIILatestNewsFromJan	ⓘen_US	1/1/10 12:00 AM	-	-	
☐	●	FSIIHotItemsFromJan	ⓘen_US	1/1/10 12:00 AM	-	-	
☐	●	FSIIHotItemsNov	ⓘen_US	11/1/09 12:00 AM	11/30/09 11:59 PM	820	
☐	●	Hot Items December	ⓘen_US	12/1/09 12:00 AM	12/31/09 11:59 PM	789	
☐	●	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 selecting the check box in the header row of the table. Then click **Approve**.

The following occurs: A message indicates that the assets are 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.

Learning Export to Disk Publishing Terminology

Compared to mirror publishing, where dependencies in both the approval and publishing stages are determined by the asset model, Export to Disk publishing is an entirely template-driven process. 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 administrator (and any users who publish) must understand which templates to call and what outcome to expect to prevent adverse effects on publishing sessions.

> **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.

This section covers the following topics:

- Page
- Publish Key
- Publish Queue
- Primary Asset

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 page.

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 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 approval template).

The initial pubkeys in Export to Disk publishing are the starting points defined by the user who initiates the publishing session. All other pubkeys are discovered and logged during publishing.

Publish Queue

A publish queue is a list of pubkeys awaiting publication. For a pubkey to be added to the publish queue, its primary asset must be approved.

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 is not exported. This can lead to the creation of a broken link on the publishing destination, depending on which approval and publishing templates were used.
>
> 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.

Using 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 Investigating Publishing Behavior and Functionality and Understanding the 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 Working with Export to Disk Publishing) do not require approval to appear on an exported page (to filter out unapproved assets, see the `<render:filter>` tag in the *Tag Reference for Oracle WebCenter Sites Reference* for more information).

This section covers the following topics:

- Using the Approval Template
- Using the Approval Queue

- Using Approval Dependencies

Using the Approval Template

Approval in Export to Disk publishing allows the developer to define dependencies that ensure the correct content is published and that 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 (and publish-time behavior) is defined by the developer who writes the approval templates (and 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 compositional dependencies. For more information, see Working with 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 (that is, 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.

Using the 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 call a feature that uses approval functionality (such as the **Publishing** tab), the approval queue is forced to run before it is rendered to keep approval information up to date.

Using Approval Dependencies

When you approve an asset, its approval template (see Using the 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 a tag listed above, an approval dependency is created from A to B. This typically 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 link 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 — to approve A, you must approve B if B has been changed. Reference dependencies are always exists dependencies—if you approved and published B one time, you are not required to approve it again 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.

Working with Export to Disk Publishing

Three main components play a role in the Export to Disk publishing process:

- Working with the Publish-Time Template

- Working with Starting Points
- Working with Compositional Dependencies

Working with the 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. When they differ, content can be published without first being approved. Bypassing the approval system requires the developer to provide a means for publishable content to be validated.

Working with 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 is not necessarily the home page. For example, it could be a sidebar with many links.) The Export to Disk publishing system crawls your site, beginning with the starting points, and logs new pubkeys as it discovers them.

Working with 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 Working with the 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 Using Approval Dependencies.

Frequently Asked Questions About the Export to Disk Publishing Process

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 is not necessarily 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 to publish the asset with? Set it as the approval template for assets of that type, and use any 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, is able to remind you which pages require updating, assuming you will want to republish those pages. 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, the modified assets do not have to be approved in order for the page to be queued for publishing; in other words, the article page is 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 have to set up `exact` dependencies.

Investigating Publishing Behavior and Functionality

Following is a compilation of simple scenarios which demonstrate Export to Disk publishing behavior:

- Investigating Template Dependencies
- Investigating 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 have to leverage the `deptype='none'` attribute in their tags if they do not wish to set up additional dependencies. For the template code, see Working with Sample Templates.

Investigating 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 is 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 42-1 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` – have 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.

Investigating 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 opens on every newly created template.

Table 42-2 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`

Table 42-2 (Cont.) Reference Dependencies

Action	Approval Status	Publish Status	Comments
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 `<render:getpageurl>` 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`.

Understanding the 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 Working with Sample Templates.

This section covers the following topics:

- Differing Templates: Example 1
- Differing Templates: Example 2
- Differing Templates: Example 3

Differing Templates: 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. Whenever asset B is changed, the affected pages are automatically re-published. (Because there are no dependencies, the automatic refresh option is chosen). So, A+T2 are placed in the publish queue automatically as soon as asset B is saved—not approved. For information about republication options, see Frequently Asked Questions About the Export to Disk Publishing Process.

Differing Templates: 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. 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.

Differing Templates: 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 the primary asset. Now, remember that primary assets must be approved before their page is exported. However, B is not approved, so its page is not 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 require re-approval. 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 with 1088466917821 to myDir/myBlob.out.

Working with Sample Templates

This section lists the template implementations used in the examples in Investigating Publishing Behavior and Functionality and Understanding the 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.

Rendering Export to Disk Pages

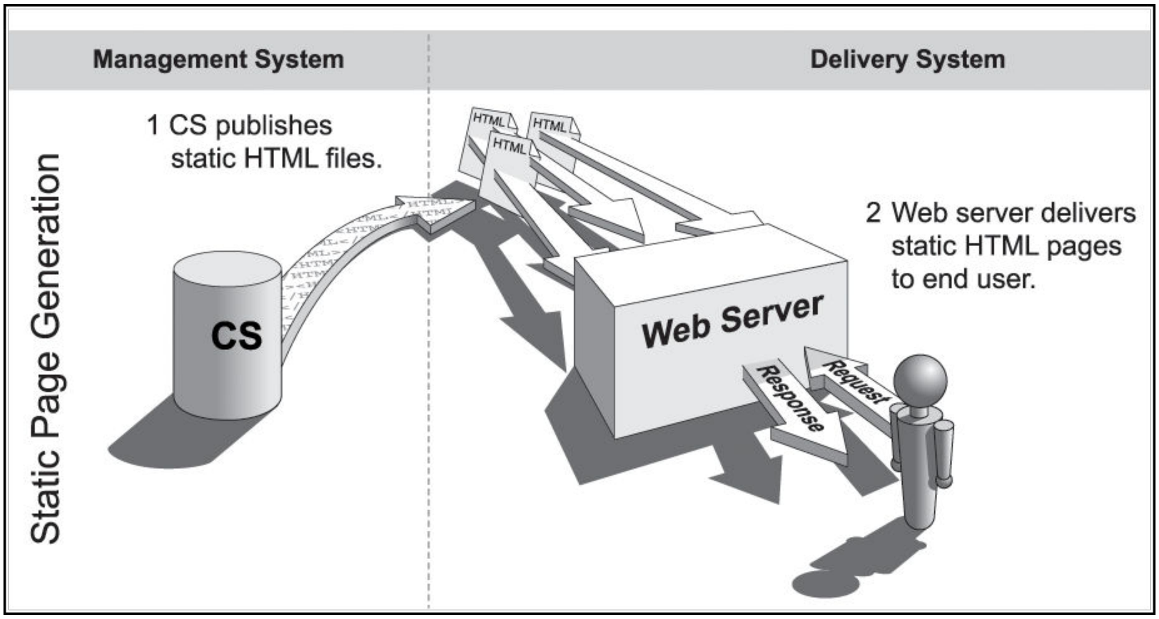
In basic terms, renders a Export to Disk page as follows:

- A page name is submitted by the client browser to (through HTTP or another protocol).

- locates the page in the SiteCatalog table, calls the root element, and renders the page to an HTML file.
- 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. This rendering is illustrated in the following figure.

Figure 42-9 Export to Disk Rendering



> **Note:**
>
> Export to Disk publishing is not recommended. Instead, use the Site Capture application of WebCenter Sites.

Understanding Export to Disk Publishing

When content is published using Export to Disk, the approval system, the publishing schedule, and your destination configurations all contribute to the export of 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 starting 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 a publishing session, exported files are cached to the page cache. If publishing sessions occur more frequently than the cache is configured to clear (expiration time), there could be exported pages in the cache that have the same names as the newly exported pages. To prevent this, the CacheManager servlet clears the page cache 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:

   - A pubkey that has been designated as an export starting point. See Working with 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.

   - 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 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 if changed dependents would require publishing the asset. The presence of this tag means that the asset must be published every 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 calls 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 are multiple export starting points, 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.

Working with the Path Naming Convention

How WebCenter Sites names paths to exported files depends on the type of asset that is published and path information that could 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 covers the following topics:

- Working with Export Path Variables and SIMPLEDIR
- Working with Export Path Construction
- Working with Paths for Links Within Exported Files
- Working with File Naming Conventions
- Working with Export Starting Points

Working with 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 `wcs_properties.json` 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 (see Configuring an Export Destination). 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 a value 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 42-3 Possible File_Dir values

Possible <file_dir> Values	Description
`<ForDestination:Path>`	Value of the For Destination: Path field (the field is shown in the **Specify Path/Filename for Destination** form for a specific destination). 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. If `<ForDestination:Path>` is not set, WebCenter Sites uses `<parent_page's_path>.`
`<parent_page's_path>`	Value of the Path field for the exported asset's parent page. 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.) If `<parent_page's_path>` is not set, WebCenter Sites uses `<ID_of_parent_page>`.
`<asset's_path>`	Value of the exported asset's Path field. 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>`.
`<ID_of_parent_page>`	ID of the asset's parent page. This value is used if `<parent_page's_path>` and `<asset's_path>` are unknown.

- **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** check box in the **Add New Destination** form (see Configuring an Export Destination).

Working with Export Path Construction

This section summarizes the export paths that WebCenter Sites can construct using the convention:

```
<cs.pgexportfolder>/<DIR>/<file_dir>
```

Note that only `<file_dir>` varies in all the examples below

This section covers the following topics:.

- Using a Forced Export Path
- Using an Export Path for Page Assets
- Using an Export Path for Assets Other than Page
- Using Sample Export Paths

Using a Forced Export Path

When `<ForDestination:Path>` (defined in Working with 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>
```

Using an Export Path for Page Assets

If `<ForDestination:Path>` is not set, constructs the export path as shown below (for examples, see Table 42-4).

- 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`, uses `<ID_of_parent_page>`:

```
<cs.pgexportfolder>/<DIR>/<ID_of_parent_page>
```

  - If `SIMPLEDIR=true`, omits `<file_dir>`:

```
<cs.pgexportfolder>/<DIR>
```

Using an Export Path for Assets Other than Page

For an asset that is not a page, constructs an export path depending on which information is available (For examples, see Table 42-5).

- When `<asset's_path>` is specified, the export path is:

```
<cs.pgexportfolder>/<DIR>/<asset's_path>
```

- When `<asset's_path>` is not specified, 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`, uses `<ID_of_parent_page>`:

```
<cs.pgexportfolder>/<DIR>/<ID_of_parent_page>
```

  - If `SIMPLEDIR=true`, omits `<file_dir>`:

```
<cs.pgexportfolder>/<DIR>
```

Using Sample Export Paths

The following tables describe the sample export paths for assets.

Table 42-4 Sample Export Paths for Page Assets when `SIMPLEDIR=false`

Page Asset	*cs.pgexportfolder*	*DIR*	*file_dir (value 1)*	*file_dir (value 2)*	*file_dir (value 3)*	Export path
N/A	N/A	N/A	*asset's_path*	*parent_page's_path*	*ID_of_parent_page*	N/A
Home	`/export`	N/A	`/abstracts`	N/A	N/A	`/export`

Table 42-4 (Cont.) Sample Export Paths for Page Assets when `SIMPLEDIR=false`

Page Asset	*cs.pgexportfolder*	*DIR*	*file_dir (value 1)*	*file_dir (value 2)*	*file_dir (value 3)*	Export path
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/998877665`
World News	`/export`	N/A	`/abstracts`	N/A	998877665	`/export/998877665`

Table 42-5 Sample Export Paths for Assets Other Than Page when `SIMPLEDIR=false`

Not Page Asset	*cs.pgexportfolder*	*DIR*	*file_dir (value 1)*	*file_dir (value 2)*	*file_dir (value 3)*	Export path
N/A	N/A	N/A	*asset's_path*	*parent_page's_path*	*ID_of_parent_page*	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/997766554`

Working with 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>
```

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.

Working with 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 about this setting, see Configuring Your System for Export to Disk Publishing. Possible file names are shown in the following table. The naming convention is explained after the table.

Table 42-6   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, these 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 multiple templates, do not select **Use Simple File Naming.**

- `<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 Using the referURL Variable in the *Developing with Oracle WebCenter Sites*.

- `<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 requires 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..

The following table 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 42-7 Export File Naming Conventions

Asset Name	SiteCatalog Pagename	packedargs	filename	asset_ID	SUFFIX	Exported File Name
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`
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 PACKEDARGS="topicword=oil"	home.html	123		`BF-Page-Home_topicword_ oil_home.htm`

Working with 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. calls 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 link, 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 have to designate multiple export starting points. This is because if there is a section on your site that isn't connected to the rest of the site, it is not rendered. For example:
>
> - If your online site has multiple top-level page assets, you require 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.

Configuring Your System for Export to Disk Publishing

The main steps when configuring a system for Export to Disk publishing are these:

- Creating the Batch User Account For Export Publishing (If One Does Not Exist)
- Specifying the Root Export Directory

- Configuring an Export Destination
- Mapping a URL Prefix for Your Web Server
- Mapping a URL Prefix for Your Web Server
- Creating the Export Starting Points
- Approving Your Assets for Export to Disk Publishing
- Setting Up the Schedule

Creating the Batch User Account For Export Publishing (If One Does Not Exist)

This procedure must be performed only one time on each source system, regardless of the number and types of publishing destinations you are configuring.

If you have not done so, create the batch user account. For instructions, see Creating the Batch User Account (If One Does Not Exist).

If an account exists for your installation, skip to Specifying 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 `wcs_properties.json` file. Additional properties are also available to you for fine-tuning your system configuration. For a complete list of properties, see Oracle WebCenter Sites JSON Property File in the *Property Files Reference for Oracle WebCenter Sites*.

Specifying 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 `wcs_properties.json` file. (If you wish to add subdirectories to this directory, you enter a value in the **Base Directory** field in the **Destination Configuration** form.)

> **Note:**
>
> Before you run an export publishing session, ensure this root export directory exists and that it has enough space for the HTML pages that will be created there.

To set the root directory for the exported files:

1. In the Admin interface, open the Property Management Tool. (For instructions on accessing the Property Management Tool, see Overview of the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Name** field, enter `cs.pgexportfolder`.
3. Click **Search**.

4. In the Properties section of the form, select the name of the property. 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 Configuring an Export Destination).

   For more information about export directories, see Working with the Path Naming Convention.

5. Click **Save**.
6. Restart the application server.

Configuring an Export Destination

To create an export destination:

1. Log in to the Admin interface on the source WebCenter Sites system.
2. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, expand the **Destinations** node, and then double-click **Add New**.

   The Add New Destination form opens.

3. Fill in the fields on the **Add New Destination** form as follows:

   - In the **Name** field, enter a unique name for the destination.

     > **Note:**
     >
     > The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), greater-than sign (>), percent (%), and question mark (?). Additionally, the name cannot end with a backslash (\).

   - 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. The names of the generated files do not include this prefix , only the links within the files use it.

  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 are relative.

  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. See Working with File Naming Conventions.

  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 multiple templates.

- **Use simple directory naming**: When selected, this parameter overrides the normal directory-naming convention. See Working with the 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.) 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 WebCenter Sites versions before 3.6 and it has not yet been redesigned for the 3.6 rendering model, you must select this parameter.

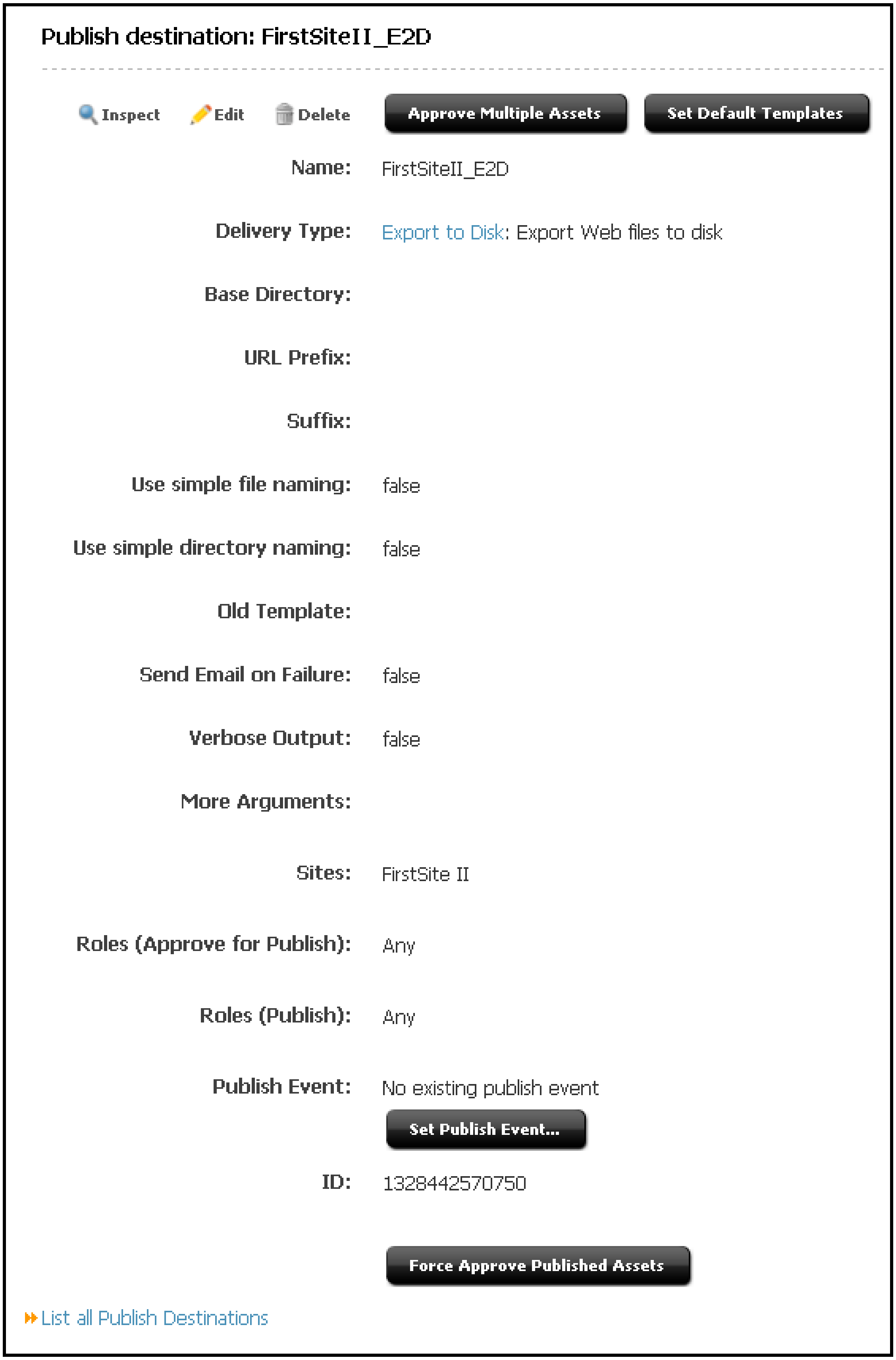
  If you select this parameter, do not set an export starting point for your site. The export starting point is determined by the templates themselves.

- **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 shown only when **Send Email on Failure** is selected). All email addresses that will receive an email on failure are shown.
- **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 can 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.

4. Click **Add New Destination**.

   The **Publish Destination** form opens. The new destination is shown on the form.

Figure 42-10 Publish Destination Form



Publish destination: FirstSiteII_E2D

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
ID:	1328442570750

5. Repeat steps 2–4 for each additional export destination.

Mapping 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 resolve.

See 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.

Creating the Export Starting Points

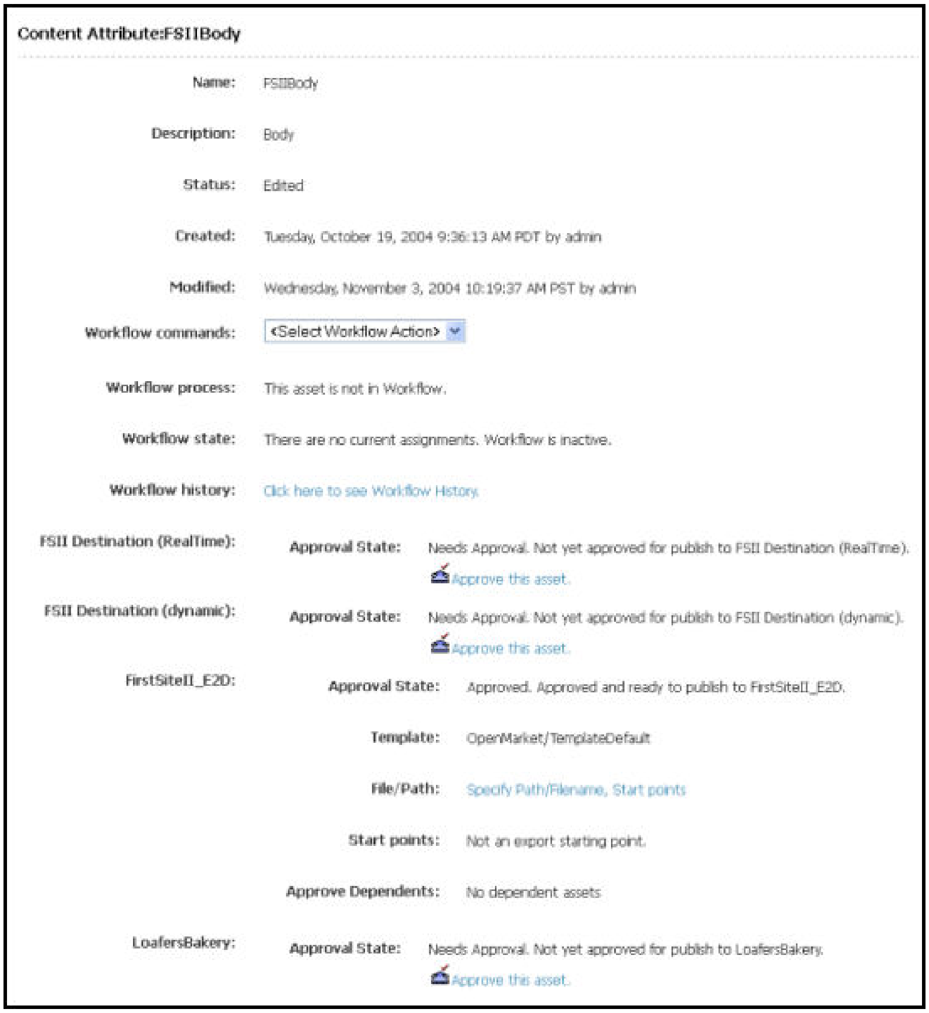
To create an export starting point:

1. Using the Admin interface on the source system, find the asset to specify as an export starting point.
2. Open the asset in its **Status** form and scroll to the Publishing Destination section.
3. Complete either:
   - 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 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**.

   calculates dependencies for all the assets linked to this asset and shows the results.
5. When the approval results are shown, 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 shows the selected destination.
7. In the **File/Path** field, click **Specify Path/Filename, Start points**.

   The **File** and **Path** fields are made available on the **Content Attribute** form.

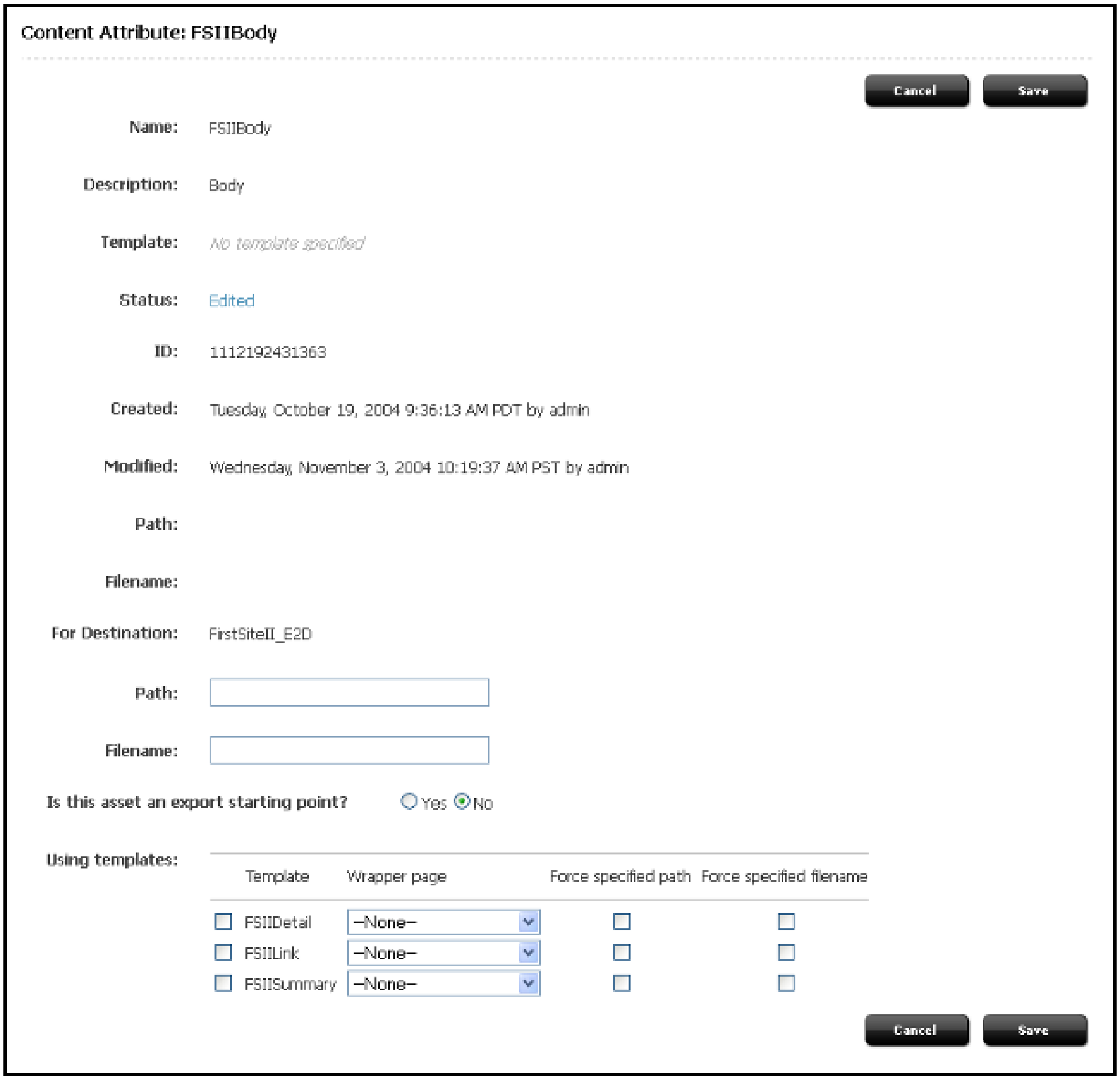
Figure 42-11   Content Attribute Form



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:	<Select Workflow Action>	
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). Approve this asset.
FSII Destination (dynamic):	Approval State:	Needs Approval. Not yet approved for publish to FSII Destination (dynamic). Approve this asset.
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. Approve this asset.

Figure 42-12 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
□ FSIIDetail	--None--	□	□
□ FSIILink	--None--	□	□
□ FSIISummary	--None--	□	□

> **Note:**
>
> If the starting point selected is a page, no templates are found, and then clicking **Save** results in the error "To make this asset a publish starting point, you must select a template," the `enableAssetForms` property must be set to `true`.
>
> From the Property Management Tool, set `advancedUI.enableAssetForms=true` then set the page as the starting point in the Content Attribute form.

8. In the **Content Attribute** form, fill in the fields:

   a. (Optional) **For Destination** section:

Path: 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.

Filename: 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:

Select a template.

Select a wrapper page.

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.

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.

Approving Your Assets for Export to Disk Publishing

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.

To complete a test publish of your entire site, approve all the assets for the site. See Approving Multiple Assets.

Publishing and Testing the Results

When the assets you want to publish to the selected destination are approved, run a test publishing session:

1. In the Admin 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**.

   shows 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 opens. Click **Yes** to proceed with publishing.

   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 shows 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.

   The page opens 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 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.

Setting Up the Schedule

After you have ensured that:

- Your destination is configured correctly,
- The file names and directory names are created according to your requirements, 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:

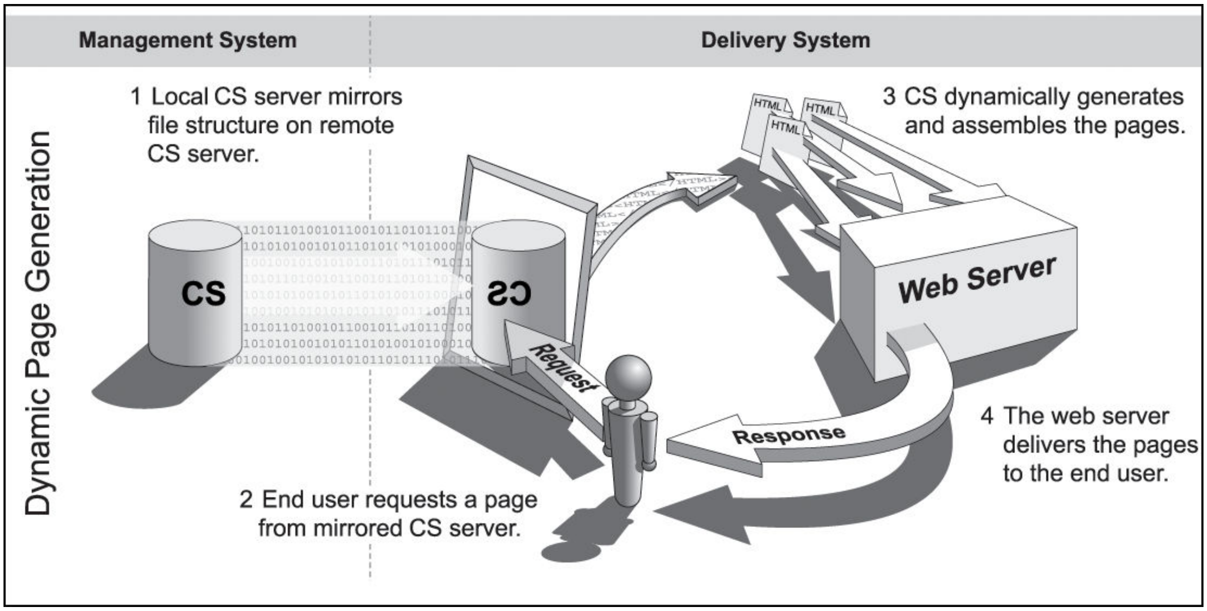
1. Create scheduled publish events for the destination. For help with this step, see Scheduling a Publish Event.

2. 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 by FTP to a testing area after a publishing session completes.

Understanding Mirror to Server Publishing

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 calls the CacheManager servlet to refresh any pages that should be regenerated to take advantage of the new content.

Figure 42-13   Mirror to Server Publishing

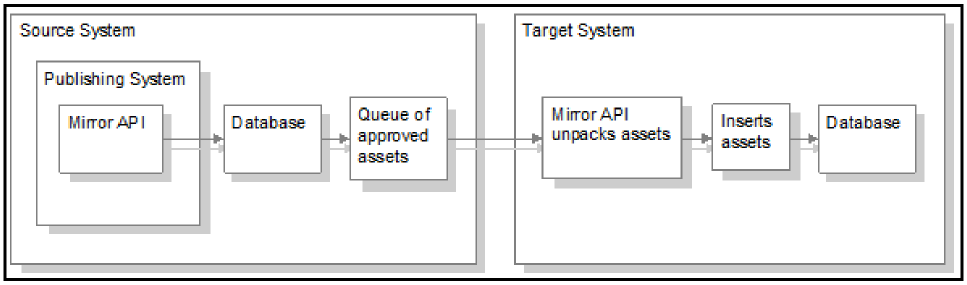

Management System
Delivery System
Dynamic Page Generation
1 Local CS server mirrors file structure on remote CS server.
CS
CS
HTML
HTML
HTML
3 CS dynamically generates and assembles the pages.
Web Server
Request
Response
2 End user requests a page from mirrored CS server.
4 The web server delivers the pages to the end user.

> **Note:**
>
> Export to Disk publishing is not recommended. Instead, use the Site Capture application of WebCenter Sites.

Whether your WebCenter 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 Troubleshooting.

When a Mirror to Server publishing session runs, the followingfigure illustrates what occurs.

Figure 42-14   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 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 must. For details, see Turning Off Asset Invalidation on the Delivery System.
   - 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 WebCenter 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.

Pre-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 provide will help to ensure a smooth configuration process.

- Working with Users and Mirror to Server Publishing
- Working with CacheManager
- Configuring a Mirror Destination
- Determining When to Use Mirror Destination Configuration

Working with 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.

Working with CacheManager

The CacheManager is a servlet that maintains the page cache on any dynamic system, including the management system.

CacheManager is important because it locks the appropriate assets on the destination during a mirror publishing session and ensures the integrity of the page cache both before and after a mirror publishing session.

See About the CacheManager in *Developing with Oracle WebCenter Sites.*

Configuring a Mirror Destination

The Mirror to Server publishing method copies information for approved assets from one 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) is 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 database to another.

Determining When to Use Mirror Destination Configuration

You use Mirror Destination Configuration in several situations:

- To set up a 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 must troubleshoot your configuration. If you can successfully initialize a mirror destination, that means that the source and the destination systems are communicating.

Configuring Your System for Mirror to Server Publishing

The main steps when configuring a system for Mirror to Server publishing are as follows:

- Creating the Batch User Account for Mirror Publishing (If One Does Not Exist)
- Setting Up the Destination System
- Identifying the Mirror User to the Source System
- Identifying the Local Proxy Server to the Source System (If One Exists)
- Creating a Mirror Destination
- Initializing the Destination
- Configuring the Mirror Destination

- Approving Your Assets
- Publishing the Assets
- Testing the Results
- Setting Up the Schedule
- Turning 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 to gather and confirm the information they prompt you to provide.

Creating the Batch User Account for Mirror Publishing (If One Does Not Exist)

This procedure must be performed only one time on each source system, regardless of the number and types of publishing destinations you are configuring.

If you have not done so, create the batch user account. For instructions, see Creating the Batch User Account (If One Does Not Exist).

If a batch user account exists on your source system, skip to Setting Up the Destination System.

Setting Up the Destination System

To mirror publish assets from one WebCenter 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 from the Property Management Tool). Be sure to restart the application server if you make changes to database properties.

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.

> **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` 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 require help with creating the mirror user, see Creating a User in the Admin Interface.

Identifying 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 `wcs_properties.json` 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 procedure steps to ensure they are set correctly.

To identify the mirror user to the source system:

1. In the Admin interface, open the Property Management Tool. (For instructions on accessing the Property Management Tool, see Accessing the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Category** drop-down menu, select **Publish**.
3. Click **Search**.
4. Set values for and save 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.
5. Complete either:
   - If there is a firewall separating your source system from the destination system, go to Identifying the Local Proxy Server to the Source System (If One Exists).
   - If there is no firewall separating your source and destination systems, stop and restart the application server. Then go to Creating a Mirror Destination.

Identifying the Local Proxy Server to the Source System (If One Exists)

Note:

This procedure must be performed only one time on each source system, regardless of the number and types of publishing destinations you are configuring. Skip to Creating a Mirror Destination if the local proxy has been identified to your source system, or a proxy server is not used.

To identify the local proxy to the source system:

1. In the Admin interface, open the Property Management Tool. (For instructions on accessing the Property Management Tool, see Accessing the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.
2. In the **Category** drop-down menu, select **Publish**.
3. Click **Search**.
4. Set values for and save 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.

5. Restart the application server.

Creating a Mirror Destination

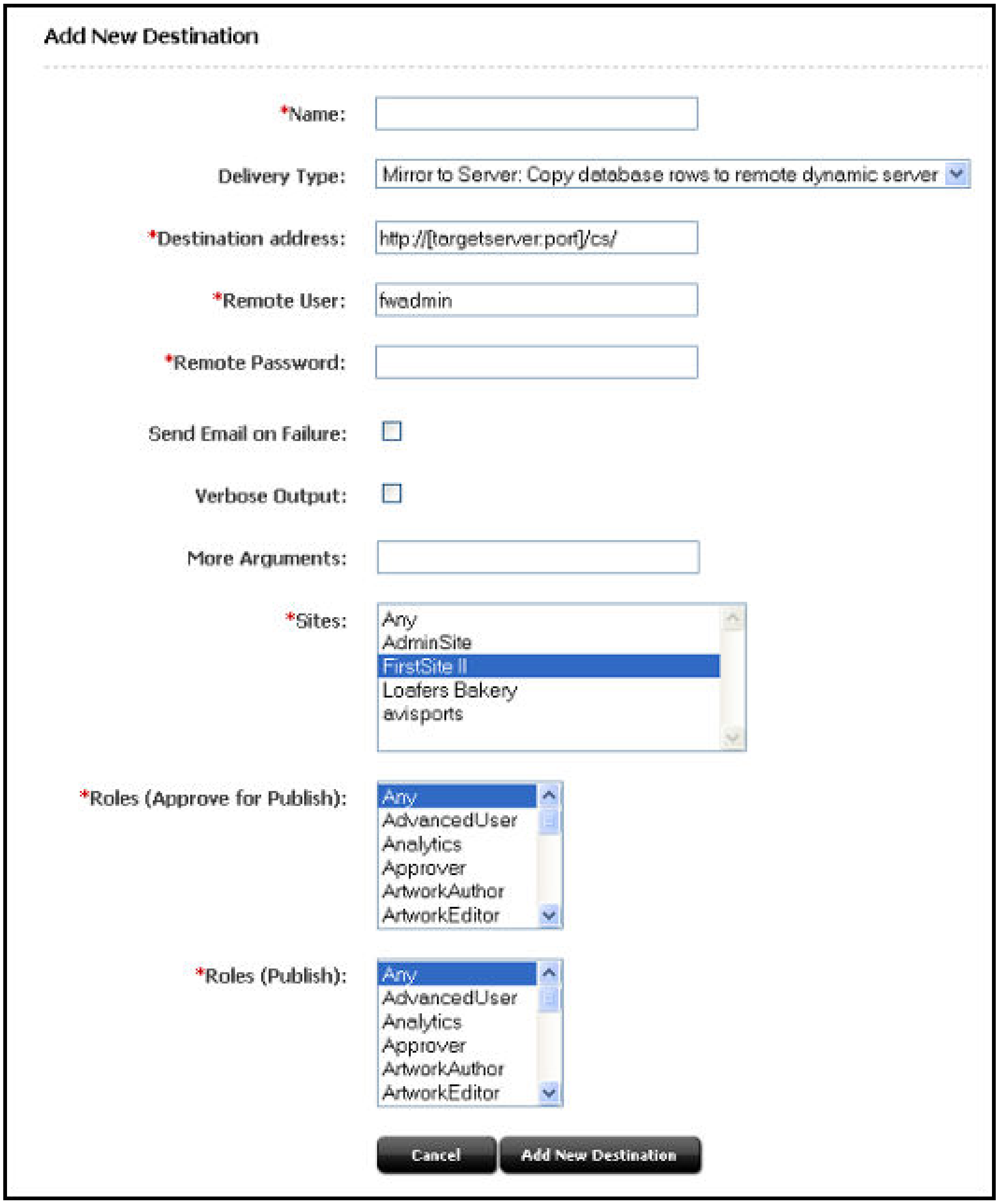
A mirror destination must be configured, just as any other part of the process of publishing.

To create a mirror destination:

1. Log in to the Admin interface on the source WebCenter Sites system.
2. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
3. Under the **Destinations** node, double-click **Add New**.

   The Add New Destination form opens.

Figure 42-15   Add New Destination Form



Add New Destination

*Name:

Delivery Type: Mirror to Server: Copy database rows to remote dynamic server

*Destination address: http://[targetserver:port]/cs/

*Remote User: fwadmin

*Remote Password:

Send Email on Failure: □

Verbose Output: □

More Arguments:

*Sites: Any, AdminSite, FirstSite II (selected), Loafers Bakery, avisports

*Roles (Approve for Publish): Any (selected), AdvancedUser, Analytics, Approver, ArtworkAuthor, ArtworkEditor

*Roles (Publish): Any (selected), AdvancedUser, Analytics, Approver, ArtworkAuthor, ArtworkEditor

4. Configure the fields on this form as follows:

   - **Name**: Enter a unique name for the destination.

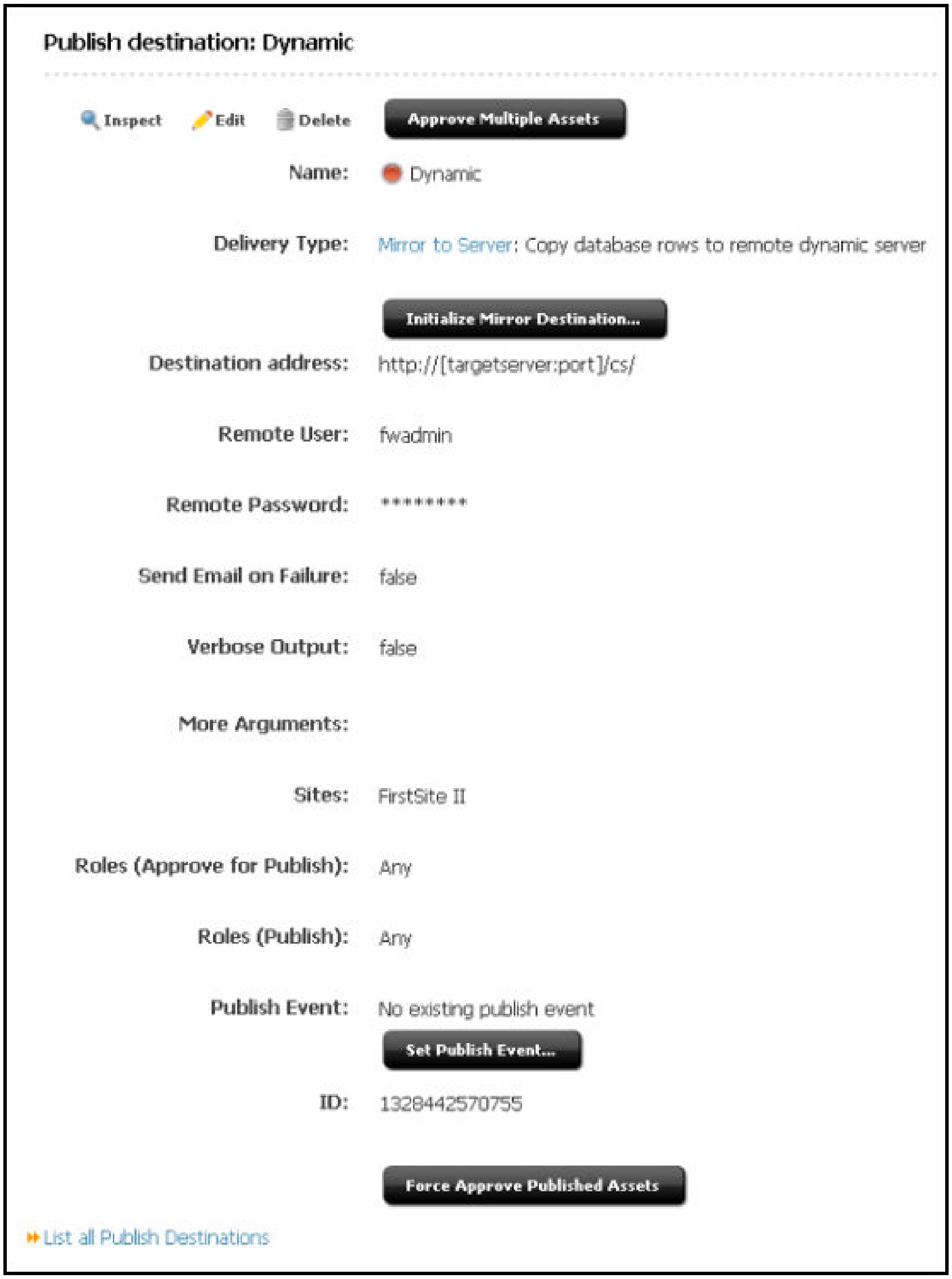
     > **Note:**
     >
     > The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), greater-than sign (>), percent (%), and question mark (?). Additionally, the name cannot end with a backslash (\).

- **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 host name or IP address of the target system and the port to be used by the destination. A slash is required after the URL because this URL is appended dynamically.
- **Remote User**: Enter the name of the mirror user created in Setting Up the Destination System. This user is called 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 check box.
  - **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 can 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 are 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 opens the Publish Destination form.

Figure 42-16 Publish Destination Form


Inspect
Edit
Delete
Approve Multiple Assets
Initialize Mirror Destination...
Set Publish Event...
Force Approve Published Assets
List all Publish Destinations

Publish destination: Dynamic

Name:	Dynamic
Delivery Type:	Mirror to Server: Copy database rows to remote dynamic server
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
ID:	1328442570755

6. 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 Initializing the Destination for detailed instructions (begin at step 4 of that procedure).

   If you must create more mirror destinations, repeat steps 1 through 5 for each additional mirror destination that you must create for your source system.

Initializing the Destination

You must initialize the destination system before you can publish to it. This creates the basic site information about 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 **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
2. Under the **Destinations** node, double-click the destination you want to initialize.
3. In the Publish Destination form, click **Initialize Mirror Destination**.

   The Initialize Mirror Destination form opens.
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 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 must 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 through 6 for the remaining tables.
6. Click **Mirror**.

   If the initialization is successful, displays a confirmation message; otherwise, an error message opens.

   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**.
>
> a. In the **Publish Console**, select your mirror destination from the drop-down list and then click **Select Destination**.
>
> b. Click the **Create Site on Target** button. When the confirmation form opens, click **Yes**. This creates the basic site information about the destination system.

Configuring the Mirror Destination

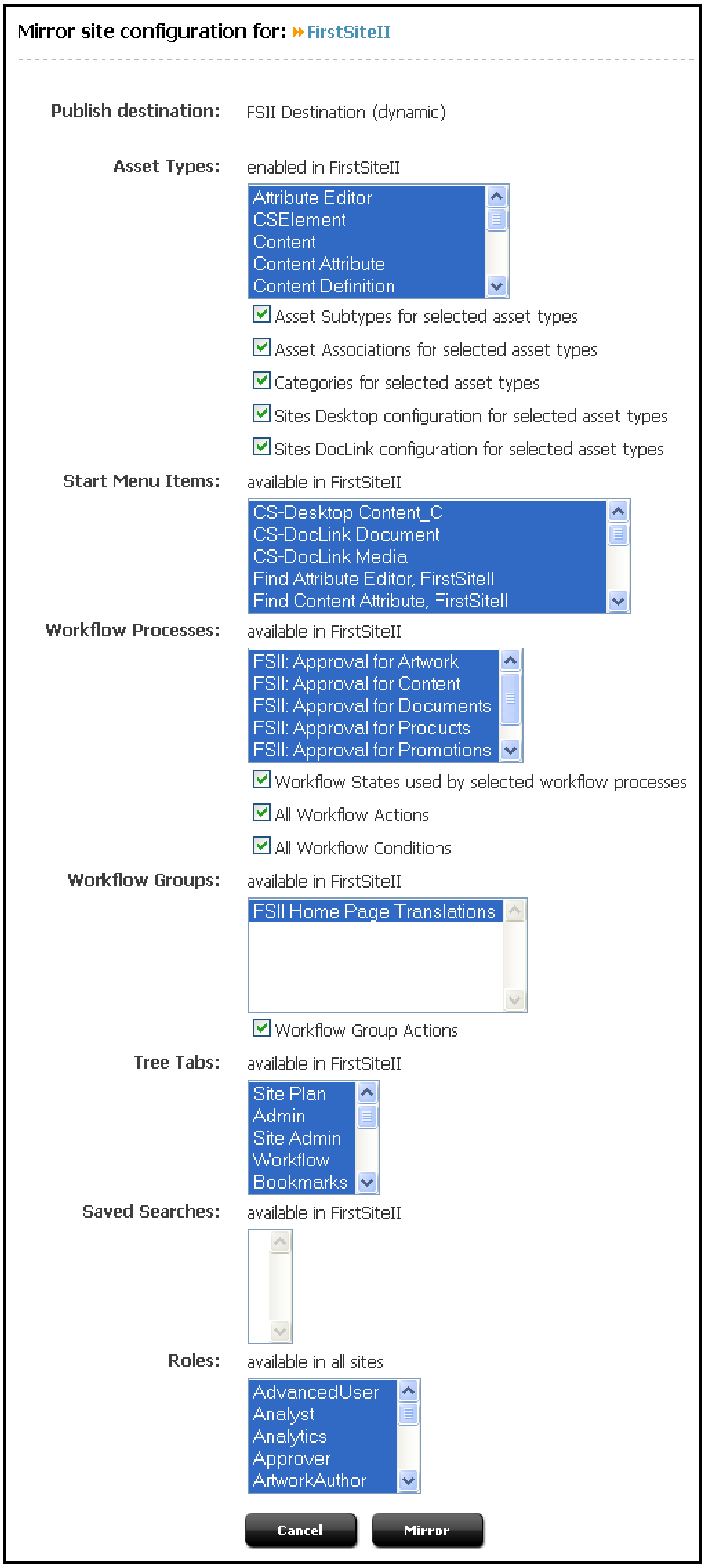
Now you must configure the data that will be on your target destination. This configuration will vary depending on the purpose of the target destination. For example, if the target destination is strictly a delivery computer, it makes sense to only include asset types when mirroring. Before proceeding with configuring the destination, you may find it helpful to see 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 **General Admin** tree, expand the **Admin** node, expand the **Sites** node, and then double-click the site whose data you want to publish on the target destination.
2. In the **Publish Destinations** field (near the bottom of the form), click **Mirror site configuration for** *destination* (dynamic), where *destination* is the target destination.

   The Mirror Site Configuration form opens.

Figure 42-17   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 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 is mirrored only if you select **Asset associations for selected asset types** and then the relevant asset types.
- **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 to make available on the target destination.

  > **Note:**
  >
  > If any of the following tree tabs 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 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 through 3 for each site whose assets you want to publish on the target destination.

Approving Your Assets

To test your published site, you must approve and publish all the assets for the site. See Approving Multiple Assets for help with approving many assets at one time.

To perform a simple test of your configuration, temporarily create a home page asset with only a few dependents.

Publishing 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 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 about the destination system.
4. Click **Publish**.

   A confirmation message opens. 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 about configuring the logs to view destination information, see Retrieving Logs From Delivery WebCenter Sites Systems.

Testing 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 done so, you must determine the asset's URL. A URL is constructed by concatenating the following values:

- The host name 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 `wcs_properties.json` 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 page criteria variables, `c`, `cid`, `tid`, and `p`. (For information about these variables, see Template, CSElement, and SiteEntry Assets in the *Developing with Oracle WebCenter Sites*.)

Complete the following steps to determine the URL of your home page and test the site:

1. In the Admin interface, open the Property Management Tool. (For instructions on accessing the Property Management Tool, see Accessing the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Category** drop-down menu, select **Core**.
3. Click **Search**.
4. Write down the values of the following properties:
   - `ft.cgipath`
   - `ft.approot`
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:

   Example for Oracle WebLogic Server and IBM WebSphere:

   ```
   http://bigfatsun.example.com:8080/servlet/
   ```
6. After 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 Oracle WebLogic Server and IBM 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.

Setting 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:

1. Create scheduled publish events for the destination. For help with this step, see Troubleshooting.
2. 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.
3. 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 CatalogMover in *Developing with Oracle WebCenter Sites*.

Turning Off Asset Invalidation on the Delivery System

By default, the publishing system is configured to select 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, it is not necessary 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. In the Admin interface, open the Property Management Tool. (For instructions about accessing the Property Management Tool, see the Accessing the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Name** field, enter `xcelerate.publishinvalidate`.
3. Click **Search**.
4. Select the name of the property. In the **Value** field, set its value to `false`.
5. Click **Save**.

Retrieving Logs From Delivery WebCenter Sites Systems

provides a way to retrieve publishing-related logging information from the delivery system and display it in the Publishing console. To achieve this, 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.ContextAwar
   eLogFactory
   ```

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 property (located under the config folder):

```
log.file.location=<path to Sites log file>
```

Exporting 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 WebCenter Sites delivery system. In contrast, Export Assets to XML creates XML files for delivery to a database or non-WebCenter Sites system.

When an Export Assets to XML publishing session runs, this is what happens:

- On the source system, the approval system provides a list of the assets approved for this destination to Export Assets to XML.
- 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-WebCenter Sites) system.

The following topics provide information about exporting assets to XML:

- Understanding the XML Output
- Understanding File Naming Conventions for Export Assets to XML

Understanding 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 | ASSETREFERNCE)>
<!ATTLIST COLUMN NAME CDATA #REQUIRED>
```

Understanding 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.

Configuring Your System for Export Assets to XML

The main steps when configuring for the Export Assets to XML delivery type are as follows:

- Creating the Batch User Account for XML Publishing (If One Does Not Exist)
- Specifying the Root Export Directory
- Configuring an Export Assets to XML Destination
- Approving Your Assets for XML Publishing
- Publishing and Testing the Results
- Setting Up the Schedule

Creating the Batch User Account for XML Publishing (If One Does Not Exist)

Creating a batch user account is required before any type of publishing can take place. This procedure must be performed only one time, regardless of the number and types of publishing destinations you are creating.

If you have not done so, create the batch user account. For instructions, see Creating the Batch User Account (If One Does Not Exist).

If a batch user account exists for your installation, skip to Specifying the Root Export Directory.

Specifying the Root Export Directory

As mentioned earlier in this chapter, the directory to which exports files is determined by the `cs.pgexportfolder` property in the `wcs_properties.json` file.

To set the root directory for exported files:

1. In the Admin interface, open the Property Management Tool. (For instructions, see Accessing the Property Management Tool in the *Property Files Reference for Oracle WebCenter Sites*.)
2. In the **Name** field, enter `cs.pgexportfolder`.
3. Click **Search**.

4. Select the name of the property. In the **Value** field, set its value to the full path of the directory to which files should be exported. This is a global setting for the system.
5. Click **Save**.
6. Restart the application server.

> **Note:**
>
> Before you run an Export to XML publishing session, ensure this destination directory exists and that it has enough space for the XML files that will be created there.

Configuring an Export Assets to XML Destination

Next, create the Export to XML destination by completing the following steps:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing**node, and then expand the **Destinations** node.
2. Double-click **Add New**.

   The Add New Destination form opens.

   Fill in the fields as described:

   - **Name:** Enter a unique name for the destination.

     > **Note:**
     >
     > The following characters are not allowed in the **Name** field: single quote ('), double quote ("), semicolon (;), colon (:), less-than sign (<), greater-than sign (>), percent (%), and question mark (?). Additionally, the name cannot end with a backslash (\).

   - **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 must configure for your source system.

Approving Your Assets for XML Publishing

To perform a simple test of your configuration, select an asset and approve it and any of its dependent assets.

To complete a test publish of your entire site, approve all the assets for the site. See Approving Multiple Assets for help with approving many assets at one time.

Publishing and Testing 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 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**.

   displays information about the assets that are ready to be published to this destination.
3. Click **Publish**.

   A confirmation message opens. Click **Yes** to proceed with the publish.

   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.

Setting Up the Schedule

After you have ensured that your destination is configured correctly, that the directory names are created according to your requirements, you can finish configuring your publishing system by completing the following steps on the source system:

1. Create scheduled publish events for the destination. For help with this step, see Troubleshooting.
2. Plan how you will move the generated files to your external, non-WebCenter 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.

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 the sites' supporting configuration (subtypes, associations, start menu items, and so on), code templates. They also assist the administrators in customizing the user interface, when necessary, by writing elements for workflow and so on. This work is done 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 both the management and the delivery systems.

This section covers the following topics:

- Moving a Site from a Development System to a Management System
- Moving a Site to a Delivery System

Moving a Site from a Development System to a Management System

The first part of moving a site from development to management requires setting up the support tables, the users, and if used, managing the flex family on the management system. This portion of the process of moving should only be done once.

Complete the following procedure only one time:

1. Log in to the Admin 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 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 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 Approving Multiple Assets.
10. Approve and then publish all of the rest of the appropriate assets for the destination system: template, page, collection, query, 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 CatalogMover in *Developing with Oracle WebCenter Sites.*

Moving a Site to a Delivery System

When you move a site to a delivery system, it is not necessary to mirror start menu items, workflow, saved searches, and so on.

To move a site to a delivery system:

1. Log in to the Admin 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 Creating a Mirror Destination.
5. Initialize the mirror destination.

   For help with this step, see Initializing the Destination.
6. Configure the mirror destination.

   For help with this step, see Configuring 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 in the **General Admin** tree. For information about this feature, see 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 CatalogMover in *Developing with Oracle WebCenter Sites.*
10. If there are any images in the online site that are not assets, be sure to copy them to the delivery system.

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.

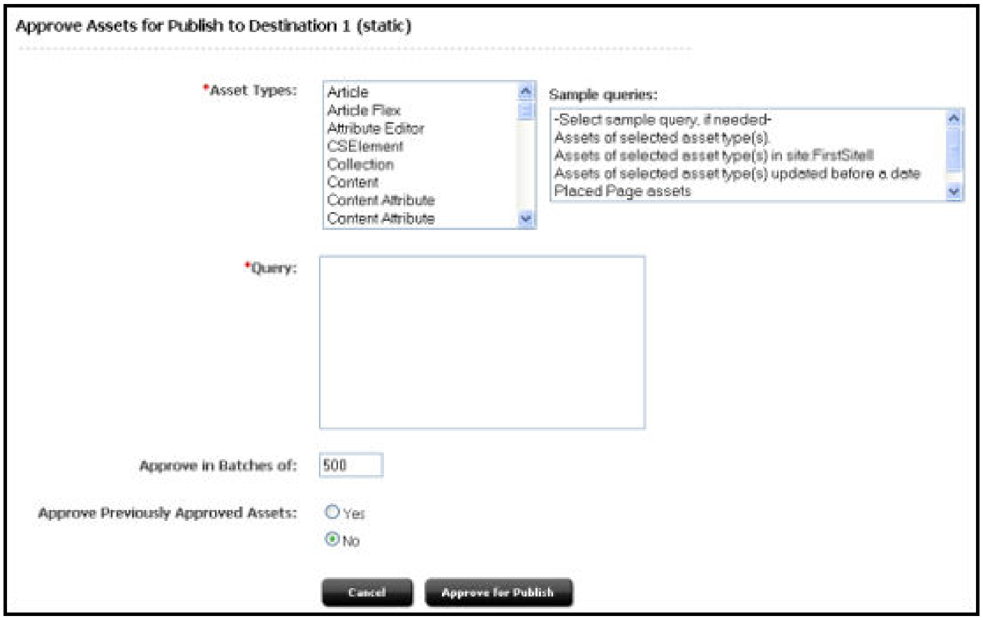
Note that the Approve Multiple Assets feature is not the BulkApprover utility. The BulkApprover utility approves only those assets that have been imported into the WebCenter Sites database with the BulkLoader utility. Both BulkLoader and BulkApprover are described in About the BulkLoader Utility and Approving Flex Assets with the BulkApprover Utility in *Developing with Oracle WebCenter Sites.*

To approve a group of assets:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, expand the **Destinations** node, and then expand the destination to approve assets for.
2. Under that destination, double-click the **Approve Multiple Assets** option.

   The Approve Assets form opens.

Figure 42-18 Approve Assets for Publish to Destination Form



Approve Assets for Publish to Destination 1 (static)

*Asset Types: Article, Article Flex, Attribute Editor, CSElement, Collection, Content, Content Attribute, Content Attribute

Sample queries:
-Select sample query, if needed-
Assets of selected asset type(s).
Assets of selected asset type(s) in site FirstSiteII
Assets of selected asset type(s) updated before a date
Placed Page assets

*Query:

Approve in Batches of: 500

Approve Previously Approved Assets: ○ Yes ● No

3. In the Asset Types section, from the list on the left, select the asset types to approve.
4. From the **Sample Queries** list on the right, select a query. If the exact query that you require is not present, select the query that is most like the one wanted.

   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 `wcs_properties.json` file, which sets the browser session timeout value. Start by setting this value to 1800 (which means 30 minutes). (See Properties in the Core Category in the *Property Files Reference for Oracle WebCenter Sites*.)
> - 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 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**.

Creating Destinations

The procedures for creating destinations are described in the previous section. For information about creating new destinations, use a procedure in the following list, as appropriate:

- To create an export destination, see Configuring an Export Destination.
- To create a mirror destination, see Creating a Mirror Destination.
- To create an export assets destination, see Configuring an Export Assets to XML Destination.

Editing Destinations

In some cases, after a publishing destination is created, it will need to be modified. The basic procedure for editing all types of publishing destinations is covered here.

To edit a destination:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
2. Double-click the destination to edit.

   The Publish Destination form opens for the selected destination.
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 Configuring an Export Destination.
- For a Mirror to Server destination, see Creating a Mirror Destination.
- For an Export Assets to XML destination, see Configuring an Export Assets to XML Destination.

Deleting Destinations

Destinations are easily deleted once they are no longer required for publication.

To delete a destination:

1. In the **General Admin** tree, expand the **Admin** node, expand the **Publishing** node, and then expand the **Destinations** node.
2. Double-click the destination to delete.

3. Click the **Delete** icon.

   displays a verification message.

4. Click **Delete Destination**.

Creating Export Starting Points

Export starting points are defined in Working with Export Starting Points. For information about creating them, see Creating the Export Starting Points.