

Content Server

Version: 6.3

Administrator's Guide

Document Revision Date: Dec. 1, 2005

FatWire[®]
S O F T W A R E

FATWIRE CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event shall FatWire be liable for any loss of profits, loss of business, loss of use of data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind, even if FatWire has been advised of the possibility of such damages arising from this publication. FatWire may revise this publication from time to time without notice. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Copyright © 2005 FatWire Corporation. All rights reserved.

This product may be covered under one or more of the following U.S. patents: 4477698, 4540855, 4720853, 4742538, 4742539, 4782510, 4797911, 4894857, 5070525, RE36416, 5309505, 5511112, 5581602, 5594791, 5675637, 5708780, 5715314, 5724424, 5812776, 5828731, 5909492, 5924090, 5963635, 6012071, 6049785, 6055522, 6118763, 6195649, 6199051, 6205437, 6212634, 6279112 and 6314089. Additional patents pending.

FatWire, Content Server, Content Server Bridge Enterprise, Content Server Bridge XML, Content Server COM Interfaces, Content Server Desktop, Content Server Direct, Content Server Direct Advantage, Content Server DocLink, Content Server Engage, Content Server InSite Editor, Content Server Satellite, and Transact are trademarks or registered trademarks of FatWire, Inc. in the United States and other countries.

iPlanet, Java, J2EE, Solaris, Sun, and other Sun products referenced herein are trademarks or registered trademarks of Sun Microsystems, Inc. *AIX, IBM, WebSphere*, and other IBM products referenced herein are trademarks or registered trademarks of IBM Corporation. *WebLogic* is a registered trademark of BEA Systems, Inc. *Microsoft, Windows* and other Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation. *UNIX* is a registered trademark of The Open Group. Any other trademarks and product names used herein may be the trademarks of their respective owners.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>) and software developed by Sun Microsystems, Inc. This product contains encryption technology from Phaos Technology Corporation.

You may not download or otherwise export or reexport this Program, its Documentation, or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations, including without limitation the United States Export Administration Act, the Trading with the Enemy Act, the International Emergency Economic Powers Act and any regulations thereunder. Any transfer of technical data outside the United States by any means, including the Internet, is an export control requirement under U.S. law. In particular, but without limitation, none of the Program, its Documentation, or underlying information of technology may be downloaded or otherwise exported or reexported (i) into (or to a national or resident, wherever located, of) Cuba, Libya, North Korea, Iran, Iraq, Sudan, Syria, or any other country to which the U.S. prohibits exports of goods or technical data; or (ii) to anyone on the U.S. Treasury Department's Specially Designated Nationals List or the Table of Denial Orders issued by the Department of Commerce. By downloading or using the Program or its Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list or table. In addition, if the Program or Documentation is identified as Domestic Only or Not-for-Export (for example, on the box, media, in the installation process, during the download process, or in the Documentation), then except for export to Canada for use in Canada by Canadian citizens, the Program, Documentation, and any underlying information or technology may not be exported outside the United States or to any foreign entity or "foreign person" as defined by U.S. Government regulations, including without limitation, anyone who is not a citizen, national, or lawful permanent resident of the United States. By using this Program and Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not a "foreign person" or under the control of a "foreign person."

Content Server Administrator's Guide

Document Revision Date: Dec. 1, 2005

Product Version: 6.3

FatWire Technical Support

www.fatwire.com/Support

FatWire Headquarters

FatWire Corporation
330 Old Country Road
Suite 207
Mineola, NY 11501
www.fatwire.com

Table of

Contents

About This Guide	13
Who Should Use This Guide	13
How This Guide Is Organized	13
Related Publications	14

Part 1. Introduction

1 Overview	17
Introduction	18
Online Sites	18
CM Sites	21
Configuration Components	22
Summary	27
Content Management Models	28
1:1 Model	28
1:n Model	28
x:n Model	29
Content Server Environment	30
Administrator's Job	31
Collaborating	31
Planning	32
Development	33
Implementation	35
The Focus of This Guide	36
Administrator's Prerequisites	36
2 Administrator's Interface	37
Types of CS Administrators	38
Logging in to the Administrator's Interface	39

Administrator Tabs	42
'Admin' Tab	43
'Site Admin' Tab	45
'Workflow' Tab	45
Non-Administrative Tabs	46
System Defaults	46
Sample Sites	46
3 Site Configuration Guidelines.....	49
Overview	50
Site Replication Option	53
Site Configuration Steps	53
Assumptions	53
Pre-Configuration Decisions	53
Configuration Steps at the Development System	54
Configuration Steps at the Management System.....	55

Part 2. CM Site Configuration Procedures

4 Working with ACLs and Roles	61
Overview	62
ACLs	62
System ACLs.....	64
Sample ACLs.....	64
Custom ACLs	64
Working with ACLs	65
Creating a New ACL	65
Editing a Custom ACL	66
Deleting a Custom ACL	66
Assigning ACLs to Custom Tables	67
Assigning ACLs to Content Server Pages (SiteCatalog Page Entries)	68
Setting the ACL Restriction Error Message	68
Roles	69
System Roles	69
Sample Roles	69
Custom Roles.....	70
Working with Roles.....	70
Creating a Role	70
Editing a Role	71
Deleting a Role	71
5 Configuring Users, Profiles, and Attributes	73
Overview	74

User Management Options	74
Native Content Server User Manager	75
LDAP Plug-In	75
Configuring Users in Content Server	75
Working with User Accounts	76
Creating a New User	77
Editing a User	78
Deleting a User from the System	79
Working with User Profiles and User Attributes	80
Creating and Editing a User Profile	80
Deleting a User Profile	81
Modifying, Adding, and Deleting User Attributes	82
6 Setting Up External Security.....	83
Overview	84
ACLs and Security.....	84
DefaultReader, secure.CatalogManager, and secure.TreeManager.....	85
BlobServer and Security	85
Security Goals	85
Implementing Security	86
Properties That Configure Security Settings	86
Users and Passwords	87
URLs and the Web Server (Delivery System Only)	88
CS Forms and Pages (Delivery System Only).....	88
Testing Security.....	89
Security Tests for All Systems	89
Additional Security Tests for the Delivery System.....	90
7 Assembling and Organizing CM Sites	91
Overview	92
Working with Sites	92
Creating a Site	92
Obtaining Site Configuration Information.....	94
Editing a Site	95
Deleting a Site	95
Managing Site Users	97
Granting Users Access to a Site (Assigning Roles to Users)	97
Viewing Site Users and Changing Role Assignments.....	98
Deleting Users from a Site.....	99
Managing Asset Types	100
Enabling Asset Types for a Site.....	100
Removing Asset Types from a Site.....	102
Enabling and Managing User Interfaces	103

8 Managing Access to CM Site Components	105
Overview	106
Start Menu Items	106
Tree Tabs	107
Managing Access to Asset Types	108
Creating Start Menu Items	108
Setting Access Permissions on Assets	114
Permission Structure	114
Types of Authorization	115
Summary	116
Setting Access Permissions	117
Configuring Other Options for Asset Types	120
Managing Access to Content Server's Tree	121
Displaying and Hiding the Tree	121
Creating Tree Tabs	122
Editing Tree Tabs	124
Deleting Tree Tabs	125
Changing the Sort Order of the Tree Tabs	125
Configuring the Number of Items Displayed Under Nodes on Tabs	126
9 Creating and Managing Workflow Processes	127
Overview	128
Workflow Participants	128
Workflow States	128
Workflow Steps: Moving Assets from State to State	129
Multiple Paths for the Asset	130
Managing Deadlocks	130
Workflow Groups	132
Delegating and Clearing Assignments	132
Placing an Asset in Workflow	133
Restricting Access to Assets While They Are in Workflow	133
Deadlines	134
Scheduling a Deadline Calculation	135
How Does a Workflow Process End?	136
Roles Required to Configure Workflow Processes	136
Planning Your Workflow Processes	137
Start with a Sketch	137
Determine Roles and Participants	137
Determine the E-mail Objects, Actions, and Conditions	139
Determine the States	145
Determine the Steps	146
Determine the Function Privileges	148
Determine Additional Workflow Process Details	152
Sample Site Workflow Processes	152
HelloArticle Process	153

BF: Normal Article Process	154
GE Lighting Process	157
Configuring Your Workflow Processes	159
Overview	159
Setting Up E-Mail Objects	160
Setting Up the Workflow Actions and Conditions	162
Setting Up the Timed Action Event	164
Setting Up the States	166
Setting Up the Workflow Processes	168
Testing Your Workflow Process	177
Moving Your Work	178
Clearing Workflow Assignments	178
10 Replicating CM Sites	181
Site Launcher Overview	182
Preparing for Replication	182
Ensuring the Source Site Meets Replication Requirements	182
Planning the New Site	183
Site Replication Steps	186
Post-Replication Tasks and Guidelines	188

Part 3. System Configuration Procedures

11 Configuring the User Interfaces	193
Setting the Locale for the Content Server Interface	194
System Default Locale for the CS Users	194
System Locale Properties	195
Single-Language Restrictions	196
Locale and Asset Types	196
Locale and the Article Asset Type	197
CS-Desktop	198
Overview	198
Configuring the Word Asset Types for CS-Desktop	200
User Accounts and CS-Desktop	203
Installing the CS-Desktop Client Application	204
Specifying Locale for the Client Application	204
Testing the CS-Desktop Configuration	205
Configuring Word Templates for the Word Assets	205
CS-DocLink	206
Overview	206
Configuring the CS-DocLink Asset Types	206
Users and CS-DocLink	211
Installing the CS-DocLink Client Application	212

Testing the CS-DocLink Configuration	212
InSite Editor	213
Enabling the InSite Editor	214
User Accounts and the InSite Editor	214
eWebEditPro	215
Configuring eWebEditPro	215
User Accounts and eWebEditPro	215
Maintaining Separate Browser Sessions for Preview	215
12 Managing the CS Publishing System	217
Overview	218
Delivery Types	219
Publishing Destinations	219
Publishing and the Approval Process	220
The Publishing Schedule (Publish Events)	225
What Happens During the Publishing Session?	227
Obtaining Information About a Publishing Session	227
The Export to Disk Delivery Type	228
How Export to Disk Works	228
Publishing Arguments for Export to Disk	230
File and Directory Naming Conventions	232
Export Starting Points	237
The Mirror to Server Delivery Type	238
How Mirror to Server Works	238
Users and Mirror to Server	240
Publishing Arguments for Mirror to Server	240
CacheManager	241
Configuring a Mirror Destination	241
The Export Assets to XML Delivery Type	242
The XML Output	242
Publishing Arguments for Export Assets to XML	244
File Naming Conventions for Export Assets to XML	244
Configuring the Publishing Process for Your CS System	245
Create the Batch User Account	245
Configuring Your System for Export to Disk Publishing	246
Configuring Your System for Mirror to Server Publishing	252
Configuring Your System for Export Assets to XML	261
Additional Publishing Procedures	264
Migrating a Site from One System to Another System	264
Approving Multiple Assets	266
Creating Destinations	268
Editing Destinations	268
Deleting Destinations	269
Creating Export Starting Points	269
Scheduling Publish Events	269

Reading the Schedule Abbreviations	270
Editing Publish Events	271
Overriding the Schedule	272
Assigning Approval or Preview Templates	272
Monitoring Sessions in the Publishing Console	275
Verifying Publishing Readiness	275
Managing Publishing History Information	276
Publishing All Approved Assets	278
Troubleshooting	278
About Publishing System Error Messages	278
Numeric Messages	279
Other Indicators of System or Configuration Issues	282
13 Revision Tracking	285
Overview	286
Tracker Tables and Storage Directories	286
The RTInfo Table	287
Revision Tracking and the Two Asset Models	287
Implicit vs. Explicit Checkin and Checkout	287
Revision Tracking and Non-Asset Tables	288
How Many Versions?	288
Enabling Revision Tracking	289
Enabling Revision Tracking for Asset Types	289
Enabling Revision Tracking for Non-Asset Tables	289
Editing Revision Tracking Settings	290
Deleting Revisions	291
Disabling Revision Tracking	292
Disabling Revision Tracking for Asset Types	292
Disabling Revision Tracking for Non-Asset Tables	292
Unlocking Revisions	293
Clearing Checkouts for Assets	293
Unlocking Versions for Non-Asset Tables	293
Additional Revision Tracking Functions for Non-Asset Tables	294
Lock	294
Commit	294
Release	294
Rollback	295
History	295
14 Search Engines	297
Overview	298
The Searching Option	298
Search Engine Properties	298
Asset Types and Search Engines	298
Location of Indexes	300

Configuring Asset Types for Your Search Engine	300
Enabling Asset Types for Search.	300
Disabling Asset Types for Search	301
Rebuilding an Index.	301

Appendixes

A. System Defaults	305
ACLs	306
Permissions	306
Accessing ACLs	306
System ACLs	308
ACLs of Default Users	311
Required ACLs for Custom Users	312
System Roles	313
System Asset Types	314
Default Tree Tabs	316
B. System Data: Content Server Database	317
Cache Management Tables (Content Server)	318
Approval System Tables (CS-Direct)	318
Publishing System Tables (CS-Direct)	319
Workflow Tables (CS-Direct)	319
Basic Asset Tables (CS-Direct)	320
Flex Asset Tables.	320
Visitor Tables (Engage).	321
Managing the Attribute Tables	321
Managing the Session Objects Table (scratch)	322
Deleting Unnecessary .class Files	323
C. Sample Site Configurations	325
Sample Users and Their ACLs	327
Sample Roles	329
Hello Asset World	331
Users, ACLS, and Roles for Hello Asset World	331
Asset Type Configuration for Hello Asset World	332
Start Menu Configuration for Hello Asset World	332
Tree Tab Configuration for Hello Asset World	334
Workflow for Hello Asset World	334
Burlington Financial	334
Users, ACLS, and Roles for Burlington Financial	335
Asset Type Configuration for Burlington Financial	336
Start Menu Configuration for Burlington Financial	338

Tree Tab Configuration for Burlington Financial	340
Workflow for Burlington Financial	340
GE Lighting	341
Users, ACLS, and Roles for GE Lighting	341
Asset Type Configuration for GE Lighting	343
Start Menu Configuration for GE Lighting	344
Tree Tab Configuration	345
Workflow	346
Burlington Financial Extension (for Engage)	347
Users, ACLS, and Roles for Burlington Financial Extension	347
Asset Type Configuration for Burlington Financial Extension	349
Start Menu Configuration for Burlington Financial Extension	350
Tree Tab Configuration for Burlington Financial Extension	352
Workflow for Burlington Financial Extension	353
CO	354
Users, ACLS, and Roles	354
Asset Type Configuration	355
Start Menu Configuration	356
Tree Tab Configuration	358
Workflow	359
D. Managing Users, Sites, and Roles in LDAP-Integrated CS Systems	361
Overview	362
LDAP Schema	362
LDAP Connectivity for Site and Role Management	363
LDAP Users and Their Permissions to LDAP Servers	364
LDAP-Integrated Operations	364
User Management Operations	365
Creating an ACL	367
Editing an ACL	367
Deleting an ACL	367
Assigning ACLs to Tables	368
Assigning ACLs to Content Server Pages (Site Catalog Page Entries)	368
Creating a User (Granting ACLs for Access Privileges)	368
Editing a User	369
Deleting a User	369
Creating/Editing a User Profile	370
Deleting a User Profile	370
Creating User Attributes	371
Editing/Deleting User Attributes	371
Site and Role Management Operations	372
Operations with Hierarchical-Schema LDAP Servers	372
Operations with Flat-Schema LDAP Servers	373
Creating a Site	374
Editing a Site	374

Deleting a Site	375
Granting Users Access to Sites	375
Removing Users' Access to Sites	376
Creating a Role	377
Editing a Role	377
Deleting a Role	378
 Index of Procedures	379
 Index	383

About This Guide

This guide describes the Content Server administrator's environment. It begins with an overview of Content Server, its add-on products, content management concepts, and the process you will follow to create your content management (CM) framework.

The rest of the guide explains your main tasks: how to configure Content Server so that writers, editors, and other content providers can electronically create and manage content, participate in workflows, and publish to the online site. Other sections describe Content Server's database and utilities for tuning and maintaining Content Server installations.

Who Should Use This Guide

This reference is written especially for administrators. It is assumed that administrators have a clear knowledge of their company's business needs, and a basic understanding of their roles in the site development process. This guide is also useful to developers, who collaborate with administrators by designing and coding the data model, presentation templates, security system, and caching options.

Administrators are not required to have programming experience, although a technical background is assumed. Developers, however, must know Java, JavaScript Pages, XML, and HTML.

How This Guide Is Organized

Information in this guide is organized by parts, where each part presents a set of chapters that are related to a particular task or function.

Part 1, "Introduction" presents an overview of Content Server, its add-on products, the interface you will be using, and the process you will follow to create your content management (CM) framework.

Part 2, "CM Site Configuration Procedures" describes Content Server's administrative environment. It provides procedures for creating and managing users, assembling and organizing content management sites (the back end of the online site), managing users' access to CM sites, creating workflow processes, and replicating sites.

Part 3, “System Configuration Procedures” presents information about configuring user interfaces, managing the publishing system, revision tracking, and implementing search engines.

The final part, “Appendixes,” contains reference information about system defaults, the Content Server database, sample site configurations, LDAP authentication and deployment options.

At the end of this guide is a standard index and an index of procedures to help you quickly locate procedures for completing administrative tasks from the Content Server interface.

As you read this guide, keep a copy of the *Content Server Property Files Reference* handy. The *Property Files Reference* provides detailed descriptions of the properties that are mentioned in this guide.

Related Publications

The FatWire library includes publications written for Content Server users, administrators, and developers. The publications are provided as product manuals with your Content Server installation. They are also posted on the Web, by version number, at the following url:

<http://e-docs.fatwire.com/CS>

Check the site regularly for updates.

Other publications, such as case studies and white papers, provide information about Content Server’s feature set and its business applications. To obtain these publications, contact sales@fatwire.com.

Part 1

Introduction

This part of the administrator's guide introduces you to Content Server, its system of managing content, the underlying concepts and constructs, and how they are implemented. The chapters in this part of the guide also provide you with guidelines for modeling your online site in the Content Server interface, as well as planning the configuration tasks that you will routinely perform in the Content Server environment.

This part contains the following chapters:

- Chapter 1, "Overview"
- Chapter 2, "Administrator's Interface"
- Chapter 3, "Site Configuration Guidelines"

Chapter 1

Overview

Welcome to Content Server, FatWire's content management system. In this chapter, you will learn about CM sites in the Content Server environment. The sites can be thought of as the back end of your online site, a place where business users work to create their electronic assets, manage them, and deploy them to their audiences.

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

This chapter also outlines the administrator's job in the Content Server environment and offers guidelines for planning configuration tasks.

This chapter contains the following sections:

- Introduction
- Online Sites
- CM Sites
- Content Management Models
- Content Server Environment
- Administrator's Job
- The Focus of This Guide
- Administrator's Prerequisites

Introduction

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

When the back end is configured, its users log on and fill in content-entry forms. The content they enter is then saved to tables in the Content Server database. When ready for delivery, the content is drawn programmatically from the database tables, formatted, and served to browsers as the online site. Figures 1 and 2 illustrate the process flow, using a dynamic publishing scenario.

In our scenario, Figure 1 depicts the smallest unit that can function as a back end: a user who has permissions to write to the Content Server database and to invoke the publishing-delivery systems. This unit, Content Server defines as a “CM site.” Content Server 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 Content Server administrator, you will be working with CM sites—configuring them, replicating them, and managing their day-to-day use. Because they are a critical part of your work, you need to understand how CM sites are defined, how they must be configured, and how they relate to the online site. This chapter provides an overview.

Online Sites

Although most Internet users don’t need a definition of “online site,” our objective in providing one is to first establish what we mean by an online site that is powered by Content Server. Later in this chapter, we explain how online sites can be modeled by CM sites.

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

Regardless of its nature, an online site originates from either a single CM site, or many CM sites, depending on which model you choose. Throughout our product guides, we use the term “online site” generically to refer to websites and portals, both of which are supported in this release.

Figure 1: CM Site

This asset was created by a user populating the fields of the content-entry form below.

Asset

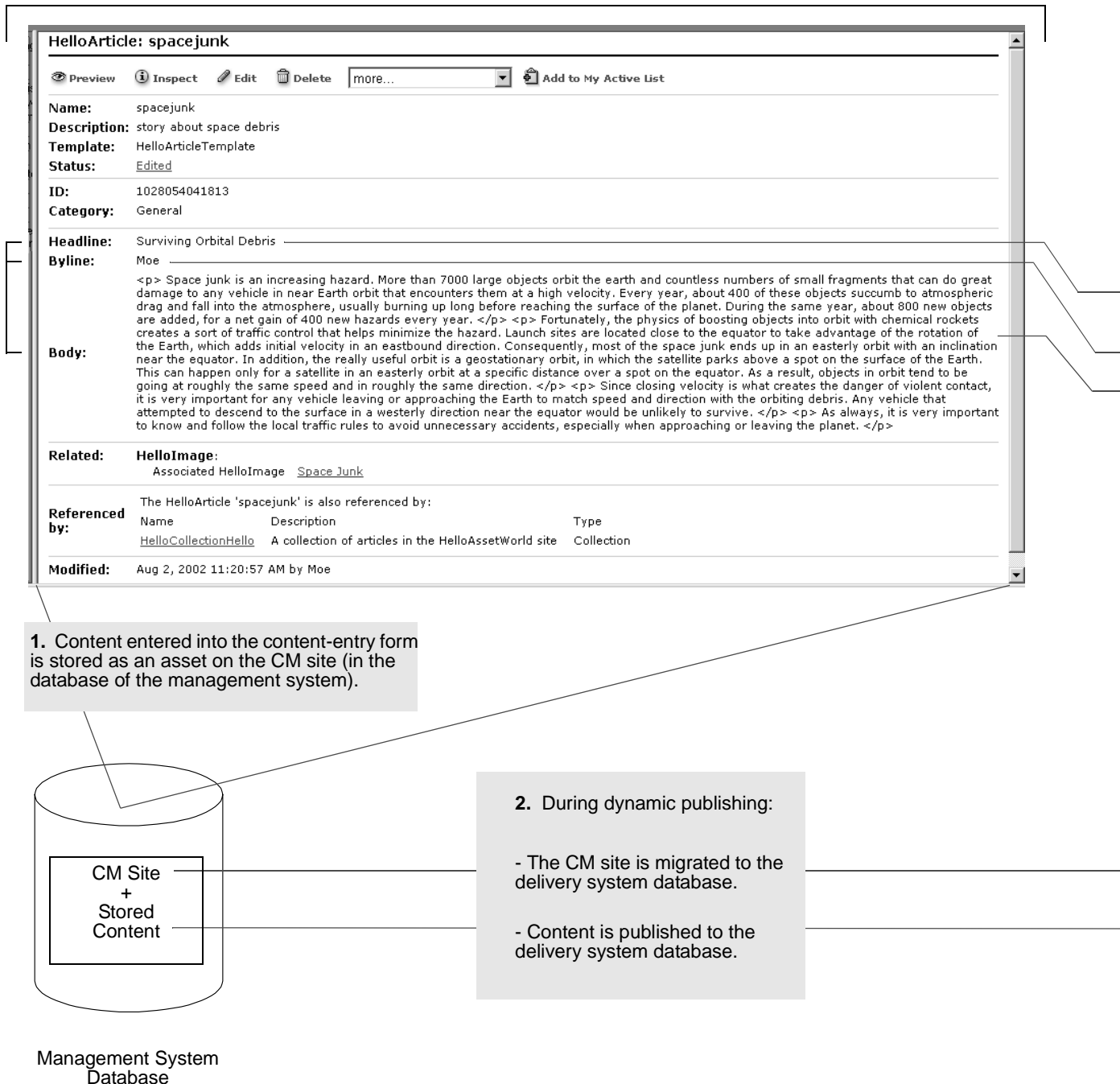
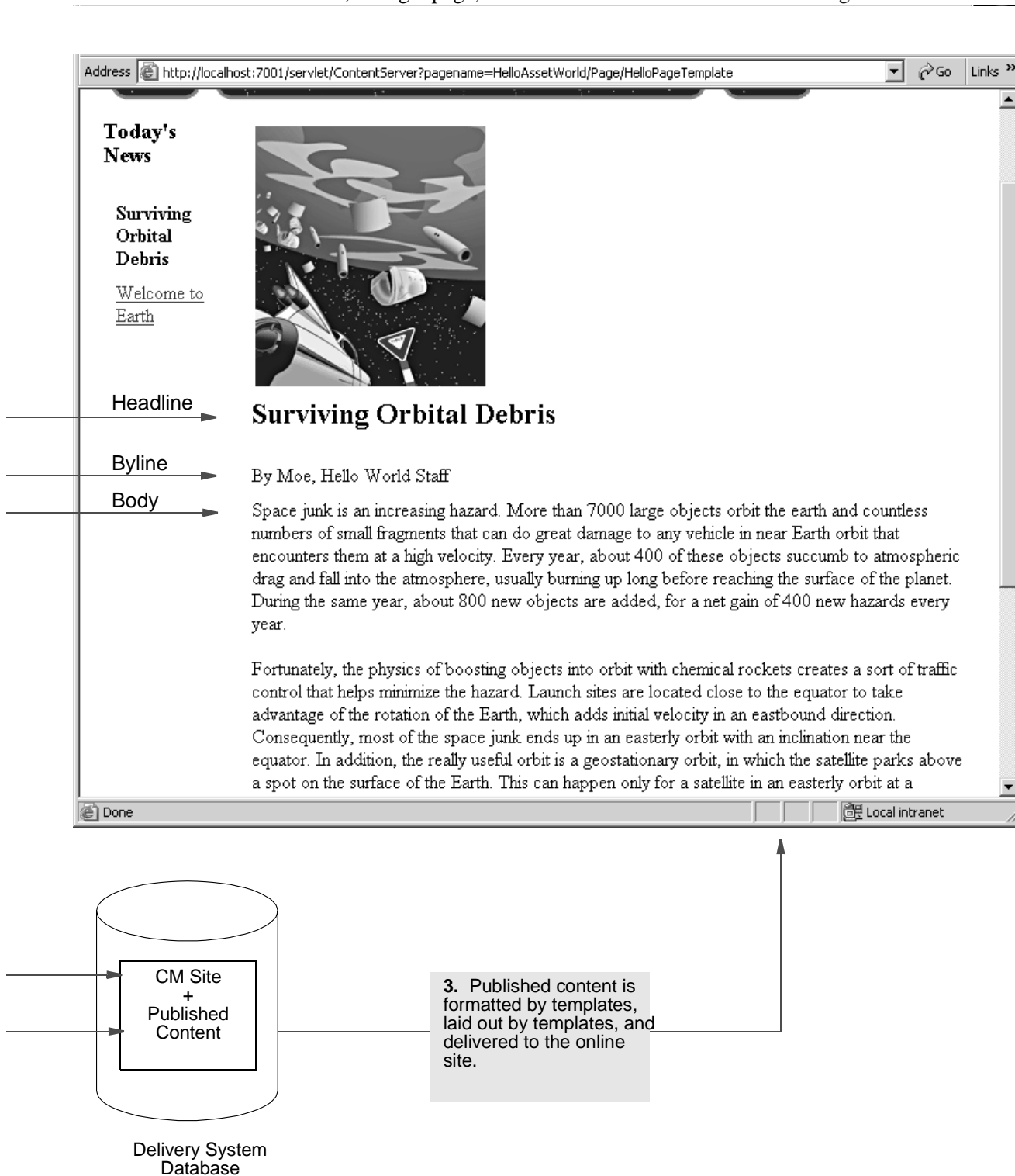


Figure 2: Delivery System and Online Site

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



CM Sites

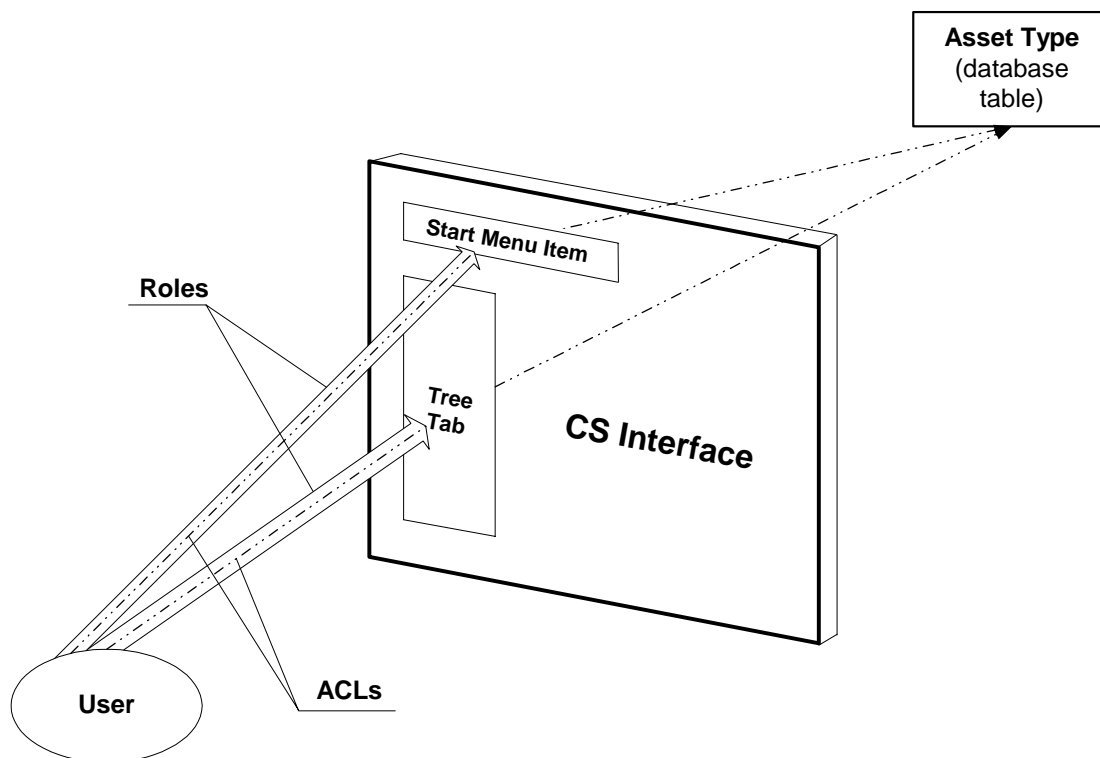
A CM site is a content management unit within Content Server. It is the source of content for the online site and can represent either an entire online site or one of its sections.

Note

From this point forward, when the term “site” is used without a qualifier, it means “CM site.”

A site is an object that you must configure in order to (1) define the authors and managers of the online site, and (2) provide them with the permissions and content management tools they will need: content-entry forms, content-rendering templates, workflow processes, start menu items, publishing methods, and a delivery system. The process of configuring a site involves creating not only the site components, but also associating the components with each other. Making the associations defines a site, similar to the one shown in Figure 3.

Figure 3: CM Site



The site is identified by a name (also configured by the administrator), stored in Content Server’s 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 Content Server interface. ACLs manage the user’s permissions to content (database tables). Through the interface functions (made accessible by roles), the user is able to actuate his permissions (as defined in the ACLs), and

therefore operate on the database tables in order to author and manage specific types of content.

A Content Server system typically has many such sites, each one unique in its function and composition. When users log in to Content Server, 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, Content Server 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 Content Server environment.

Configuration Components

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

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

Site components are generally complex constructs, especially the data model. They are described in the sections that follow. Data modeling is described in detail in the *Content Server Developer’s Guide*.

Table 1: Site components

Component	Required	Created by	See page ...
CM site definition	✓	CS administrator	22
Data model	✓	Developers	23
Users	✓	CS administrator	24
Roles	✓	CS administrator	25
Start menu items	✓	CS or its administrator	26
Workflow processes	–	CS administrator	26
Publishing system	✓	CS administrator	27
User interface options	–	CS administrator	27

CM Site Definition

A CM site definition is one of the components of a CM site. It consists of a site name and optionally a description, both specified by the administrator. The site name can represent an online site, a business topic, the work of a department, or yet another type of content. In

any case, the site name establishes a business theme that users in the site are responsible for developing and maintaining.

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

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

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, Content Server 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 CS interface as content-entry forms (such as the one in Figure 1), whose fields prompt users for information that will be delivered to the online site (or reserved for internal use, if necessary). The set of fields defines the asset type; users’ field entries define the asset (an instance of the asset type).

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

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

Note

Content Server provides a number of default asset types and allows developers to create their own. The asset types, their definitions, and the assets themselves are stored in the Content Server database as tables or table entries, and loosely referred to as “content” in this guide.

For information about data modeling, see the *Content Server Developer’s Guide*. 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 in order to ensure its quality. Examples of content reviewers include editors and art directors who review and modify the copy and designs that are submitted to them.
- Content publishers, who ensure that content is ready to be delivered to the online site, and approve the content for delivery. An editor-in-chief could be a content publisher.
- Content managers, who oversee the authoring, review, and publishing processes.

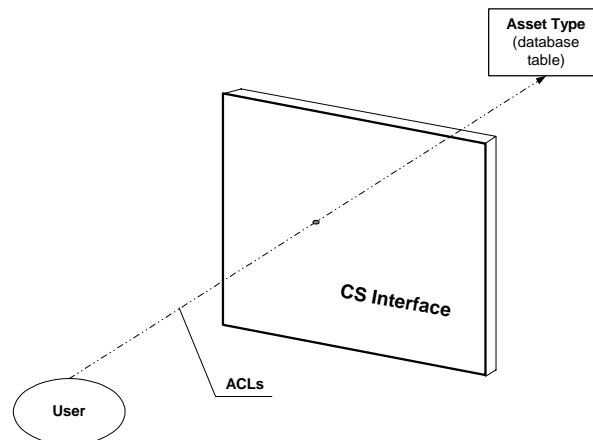
Each content provider must be identified to Content Server through a user account, which consists of a login name, a password, and Access Control Lists (ACLs), the foundation of Content Server’s security system.

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

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

Be aware, that while ACLs give the user *permissions* to operate on tables, they do not give the user *the means* to operate on the tables. For example, in our preceding scenario, the user’s permissions to read and edit the table translate into permissions to use the “view” and “edit” functions in the Content Server interface. However, the functions are hidden from the user, unless the same roles are assigned to both the user and the functions. (For information about roles, see the next section, “Roles,” on page 25.)

ACLs give the user permission (but not the means) to operate on database tables



In general, Content Server uses ACLs at two levels:

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

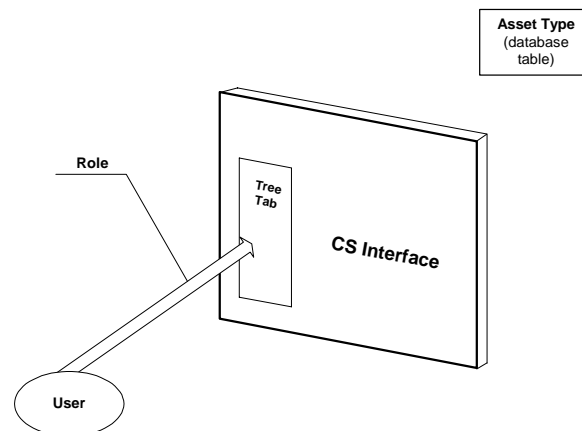
Content Server provides a number of default system ACLs and pre-assigns them to system tables. You can re-use the ACLs, or configure your own and assign them to custom tables, as necessary. For more information about ACLs, see Chapter 4, “Working with ACLs and Roles” and Appendix A, “System Defaults.”

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 (on page 24), where a user is given editorial permissions to the `SystemUser` table through the `UserReader` and `UserEditor` ACLs. To exercise those permissions, the user needs access to the “edit” function in the Content Server interface. The “edit” function is located in the “Admin” tab. To view the “Admin” tab, however, the user must be assigned the same role as the tab. By default, the tab is assigned the `GeneralAdmin` role; the same role must also be assigned to the user.

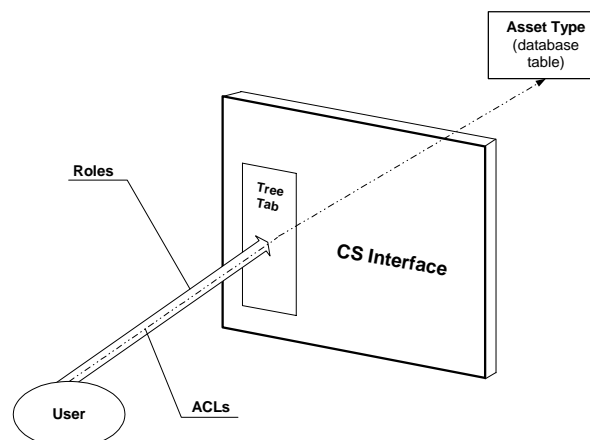
Roles, without ACLs, give the user the means (but not the permission) to operate on database tables



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

In addition to displaying interface functions to site users, roles provide a way of grouping users according to their responsibilities on the

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



site. Users with similar responsibilities can be assigned the same role(s). For example, administrative users need access to the “Admin” tab. All the administrators can be assigned the GeneralAdmin role and thereby be given access to the “Admin” tab.

Content Server defines several default system roles, all of which are pre-assigned to various functions in the CS interface. Content Server 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 (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 Menu Items

A start menu item is a site component. Start menu items 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.

Content Server defines several start menu items: “New” which allows the user to create assets on the site; “Search” which allows the user to look for assets on the site; “CS-Desktop” which enables Microsoft Word as an alternative interface to CS content-entry forms; and “CS-DocLink” which enables the document management interface.

A start menu item specifies:

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

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

Content Server 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 needs, and used to regulate collaborations among content providers.

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

Once workflows are configured, they can be specified in start menu items as processes that automatically begin the moment newly created assets are saved. Workflow processes can also be omitted from start menu items, to be invoked by users as necessary. For more

information about workflow processes, see Chapter 9, “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. Two basic publishing methods can be configured: dynamic publishing (mirror to server) and static publishing (export to disk, and export to XML). The structure of the site (that is, the database schema) is migrated by means of the mirror to server publishing method. The content itself can be published either statically or dynamically, at the administrator’s discretion.

User Interface Options

Content Server supports a number of optional interfaces through which users can interact with content:

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

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

- **Content Server DocLink (CS-DocLink)**, which supports unstructured content in the flex asset family.

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

- **InSite Editor**, which supports the editing of content directly on the rendered page. Regular Content Server users can make quick edits in context, while infrequent users can accomplish their work without having to learn the Content Server interface.
- **eWebEditPro**, a third-party HTML editor from Ektron.
- **Portal interface**, which re-interprets the Content Server interface as a set of portlets in the user’s workspace. One set of portlets is provided for the management of structured content, another set for the management of documents.

Summary

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

Site components are constructs that either exist or must be created in a system-wide pool. From this pool, you select the components that must interact with each other in order to form 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 form one site can be used to form 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 the next section “Content Management Models.”

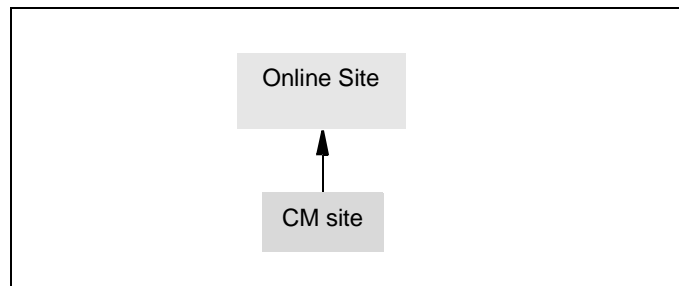
Content Management Models

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

1:1 Model

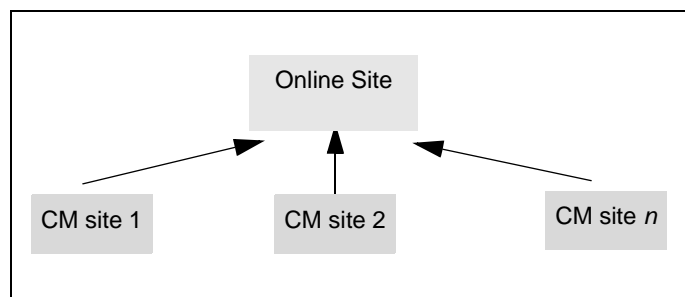
The 1:1 model maps the online site directly to a single CM site. The CM site is the sole source of the online site.

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.



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

section, while the financial writers have a separate site that represents the financial news section. In each design, two sites contribute to one online site.

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

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

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

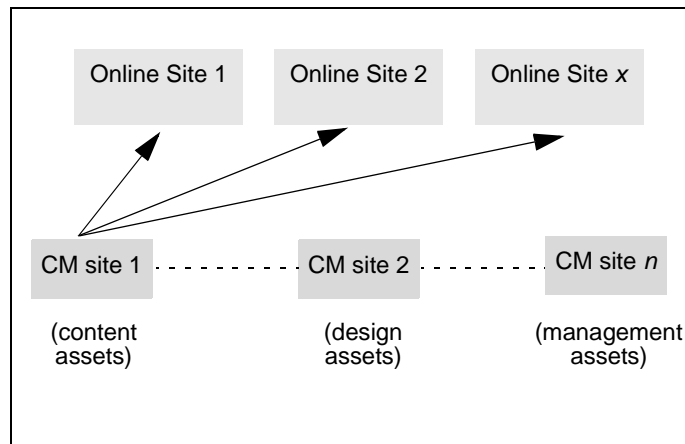
For example, consider the following scenario. A growing e-business specializing in mountain climbing disseminates mountain-climbing news and publishes a photograph album. It also sells gear and answers FAQs. The online site is clearly segmented into four sections: news, photos, gear, and FAQs. On many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

$x:n$ 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

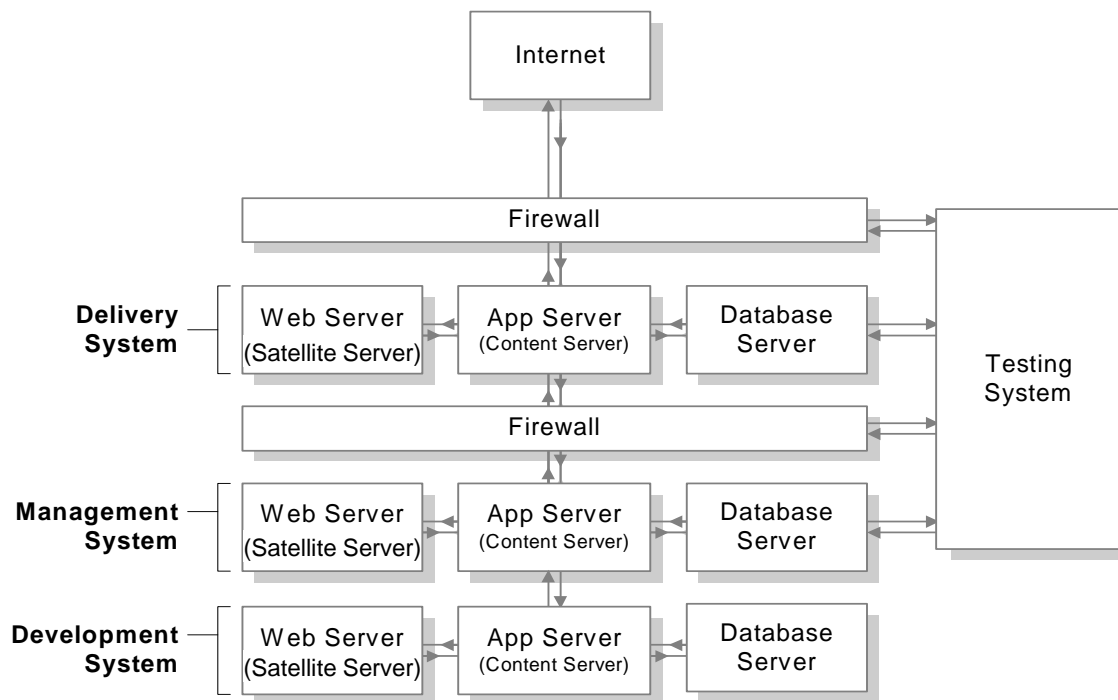
To help administrators and other users understand sites and their relation to the online site, Content Server provides sample sites. Each sample site maps 1:1 to its online site. For sample site specifications, see Appendix C, “Sample Site Configurations.” To log in to the sample sites, follow the login procedure in Chapter 2.

Content Server Environment

Site configuration is a collaborative process that takes place in a Content Server environment rather than an isolated Content Server system. An enterprise-level environment typically consists of four different CS-powered systems: development, management, delivery, and testing, as shown in Figure 4. Each system runs on its own database and at some point interacts with the other systems. One of the systems, testing, is an optional system that large organizations typically install for increased quality assurance. The remaining systems, however, are required.

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

Figure 4: Content Server Environment



Note

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

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

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

Administrator's Job

Collaborating

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

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

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

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

Much of the preceding information is outlined in Chapter 1 of the *Content Server Developer's Guide*. The information is presented in this *Administrator's Guide* in the context of the administrator's job.

Planning

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

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

- Content management model
 - Determine which of the models (1:1, $n:1$, and $x:n$) is most appropriate for your operation
 - Determine which content assets must be shared and which must copied. Do the same for template assets.
- Site replication

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

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

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

Determine the following:

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

(Note that you may also need to create ACLs to assign to visitors of the online site in order to restrict them from accessing certain database tables.)
 - How many roles are needed

- What types of roles are needed (writers, editors, illustrators). Make sure that the roles are compatible with the users' ACLs.
 - To which users the roles should be assigned.
- Workflow processes

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

 - Should workflow be implemented
 - If so, which asset types need workflow processes
 - How must the workflow processes be designed
 - Which users and roles need to participate in the workflow processes
- Publishing system options
 - Which type of publishing needs to be configured—static or dynamic
 - Whether publishing schedules need to be approved
- User interface options

Determine which interfaces the users will need:

 - CS-Desktop
 - DocLink
 - InSite Editor
 - eWebEditPro
 - Portal interface

Use both this book and the *Content Server Developer's Guide* to help you make these decisions.

Development

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

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

At the Development System

The development system is where coding takes place.

- Your job is to provide developers with the specifications of the management system and to assist with content management operations, such as creating sites.
- The developers' job is to:
 - Create sites with the same names as those that will be used on the management system
 - Code the data model (bearing in mind site-replication requirements:

- Content asset types
- Design asset types
- Management asset types
- Create sample assets of each type.
- Test the data model.
- Code templates and, in general, the framework of the online site.

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

Note

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

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

At the Management System

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

Note

On occasion, you will need help from developers; for example, you might want to add custom functionality to the tree tabs in Content Server's interface.

At the management system, your job is the following:

- Create the sites

Note

In this release of Content Server, you can quickly spin off new sites by using the newly added Site Launcher feature to replicate source sites. You can then modify the replicates as necessary. For information about site replication and selecting or creating the appropriate source sites, see Chapter 10, "Replicating CM Sites."

- Configure the users
 - Create their ACLs in Content Server, as necessary

- Create user accounts, either through Content Server, or external user managers, such as LDAP plug-ins
 - Create roles in Content Server, as necessary
 - Assign users their roles for each site
- Assemble the sites by
 - Associating the users with the sites
 - Associating the correct asset types and publishing system with the sites
- Enable the users by
 - Assigning users their roles for each site
 - Using roles to associate users and asset types with start menu items
 - Using roles to associate users with interface functions such as tree tabs
 - Creating tree tabs, as necessary
- Create and manage workflows
- Enable revision tracking
- Configure site and workflow administrators, as necessary
- Configure the publishing process
- Configure the users' interfaces, as necessary

At the Delivery System

The delivery system is configured by developers and typically does not involve an administrator until the system is ready for use. For dynamic delivery, the delivery system is the entire Content Server installation. For static delivery, the delivery system is simply the web server component of the Content Server installation.

At the Testing System

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

Implementation

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

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

The Focus of This Guide

This guide focuses on the configuration tasks that you execute at the management system during the development stage, as described in “Development,” on page 33. It is assumed that the data model resides on the management system by the time you begin your configuration tasks.

Administrator's Prerequisites

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

- Technical aspects of content management

Although you do not have specific technical requirements, a basic knowledge of the following is desirable:

- Java
- JSPs
- Static and dynamic publishing
- Databases and database management
- Browsers

- Your collaborators

For detailed information, see “Collaborating,” on page 31 and the *Content Server Developer's Guide*.

- Online site specifications

Online site specifications are typically prepared by site designers or the corporate communications department to define the site's audience, the site's size, its message and goals, the nature of the content, and the presentation style. Having an understanding of the online site will help you model the online site, plan sites, and collaborate efficiently with developers.

- The Content Server environment

As the Content Server administrator, you can work with up to four different systems: development, management, delivery, and testing. Make sure you have a good understanding of the systems and your function at each of them. For detailed information, refer to the sections “Content Server Environment” and “Administrator's Job” (both in this chapter), and the *Content Server Developer's Guide*.

- Content Server's administrative interface and system defaults

The administrative interface provides you with site configuration tools, and access to system defaults such as ACLs, roles, and asset types. For information about the administrator interface, see Chapter 2, “Administrator's Interface.” For information about system defaults, see Appendix A, “System Defaults” and Appendix B, “System Data: Content Server Database.”

Chapter 2

Administrator's Interface

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

This chapter contains the following sections:

- Types of CS Administrators
- Logging in to the Administrator's Interface
- Administrator Tabs
- Non-Administrative Tabs
- System Defaults
- Sample Sites

Types of CS Administrators

Content Server defines three types of administrators:

- The general administrator, for whom this guide is written.

The general administrator typically administers all systems in the CS environment and therefore has unrestricted access to each system's Content Server interface. The jobs of the general administrator at each system are outlined in Chapter 1, "Overview."

- Site administrators, who administer only certain sites on the management system.
- Workflow administrators, who create workflow processes on the management system

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

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

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

Note

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

Logging in to the Administrator's Interface

This section shows you how to log in to Content Server as the general administrator.

Note

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

To log in to Content Server as the general administrator

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

`http://servername/servlet/Xcelerate/LoginPage.html`

where *servername* is the name of the server that hosts the system that you want to log in to, and *servlet* is the name of the web application on the same server. The system can be development, management, delivery, testing, or another system in your CS environment. The value of *servername* might be either the host name or the IP address. Depending on how the system was set up, you might also need to include the port number (*serverX:8080* for example).

Content Server displays the login form, where the lower left-hand corner lists the products that are installed on the system you are logging in to.



Note

CS-Direct and CS-Direct Advantage are Content Server modules, packaged with the Content Server core product. Whether these modules are listed in the login form depends on whether they were installed on your system.

4. Enter your user name and password.

Note

If you are the 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 to a value other than “admin.” For instructions on changing your password, follow the procedure “To create a user,” on page 77.

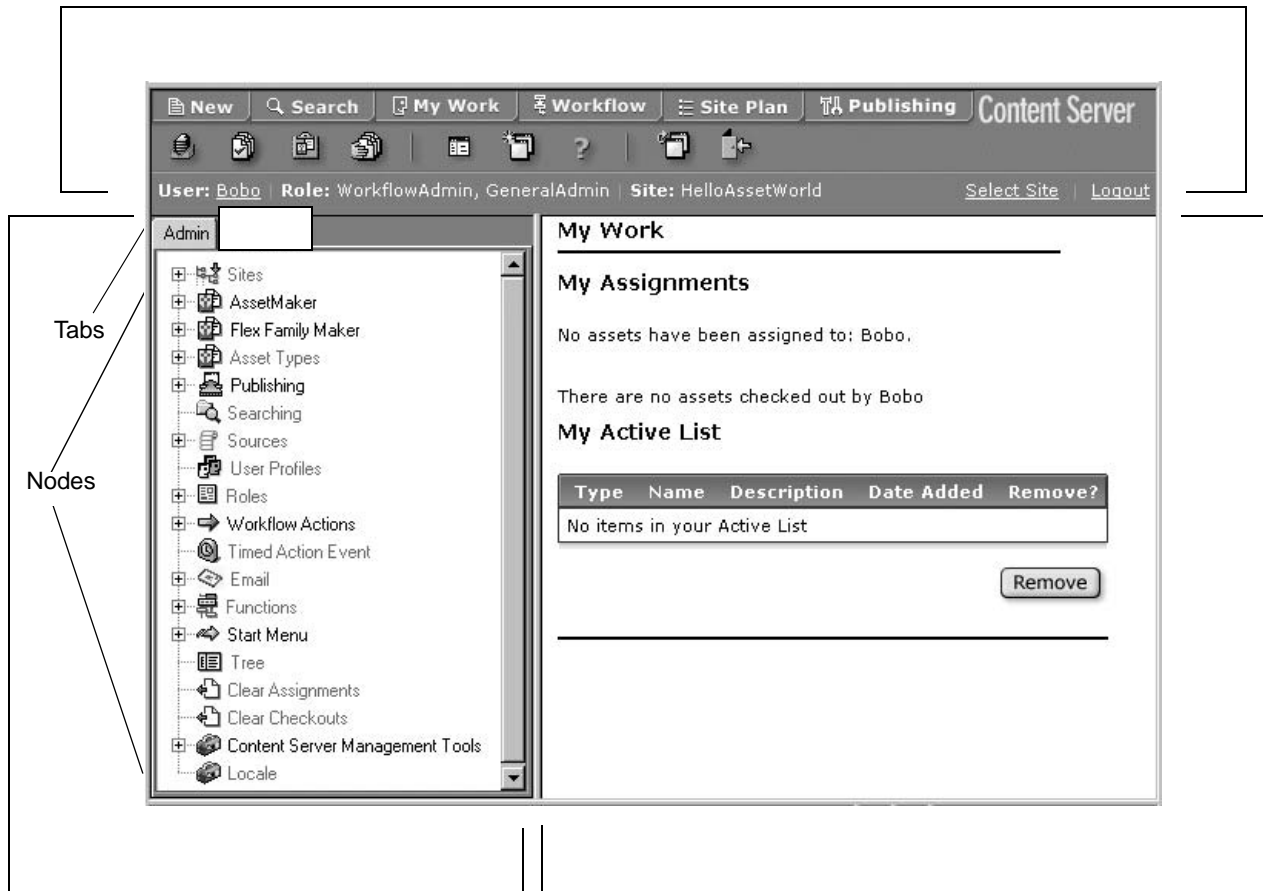
Because you have logged on as an administrator, your user name must have the xceladmin ACL (Access Control List) assigned to it. See Chapter 5, “Configuring Users, Profiles, and Attributes” for information about user accounts and their ACLs.

5. Click **Login**.
6. Content Server responds according to the number of sites that have been created in its interface. To continue, complete one of the following steps:
 - If no sites have been created, you are automatically logged in to the default management site, which enables you to create sites. (Note that the situation of no sites is rare. Most new installations run the sample sites that are provided with Content Server.) Continue with step 7.
 - If one site has been created, you are automatically logged in to that site. Continue with step 7.
 - If two or more sites have been created, they are listed in your CS interface, along with the sample sites that your installation engineer has elected to install.
 - a. Select the site that you want to work on. (For the list of sample sites and their specifications, see Appendix C, “Sample Site Configurations.”)
 - b. Continue with step 7.
7. At this point, you are logged in to either the site of your choice, or the default site. All the tasks that you complete pertain to the site, until you switch to a different site. The one exception is the set of administrative tasks that you execute from the “Admin” tab, which is not site specific. The “Admin” tab contains data for all sites in the system.

Within your selected site, Content Server displays the administrator's interface, consisting of the three frames shown in the next figure:

Top Frame: Button bar

Displays icons and buttons for navigating Content Server.

**Left Frame: Content Server Tree**

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

Right Frame: Site-Building Form

The right frame displays the forms that Content Server calls when you select nodes from the left frame. Each form is used to enter data into the selected node and manage the data.

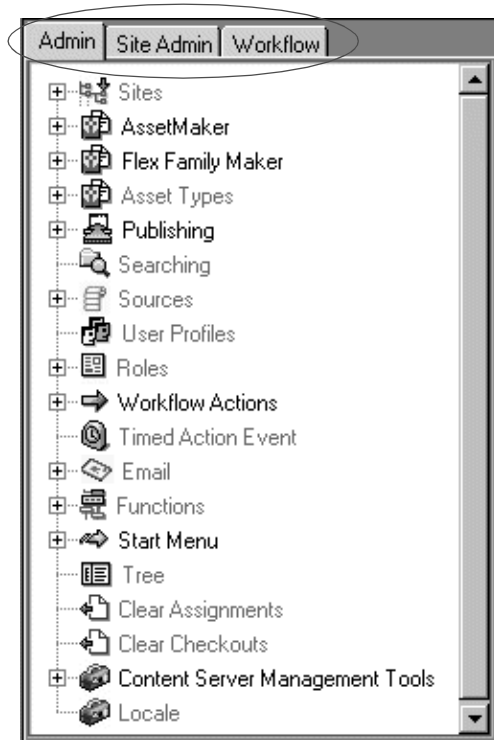
If you are a first time CS user, we recommend that you familiarize yourself with the following features:

- The administrator's interface—the “Admin” tab in particular. You will be using the “Admin” tab almost exclusively throughout this guide to create, configure, and manage sites. For more information, see “Administrator Tabs,” on page 42.
- System defaults, such as ACLs, which you will need in order to configure users, their access rights, and so on. For more information about system defaults, see Appendix A, “System Defaults.”

Administrator Tabs

Up to three administrator tabs can be displayed in Content Server's tree:

- Admin
- Site Admin
- Workflow



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

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

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

- If you log in to a Content Server installation where no sites have been created, you are automatically logged in to the default management site. Here, you see only the “Admin” tab, a system-wide tab for managing the Content Server installation.
- If you are logged in to a custom site, both the “Admin” and the “Site Admin” tabs are displayed in your interface. The “Admin” tab remains system-wide. The “SiteAdmin”

tab is specific to the site you are logged in to. It is therefore a sub-set of the “Admin” tab.

- If workflow processes have been created for any site (except the default site), the “Workflow” tab is also displayed in your interface. The “Workflow” tab accounts for all workflow processes that have been configured for all sites in the system.

Whether administrator tabs are exposed to other users depends on whether you assign the users the associated ACLs and roles. For more information about ACLs and roles, see Chapter 4, “Working with ACLs and Roles.”

‘Admin’ Tab

The “Admin” tab is the main administrative tab, used to create, configure, and manage sites. The “Admin” tab accounts for all content management components in the Content Server installation.

The “Admin” tab displays the following nodes:

- **Sites**, that is, CM sites (described in Chapter 1, “Overview”).

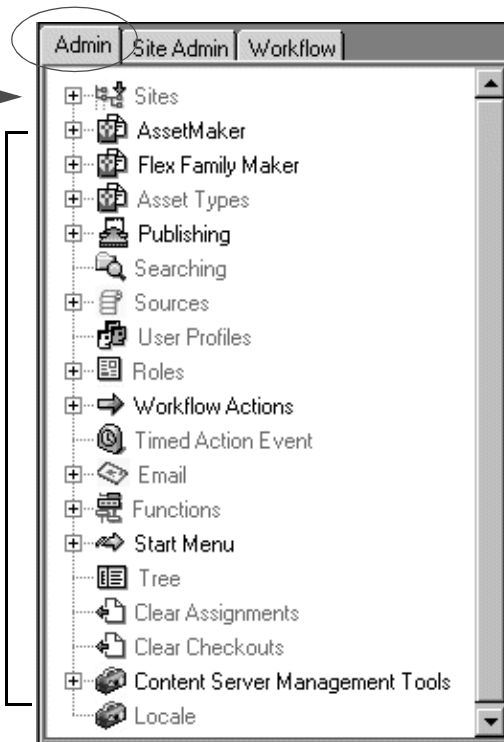
The **Sites** node has the following functions:

- Lists sites that have been created in the system.
- Enables you to define sites and associate to them the components from the system-wide configuration pool, below the “Sites” node.

Content Server provides several sample sites as a learning tool for you. For a description of how they are configured, see Appendix C, “Sample Site Configurations.”

The configuration pool consists of the following nodes:

- **AssetMaker**, for site developers to create new asset types that use the basic asset data model. See the *Content Server Developer's Guide* for information.
- **Flex Family Maker**, for site developers to create new flex families and flex asset types. See the *Content Server Developer's Guide* for information.
- **Asset Types**, for configuring asset types:
 - To add subtypes or asset associations for basic asset types.
 - To create start menu items for asset types that appear on the **New** and **Search** lists.
 - To enable or disable revision tracking for specific asset types. See Chapter 13, “Revision Tracking” for information.



- To enable assets for Content Server Desktop. For information, see Chapter 11, “Configuring the User Interfaces.”
- To enable assets for CS-DocLink. For information, see Chapter 11, “Configuring the User Interfaces.”
- **Publishing**, for setting up the publishing system on the development and management systems. This release provides the “mirror to destination” feature that is managed from the Content Server interface. See Chapter 12, “Managing the CS Publishing System” for information.
- **Searching**, for configuring asset types so that they are indexed correctly when you are using one of the supported third-party search engine modules. See Chapter 14, “Search Engines” for information.

Note

The **Searching** node is displayed only if a supported third-party search engine is installed.

- **Sources**, which you or your site developers use to add new sources if your asset types are designed to use them. See the *Content Server Developer's Guide* for information.
- **User Profiles**, which you use to specify e-mail addresses for your users. See Chapter 5, “Configuring Users, Profiles, and Attributes” for information.
- **Roles**, which you use both for determining the recipients of workflow assignments and to control users' access to functions in the Content Server interface.
- **Workflow Actions**, which you use to create the workflow building blocks called actions and conditions. These items can be used by any workflow process. See Chapter 9, “Creating and Managing Workflow Processes” for information.
- **Timed Action Event**, which you use to configure the frequency with which due dates are calculated for assets that are in workflow. See Chapter 9, “Creating and Managing Workflow Processes” for information.
- **Email**, which you use to create the workflow e-mail messages that can be used by any workflow process. See Chapter 9, “Creating and Managing Workflow Processes” for information.
- **Functions**, which your developers or FatWire professional services staff use to add customized functions to the Content Server interface.
- **Start Menu**, which you use to provide links from the form that is displayed when you click the **New** button in the Content Server interface and the links from the form that is displayed when you click the **Search** button. You use them to create new assets or search for existing ones. Start Menu items can also be created for managing access to the CS-Desktop and CS-DocLink interfaces. See Chapter 8, “Managing Access to CM Site Components” for information.

Note

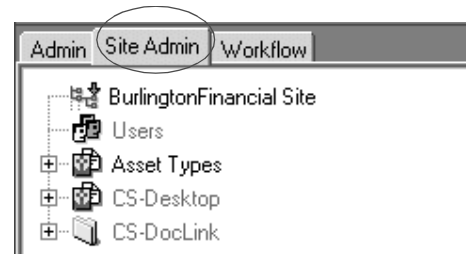
This release of Content Server supports automatic creation of the **New** and **Search** Start Menu items.

- **Tree**, which you use to control access to the existing tree tabs or to create new ones. See Chapter 8, “Managing Access to CM Site Components” for information.
- **Clear Assignments**, which you use to clear workflow assignments. See Chapter 9, “Creating and Managing Workflow Processes” for information.
- **Clear Checkouts**, which you use to unlock and check back in to the database the assets that are checked out by specific users. See Chapter 13, “Revision Tracking” for information.
- **Content Server Management Tools**, which you use to create ACLs and users, and to enable or disable revision tracking for non-asset tables. For information, see Chapter 4, “Working with ACLs and Roles,” Chapter 5, “Configuring Users, Profiles, and Attributes,” and Chapter 13, “Revision Tracking.”
- **Locale**, which you use to set the default locale for the CS system. See Chapter 11, “Configuring the User Interfaces” for information.

‘Site Admin’ Tab

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

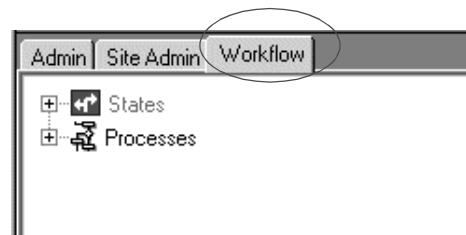
The “Site Admin” tab holds a subset of the administrative functions in the “Admin” tab. It allows you to manage existing users of the site; enable or disable asset types; enable or disable Content Server Desktop and CS-DocLink (if it is installed). It does not allow you to create users, asset types, or sites.



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.

‘Workflow’ Tab

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



Note

While the “Workflow” tab itself is site-specific, its content is not.

The tab. When a user logs on to a site, the user can access the “Workflow” tab if he 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 of the “Workflow” tab is not site-specific.

Non-Administrative Tabs

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

For information about the default tabs, see “Default Tree Tabs,” on page 316. For information about creating and configuring tabs, see “Managing Access to Content Server’s Tree,” on page 121.

System Defaults

Content Server has a number of system defaults:

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

You must be familiar with the defaults in order to configure users and sites.

Information about system defaults is given in Appendix A, “System Defaults” and Appendix B, “System Data: Content Server Database.”

Sample Sites

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

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

- To identify the sample sites and obtain their specifications, see Appendix C, “Sample Site Configurations.”

- To work on the sample sites, follow the steps in “Logging in to the Administrator’s Interface,” on page 39. You can access the corresponding online sites by pointing your browser to each site’s URL, available from your installation engineer.

Note

As mentioned earlier, Content Server now supports the portal environment and provides its business users with two set of portlets: one for managing structured content, the other for managing document-based content. Content Server also provides the “Spark” sample site, which can be optionally installed to demonstrate the portal environment (described in the *Content Server User’s Guide*). Note that the CS administrator continues to use the standard interface to manage the portal environment and its users.

Also note that while some administrative portlets are packaged with Content Server, they are specific to the Spark sample site and are there to help CS administrators manage the Spark sample site, if it is installed. Readers who are interested in learning more about the Spark product can contact their sales representatives for information.

Chapter 3

Site Configuration Guidelines

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

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

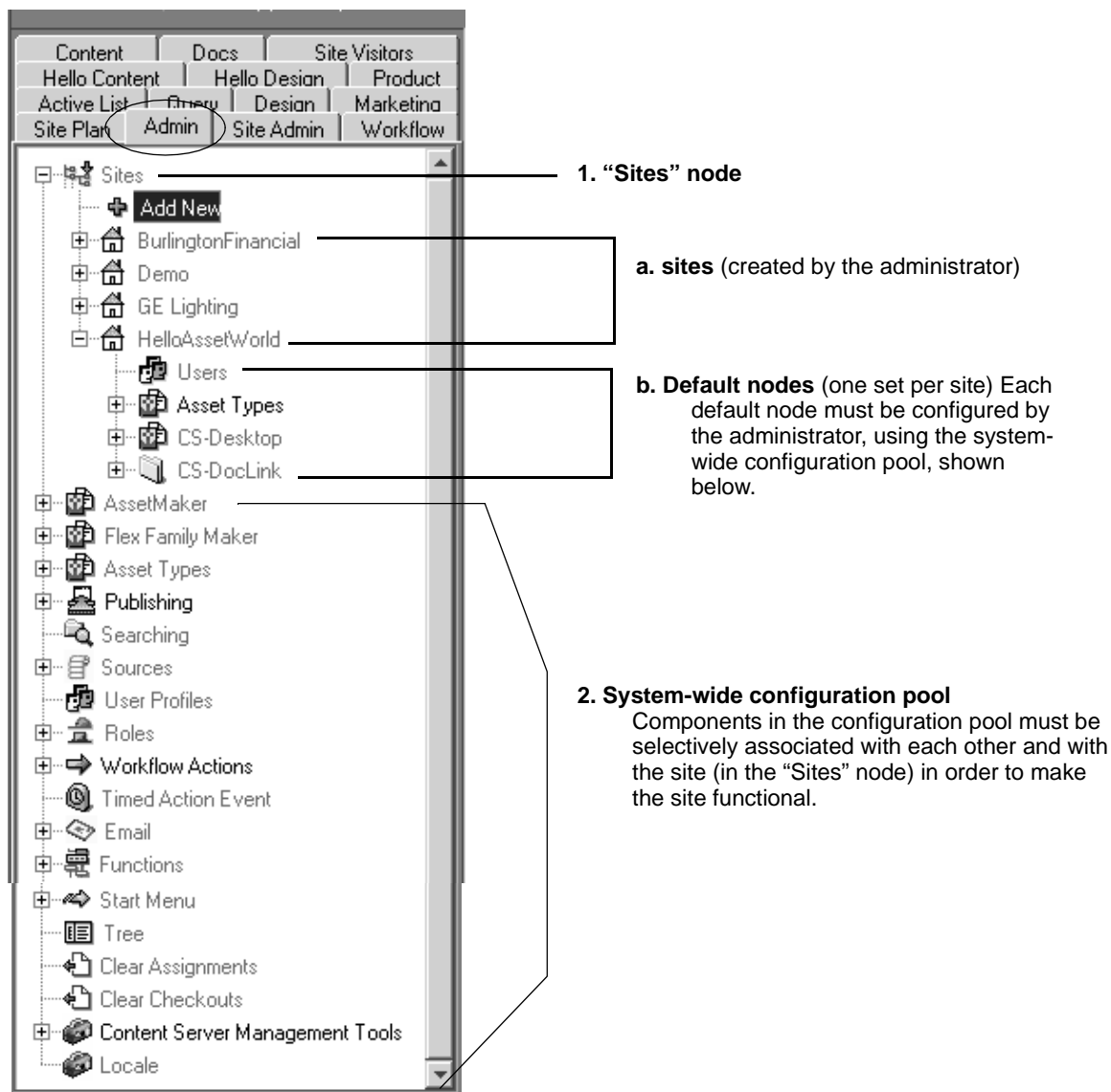
This chapter contains the following sections:

- Overview
- Site Replication Option
- Site Configuration Steps

Overview

When creating and configuring sites, you will be using the “Admin” tab, shown in the figure below. The tab is logically separated into two sections: (1) the “Sites” node, where sites are first established, and (2) the system-wide configuration pool, where site components are created, associated to each other, and managed. The configuration pool begins with the “AssetMaker” node and ends with “Content Server Management Tools.”

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



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

Note

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

Step	Description
<p>1. Add sites to Content Server:</p> <ol style="list-style-type: none"> From the Admin tab, select Sites > Add New. Enter the site name and description. Select a preview method by which you will preview the site’s content (when the content is created). 	<p>The “Sites” node (rather than the configuration pool) can be the starting point for creating sites. When you execute step 1, Content Server adds the site to the “Sites” node. Under the newly added site, Content Server creates a set of sub-nodes that prompt you for site components:</p> <ul style="list-style-type: none"> The “Users” sub-node prompts you for users. The “Asset Types” sub-node prompts you for asset types (created by developers). The “CS-Desktop” sub-node prompts you to enable CS-Desktop. The “CS DocLink” sub-node prompts you to enable CS-DocLink. <p>Each sub-node is equipped with a linking mechanism that allows you to associate the sub-node, and therefore the site, with data in the configuration pool. Because of this linking mechanism, the sub-nodes cannot be deleted or supplemented with custom sub-nodes. All data required by the sub-nodes must be entered in to the configuration pool, described next.</p>
<p>2. Populate the configuration pool with components, such as users, asset types, and roles.</p>	<p>The configuration pool begins with the AssetMaker node and ends with Content Server Management Tools, as shown in the figure on page 50.</p> <p>The configuration pool enables you to select and create components for the site and to associate the components with each other. Data that you establish in the configuration pool remains independent of any sites (and therefore inaccessible to users) until you associate the data with the sites.</p>

Step	Description
<p>3. Associate and enable components in the configuration pool with the site:</p> <ul style="list-style-type: none">a. From the Admin tab, select Sites and the Site name.b. Add users and asset types to the site.c. Enable the users and asset types by means of start menu items, access permissions, and permissions to tree tabs.	<p>The “Sites” node is the endpoint of the site building process. In this step, you once again use the “Sites” node, this time to associate components in the configuration pool to the site that you added in step 1.</p> <p>Note that components in the configuration pool are reusable; they can be associated with as many sites as necessary.</p>

Site Replication Option

If you plan to create many sites that are similar in content and structure, you can shorten the creation process. You can configure the appropriate source sites, then replicate the sites as often as necessary, and finally modify the replicates as necessary. The source sites must conform to certain replication requirements. For more information, see Chapter 10, “Replicating CM Sites.”

Site Configuration Steps

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

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

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

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

Assumptions

We assume that you and developers will code the data model at the development system, and after successful testing, migrate the data model to the management system. At the management system, you will complete the site configuration tasks, test the management system, and finally deploy the system. We also assume that before beginning the configuration tasks, you have read Chapter 1, “Overview” to obtain a general understanding of Content Server concepts, components, and content management models.

Pre-Configuration Decisions

Step 1. Determine the user management method

Your options are Content Server's native user manager, an LDAP plug-in, or an external manager.

While this guide is based on the native Content Server user manager, it does describe system behavior for LDAP-integrated systems. For information, see Appendix D, “Managing Users, Sites, and Roles in LDAP-Integrated CS Systems.” For information about configuring users in LDAP, refer to the LDAP product documentation.

Step 2. Determine users' table-level permissions (ACLs)

Regardless of which user management method you will be implementing, you must have ACLs defined in Content Server. Examine the list of system ACLs (and custom ACLs, if

any have been created) to determine which ACLs to assign to users, and whether you need to create additional ACLs. Typically, the system ACLs are sufficient. For general information about ACLs, see Chapter 4, “Working with ACLs and Roles.” For specifications, see Appendix A, “System Defaults.”

Step 3. Determine the users’ roles

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

For information about roles, see the following:

- Chapter 1, “Overview” and the section “Roles,” on page 69 for general information.
- Appendix A, “System Defaults,” for the list of system roles and their specifications.
- Appendix C, “Sample Site Configurations” for a list of sample roles, their descriptions, and their usage on the sample sites.

Step 4. Determine whether you will be replicating the sites you are configuring

If you plan to replicate the sites you are configuring, make sure that template-coding and other requirements are satisfied for the source sites (the sites you are currently configuring). For an overview of site replication as well as replication options and requirements, see Chapter 10, “Replicating CM Sites.”

Step 5. Determine system security

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

Configuration Steps at the Development System

Configuration steps at the development system are also given in detail in Chapter 2 “CS Development Process,” in the *Content Server Developer’s Guide*. When coding at the development system is complete, the data model is migrated to the management system, where site configuration is completed.

Step 1. Create the sites that will be used on the management system

In this step, you will create sites with the names that will be used on the management system. For information and procedures on creating sites, see Chapter 7, “Assembling and Organizing CM Sites” (in particular “Creating a Site,” on page 92).

Step 2. Create the data model (asset types and their definitions)

1. Create content asset types.
2. Create design asset types (templates), bearing in mind site-replication requirements, if any.
3. Create management asset types.

4. Create sample assets of each type.

For information about creating asset types, see the *Content Server Developer's Guide*.

Step 3. Test the data model

Step 4. Migrate the data model to the management system

To migrate the data model, mirror publish it to the management system. For instructions, see Chapter 12, “Managing the CS Publishing System.”

Configuration Steps at the Management System

Step 1. Create ACLs, if necessary

For procedures on creating ACLs, see Chapter 4, “Working with ACLs and Roles” (in particular “Creating a New ACL,” on page 65).

Notes

- Under no circumstances should you modify or delete a system ACL.
- **Database tables.** When the data model is migrated to the management system, assign ACLs to the custom database tables (asset types), as necessary. For procedures on assigning ACLs to database tables, see Chapter 4, “Working with ACLs and Roles” (in particular “Assigning ACLs to Custom Tables,” on page 67).
- **Content Server page entries.** Typically there is no need to assign ACLs to Content Server page entries on the management system because you use roles to control access to assets and interface functions. However, if you need instructions for assigning ACLs to Content Server page entries, see “Assigning ACLs to Content Server Pages (SiteCatalog Page Entries),” on page 68.

Step 2. Configure the external user manager, if you are using one

For procedures on configuring an LDAP server, see our configuration guide, *Third-Party Software*.

Step 3. Create roles

For procedures on creating roles, see Chapter 4, “Working with ACLs and Roles” (in particular “Creating a Role,” on page 70).

Step 4. Create the users (the content providers)

For instructions on creating user accounts in Content Server, see “Creating a New User,” on page 77. If you are creating user accounts in external managers, refer to the product documentation.

Step 5. Create user profiles, if necessary

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

- CS content applications (CS-Direct, CS-Direct Advantage, and Engage)
- Language packs and setting a default language
- Workflow processes, in which e-mail messages will be sent to notify workflow participants of their assignments. The user profile supports workflow actions by mapping a user name to an e-mail address.

User profiles must be created in Content Server. For instructions, see “Creating and Editing a User Profile,” on page 80.

Step 6. Set user attributes, if necessary

If users require attributes beyond the locale and e-mail attributes that are specified in the user profile, you can create the attributes. For instructions, see “Modifying, Adding, and Deleting User Attributes,” on page 82.

Step 7. Create workflow processes for asset types, as necessary

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

1. Create the “Workflow” tab for the site, if one does not already exist. For instructions, see “Creating Tree Tabs,” on page 122.
2. Plan your workflow process by sketching it. See Chapter 9, “Creating and Managing Workflow Processes” for help with this step. Then refer to your sketch and notes throughout this section.
3. If you have not already done so, create the roles that are needed for the workflow processes. In you need instructions, see “Creating a Role,” on page 70.
4. Ensure that users are set up to participate effectively in the workflow:
 - a. Ensure that users who will participate in workflow have user profiles created for them. Otherwise, they will not receive e-mail messages from the workflow process.
 - a. For shared assets, if you want the pool of workflow candidates to include people from all the sites that an asset is shared to, enable the cross-site assignments feature. See “Cross-Site Assignments and Participants,” on page 138 for details.
5. Create the e-mail objects that you need for your actions and enable the `xcelerate.emailnotification` property in the `futuretense_xcel.ini` file. For instructions, see Chapter 1, “Overview.”
6. Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions that you need. For instructions, see “Setting Up the Workflow Actions and Conditions,” on page 162.
7. If your states have deadlines, be sure to configure the Timed Action Event so that the deadlines of assets are calculated regularly and the appropriate timed actions (if any) are invoked in a timely way. For instructions, see “Setting Up the Timed Action Event,” on page 164.
8. Create your states. For instructions, see “Setting Up the States,” on page 166.

9. Create your process. While creating your workflow process, you create the steps for that process. The steps link together the states so they occur in the proper order. Additionally, while creating your process, you configure any function privileges that you need. For instructions, see “Setting Up the Workflow Processes,” on page 168.
10. Test your workflow processes. For instructions, see “Testing Your Workflow Process,” on page 177.

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 “Creating Start Menu Items,” on page 108.

Step 8. Re-create the sites that were created on the development system

Again, for information and procedures on creating sites, see “Assembling and Organizing CM Sites,” on page 91 (in particular “Creating a Site,” on page 92).

Step 9. Migrate the data model, if you have not already done so

At this point, you will need the data model in order to complete most of the remaining steps. Mirror publish the data model from the development system to the management system by using procedures in Chapter 12, “Managing the CS Publishing System.”

Step 10. Assign custom ACLs (if any) to the custom tables, if you have not already done so

For procedures on assigning ACLs to database tables, see Chapter 4, “Working with ACLs and Roles” (in particular “Assigning ACLs to Custom Tables,” on page 67).

Step 11. Assemble the sites

1. Grant users access to the sites. For instructions, see “Granting Users Access to a Site (Assigning Roles to Users),” on page 97.
2. Enable asset types for the sites by associating the asset types with the sites. For instructions, see “Enabling Asset Types for a Site,” on page 100.

Note

This step provides you with the option of allowing Content Server to automatically create “Start Menu” items for the asset types you are enabling. The “Start Menu” items **New** and **Search** items allow site users to create and search for assets of those types. You can also create your own “Start Menu” items, as summarized in the next step.

Step 12. Make 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,” on page 108 for instructions.
2. Set access permissions on assets. For instructions, see “Setting Access Permissions on Assets,” on page 114.
3. Give users access to the tabs in Content Server’s tree. For instructions, see “Managing Access to Content Server’s Tree,” on page 121.

Step 13. Configure the CS publishing process

1. Determine which type of publishing your management system is to execute:
 - Static (Export to Disk)
 - Dynamic (Mirror to Server)
 - Data transformation (**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 Chapter 12, “Managing the CS Publishing System.”

Step 14. Configure user interfaces, as necessary

1. The user interface options are CS-Desktop, CS-DocLink, InSite editor, and the portal interface. For instructions on making these interfaces available to content providers, see Chapter 11, “Configuring the User Interfaces.”
2. If you have more than one language pack installed on your system, configure the default locale for the system. For instructions, see Chapter 11, “Configuring the User Interfaces.”

Step 15. Set up revision tracking, as necessary

Revision tracking prevents assets from being edited by more than one user at a time. When you enable revision tracking for a database table, Content Server maintains multiple versions of the assets in that table. For instructions on setting up revision tracking, see Chapter 13, “Revision Tracking.”

Step 16. Enable asset types for search engines, as necessary

If you purchased a supported third-party search engine modules and you are using basic asset types, configure the asset types to be searched, so that they can be indexed correctly and found by your search engine. For instructions on configuring the asset types, see Chapter 14, “Search Engines.”

Step 17. Set up the delivery system

For more information, see Chapter 2 “CS Development Process,” in the *Content Server Developer's Guide*.

Step 18. Test all systems**Step 19. Give users the URL to the CS installation and their login information**

Part 2

CM Site Configuration Procedures

This part contains detailed procedures for configuring and replicating sites on your Content Server system.

This part contains the following chapters:

- Chapter 4, “Working with ACLs and Roles”
- Chapter 5, “Configuring Users, Profiles, and Attributes”
- Chapter 6, “Setting Up External Security”
- Chapter 7, “Assembling and Organizing CM Sites”
- Chapter 8, “Managing Access to CM Site Components”
- Chapter 9, “Creating and Managing Workflow Processes”
- Chapter 10, “Replicating CM Sites”

Chapter 4

Working with ACLs and Roles

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

ACLs can be used on both your management and your delivery systems. However, the main focus of this chapter is user management on the management system. For information about user management on your delivery system, see the “Site Development” section in the *Content Server Developer's Guide*.

This chapter contains the following sections:

- Overview
- ACLs
- Working with ACLs
- Roles
- Working with Roles

Overview

ACLs and roles are of paramount importance in Content Server.

- ACLs are used to regulate entry in to the Content Server system. The assignment of ACLs to users, database tables, and Content Server Pages determines users' permissions to operate on Content Server's database tables. If the user's ACLs match the database ACLs, the user has certain permissions (defined by the ACLs) to operate on the database tables. ACLs therefore serve as the foundation of the security and user management model. Without ACLs user accounts cannot be created.
- Roles are used to manage access to sites and their components. The assignment of roles to users and interface functions on a given site determines whether the interface functions are enabled for the users or hidden from them. If the user's roles match the roles that are assigned to the interface functions, the functions are enabled for the user. Otherwise, the functions are hidden.

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

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

ACLs

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

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

- Individual database tables
- Individual Content Server pages

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

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

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

A user must always have at least the Browser ACL in order to view CS pages. However, additional ACL restrictions are enforced only when the `cc.security` property in the `futuretense.ini` file is set to `true`. For information about the `cc.security`

property, see `futuretense.ini` in the *Property Files Reference*. For information about configuring security on your CS system, see Chapter 6, “Setting Up External Security.”

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

Note

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

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

User Accounts

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

While users have one user account and one set of ACLs, no matter how many sites they have access to, they can have one set of roles for one site and a different set of roles for another site. Therefore, users must be assigned all the ACLs necessary to provide them the permissions that they need to fulfill all of their site-specific roles.

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

Database Tables

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

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

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

Note

Do not add ACLs to database tables through the Content Server Explorer application. Instead, use the **Content Server Database** form on the **Admin** tab.

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

For information about:

- Assigning ACLs to database tables, see “Assigning ACLs to Custom Tables,” on page 67
- Registering foreign tables, see the “Database” chapter in the *Content Server Developer’s Guide*

Page Entries in the SiteCatalog Table

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

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

System ACLs

Content Server and the CS content applications use a number of system ACLs to control user access to their features and functions. Different combinations of these ACLs must be assigned to users. Information about system ACLs and their permissions are given in Appendix A, “System Defaults.”

Sample ACLs

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

Custom ACLs

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

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

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

Working with ACLs

This section shows you how to create ACLs, edit, and delete custom ACLs; apply ACLs to the database tables and Content Server pages; and customize “access restricted” messages.

Note

When using an LDAP integration option, be aware of system response to user and site management operations. For information about system response, see Appendix D, “Managing Users, Sites, and Roles in LDAP-Integrated CS Systems.”

Creating a New ACL

Note

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

To create a new ACL

1. On the **Admin** tab, expand the **Content Server Management Tools** tree and then double-click on **ACLs**.
2. In the “ACL Select Operation” form, select **Add ACL** and click **OK**.

The “Add ACL” form appears.

Add ACL

ACL Name	<input type="text"/>
Description	<input type="text"/>
Access Privileges	<input type="checkbox"/> Read <input type="checkbox"/> Retrieve <input type="checkbox"/> Write <input type="checkbox"/> Create <input type="checkbox"/> Delete <input type="checkbox"/> RevisionTracking Audit <input type="checkbox"/> RevisionTracking Admin

Add

3. In the “ACL Name” field, enter a unique name.
4. Select each of the access privileges that you want to assign to this ACL. For information about the listed access permissions, see “Permissions,” on page 306).
5. Click the **Add** button.

Content Server creates the ACL, writing it to the `SystemACL` table.

The ACL that you created now appears in the drop-down list of ACLs in the “ACL Select Operation” form.

6. If you are using a user manager plug-in such as LDAP, be sure to use your LDAP software to first create a group that exactly matches the ACL that you created and then assign that group 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,” on page 308.

To edit a custom ACL

1. On the **Admin** tab, expand the **Content Server Management Tools** tree and then double-click on **ACLs**.
2. In the “ACL Select Operation” form, from the drop-down list, select the name of the ACL that you want to edit.
3. Select the **Modify ACL** option and click OK.
4. Adjust the set of permissions configured for the ACL. (For information about the permissions listed on the form, see “Permissions,” on page 306.)
5. Click **Modify**.

Content Server writes the changes to the `SystemACL` table.

If you are using a user manager plug-in, be sure to use the manager plug-in software to first delete the group that exactly matches the ACL you are deleting.

Deleting a Custom ACL

Caution

Never delete any of the system ACLs. For a list of these ACLs, see “System ACLs,” on page 308.

To delete a custom ACL

1. On the **Admin** tab, expand the **Content Server Management Tools** tree and then double-click on **ACLs**.
2. From the “ACL Select Operation” form, from the drop-down list, select the ACL that you want to delete.
3. Click **OK**.

A message appears that asks you to verify that you intend to delete the ACL.

4. Click **OK** to delete the ACL.

Assigning ACLs to Custom Tables

If you or the site designers create new tables, you might need to restrict access to those tables by assigning ACLs to them. Typically, you assign ACLs to new tables when you create those tables. For information, see the *Content Server Developer's Guide*.

Note

Do not assign additional ACLs to the system tables or to any of the core product tables installed by the CS content applications.

To assign ACLs to an existing table

1. On the **Admin** tab, expand the **Content Server Management Tools** and then double-click on **Content Server Database**.

The “Enter Table Name” form appears.

Enter Table Name
(use "%" as a wild card)

Select Operation:

- ☒ Modify Table
- ☐ Add Table
- ☐ Mirror Table
- ☐ Delete Table

OK

2. Enter the name of the table that you want to assign ACLs to. You can use the percentage character (%) as a wildcard to obtain a list of all tables or you can enter a partial name, ending with the wildcard character, for a set of tables that match the partial name.

3. Select **Modify Table** and click **OK**.

The “Modify Table Select” form appears with a list of tables that match the name you entered. It might be a long list with a vertical scroll bar.

4. Click on a table to select it.

The “Modify Catalog” form appears.

Do not change the value in the **File Storage Directory** field. For information about this field, see the description of `defdir` in the chapter about the Content Server database in the *Content Server Developer's Guide*.

5. From the drop-down list, select the ACLs that you want to apply to this table. To select more than one, use the shift key.
6. Click **Modify**.

Assigning ACLs to Content Server Pages (SiteCatalog Page Entries)

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

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

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

1. Open the **Content Server Management Tools** item.
2. Double-click the **Site** option.

The “Page Operation” screen appears in the work area.

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

Select Operation:

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

3. Enter the path name of the page entry that you want to assign ACLs to. You can use the wildcard character by itself to retrieve all the page entries in the SiteCatalog table, or with a partial name, ending with the wildcard character, to narrow the list that is returned.
4. Select **Modify ACLs** and click **OK**.
 The “Modify ACLs” form appears with a list of pages that match the name you entered. It can be a long list with a vertical scroll bar.
5. From the list, select the ACL that you want to assign. You can select more than one.
6. Scroll through the list of pages. To assign the selected ACLs to a page, select **Apply** (to the right of the **Page Name**). Any ACLs currently assigned to the page are shown in the **ACL** column. You can use the **All** button to select all the pages shown. You can also use the **None** button to clear all selections.
7. Click **Apply**.

Setting the ACL Restriction Error Message

When users attempt to access a page but their ACLs do not give them the correct privileges, the system displays a message. The source of that message is located in the following path:

WebServerInstallDir/doc/futuretense_cs/formpriv.html

You can customize this page and the other HTML error message pages in this directory, but with the following restrictions:

- Do not change the file name of any of the files.
- Do not alter the {0} string anywhere in any of the files.

Content Server uses the {0} string to automatically generate the error messages.

Roles

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

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

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

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

System Roles

Several system roles are installed by Content Server. One role is required for the CS content applications to function, and three are required for the Content Server administrators to function. For descriptions of system roles, see "System Roles," on page 313.

Sample Roles

Content Server provides a number of sample roles that were created to support the sample users on the sample sites. The sites are configured to allow access to tree tabs, based on the roles assigned to users. You can use the sample roles as examples of how you can

configure access control on your sites. For descriptions of sample roles, see “Sample Roles,” on page 329.

Note that if sample sites were not installed on your system, the sample roles will not be available.

Custom Roles

Unlike ACLs, roles are objects that you will most likely need to create in order to account for the full range of users’ responsibilities on the sites. To create roles, follow instructions in the next section “Working with Roles.”

Working with Roles

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

Note

When using an LDAP integration option, be aware of system response to user and site management operations. For information about system response, see Appendix D, “Managing Users, Sites, and Roles in LDAP-Integrated CS Systems.”

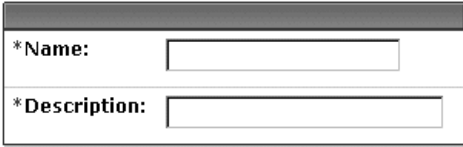
Creating a Role

To create a new role

1. On the **Admin** tab, double-click the **Roles** option.
2. Double-click the **Add New** option in the **Roles** item.

The “Add Role” form appears:

Add New Role



Cancel

Add New Role

3. Click in the **Name** field and enter a unique name of up to 32 characters.
4. Click in the **Description** field and enter a descriptive phrase using up to 255 characters for the role.
5. Click **Save**.
6. After you create a role, be sure to edit the default tree tabs (**Workflow**, **Site Admin**, **Admin**, **Site Plan**, and **Active List**) and add your new roles to the appropriate tabs. For more information, see “Editing Tree Tabs,” on page 124.

Editing a Role

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

To edit the description of a role

1. On the **Admin** tab, double-click the **Roles** option.
2. In the list of roles, double-click the name of the role that you want to edit.
3. In the “Role” form, click **Edit**.

The **Edit Role** screen appears.

4. Click in the **Description** field and enter the new description.
5. Click **Save**.

Deleting a Role

Caution

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

To delete a role

1. On the **Admin** tab, double-click the **Roles** option.
2. In the list of roles, double-click the name of the that role you want to delete.
3. In the **Role** form, click **Delete**.

The system displays a confirmation message.

4. Click **OK**.

Chapter 5

Configuring Users, Profiles, and Attributes

Users can be created through one of the following options: Content Server's native user manager, or external user managers. However, to configure user profiles and attributes, you must use the Content Server interface.

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

This chapter contains the following sections:

- Overview
- User Management Options
- Configuring Users in Content Server
- Working with User Accounts
- Working with User Profiles and User Attributes

Overview

CS users can be created through the native Content Server interface, or through external user managers such as LDAP.

Every CS user is defined by the following set of data:

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

Once users are created and configured, they must be associated, by means of roles, with the sites they are to work in. This chapter shows you how to create and configure users. Procedures for associating users with sites are given in Chapter 7, “Assembling and Organizing CM Sites.”

User Management Options

Content Server’s Directory Services API enables your CS system to connect to external directory servers or user managers that contain authentication information, user information, and so on. The following connection options are available:

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

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

Information about configuring the plug-ins is given in our configuration guide, *Third-Party Software*. Properties that configure the plug-ins are located in the files `futuretense.ini` (the **Authentication** tab), `ldap.ini`, and `dir.ini` files. The files are described in the *Property Files Reference*.

Note

This guide uses the native Content Server user manager throughout.

Native Content Server User Manager

If you are using the native Content Server user manager, follow the guidelines in Chapter 3, “Site Configuration Guidelines” to create and configure users, and then grant them access to the management system.

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 Content Server interface. However, you must still use the Content Server interface to create ACLs and roles in the Content Server database. Instructions for granting users access to a CS management system integrated with LDAP are given in our configuration guide, *Third-Party Software*.

Configuring Users in Content Server

Each Content Server user is completely defined by a user account, user profile, and, if necessary, user attributes.

- A user account is required for anyone who is to work with Content Server.
- A user profile is required for users who will be working with CS modules and products, setting a default language, and participating in workflow processes in which e-mail messages will be sent.
- User attributes, in addition to the locale and e-mail attributes in the user profile, may also be required for your operation. If so, the additional attributes can be created.

Once you have created the user, you must enable that user for the appropriate sites by assigning roles to the user name for each site the user will work in. For information about enabling users after you have created them, see “Granting Users Access to a Site (Assigning Roles to Users),” on page 97.

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 CS applications on the management system:

`http://servername/servlet/Xcelerate/LoginPage.html`

where

servername is the name of the server that hosts the system that you want to log in to (this value might be either the host name or the IP address. Depending on how the system was set up, you might also need to include the port number— *serverX:8080* for example); and

servlet is the name of the web application on the same server.

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

Working with User Accounts

This section shows you how to create, edit, and delete user accounts in Content Server.

Note

This section provides procedures for creating, editing, and deleting users in the Content Server interface. If you are using an LDAP user manager plug-in, refer to the LDAP product documentation. Also, be sure to substitute the word “group” for the word “ACL” and create users who belong to the groups with these names.

Creating a New User

Note

This section shows you how to create users in the Content Server interface. If you are using an LDAP integration option to create users, refer to the LDAP product documentation.

Before creating a user

1. Before creating a user, determine the user's:
 - Login name
 - Password
 - ACLs, which regulate the user's access to Content Server's database tables.
 - To determine the user's required ACLs, see "Required ACLs for Custom Users," on page 312.
 - To determine the user's additional system ACLs, see "System ACLs," on page 308.
 - To determine which ACLs are assigned to the database tables the user must access, follow the steps in "Assigning ACLs to Custom Tables," on page 67.

To create a user

2. On the **Admin** tab, expand **Content Server Management Tools** and double-click on **Users**.
3. In the "Select Operation" form, select **Add User** and click **OK**.
4. Fill in the "Add User" form.

Add User

Login Name (Required)

Access Privileges (Required)
 Browser
 SiteGod
 ContentEditor
 ElementReader

Password (Required)

Re-enter Password (Required)

Add

- a. In the "Login Name" field, enter a unique name. Do not include spaces or special characters, such as punctuation. The underscore character (**_**) is allowed.
- b. From the "Access Privileges" list, select ACLs for the user. (If you are selecting multiple ACLs, use the **Shift** key to select a block of ACLs and the **Ctrl** key to select additional ACLs.)
- c. Enter the same password in to the **Password** and **Re-Enter Password** fields.

d. Click **Add**.

The user has been created.

To follow up with post-creation procedures

5. If the user will be implementing any of the following options:
 - Content Server products such as Engage
 - Language packs and different languages
 - Workflow processes that send e-mail messagescreate a profile for the user. For instructions, see “Creating and Editing a User Profile,” on page 80.
6. If the user requires attributes in addition to or in place of locale and e-mail (specified in the user profile), create the attributes. For instructions, see “Modifying, Adding, and Deleting User Attributes,” on page 82.
7. Once the user has been completely defined, you must associate the user with a site by means of roles.
 - a. If you have not already done so, create roles for the user, following instructions in “Creating a Role,” on page 70.
 - b. To associate the user to the site, create the site and add the user to the site. For instructions, see “Creating a Site,” on page 92, and “Granting Users Access to a Site (Assigning Roles to Users),” on page 97.

Editing a User

Note

This section shows you how to edit users in the Content Server interface. If you are using an LDAP integration option to edit users, refer to the LDAP product documentation.

To edit a user

Caution

Do not change the ACLs or the names of any of the system users: *DefaultReader*, *Content Server*, *xceladmin*.

1. On the **Admin** tab, expand the **Content Server Management Tools** tree and then double-click on **Users**.
2. Fill in the “Select Operation” form:
 - a. Enter the name of the user that you want to edit. (If you do not know the exact name of the user, use the percent (%) character as a wildcard.)
 - b. Select **Modify User** and click **OK**.
3. In the “Modify User Select” form:
 - a. Click on the name of the user to be edited.

Working with User Profiles and User Attributes

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

- CS modules and products
- Language packs (and languages must be enabled)
- Workflow processes in which e-mail messages will be sent to notify workflow participants of their assignments. The user profile supports workflow actions by mapping a user name to an e-mail 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 e-mail address as a value. (You can create workflow actions that send workflow participants e-mail about the assets that are assigned to them.)
- The **locale attribute**, which is used to determine which language to use when your Content Server system has a language pack installed. This attribute takes the user's preferred location as a value.

You can add more user attributes and store values for them in the Content Server user management tables if you want to. However, to use these values in the Content Server interface requires you to customize the elements that display the user profile forms. For information about customizing elements for the Content Server user interface, see the *Content Server's Developer's Guide*.

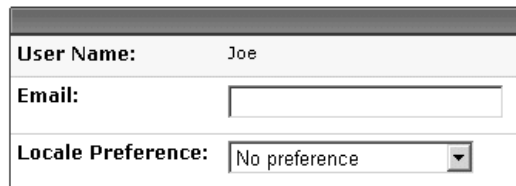
Creating and Editing a User Profile

Note

When using an LDAP integration option, be aware of system response to user and site management operations. For information about system response, see Appendix D, "Managing Users, Sites, and Roles in LDAP-Integrated CS Systems."

To create or edit a user profile

1. On the **Admin** tab, double-click the **User Profiles** option.
The "User Profile Management" form appears.
2. Enter the name of the user and click **Select**.
The "User Profile Management" form appears.
3. Click **Edit**.
4. The "Edit User Profile" form appears.

Edit User Profile

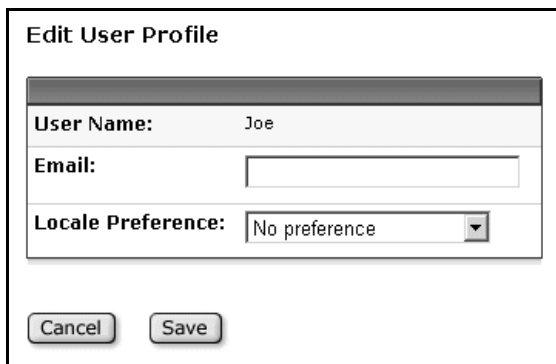
5. Enter an e-mail address.
6. (Optional) If your CS system has any language packs installed, select a locale preference for this user.
7. Click **Save**.
8. Enable this user for the sites the user needs to work with. See “Granting Users Access to a Site (Assigning Roles to Users),” on page 97.

Deleting a User Profile

To delete a user profile

1. On the **Admin** tab, double-click **User Profiles**.
2. Fill in the “User Profile Management” form:
 - a. Enter the name of a user, then click **Select**.
 - b. In the “User Profile” form, click **Delete**.

The system displays a confirmation message.
 - c. Click **OK**.
3. Fill in the “Edit User Profile” form:



- a. Enter the user’s e-mail address.
 - b. (Optional) If language packs are installed on your CS system, select a locale preference for this user.
4. Click **Save**.

Modifying, Adding, and Deleting User Attributes

By default, the only user attributes that the CS content applications need are an e-mail address and locale preference. You use the user profile feature to assign these attributes to a user, as shown in “Creating and Editing a User Profile,” on page 80. If you need to, you can store and use additional user attributes for your users in this table, even if you are using LDAP.

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

To modify a user’s attributes

1. On the **Admin** tab, expand the **Content Server Management Tools** tree and then double-click on **Users**.
2. In the “Select Operation” screen, go to the user name field and do one of the following:
 - If you want to retrieve all users, leave the user name field blank.
 - If you want to retrieve a specific user, enter the specific user name.
3. Select **Modify User Attributes** and then click **OK**.

The “Modify User Select” form appears, listing either all the current users of the system or the one whose user name you entered.

4. Select the name of the user whose attributes you want to modify.
5. Fill in the “User Attributes” form, as explained below:

Attribute name	Attribute Values
<input type="text"/>	<input type="text"/>

Add new attribute and value .

Modify Attributes

Do one of the following, as required:

- Change the current value (or values) assigned to an attribute by editing the attribute value(s) (in the **Attribute Values** field next to the **Attribute Name**).
 - Delete any unwanted attributes by deleting the associated value (in the **Attribute Values** field).
 - Add an entirely new attribute by entering its name and at least one value in the boxes at the bottom of the display.
6. Click **Modify Attributes**.

The attributes are modified (deleted, or added), and the user profile is dynamically updated.

Chapter 6

Setting Up External Security

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

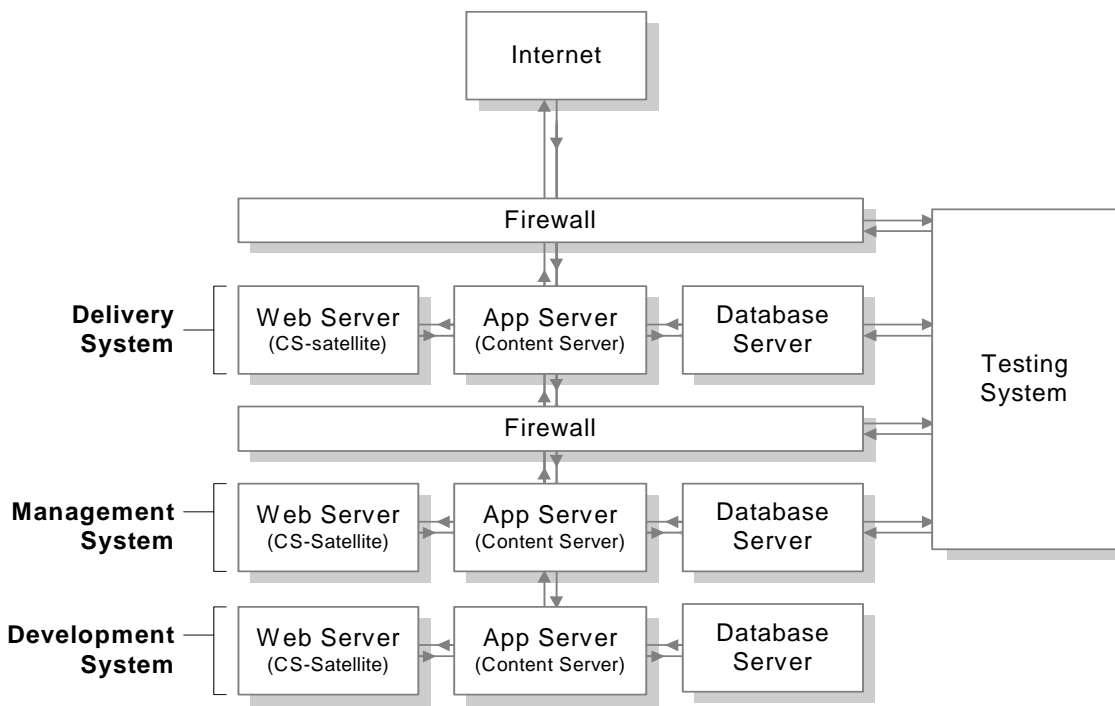
This chapter contains the following sections:

- Overview
- Implementing Security
- Testing Security

Overview

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

At regular intervals, you should review your systems to determine whether they are working as they should. The following figure shows an example of a secure CS system:



ACLs and Security

As mentioned in Chapter 4, “Working with ACLs and Roles,” ACLs (access control lists) serve as the foundation of the security and user management model in your CS system. ACLs are sets of permissions that restrict access to both database tables and Content Server pages.

ACL restrictions are enforced only when the `cc.security` and the `security.checkpagelets` properties in the `futuretense.ini` file are set to `true`. These properties are set to `true` by default. If you need to disable security for any reason, open the Property Editor and set the `cc.security` property to `false`. However, FatWire 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 (they are described in Appendix A, “System Defaults”), and determine whether you need

additional ACLs. You might need to create an ACL with a different combination of permissions than any of the system ACLs provide if your developers plan to create new tables or to make additions to the user interface.

Note

Under no circumstances should you modify a system default ACL or modify the ACLs assigned to a Content Server system table or a CS content application table.

DefaultReader, secure.CatalogManager, and secure.TreeManager

Content Server and the CS content applications are delivered with several default user accounts, one of which is named DefaultReader. Because anyone who visits a site hosted by Content Server must have a Content Server user identity, the DefaultReader user is assigned to any unidentified (unauthorized) visitor.

The DefaultReader user account has one ACL: Browser. Because many of the Content Server database tables have the Browser ACL assigned to them, this means that someone could log in to a Content Server database as DefaultReader using Content Server Explorer and examine information about your system (although they cannot write to any tables as this user).

To prevent someone from using the DefaultReader user account to see tables in your Content Server database, set the following properties in the `futuretense.ini` file to `true`:

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

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

BlobServer and Security

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

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

For information about how those links should be coded, see the *Content Server Developer's Guide*.

Security Goals

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

Security Goals for the Development System

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

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

Security Goals for the Management System

Security on the management system encompasses two main concepts:

- Ensuring that only valid CS users can access the system (described in this chapter).
- Ensuring that those valid users can access only the functions that are appropriate for them. For information, see Chapter 4, “Working with ACLs and Roles.”

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 Content Server user interface to prevent anyone from adding assets or code through CS-Direct or with any of the developer tools. Additionally, you restrict access to some of the Content Server servlets.

Implementing Security

This section describes the steps you must take to configure the security measures you have decided to implement: setting properties, changing passwords for default user accounts, mapping URLs for specific Content Server servlets, and disabling certain parts of the applications on the delivery system.

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

Properties That Configure Security Settings

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

For All Systems

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

- `cc.security`
- `security.checkpagelets`

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

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

`bs.security`

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

`cs.timeout`

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

For the Delivery System

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

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

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

For more information, see “CS Forms and Pages (Delivery System Only),” on page 88.

Users and Passwords

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

For All Systems

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

- Change the default password for the `fwadmin` user account (**Users** node under **Content Server Management Tools** on the **Admin** tab in the Content Server interface).
- If the Content Server user that was created during the installation is named `ContentServer`, verify that its password is not `FutureTense`. If the password is `FutureTense`, change it (**Users** node under **Content Server Management Tools** on the **Admin** tab in the Content Server interface).
- Change the default administrator username/password for your application server.
- Change the default administrator username/password for your web server.
- If you have the sample sites installed, a mirror user was created for the Mirror to Server publishing method (name: `mirroruser`; password: `mirroruser`). Change the password for this user.
- For any user accounts that have the SiteGod ACL, change the password frequently (**Users** node under **Content Server Management Tools** on the **Admin** tab in the Content Server interface). Handle any user account that has the SiteGod ACL as you would the UNIX `root` user.

Note

Do **not** change the default password for the DefaultReader user account.

For the Management and Delivery Systems

It is best if the sample sites (HelloAssetWorld, Burlington Financial, and GE Lighting) are not installed on either the management or delivery systems.

If any of the sample sites were installed on such a system, **delete all the sample users including editor**, but do **not** delete the xceleditor ACL and do **not** delete the fwadmin user. Additionally, change the password for the mirroruser user.

For the Delivery System

Do **not** assign the xceleditor ACL to any user on the delivery system other than the system administrator of that system. This ACL allows access to the CS content applications installed on that system.

URLs and the Web Server (Delivery System Only)

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

- ContentServer
- BlobServer
- Satellite
- CookieServer

Do **not** map URLs to the application server for any of the other Content Server 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

CS Forms and Pages (Delivery System Only)

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

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

Note

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

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

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

Testing Security

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

Security Tests for All Systems

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

1. Try to log in to the database with Content Server Explorer using the default user accounts:
 - `DefaultReader`
If you can log in using `SomeReader` as the password, the `secure.CatalogManager` and `secure.TreeManager` properties are set to `false`. Change them to `true`.
 - `ContentServer`
If you can log in using `FutureTense` as the password, change the password immediately.
 - `editor`
If you can log in using `xceeditor` as the password, change the password immediately.
 - `fwadmin`
If you can log in using `xceladmin` as the password, change the password immediately.
2. Verify that the sample site users do not exist on the management or delivery systems.
3. Verify that you cannot log in as `ContentServer/FutureTense` using a `CatalogManager` URL:

```
http://servername/pathToServlet/  
CatalogManager?ftcmd=login&username=ContentServer&password=  
FutureTense
```
4. Verify that you cannot flush the entire cache as `ContentServer/FutureTense` using a `CacheServer` URL:

```
http://servername/pathToServlet/  
CacheServer?all=true&authusername=ContentServer&authpassword=FutureTense
```

5. Verify that you cannot log in to the application server as the default administrator user.
6. Verify that you cannot log in to the database as the default administrator user.
7. Verify that you cannot log in to the web server as the default administrator user.

Additional Security Tests for the Delivery System

In addition to the preceding six steps, complete the following tasks to test your security setup on the delivery system:

1. Verify that you removed the developer forms from the web server:

```
http://yourhost/futuretense_cs/Dev
```

If the `futuretense_cs` directory is present, delete it and then set the `cs.wrapper` property to `false`.

2. Verify that you cannot log in to CS-Direct:

```
http://yourhost/Xcelerate/LoginPage.html
```

If you can log in, change the names of the `LoginPage` and `LoginPost` pages.

3. Verify that you mapped URLs for all servlets other than `ContentServer`, `BlobServer`, `CookieServer`, and `Satellite` to display a “404 Page Not Found” message. If you can send a request to any other servlet, you should map that URL to an error page immediately.

Chapter 7

Assembling and Organizing CM Sites

CM sites can be assembled and organized once the components are created. The components are the data model, ACLs, roles, and users. This chapter shows you how to work with sites (create, edit, and delete sites); how to manage site users (add, edit, and delete users), and how to manage asset types for the sites (enable and remove asset types).

This chapter contains the following sections:

- Overview
- Working with Sites
- Managing Site Users
- Managing Asset Types
- Enabling and Managing User Interfaces

Overview

Users' access to sites is regulated by roles. For a user to gain access to a site, the user must be associated to the site by one or more roles. The roles allow the user to log on to the site and use certain interface functions. Roles also describe the user's place in workflow processes that might have been created for that site.

This section presents the basic procedures for creating, editing, and deleting sites, for associating users to sites, and for deleting users from the sites.

Working with Sites

Creating a Site

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

To create a site


1. In the Content Server interface, select the **Admin** tab and expand the **Sites** node. (This node is not available on the **Site Admin** tab.)



2. Select **Add New**.

The “Add New Site” form appears.

Add New Site



3. Click in the **Name** field and enter a unique name of up to 255 characters.




Note

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

4. In the **Description** field, enter the name of the site as you want it to appear in the Content Server interface (you can use up to 64 characters). This is the name that will be used in site lists throughout the user interface.
5. In the **Preview method** field, select the method by which you will preview the site's content.
6. Click **Add**.

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

Site: NewSite

 Inspect  Edit  Delete	
Name:	NewSite
Description:	For replication
Publication ID:	1090441988963
Preview method:	Use default (Standard servlet preview)
Users:	Manage users for this site
Enabled Asset Types:	No Asset Types are enabled for this site <div> <input type="button" value="Enable Asset Types..."/> <input type="button" value="Disable Asset Types..."/> </div> List all Start Menu items for this site
Workflow Processes:	List all Workflow Processes for this site
CS-Desktop:	Configure CS-Desktop for this site
CS-Site Launcher:	Configure CS-Site Launcher for this site

[List all sites](#)

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

- The “Users” sub-node prompts you for users.
- The “Asset Types” sub-node prompts you for asset types (created by developers).
- The “CS-Desktop” sub-node prompts you to enable CS-Desktop.
- The “CS DocLink” sub-node prompts you to enable CS-DocLink.

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 Content Server system, use the Mirror to Server publishing method. See “Troubleshooting,” on page 278.




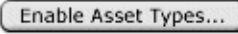

Obtaining Site Configuration Information

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

To obtain site configuration information

1. In the Content Server interface, select the **Admin** tab and expand the **Sites** node.
2. Double-click the site of interest.
3. In the **Site** screen, select the option that provides the information you are looking for.

Site: NewSite

 Inspect  Edit  Delete	
Name:	NewSite
Description:	For replication
Publication ID:	1090441988963
Preview method:	Use default (Standard servlet preview)
Users:	» Manage users for this site
Enabled Asset Types:	No Asset Types are enabled for this site <div>   </div> List all Start Menu items for this site
Workflow Processes:	» List all Workflow Processes for this site
CS-Desktop:	» Configure CS-Desktop for this site
CS-Site Launcher:	» Configure CS-Site Launcher for this site

» [List all sites](#)

Editing a Site

Editing a site means changing its description, the only part of a site definition that you can change. If you need to change the name of a site, you must delete the site and create a new one with the correct name.

To edit a site description

1. In the Content Server interface, select the **Admin** tab and expand **Sites**.
2. Select the site that you want to change.
3. Click **Edit**.
4. In the **Edit** form:
 - a. Click in the **Description** field and make the appropriate changes.
 - b. In the **Preview Method** field, select a different preview method, as necessary.
5. Click **Save**.

Deleting a Site

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

- Start Menu items that were enabled for the site are deleted or unshared
- Saved searches are deleted or unshared
- Publish destinations are unshared
- Users are removed from the site
- Workflows that were shared are unshared or deleted
- Template assets that were shared have their SiteEntry removed
- Asset subtypes that were enabled for that site as well as tree tabs are removed or unshared


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

To delete a site

1. In the Content Server interface, select the **Admin** tab and expand **Sites**.
2. Select the site that you want to delete. If you want to select a newly created site that does not appear on the tree yet, refresh the tree.
3. Click **Delete**.

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

Delete Site

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

Name:	NewSite
Description:	For replication
Publication ID:	1090441988963

4. Click **Delete Site**.

Managing Site Users

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

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

For a user to have access to a site, the user and the site must be assigned the same role (or roles). The assigned roles associate the user with the site, allow the user to log on to the site, and describe the user's place in the workflow processes that were (or will be) created for that site.

To give a user access to a site

1. On the **Admin** tab, expand the **Sites** option.
2. Expand the site that you want the user to have access to.
3. Double-click the **Users** option for that site.

The “User Role Management” form appears.



User Role Management
Site: » [HelloAssetWorld](#)

Username:

(leave blank to select all users)

Select

4. Enter the user name and click **Select**.

The “User Role Management” form displays the user name with no assigned roles.

5. Select the edit icon next to the user name.

The “Edit Roles” form appears:

Edit Roles for User: Flo

User Name: Flo

Site: HelloAssetWorld

***Roles:**

- Analyst
- Approver
- Author
- Checker
- Designer
- Editor
- Expert
- GeneralAdmin
- HelloAuthor
- HelloDesigner
- HelloEditor**
- Marketer
- Pricer
- SiteAdmin
- WorkflowAdmin

6. In the “Roles” list box, select a role or any combination of roles from the list.

Note

Remember that the user cannot perform the functions of this role unless the user has adequate ACL permissions.

7. Click **Save**.
8. Give user access to site components by following procedures in Chapter 8, “Managing Access to CM Site Components.”

Note

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 Content Server (and the CS content applications) on the management system.

Viewing Site Users and Changing Role Assignments

To view the list of users and roles currently defined for a given site, and to change role assignments


1. On the **Admin** tab, expand the **Sites** option, and then expand the site.
2. Double-click the **Users** option for that site.

3. In the “User Role Management” form, leave the user name field blank and click **Select**.

The system displays a list of all the users currently assigned roles at that site:

User Role Management
Site: » [HelloAssetWorld](#)

Select the user to modify:

	User Name	Roles
  	admin	Designer, WorkflowAdmin, SiteAdmin, GeneralAdmin
  	Coco	HelloDesigner, GeneralAdmin
  	Bobo	WorkflowAdmin, GeneralAdmin
  	Flo	HelloEditor
  	Joe	HelloAuthor
  	Moe	HelloAuthor

» [Manage users for this site](#)

4. If you want to make any changes, click the edit icon next to the user name.
5. In the “Edit Roles” form for the user, select a role or any combination of roles from the list and click the **Save** button.
6. Repeat the previous two steps as needed for other user changes.

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. On the **Admin** tab, expand the **Sites** option, and then expand the site.
2. Double-click the **Users** option for that site.
3. In the “User Role Management” form, enter the user’s name and click **Select**.
4. In the next form, click the delete icon (the trash can) next to the user’s name.
The system displays a confirmation message.
5. Click **Delete User from Site**.

Managing Asset Types

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

Enabling Asset Types for a Site

Enabling an asset type means associating the asset type with a chosen site or sites. Once the assignment is made, the asset type itself must be made available to users through Start Menu items, access permissions, and tree tabs, as shown in Chapter 8, “Managing Access to CM Site Components.”

Note

The following procedure describes how to enable an asset type for use within the standard Content Server interface. Enabling an asset type so that it is available within Microsoft Word on the Content Server toolbar is described in “Configuring the Word Asset Types for CS-Desktop,” on page 200.

To enable asset types for a site

1. In the Content Server interface, select the **Admin** tab, expand the **Sites** option, and then expand the site that you want to enable asset types for.

Note

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

2. Select **Asset Types > Enable**.

The **Enable Asset Types** form appears. It displays all the asset types that exist on this system that have not yet been enabled for this site:

Enable Asset Types: HelloAssetWorld

Name:	HelloAssetWorld
Description:	Hello Asset World
Publication ID:	1024593690882

Enable Asset Types:

Name	Description
<input type="checkbox"/> Article	Article
<input type="checkbox"/> AArticles	Article (Flex)
<input type="checkbox"/> AttrTypes	Attribute Editor
<input type="checkbox"/> CAttributes	Content Attribute
<input type="checkbox"/> ContentTmpls	Content Definition
<input type="checkbox"/> ContentGroups	Content Parent
<input type="checkbox"/> CGroupTmpls	Content Parent Definition
<input type="checkbox"/> Image	Image
<input type="checkbox"/> AImages	Image (Flex)
<input type="checkbox"/> ImageFile	ImageFile
<input type="checkbox"/> Link	Link
<input type="checkbox"/> Linkset	Linkset
<input type="checkbox"/> PAttributes	Product Attribute
<input type="checkbox"/> ProductTmpls	Product Definition
<input type="checkbox"/> ProductGroups	Product Parent
<input type="checkbox"/> PGroupTmpls	Product Parent Definition
<input type="checkbox"/> Products	Products
<input type="checkbox"/> StyleSheet	StyleSheet

3. Select the asset types that you want to enable for this site.
4. Click **Enable Asset Types**.
5. Content Server can automatically create a **New Start Menu Item** and/or a **Search Start Menu Item** for the asset types you are enabling. Check the box next to any available Start Menu Item that you would like Content Server to create. (For more information about Start Menu items, see “Managing Access to Asset Types,” on page 108.)

Enable Asset Types: HelloAssetWorld

Start Menu selection

Asset Type	Available Start Menus
HelloArticle	<input checked="" type="checkbox"/> Create Search start menu for HelloArticle <input checked="" type="checkbox"/> Create New start menu for HelloArticle

If you choose not to generate these menu items at this time, you must manually create them later (no one can create assets of the enabled asset types until Start Menu items are created for them).

6. Click **Enable Asset Types**.
7. The asset types are now enabled for the site. If you need to create Start Menu items for them, continue to “Managing Access to Asset Types,” on page 108.

Removing Asset Types from a Site

To remove asset types from a site

1. In the Content Server interface, do one of the following:
 - For the general admin: If you have the GeneralAdmin role, select the **Admin** tab, expand the **Sites** node, and then expand the site you want to disable asset types for.
 - For site admins: If you have the Site Admin role, but not the GeneralAdmin role, select the **Site Admin** tab.
2. Select **Asset Types > Disable**.

The “Disable Asset Types” form appears. It displays the asset types that have been added for this site:

Disable Asset Types: HelloAssetWorld

Name:	HelloAssetWorld																				
Description:	Hello Asset World																				
Publication ID:	1024593690882																				
Disable Asset Types:	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> CSElement</td> <td>CSElement</td> </tr> <tr> <td><input type="checkbox"/> Collection</td> <td>Collection</td> </tr> <tr> <td><input type="checkbox"/> HelloArticle</td> <td>HelloArticle</td> </tr> <tr> <td><input type="checkbox"/> HelloImage</td> <td>HelloImage</td> </tr> <tr> <td><input type="checkbox"/> Page</td> <td>Page</td> </tr> <tr> <td><input type="checkbox"/> Query</td> <td>Query</td> </tr> <tr> <td><input type="checkbox"/> SiteEntry</td> <td>SiteEntry</td> </tr> <tr> <td><input type="checkbox"/> StyleSheet</td> <td>StyleSheet</td> </tr> <tr> <td><input type="checkbox"/> Template</td> <td>Template</td> </tr> </tbody> </table>	Name	Description	<input type="checkbox"/> CSElement	CSElement	<input type="checkbox"/> Collection	Collection	<input type="checkbox"/> HelloArticle	HelloArticle	<input type="checkbox"/> HelloImage	HelloImage	<input type="checkbox"/> Page	Page	<input type="checkbox"/> Query	Query	<input type="checkbox"/> SiteEntry	SiteEntry	<input type="checkbox"/> StyleSheet	StyleSheet	<input type="checkbox"/> Template	Template
Name	Description																				
<input type="checkbox"/> CSElement	CSElement																				
<input type="checkbox"/> Collection	Collection																				
<input type="checkbox"/> HelloArticle	HelloArticle																				
<input type="checkbox"/> HelloImage	HelloImage																				
<input type="checkbox"/> Page	Page																				
<input type="checkbox"/> Query	Query																				
<input type="checkbox"/> SiteEntry	SiteEntry																				
<input type="checkbox"/> StyleSheet	StyleSheet																				
<input type="checkbox"/> Template	Template																				

[Cancel](#)

[Disable Asset Types](#)

3. Select the asset types that you want to remove.

4. Click **Disable**.

Note

If there are start menu items or workflow processes that apply to an asset type that you disabled for a site, be sure to edit the start menu item or workflow process so that it reflects the change as well.

Enabling and Managing User Interfaces

If you plan to enable the Desktop and DocLink interfaces for users on certain sites, follow instructions in Chapter 11, “Configuring the User Interfaces.”

Chapter 8

Managing Access to CM Site Components

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

This chapter contains the following sections:

- Overview
- Managing Access to Asset Types
- Setting Access Permissions on Assets
- Managing Access to Content Server's Tree

Overview

Once users are granted permission to access a site (as shown in Chapter 7, “Assembling and Organizing CM Sites”) their ability to work with the site’s content is managed through

- start menu items, which determine the users’ access to asset types on the site; and
- permissions to the Content Server tree, including
 - tabs in the tree, and
 - nodes in the tabs.

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

Start Menu Items

Start Menu items are the links from the form that is displayed when you click the **New** button in the Content Server interface and the links from the form that is displayed when you click the **Search** button. You use them to create new assets, search for existing assets, or copy assets.

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

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

For example, for basic assets, it might be useful to have the start menu item set a pre-determined value for the Template, Category, Subtype, or Source field (or some combination). If your basic assets have named 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 named 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. Note that if you are upgrading from a version of CS-Direct previous to 4.0, this functionality replaces the default workflow property.

Start menu items also determine which options are available for an asset type that is listed on a tree tab. If an asset type is displayed on a tree tab, but there is no start menu item created for it, the options listed in the right-mouse menu for the asset type are **Refresh**, **Edit**, and **Inspect** only. To have the **New** option included in the right-mouse menu, there must be a **New** start menu item created for it.

For Different Kinds of Users

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

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

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 definition by mistake.

Tree Tabs

A tree tab is a tab that appears in the left-hand panel of the Content Server interface. Access to tree tabs is, again, controlled through roles.

There are several default tree tabs that are core to Content Server:

- **Active List** – displays the assets that you selected and placed on your Active List. All users see this tab as soon as they use the **Move to Active List** button to place an asset on their active list.
- **Admin** – displays the administrative functions that affect all of the sites in the system. By default, only users with the default system role named GeneralAdmin have access to this tab.
- **Design** — serves as the source for creating pages on your site. Some of these sources are: Templates, Product Definition, Content Definition, and other sources for the creating pages.
- **History** – displays the assets that you worked with during the current session. All users see this tab as soon as they create, inspect, edit, or copy their first asset.
- **Query** – provides access to the query asset type. You can run queries and see their results on the tab.
- **Site Admin** – holds a subset of the administrative functions that apply only to the site that you are currently logged in to. By default, only users with the default system role named SiteAdmin have access to this tab. This tab is useful if you have individuals who manage which existing users have access to individual sites, but who do not need to create new users or new sites.
- **Site Plan** – displays a graphical and hierarchical representation of the pages, collections, queries, and content assets that make up the online site that you are making available on the delivery system. By default, all of the 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.

You can create new tree tabs that give your users another method of creating or editing assets of specific types (in addition to the start menu lists, that is.)

For example, the **Design** tab (a core tab displayed on the GE Lighting sample site) gives users with the Designer role access to attribute editors, product definitions, and so on. The **HelloContent** tab in the HelloAssetWorld sample site gives users with the HelloAuthor, HelloEditor, and HelloDesigner roles access to HelloArticles and HelloImages.

When you create a new site, CS-Direct adds that site to the list of sites that are enabled for all the tree tabs. However, the roles that create for your site cannot be automatically assigned to a tab—you must specify which tabs a new role has access to.

Managing Access to Asset Types

Managing user's access to asset types on a site entails creating start menu items for the asset types.

Creating Start Menu Items

A start menu item is a way of coupling a user with the asset types he is permitted to work with. When you create a start menu item, you specify

- the sites the item is for, and
- the roles at those sites that are allowed to create or search for assets of those types.

In other words, start menu items determine which users can create and search for assets of specific types on a specific site.

CS supports four kinds of start menu items:

- **New** – creates a link for the asset type on the list that is displayed when you click the **New** button located in the button bar at the top of the Content Server interface. If you do not create **New** items for the asset types that are enabled for a site, no one can create assets of those types. For information about creating **New** start menu items, see “Creating Start Menu Items for the New List,” on page 108.
- **Search** – creates a link for the asset type on the list that is displayed when you click the **Search** button located in the button bar at the top of the Content Server interface. If you do not create **Search** items for the assets that are enabled for a site, no one can search for assets of those types. For information about creating **New** start menu items, see “Creating Start Menu Items for the Search List,” on page 112.
- **CS-Desktop** – makes the asset type available to content providers who use the CS-Desktop interface instead of the Content Server interface. For information about creating CS-Desktop start menu items, see “Configuring the Word Asset Types for CS-Desktop,” on page 200.
- **CS-DocLink** – makes the asset type available to content providers who use the CS-DocLink interface instead of the Content Server interface; also configures document assets for the portal. For information about creating CS-DocLink start menu items, see “Configuring the CS-DocLink Asset Types,” on page 206.

Creating Start Menu Items for the New List

Before you begin creating start menu items, note the following:

- Be sure to create start menu items for only the asset types that are enabled for that site. If the assets are not enabled, go back to Chapter 7, “Assembling and Organizing CM Sites” and complete the steps in “Enabling Asset Types for a Site,” on page 100.

- If you want to specify a default workflow process, you should create the workflow process first. For information about workflow, see Chapter 9, “Creating and Managing Workflow Processes.”
- Talk to your developers to find out which fields are used by queries, collections, or other design elements for the online site. That way, you can be sure to specify default values for those fields.
- Talk to 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.
 - Date fields
 - Fields that accept more than one value.
 - Association fields.
 - Flex attribute fields for flex and flex parent assets.

To create a start menu item for the **New** list

1. In the Content Server interface, select the **Admin** tab.
2. Expand the **Start Menu** node, and double-click **Add New**.
3. In the “Start Menu” form, do the following:



- a. In the **Name** field, enter the name of the item, using up to 64 characters. Although it is not required, it is a good idea to start the name with the word “New” (for example, “New Article” or “New Template”). The name that you enter will be displayed on the **New** list.
 - b. In the “Asset Type” field, select the type of asset for which you are creating the Start Menu item.
 - c. Click **Continue**.
4. If you have chosen a flex asset type, continue with this step. Otherwise skip to step 5.

If you have chosen to create a Start Menu item for a flex asset type, you are presented with the following form:

The form is titled "Start Menu". It contains three input fields: "Name" with the value "New SMI", "*Asset Type" with a dropdown menu showing "Article (Flex)", and "*Flex Definition" with a dropdown menu showing "Any". At the bottom right, there are two buttons: "Cancel" and "Continue".

- a. In the **Flex Definition** field, select the asset subtype.
 - b. Click **Continue**.
5. In the next "Start Menu" form, fill in the fields as explained below:

The form is titled "Start Menu". It contains the following fields and controls:

- *Name : Text field with "New SMI".
- Description : Text field.
- Type : Dropdown menu with "New" selected.
- Asset Type : Text field with "Article (Flex)".
- Flex Definition : Text field with "Story".
- Default Values : A section with a "Field" dropdown (showing "Attribute_body"), a "Value" text field (showing "[Value]"), and a list box. Below this are "Add" and "Remove" buttons.
- *Sites : List box containing "Any", "CO", "HelloAssetWorld", "BurlingtonFinancial", and "GE Lighting".
- *Roles : List box containing "Any", "Analyst", "Approver", "Author", "Checker", "Designer", "Editor", and "Expert".
- Workflow Process : Button labeled "Select Workflow...".

 At the bottom left, there are "Cancel" and "Save" buttons.

- a. In the **Description** field, enter a short description, using up to 255 characters.
- b. In the **Type** field, select **New**.
- c. Note that the **Asset Type** field is an information-only field that displays your choice of asset type.

- d. (Optional). In the **Default Values** section, complete the following steps if you want field values to be set automatically when a content provider creates an asset with this start menu item.
- 1) Select a field from the **Field** drop-down list. This list displays all the fields for assets of this type; for example, category, source, and template.
 - 2) In the **Value** field, enter or select a value.

Notes

- Selecting certain default fields, such as category and source, automatically produces a drop-down list of options in the **Value** field. For other fields, you must provide your own value. **All values are case-sensitive.**
- Content Server 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.

- 3) Click the **Add** arrow to add the value to the field.
- 4) Repeat the previous three steps, as necessary.

Note

When you set a field value through a start menu item, the field does not become a read-only field. That is, no matter what value you set in the “Start Menu” form, that value can be overridden by the content provider who creates an asset through the start menu item. The values that you specify are for convenience only.

- e. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl + click** to select more than one site.
- f. In the **Roles** field, select all the roles that can have access to the start menu item. Use **Ctrl + click** to select more than one role.

Note

If you are assigning a default workflow with this start menu item, the roles that you select in this step must match the roles that are (or will be) authorized for the start step in the workflow process.

- g. (Optional) To configure workflow process details for assets created with this start menu item, complete the following steps:
 - 1) Click **Select Workflow**.
 - 2) In the “Select Workflow” form, select the appropriate workflow process from the drop-down list and then click **Select Workflow**.

3) In the “Set Participants” form, select at least one user for each role that appears, and click **Set Participants**.

4) Click **Continue**.

The system saves the workflow information and displays the “Start Menu” form again.

6. In the “Start Menu” form, click **Save**.

Creating Start Menu Items for the Search List

To create a start menu item for the Search list

1. In the Content Server interface, select the **Admin** tab.
2. Expand the **Start Menu** node, and double-click **Add New**.
3. In the “Start Menu” form, do the following:

The screenshot shows a web form titled "Start Menu". It contains two input fields: "Name :" with an empty text box, and "* Asset Type :" with a dropdown menu currently showing "Article". Below these fields are two buttons: "Cancel" and "Continue".

- a. In the **Name** field, enter the name of the item, using up to 64 characters. Although it is not required, it is a good idea to start the name with the word “Find” (for example, “Find Article” or “Find Template”). This is the name that is displayed on the **Search** list.
 - b. In the “Asset Type” field, select the type of asset for which you are creating the Start Menu item.
 - c. Click **Continue**.
4. If you have chosen a flex asset type, continue with this step. Otherwise skip to step 5.

If you have chosen to create a Start Menu item for a flex asset type, you are presented with the following form:

The screenshot shows a web form titled "Start Menu". It contains three input fields: "Name :" with the text "New SMI" entered, "* Asset Type :" with a dropdown menu showing "Article (Flex)", and "* Flex Definition :" with a dropdown menu showing "Any". Below these fields are two buttons: "Cancel" and "Continue".

- a. In the **Flex Definition** field, select the asset subtype.
 - b. Click **Continue**.
5. In the next “Start Menu” form, fill in the fields as explained below:

Start Menu

*Name :	<input type="text" value="New SMI"/>
Description :	<input type="text"/>
Type :	<input type="text" value="New"/>
Asset Type :	Article (Flex)
Flex Definition :	Story
Default Values	<div>Field : <input type="text" value="Attribute_body"/></div> <div>Value : <input type="text" value="[Value]"/></div> <div><input type="button" value="Add"/> <input type="button" value="Remove"/></div>
*Sites :	<div>Any</div> <div>CO</div> <div>HelloAssetWorld</div> <div>BurlingtonFinancial</div> <div>GE Lighting</div>
*Roles :	<div>Any</div> <div>Analyst</div> <div>Approver</div> <div>Author</div> <div>Checker</div> <div>Designer</div> <div>Editor</div> <div>Expert</div>
Workflow Process :	<input type="button" value="Select Workflow..."/>

6. Click in the **Description** field, and enter a short description for the start menu item.
7. In the **Type** field, select **Search**.
8. In the **Roles** field, select all the roles that can have access to the start menu item. Use **Ctrl + click** to select more than one role.
9. In the **Sites** field, select which sites can use this start menu item. Use **Ctrl + click** to select more than one site.
10. Click **Save**.

Creating Start Menu Items for CS-Desktop

For information about creating CS-Desktop start menu items, see “Configuring the Word Asset Types for CS-Desktop,” on page 200.

Creating Start Menu Items for CS-DocLink

For information about creating CS-DocLink start menu items, see “Configuring the CS-DocLink Asset Types,” on page 206.

Displaying a List of All Start Menu Items

In addition to the individual start menu nodes that appear for each of the asset types that exist on your CS system, there is a top-level **Start Menu** node on the **Admin** tab. Use this node when you want to see a list of all the start menu items in the system.

Additionally, there is a link to the list of all the start menu items from the “Inspect” form for any individual start menu item.

Setting Access Permissions on Assets

Content Server now provides a system by which you can set access permissions on assets that are not part of a workflow process. For such assets, you can create policies and, within them, define which roles have permissions to perform which functions on assets. For example, a policy for the function “copy” might specify which roles are allowed to copy an asset. A different policy for “copy” might specify which roles are denied the ability to copy an asset. For flex assets, inheritable policies can be set on parents, meaning that the policies will be inherited by the children.

Functions that can be performed on assets, by default, are the following: checkout, copy, edit, delete, rollback, share, approve, and build. (For definitions of the functions, see “Determine the Function Privileges,” on page 148).

All of the above-listed functions are defined in the `futuretense_xcel.ini` file, as values of the property `xcelerate.authorizefunction` (located on the “Authorization” tab, if you are using the Property Editor). You can specify additional functions (which are listed on the “Authorization” tab).

To summarize, access permissions serve a three-fold purpose:

- Extend the workflow function privileges system, making it possible to set permissions for assets when they are not in workflow
- Introduce new functions (authorize, inspect, and preview) for viewing the contents of assets that are not in workflow
- Introduce a hierarchical permission inheritance model for assets that are not in workflow.

Note

The “access permissions” feature does not apply to other entities, such as pagelets.

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. Content Server implements a three-level permission structure: Grant, Deny, and Inherited. “Grant” and “Deny” denote the explicit permissions, that is, grant or deny the permission to perform a function. “Inherited” means that permissions are neither granted nor denied. Instead, the permissions are inherited from the parent asset's policy, if one exists. If the parent's policy does not exist, then the asset's permissions are inherited from the policy of the parent's parent, and so on.

If permissions are ambiguous, the following considerations may help resolve the ambiguity:

- A permission is granted if at least one of the user's roles grants the permission.
- A permission is denied if at least one of the user's roles denies the permission, and none of the user's roles grant the permission.
- "Grant" overrides "Deny."

If one role grants a permission, but another role denies the same permission, the permission is granted.

The preceding scenarios are summarized in the following table:

	Role 1	Role 2	Result
Permission	Granted		Granted
	Denied		Denied
	Granted	Denied	Granted

- Assets can inherit permissions from parent assets.

For example, if a parent asset named `Money Market Funds` grants the "delete" permission to a role named "Editor," and user Bob has that role, then Bob can delete the `Money Market Funds` asset and all of its child assets.

- Asset permissions can override the parent assets' permissions.

A `Prime Money Market semi-annual report` asset denies the "delete" permission to a certain role. That same role is granted the "delete" permission in the `Prime Money Market` parent asset. This means that Bob cannot delete the report, even though the parent asset grants the "delete" permission.

- If there are no applicable permissions found on a particular asset, then the administrator needs to examine the `futuretense_xcel.ini` file for this permission. In the `futuretense_xcel.ini` file, permissions for a particular function are set for all assets, rather than for a single asset.

For more information about setting permissions in the `futuretense_xcel.ini`, see "To set access permissions from `futuretense_xcel.ini`," on page 119.

- Asset permissions overrides `futuretense_xcel.ini` settings.

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

Types of Authorization

Three types of authorization can be defined on a particular function and a particular asset: workflow function privileges, access permissions, and authorization properties in `futuretense_xcel.ini`, all independent of each other. (For information about workflow function privileges, see Chapter 9, "Creating and Managing Workflow Processes.")

When a user attempts to perform a function on an asset, the system performs a series of authorization checks in the following order, for that function:

For assets in workflow:

1. The system checks the workflow function privileges for the function in the workflow process in which the asset is entered.
2. 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.
3. If no access permission is set for the function (including inheritance of the parent's policy) as defined in step 2, then the system checks the authorization properties in `futuretense_xcel.ini`.
4. If none of the above types of authorization are defined, permission to perform the function on the asset is granted, by default.

Note

The above types of authorization are checked in the order shown. If two or more types of authorization are defined, the authorization that is found first overrides all other types of authorization. For example, for a specific function such as Edit, if both workflow function privileges and access permissions are defined on an asset, then function privileges on Edit override access permissions on Edit.

For assets that are not in workflow:

1. The system checks the access permissions policy for the function, as defined in the asset's access permissions interface.
2. If no access permission is set for the function (including inheritance of the parent's policy), then the system checks the authorization properties in `futuretense_xcel.ini`.
3. If none of the above levels of authorization are defined, permission to perform the function on the asset is granted, by default.

Summary

The different types of authorization described above are independent of each other, which means that only one type of authorization can apply to an asset at any given time for any given function. The following guidelines can help you decide which type of authorization to use:

1. If a function does not require authorization, do not use any of the authorization types. All roles will have permission to perform that function.
2. If you need a simple authorization policy across multiple assets, set authorization properties in `futuretense_xcel.ini`.
3. If you need a simple authorization policy on a particular asset, set access permissions.
4. If you need a more complex type of authorization that allows workflow to control the permissions on an asset, set workflow function privileges.

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 Content Server interface, to set permissions on individual assets
- Using the `futuretense_xcel.ini` file, to set permissions on all assets at once

Note

If your asset is entered in to a workflow for which function privileges are defined, the function privileges override the asset's access permissions. For information about function privileges, Chapter 9, "Creating and Managing Workflow Processes."

Setting Access Permissions from the CS Interface

To set access permissions from the Content Server interface

1. Locate the asset for which you will set access permissions.
2. In the asset's "Inspect" form, select **Access Permissions** from the **more...** drop-down menu.

3. Select one of the radio boxes, click **View by role** or **View by function**. In this example, we use **View by Function**.

Article:100-A637-2001Mar9

Preview Inspect Edit Delete more... Item is in your Active List

Name: 100-A637-2001Mar9
 Description: 100 Blank Passports Still Missing in D.C.
 Status: Received
 Modified: Mar 18, 2001 8:15:17 PM by user_author

☐ View by role ☒ View by function

Functions

	Inherited	Roles Granted	Denied
Approve for Publish Build Checkout Copy Delete Edit Inspect Preview Rollback Share Assets			
	Grant Deny	Remove	Remove

Cancel Save

Note

In this procedure, we are using **View by function**, which allows you to select a function and then specify the roles that are granted or denied permission to perform it. If you select **View by role**, the order will be reversed: you will have to select the role first, then specify the functions it is granted or denied.

4. Select a function.
- Roles that can perform the function are now displayed in the “Inherited” list box, by default. “Inherited” refers to the permissions policy that the asset inherits from its parent, or its parent’s parent, and so on, depending on which asset in the hierarchy has a policy.
5. Select a role (or roles) to which you will deny or grant the permission to perform the function. (To select a block of roles, **Ctrl-Shift-click** the extremes of the block. To select non-adjacent roles, **Ctrl-click** each role).
6. Select either **Grant** or **Deny** to allow or forbid the role(s) to perform the function:
- **Grant** explicitly allows the selected role(s) to perform the selected functions on the specific asset.
 - **Deny** explicitly forbids the selected role(s) to perform the selected function on the specific asset.

- Selecting neither **Grant** nor **Deny** leaves the roles in the “Inherited” list box to indicate that access permissions are inherited.
7. Repeat steps 4 to 6 for each function and role that must be granted or denied or which you want to grant or deny permission.

When you have completed selecting functions and roles, your access permissions screen will look similar to the one below:

Article:100-A637-2001Mar9

Preview Inspect Edit Delete more... Item is in your Active List

Name: 100-A637-2001Mar9
 Description: 100 Blank Passports Still Missing in D.C.
 Status: Received
 Modified: Mar 18, 2001 8:15:17 PM by user_author

☐ View by role ☒ View by function

Functions	Inherited	Roles Granted	Denied
Approve for Publish	Analyst	Designer	HelloDesigner
Build	Approver	Editor	
Checkout	Author		
Copy	Checker		
Delete	Expert		
Edit	HelloAuthor		
Inspect	HelloEditor		
Preview	Marketer		
Rollback	Pricer		
Share Assets	SiteAdmin		

Grant Deny Remove Remove

Cancel Save

8. Click **Save** when finished.

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

Setting Access Permissions from Property Files

To set access permissions from `futuretense_xcel.ini`

In `futuretense_xcel.ini`, either grant or deny permissions by setting the following properties:

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

where

functionname is the name of the function, as shown in the `futuretense_xcel.ini` file, on the “Authorization” tab if you are using the Property Editor.

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 the `futuretense_xcel.ini`, it is granted or denied for all assets. These permissions can be overridden by specifically setting access permissions on an asset-by-asset basis, using Content Server's access permissions feature.

Configuring Other Options for Asset Types

There are several other items that configure asset types on the **Admin** tab. These items contribute to the design and implementation of the asset types, which means that they are typically used by the developers. Administrators typically do not manage the items.

However, after an asset type is designed and installed, you might be asked to maintain the following items:

- sources
- categories
- subtypes
- associations
- mimetypes

For information about these items and procedures for creating and editing them, see the section called “Data Design” in the *Content Server Developer's Guide*.

Managing Access to Content Server's Tree

You have several ways to configure the tree that is displayed in the left-hand panel of the Content Server interface:

- You can configure the tree to be either displayed by default in the main window or hidden by default.
- You can control whether users who toggle a tree off are allowed to toggle the tree back on.
- You can configure tabs within the tree, adding items or functionality to them, and determining which users have access to them based on their roles.
- You can configure the tabs' nodes to specify the number of items that are displayed under any given node.

Content Server supports two general categories of tree tabs:

- System default tabs, which provide administrative functionality (Admin, Site Admin, Workflow) and access to individual assets (History, Active List, Site Plan).
- Custom tabs
 - Tabs that you create to make it more convenient for users in different roles to create the most commonly used asset types.
 - Tabs that developers create to offer customized behavior or new functionality.

Note

The Workflow Groups tab is a hybrid between a system default and a custom tab. Although the Workflow Groups tab is not installed by default and you must create it to use it, the element that provides its logic is installed. That is, while you must manually create this tab, your developers do not have to code a new element for you.

This section describes how to add roles to the system default tabs, how to create basic tree tabs that provide access to asset types, and how to configure whether the tree is displayed by default.

For information about creating custom tree tabs that implement new functionality, see the section called "Management System Features" in the *Content Server Developer's Guide*.

Displaying and Hiding the Tree

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

If you want, you can configure your CS system so that the tree is toggled off by default when users first log in to the system. You can also control whether users are allowed to toggle the tree back on.

The CS-Direct features for working with assets are available in right-mouse menus on the tabs as well as through the icons at the top of the window and the action bars displayed in the **New**, **Edit**, and **Inspect** forms for individual assets. Therefore, it is not necessary to display the tree for content providers. However, because the administrative functions are

available only through the **Admin**, **Site Plan**, and **Workflow** tabs, administrative users must always have access to the tree.

To configure display of the tree and toggling rights

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. Select the **Preference** tab.
3. To specify that the tree should be toggled off by default when users first log in to the Content Server interface, set the value of the `xcelerate.showSiteTree` value to `false`.
4. (Optional) To configure the system so that only administrative users (that is, users who have the `xceladmin` ACL assigned to their user accounts) are allowed to toggle the tree back on, set the `xcelerate.restrictSiteTree` property to `true`.
5. Select **File > Save**.
6. Select **File > Close**.
7. Stop and restart the application server.

Creating Tree Tabs


































This section shows you how to create new tree tabs, and the Workflow Groups tab.

Creating a New Tree Tab

To create a new tree tab

1. In the Content Server interface, select **Admin > Tree**.
CS-Direct displays a list of the tabs.

Tree Tabs

	Title	Tooltip
  	Site Plan	Site Plan Tab
  	Admin	Administrative Functions
  	Site Admin	Site Administration
  	Workflow	Workflow Administration
  	Active List	Active List
  	Query	Query Tab
  	Design	Administrative Assets
  	Hello Content	Hello Content
  	Hello Design	Hello Design
  	Product	Product Catalog
  	Content	Content Catalog

Add New Tree Tab

Order Tree Tabs

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

Add New Tree Tab

***Title:**

Tooltip:

***Sites:** Any
HelloAssetWorld
BurlingtonFinancial
GE Lighting
Management Site

***Required Roles:** Any
Analyst
Approver
Author
Checker

Tab Contents:

Available: Article
Article (Flex)
Attribute Editor
CSElement
Collection
Content Attribute
Content Definition
Content Parent

Selected:

Display Order: ▲▼

▶ Add Selected Items Remove ◀

Choose the tab contents from above and/or make a new section with a custom element

Section Name:

Element Name:

▶ Add New Section

- a. Click in the **Title** field and enter a short, descriptive name of up to 64 characters.
 - b. Click in the **Tooltip** field and enter a short, descriptive phrase of the tab, using up to 255 characters. This phrase is displayed when the mouse hovers over the tab in the tree.
 - c. In the **Site** field, select the sites that will display the tab.
 - d. In the **Required Roles** field, select which roles can access the tab.
 - e. In the **Tab Contents** field, select the asset types that will be displayed on the tab and then click **Add Selected Items**.
 - f. (Optional) If you want to add custom functionality to this tab, use the **Section Name** and **Element Name** fields at the bottom of the form. You need help from your developers for this step. See the *Content Server Developer's Guide* for more information.
3. Click **Save**.

The new tab is displayed in the tree (refresh the display, if necessary).

Creating the Workflow Groups Tab

The Workflow Groups tab displays the workflow groups that exist for a site and the assets that are included in those groups. For information about workflow groups, see “Workflow Groups,” on page 132. This tab does not appear by default. If you want to use it, you must create it.

To create the Workflow Groups tab

1. In the Content Server interface, select **Admin > Tree**.
CS-Direct displays a list of the tabs.
2. In the “Tree Tabs” form, click **Add New Tree Tab**.
3. Click in the **Title** field and enter a short, descriptive name of up to 64 characters. For example: Groups
4. Click in the **Tooltip** field and enter a short, descriptive phrase using up to 255 characters. For example: **Workflow Groups**.
5. In the **Site** field, select the sites that will display the tab.
6. In the **Required Roles** field, select which roles can access the tab.
7. In the **Tab Contents** area, scroll down to the section area. In the **Section Name** field, enter the following name exactly as it appears here: Groups.
8. In the **Element Name** field, enter the following name exactly as it appears here:
OpenMarket/Gator/UIFramework/LoadWorkflowGroups
9. Click **Add New Section**.
The word “Groups” appears in the **Selected** column.
10. Click **Save**.

Editing Tree Tabs

After you create roles for your sites, you must edit the following system default tabs to add your roles, so that your users have access to the tabs:

- Site Plan
- History
- Active List

To edit a tree tab

1. In the Content Server interface, select **Admin tab > Tree**.
2. In the tree tab list, click the **Edit** icon for the tab that you want to edit.
3. In the “Edit Tree Tab” form, make the necessary changes. For example, select your roles from the **Roles** list.
4. Click **Save**.

Be sure to add your roles to all of the default, system tabs so that your users have access.

Deleting Tree Tabs

To delete a tree tab

1. In the Content Server interface, select **Admin tab > Tree**.
2. In the tree tab list, click the **Delete** icon for the tab that you want to edit.
The system displays a confirmation message.
3. Click **Delete Tree Tab**.

Changing the Sort Order of the Tree Tabs

You can change the order in which the tree tabs are displayed.

To change the order of tree tabs

1. In the Content Server interface, select **Admin > Tree**.
2. At the bottom of the list of tree tabs, click **Order Tree Tabs**.

CS-Direct displays a form like this one:

Order Tree Tabs

The dialog box titled "Order Tree Tabs" contains a section labeled "Tab Order:" with a list of the following tabs: Site Plan, Admin, Site Admin, Workflow, Active List, Query, Design, Hello Content, Hello Design, Product, and Content. To the right of this list are two arrow buttons (up and down) under the label "Display Order:".

Cancel

Save

3. Select a tab from the list and then use the arrow buttons to move the tab to the correct position.
4. Repeat step 3 for each tab that you want to reposition.
5. Click **Save**.

Configuring the Number of Items Displayed Under Nodes on Tabs

To optimize Content Server's response time, you may want to limit the number of asset names that are displayed under their respective node in a tree tab. Specifically, when the Content Server database holds many assets of any one type, Content Server can take a long time to load all the assets and then display the asset names under their node in the tab. To minimize the loading time, you can limit the number of items to be displayed under the node by setting a threshold. Then, when the number of assets or other items for a node exceeds the threshold, and a user attempts to access the assets, the Content Server interface prompts the user to enter search criteria that restricts the number of items.

For example, suppose that a node contains more than 300 articles, and the display threshold is set to 100. When a user expands the node icon that represents the articles, Content Server displays a small search criteria form that prompts the user to enter criteria that Content Server uses to limit the number of article assets that it returns.

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

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. Select the **Preference** tab.
3. Select the `xcelerate.treeMaxNodes` property and specify a value for it. By default, this property is set to 100, which means that up to 100 items can be displayed under a node.
4. Select **File > Save**.
5. Select **File > Close**.
6. Stop and restart the application server.

Chapter 9

Creating and Managing Workflow Processes

Workflow is a feature provided by CS-Direct that you use to manage the work on an asset when more than one person participates in its creation. For example, if assets of a certain type must be reviewed by an editor or a legal representative before it can be approved for publishing, the workflow feature can route those assets to the appropriate people at the appropriate time.

This chapter contains the following sections:

- Overview
- Planning Your Workflow Processes
- Sample Site Workflow Processes
- Configuring Your Workflow Processes
- Moving Your Work
- Clearing Workflow Assignments

Overview

The end goal for any content provider is to have content assets published. Before an asset can be published, it must be approved for publishing. The workflow feature routes assets through whatever series of tasks you deem necessary in a **workflow process** that ushers assets from creation to approval.

You can configure a workflow process with as many or as few tasks as necessary to reflect the way the work at your organization is accomplished. You can configure e-mail messages that CS-Direct sends to notify people when assets are assigned to them and to remind them that a deadline is approaching or has been missed.

Because there are so many configuration possibilities, it is typical to create a separate workflow process for each asset type that you plan to use workflow with rather than attempt to create one process for more than one asset type.

Workflow Participants

When you begin creating a workflow process, the first general question is this: what are the job titles of the people who work on assets of this type? For example, are they authors, editors, marketers, graphic artists, product managers, lawyers?

The job titles of the people who participate in a workflow process are considered **roles** in the Content Server interface. 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**.

Workflow States

Next, what are the tasks that are performed for assets of this type? For example, writing, pricing, editing, fact checking, legal review, and so on.

These tasks are called workflow **states**. A state is a point in the workflow process that represents the status of the asset at that point. For example, writing article, reviewing image, legal review, and so on. Participants complete the work that the state represents while the asset is in that state.

An asset that a participant is working on (or is supposed to be working on) is called an **assignment**. A user's **assignment list** is displayed on the "My Work" form in the Content Server interface. An asset appears on a user's assignment list as soon as it enters a state for which the user has a role to fulfill as a workflow participant.

Should an asset in a specific state remain in that state for only a specific amount of time? If so, assign a deadline to the state. You can then configure the workflow process to send e-mail messages that remind a participant when the deadline is approaching or has been missed. These e-mail messages are examples of **timed actions**.

You can create one or more timed action for each state.

Workflow Steps: Moving Assets from State to State

Next, how does the asset move from state to state? Does a marketing writer send a prospectus asset to a legal reviewer? Does a graphic artist send an image asset to an editor? And then what?

The movement of an asset between states is called a **step**. Because creating the steps in a process links together states in a specific order, creating the steps in your workflow process is what organizes your process.

When you create a step in a process, you specify which state a step moves the asset from (the **From State**), which state that step moves the asset to (the **To State**), and which roles can take the step.

How Steps Work

A step places an asset in a state and notifies participants from the appropriate roles. This operation creates the assignment. You then need a step for the participants from those roles to take that moves the asset to the next state. In other words, the roles notified by the previous step should be the roles authorized to take the next step. For example:



How does a user take a step in a workflow process? By specifying that he or she has finished the assignment (except for the start step, which is described below). When a user selects the **Finish My Assignment** option for an asset that is assigned to him or her, that option invokes the step, moves the asset to the next state, and assigns the asset to the next participants.

When more than one participant is assigned an asset in the same state, using the **Finish My Assignment** option is also referred to as **voting**. Each participant “votes” to move the asset to the next state.

The first step in a workflow process is a **start step**. A start step is one which has no From State. That is, the start step begins the workflow process, moving the asset to the first state in the series of states. This is the step that is invoked when the workflow process is assigned to the asset. Only users who have the roles authorized for the start step can start the workflow process for an asset.

Step Actions and Conditions

Step actions are events that occur when the step is completed. For example, when a step occurs, the asset is assigned to a participant. Should the participant be notified that an asset has been assigned to him or her? If so, you can configure a step action to send an e-mail message to the new participant. You can specify one or more step actions for each step.

Another example of a step action is the **ApproveForPublish** action. It is a default action delivered with CS-Direct that approves the asset. Typically you use this action in the final step of a workflow process.

Finally, are there any requirements that must be met for an asset in a state before the step can move the asset to the next state? If so, configure and assign a **step condition** to the step. For example, you could configure a condition that verifies that the asset has all of its

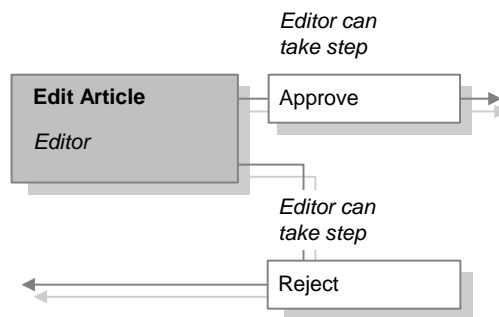
association fields filled—that an article has the associated images that it needs—before it progresses to the next state.

Multiple Paths for the Asset

When a workflow process includes a state in which people are reviewing an asset, typically there is more than one path possible for an asset in that workflow because the reviewer can either accept it the way it is or reject it.

In such a case, you create two steps for moving the asset from the review state. When the asset is in that state, the “Finish My Assignment” form lists both options and the participants select the appropriate step when they finish their assignments.

For example:



What happens if more than one participant are reviewing the same asset in the same state and they choose different steps (that is, vote differently)? That depends on how you configured each step. There are several possibilities:

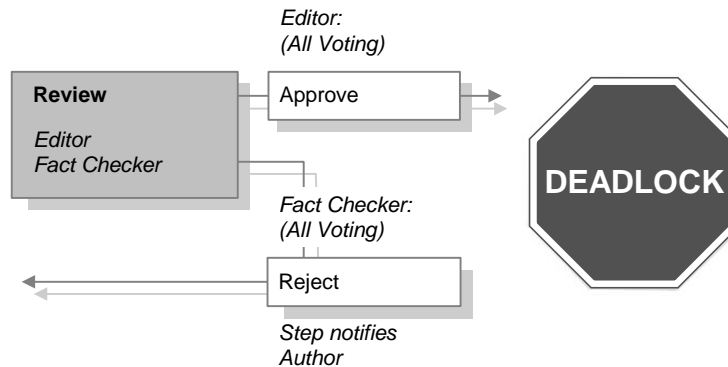
- Configure the steps so that the first participant to finish the assignment (vote) determines the direction the asset takes.
- Configure the steps so that all participants have to select the same step (vote the same way) in order for the asset to progress in either direction. A step that all the participants must select before that step can be completed is called an **all-voting step**. Note that this option can result in **deadlocks**, described in the next section.
- Use a combination of the preceding possibilities: configure one all-voting step (all participants must agree), and configure the other step(s) so that if any participant selects it, the step completes and the asset takes that path.

Managing Deadlocks

A deadlock occurs when the following conditions are true:

- There is more than one step from a state.
- Two or more of the steps require the participants to perform that step. That is, they are all-voting steps.
- An asset in that state is assigned to more than one participant.
- The participants select different all-voting steps when they finish the assignment.

This diagram illustrates a deadlock:



Note that if even one of the steps is not all-voting and a participant selects that step, the asset will not become deadlocked.

Resolving Deadlocks

An asset that is in a deadlocked state cannot progress through the workflow process until the deadlock is resolved. To resolve the deadlock, the participants must confer with each other and agree on that path that the asset should take. Then, the participants who must change their selection can do one of the following to resolve the deadlock:

- Finish the assignment and select the step that they all agreed to take.
- Select the **Abstain from Voting** option from the **Workflow Commands** drop-down list on the asset's "Status" form.

Preventing Deadlocks

Before configuring a workflow process that can result in a deadlock, be sure that it is absolutely necessary to have complete agreement on all the possible steps from the state. As you can see from this description, deadlocks cause additional work for all the participants so be sure that you use this feature only when you need to.

For example, consider a review state with two possible steps: "return for revisions" and "approve for publish." If you configure the steps so that "return for revisions" does not require a unanimous vote but "approve for publish" does, you have created a desirable control—all the reviewers must agree before the asset can be published and any rejection stops the asset from being published—without risking a deadlock.

Notifying Participants When Deadlocks Occur

If you do need to create a workflow process that can result in a deadlock, be sure to configure and assign a **deadlock action** that notifies the participants of the deadlock for each of the steps that can cause the deadlock. The default deadlock action is an e-mail message that CS-Direct sends to the appropriate participants when a step causes a deadlock.

Workflow Groups

Is there ever a situation in which several assets are so closely connected that they need to be thought of as one unit of work or they need to be approved at the same time? In such a case, you can use the **workflow group** feature.

Using Workflow Groups

Workflow groups enable content providers to send a defined set of assets through the workflow together. While it is the content providers who create workflow groups and select the assets that are assigned to the group, you, the administrator, still need to know what kind of assets will be included in workflow groups. Why? So that you can configure the workflow processes appropriately.

For example, you can configure workflow steps that allow each asset in the group to progress to the next state when it is finished or you can configure a step that requires all the assets in the group to reach that point before any of them can progress. (This second example, called a **synchronize step**, is described next.)

Adding a Synchronize Step

When creating a workflow process that will be used with workflow groups, it is usually best to configure the process so that it has only one synchronize step. Multiple synchronize steps can slow down the work on those assets unnecessarily. Assess the business process that is reflected by the workflow process and determine which steps must truly be synchronized: perhaps all the assets should go to legal review in one batch, or perhaps all the assets should be approved for publishing at the same time, for example.

Managing Group Deadlocks

If you create a workflow process to be used with workflow groups and any of the steps can result in a deadlock, be sure to configure and assign a **group deadlock action** that notifies participants when there is a group deadlock and assign it to the process.

Delegating and Clearing Assignments

People go on vacation, get reassigned to new work groups, and move on to different jobs. What happens to the assets that they are working on? They can **delegate** their assignments to other participants who have the appropriate roles.

Additionally, each workflow process can have a **workflow administrator**. The administrator of a workflow process can delegate assignments on behalf of the other participants.

When an asset is delegated to a new participant, should that person receive an e-mail notice? If so, configure a **delegate action** to send an e-mail message to new assignees when assets are delegated to them. You can specify one or more delegate actions for each workflow process.

If an asset no longer needs to be assigned or if it is easiest to clear the assignment and then start over, you can use the **Clear Assignments** feature on the **Admin** tab.

Placing an Asset in Workflow

An asset begins its participation in a workflow process in one of the following ways:

- A user selects a workflow process from the **Workflow Commands** field on the “Status” form for the asset.

Selecting the workflow process invokes the start step for the process, which places the asset into the first state.

- A user creates an asset and the start menu **New** item for assets of that type is configured such that there is a default workflow process.

In this case, saving the asset invokes the start step for the process, which automatically places the asset into the first state.

Because steps are enabled for specific roles, only the users who have a role that is assigned to the start step of a process can select that process. This means that if you are using start menu items to place assets in workflow, you must be sure that the roles assigned to the start menu item are the same roles that are assigned to the start step of the default workflow.

Note

Versions of CS-Direct prior to Version 4 assigned a default workflow to assets through a property in the `futuretense_xcel.ini` file. The ability to assign a default workflow through a start menu item has replaced that method of determining a 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, the fact that the asset is assigned to a specific user does not stop other users from modifying or even deleting that asset.

If you want to restrict who has access to an asset while it is in workflow, use the workflow feature called **function privileges**. These are restrictions set on functions such as edit, copy, approve, delete, show versions, and so on in the context of workflow states and workflow roles.

There are three parts to a function privilege:

- The function being restricted.
- The roles allowed or not allowed to perform the function.
- The state during which users with those roles are allowed or not allowed to perform the function.

When a function privilege is in effect, it means that a user can perform that function only when the following conditions are true:

- The user has an appropriate role.
- The asset is in the correct state.
- The asset is **assigned** to the user.

This means that even if the user has the correct role and the asset is in the correct state, the user cannot perform that function on that asset unless the asset is assigned to that user.

Function Privileges and Step Actions

Function privileges restrict access to a function from the user interface only. This means that you can program step actions that invoke a function when a step is taken regardless of what the function privilege is set to at that moment.

The ApproveForPublish step action is an example. Even if you specify function privileges that restrict users from using the **Approve** option in the user interface, those same users can approve an asset with a workflow step if the workflow step invokes the ApproveForPublish action and the user has the correct role to take the step.

In other words, you can use function privileges to prevent users from selecting and changing assets by mistake and use actions to invoke those functions in a highly controlled way.

Function Privileges: All or None

You cannot create just one function privilege: if you create even one function privilege that allows or restricts access to a function, you must create function privileges that cover all the other possible conditions for your workflow process.

For example, suppose that the only function that you want to restrict is the Delete function—you want to allow only the editors to delete article assets and only then if the article is in the Review state. If you create a function privilege that allows editors to delete article assets while it is in the Review state but you do not create any other function privileges, the only task that can be completed for an article when it is in the workflow is that editors can delete it while it is in the Review state. That is, no one can edit it, approve it, copy it, or even finish their assignments.

To implement the preceding condition—the Delete function is accessible to editors only when the article is in the Review state—you need to create the following function privileges:

- Delete – allowed for editors when the asset is in the Review state.
- Edit – allowed for all roles in all states.
- Abstain from Voting – allowed for all roles in all states.
- Copy – allowed for all roles in all states.

Continue down the list of functions, until you have at least one privilege specified for each function privilege listed in the “Functions” form for the workflow process.

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.

Setting and Overriding State Deadlines

When you create a workflow state, you can set an Estimated Time for it and determine whether reminder e-mails 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.

Setting Process Deadlines

When you create a workflow process, you determine whether a process deadline can be set. A process deadline is set when an asset is first placed in workflow, in the “Select Workflow” form. Unlike an assignment deadline, however, you cannot configure the process to send reminder e-mails 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 invoked by a step. These actions are events that CS-Direct completes when a step is taken.
- Actions that are triggered by a deadline. These actions are queued and are triggered only after CS-Direct calculates the deadlines for the assets and determines which timed actions (if any) should be invoked.

You specify how often the deadlines are calculated by configuring the **Timed Action Event**. This is an event that invokes a background calculation process of all deadlines. It is similar to the publishing event that invokes the background approval calculation process.

Just as the publishing process calculates approvals to determine which assets should be published, the deadline calculation process that is invoked by the **Timed Action Event** calculates the deadlines for all assets that are participating in workflow processes—to determine whether any reminder messages should be sent for assignment deadlines and to determine the times that should be displayed for assets in the Due and Process Deadline columns of assignment lists.

You can configure the **Timed Action Event** on your management system to run as often as you find it necessary.

How Does a Workflow Process End?

A workflow process ends when there are no more states for the asset to progress through. This occurs when the final participant takes the **end step** for the workflow process.

An end step is the opposite of a start step—it has a From State but no To State. When a user takes the end step, it is moved to a “stateless” state, which means the asset is no longer in workflow and any function privileges set for that workflow process no longer apply.

Roles Required to Configure Workflow Processes

The workflow building blocks are located on two tree tabs in the Content Server interface:

- **Admin** – holds e-mail objects, actions, conditions, and the **Timed Action** Event.
- **Workflow** –holds workflow states and processes.

For access to the **Admin** tab, you must be assigned the GeneralAdmin role and have the xceladmin ACL assigned to your user account. For access to the **Workflow** tab, you must be assigned the WorkflowAdmin role for the site you want to create workflow processes for.

For more information about access rights in the Content Server interface, see Chapter 4, “Working with ACLs and Roles” and Chapter 5, “Configuring Users, Profiles, and Attributes.”

Planning Your Workflow Processes

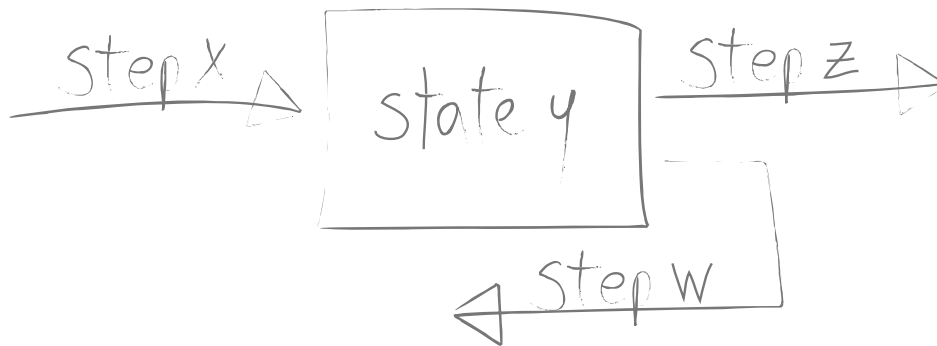
When you create a workflow process, you create steps that link together the states for that process. This means that you or someone else must create the workflow components—roles, e-mail messages, the various kinds of actions, step conditions, and states—before you can create a workflow process.

This section provides more details about the configuration of each of the workflow components so that you can plan and implement your workflow processes.

Start with a Sketch

Where do you begin? With sketches of the business processes that you need to implement as workflow processes:

- Use boxes to represent the states.
- Use arrows to represent the steps that connect the states.



As you read through the descriptions in this section, write on your sketches the details about which roles, actions, conditions, deadlines, and so on are appropriate for each state or step in the process.

Then, refer to your sketches as you use the Content Server interface to create your workflow processes, described in “Configuring Your Workflow Processes,” on page 159.

Determine Roles and Participants

When you start planning your workflow processes, begin with the roles. What kind of functional groups participate in workflow? Then, determine how the individual users from those roles will become the participants in a workflow process for a specific asset.

Planning Roles

To determine the roles that you should create for your processes, ask yourself the following questions:

- What are the job titles or roles of the content providers who are using your CS management system?
- What do the people in each role do? (You can map the tasks that they complete to your states.)

- How are the roles organized? For example, is there one group of reviewers who review everything but several groups of content providers who are organized by subject (for example, sports writers, financial writers, marketing writers, and so on)
- Do certain groups of people work with specific asset types but not with others? If so, which roles should have access to each asset type that you plan to create a workflow process for?
- Would you ever need to notify someone who is not participating in a workflow about the status of an asset that is in workflow? If so, that person will need a role so a step or timed action can send an e-mail message to that person.

On your sketches of your workflow processes, write the names of the roles that will have the asset assigned to them in each state.

For information about creating roles, see Chapter 4, “Working with ACLs and Roles.”

Selecting Individual Participants

When you configure a workflow process, there are several ways to determine which specific users participate in the workflow for each individual asset that goes through the workflow:

- When the workflow is assigned.
You can configure the process so that the person who first assigns the workflow to the asset must select all the users who will participate. For each state, they select users from a list. The list includes all the users who have the correct role for that state.
- When a participant finishes the assignment.
You can configure the process so that each participant determines who the next participant is when he or she finishes the assignment. That participant selects a user name from a list that includes all the users who have the correct role for the next state.

You specify how participants will be selected when you configure the steps.

Cross-Site Assignments and Participants

Assets can be shared between sites. If a shared asset is entered into a workflow process, should users from all the sites that have access to the asset be considered as candidates for participating in the workflow? If the answer is yes, enable the cross-site assignments feature.

When you use the cross-site assignments feature, users see all of their assignments from all the sites that they have access to in their assignment list no matter which site they are currently logged into. Having one, consolidated assignment list is very convenient for your users.

Keep in mind, however, that if your users have different roles in different sites and you are using function privileges in your workflow processes, they might not be able to work on an asset that they can see in their assignment lists. For example, say that some of your users have the author role in one site and the editor role in another. If you are using function privileges to restrict editing to editors in a certain state, they can see an asset that they aren't allowed to edit in their assignment list when they are logged in to the site where they function as authors.

To enable this feature, you set the value of the `xcelerate.crosssiteassign` property to true. This property is in the `futuretense_xcel.ini` property file. For information about the property, see the *Content Server Property Files Reference*.

Determine the E-mail Objects, Actions, and Conditions

An **action** is an event that is triggered in one of three ways:

- A step invokes it (step action).
- A deadline triggers it (timed action).
- A workflow situation triggers it (deadlock, group deadlock, and delegate actions)

A **condition** is an event that is assessed when a step is attempted. If the condition is not met, the step cannot be completed.

When you create an action or a condition, you identify an element. **Elements** are named pieces of code that are stored in the `ElementCatalog` table. It is the element that invokes the function represented by the action. If the element is coded to expect variables or arguments, you identify values for them when you create the action and those variables are then passed to the element when the action is triggered.

As an administrator, you are not responsible for coding elements. If your workflow needs cannot be met by the default workflow elements that are provided, your developers can code the functionality that you need. See the *Content Server Developer's Guide* for information about customizing workflow.

CS-Direct provides the following default workflow elements:

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 e-mail object to send.
OpenMarket/Xcelerate/Actions/Workflow/StepConditions/ExampleStepCondition	
OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail	emailname The name of the e-mail object to send.
OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees	emailname The name of the e-mail object to send.
OpenMarket/Xcelerate/Actions/Workflow/GroupActions/SendEmailToAssignees	emailname The name of the e-mail object to send.

These elements are described in detail in the *Content Server Developer's Guide*.

With the exception of `ApproveForPublish` and `ExampleStepCondition`, these elements take a variable called `emailname` and send the e-mail message that is identified by that variable.

How do you determine the value of the `emailname` variable? By creating workflow **e-mail objects**. The names of the e-mail objects that you create on the **Admin** tab in the

Content Server interface are the names that you specify as the arguments with the `emailname` variable when you create actions that send e-mail messages.

About E-Mail Objects

E-mail objects are building blocks separate from the actions so that you can use them with more than one action. For example, the default deadlock action and the default group deadlock action use the same e-mail message (named Deadlock Message).

You create e-mail objects by giving them a name, a description, a subject line, and text for the body. When the message is sent, the text provided for the subject is placed in the subject line and the text provided for the body is placed in the body of the message.

Workflow E-Mail Variables

There are several variables that you can use in the subject line and body text which makes it easier for you to write e-mail messages that are personalized for each recipient.

The following table lists the default workflow variables that you can use with any e-mail message:

Variable Name	Description
<code>Variables.assetname</code>	The name of the asset. Use this variable in every e-mail message so the recipient knows which asset is being referred to.
<code>Variables.assigner</code>	The user name of the participant who assigned the asset to the person receiving the e-mail. Use this variable in e-mail messages for step actions that notify participants that a new asset has been assigned to them.
<code>Variables.time</code>	The time specified in the Estimated Time field for a state. Use this variable for timed actions.
<code>Variables.instruction</code>	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 e-mail messages for step actions that notify participants that a new asset has been assigned to them.

For examples of custom e-mail variables, see the e-mail object named Deadlock Message and the corresponding deadlock action element named `OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees`.

Default Workflow E-Mail Objects

The default workflow e-mail objects are these:

- Assignment Due Reminder – specifies the asset and the time that it is due.
- Assignment Message – specifies the asset, the person who assigned it, and in the message from the **Action To Take** box on the “Finish My Assignment” form.

- **Deadlock Message** – describes how the deadlock occurred by listing the users and the steps that they took.
- **Rejection Message** – is similar to the Assignment Message, but states that the asset was rejected by the previous participant (the assigner).

Planning Your E-mail Objects

To determine the e-mail objects that you need to create, you must also determine the actions that will send the e-mail messages held in the objects. By using the e-mail variables in the subject and body, it is likely that you can use the same e-mail object with more than one timed action or step action.

Typically, you need to compose e-mail 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 Content Server interface, select **Admin > Email**. Examine the default e-mail objects to determine whether you can use them. You can modify the default messages or create your own e-mail messages.

In one corner of your sketches of the workflow processes, list the e-mail messages that you will need for that workflow.

About Timed Actions

Timed actions are actions that are based on the deadline of a state. If you specify a deadline for a state, you can specify a timed action to remind the participants about the deadline.

You create a timed action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

To create a timed action that sends e-mail notices about a deadline, specify the `OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail` element and use the **Argument** field to specify which e-mail message to send.

When you create a state, you specify which timed actions to use and when to trigger them, relative to the deadline specified for the state. You can specify more than one timed action for each state.

When the Timed Action Event runs and calculates the deadlines for all the assets currently participating in workflow, the appropriate timed actions are triggered.

Default Timed Actions

There is one default timed action: Send Email. It uses the `OpenMarket/Xcelerate/Actions/Workflow/AssignmentActions/SendEmail` element to send the Assignment Due Reminder e-mail message.

Planning Timed Actions

When planning your timed actions, you must also consider the following workflow components:

- The states that you will create, because these actions are triggered in terms of the deadlines that you set for your states.
- The e-mail objects that you need to create, because it is likely that your timed actions will send e-mail 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 e-mail message they send—and use them repeatedly with your states.

In the Content Server interface, select **Admin > Workflow Actions > Timed Actions**. Examine the Send Email action to determine whether you can use it. You can modify the default timed action or create your own.

On each of your sketches of your workflow processes, list the timed actions that you need for that process near the list of the e-mail messages.

About Delegate Actions

When you assign a delegate action to a workflow process, the action is triggered whenever a participant (or the workflow administrator) delegates an assignment to another participant.

You create a delegate action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

You can specify one delegate action per workflow process.

Default Delegate Action

There are no default delegate actions provided.

Planning Delegate Actions

It is likely that your delegate action will send an e-mail 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 e-mail and create a new e-mail 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.)

About Step Actions

When you assign a step action to a step, the action is triggered when a participant takes the step.

You create a step action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

There are two general categories of step actions: those that complete functions and those that send e-mail 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 CS-Direct function, the action does not check the function privileges of the participant taking the step. This means that you can restrict access to a

function from the user interface and require participants to use a step in a workflow in order to complete a specific function (approve for publish, for example).

You can specify one or more step actions for each step.

Default Step Actions

The default step actions are these:

- **Approve for Publish**, which uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/ApproveForPublish` element to approve the asset. Then, the next time a publishing process runs, the asset is published (as long as all of its dependencies are also approved).

To use this action for your workflow processes, you must specify the publishing destination(s) that the asset is to be approved for by using the `targets` argument.

- **Send Assignment Email**, which uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees` element to send the Assignment Message e-mail 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 e-mail message to the new assignee (the participants specified in the **Assignment Method** for the step).

Planning Step Actions

To determine what kind of step actions you need to create, you must consider the following workflow components:

- The steps that you will create for the process. Do people need to be notified that a step has assigned an asset to them? If so, you need a step action that sends an e-mail message.
- The states that the steps move the assets into. Does the state represent the asset after a function has been completed? If so, you need a step action that implements that function.

In the Content Server interface, select **Admin > Workflow Actions > 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.

About Step Conditions

When you assign a step condition to a step, the condition is assessed when the step is taken. The condition determines whether the step can be completed or not.

You create a step condition by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary. If you want to use conditions in your workflow processes, talk to your developers about the conditions you need them to implement through elements.

You can specify one or more conditions for each step.

Default Step Conditions

There is one default step condition: Example Step Condition. It is a “hello world”-style example that illustrates how to code an element to check for a condition. You and your developers should examine it to learn how it works and then your developers should create their own condition elements.

Planning Step Conditions

When thinking about step conditions, you must consider the steps and the state. If you determine that conditions are necessary for your workflow processes, ask your developers to create them.

On each of your sketches of your workflow processes, write the name of any step conditions that you need next to the appropriate steps.

About Deadlock Actions

When you assign a deadlock action to a step that can result in a deadlock, the action is triggered whenever an asset becomes deadlocked during the step.

You create a deadlock action by giving it a name and a description, specifying an element, and then providing argument values for the element, if necessary.

You can assign one or more deadlock actions per step.

Default Deadlock Action

There is one default deadlock action: Send Deadlock Email. It uses the `OpenMarket/Xcelerate/Actions/Workflow/DeadlockActions/SendEmailToAssignees` element to send the Deadlock Message e-mail message.

Planning Deadlock Actions

When thinking about possible deadlock actions, you need to consider the steps that you will create for your workflow processes. If you plan to design your steps so that deadlocks cannot occur, there is no need to create any deadlock actions.

In the Content Server interface, select **Admin > Workflow Actions > 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.

About Group Deadlock Actions

A group deadlock action is triggered when workflow group becomes deadlocked. Whoever creates the workflow group selects the group deadlock action. That person can select one or more for each group.

The group deadlock actions are invoked in addition to any other actions that are associated with the states or the steps in the workflow process.

Default Group Deadlock Action

There is one default group deadlock action: Send Deadlock Email. It is identical to the default deadlock action.

Planning Group Deadlock Actions

The same considerations apply for group deadlock actions as for deadlock actions. However, because it is a content provider who selects the group deadlock action for a workflow group, be sure that you give a group deadlock action a meaningful, unambiguous name.

Determine the States

When you create a state, you specify the following kinds of information:

- Name and description. The name of a state should be meaningful and should represent the kind of work that is being done while the asset is in that state.
- Amount of time an asset should remain in this state (the **Estimated Time**). The deadline is calculated in terms of the number of hours or days since the asset entered the state.

Note that you set the estimated time (the deadline) in terms of hours or days rather than as a specific date because a state deadline is calculated for each asset as it passes through the state.

- The timed actions that should occur and when they should occur, in terms of days or hours before or after the deadline.

There are no default states, although there are example states provided with all of the sample sites. On a system where the sample sites are installed, select **Workflow > States** and then select a state to examine to learn about how the sample site states are configured.

Planning Your States

To determine the states that you need to create, ask yourself the following kinds of questions:

- Which roles participate in each state? On your sketches, write the names of the roles next to the states.
- Do any of these states apply to more than one asset type? If so, it's possible that you can reuse the same state in more than one workflow process.
- How long should it take to complete each state?
- Does this amount of time differ by asset type or by site? If yes, you cannot reuse the same state in more than one workflow process or share the workflow process with another site.
- If there is a deadline, should CS-Direct notify the participant when the deadline is approaching? If yes, when? The day before? Several hours before? And which timed action (which determines which e-mail message) should be used?
- Should there be a notice when a deadline is missed? If yes, when? And which timed action should be used?

On the sketches of your workflow processes, list all of this information next to each state.

Depending on your goals, you might consider creating a workflow process that does not really end. For example, the Hello Asset World sample site workflow process ends after a HelloArticle asset has been approved. However, in a case like this, someone could open and save the asset by mistake, which would mean that it is no longer approved and won't get published.

To keep assets from being edited after they have been approved and before they have been published, you could create a final state that holds assets after they have been approved with a function privilege on the state that restricts anyone from editing the asset.

Be sure that if you do create a state like this that you create a step that can move the asset back into an editing state in the workflow so someone can edit it on purpose.

Determine the Steps

You create steps within a process. You use the steps to link together the states. This is how the process is actually created.

You specify the following kinds of information for each step in your process:

- The name of the step. The name should be meaningful and should describe the path being taken. For example, Send for Review, Send for Approval, and so on.
- The **From State**, which is the state that the step is moving the asset from. If the step has no **From State**, it is the start step that begins the workflow process. A **From State** is required for each step other than the start step.
- The **To State**, which is the state that the step is moving the asset to. If the step has no **To State**, it is the end step that ends the workflow process.
- Which roles are authorized for the step.

If this is the very first step, the roles you specify determine which users can select this workflow for an asset. If you are using a Start Menu item to assign the workflow process to an asset, be sure that the roles assigned to this first (start) step match the roles assigned to the Start Menu item.

For subsequent steps after the first (start) step, the roles that you select must either match or be a subset of the roles that were selected to be notified by the previous step. Otherwise, the asset cannot leave the state because no one who is assigned the asset is allowed to move it to the next state.

- Who gets the asset next? (That is, when this step is completed). You specify the assignee for the To State by selecting one of the following assignment methods:
 - Retain “From State” assignees.

This option assigns the asset to its current assignees—that is, to the user who completes the step.

This option is useful when you are creating a step that returns to the same state (which creates an iterative state) or when it is the start step and you want the asset to be assigned to the person who selects the workflow process (whether through a start menu item or by selecting it on the “Status” form).
 - No assignments; control actions with function privs.

This option keeps the asset in the workflow process, which means that function privileges are enforced, but the asset is not actually assigned to anyone.

Note

When an asset is assigned to the current assignees (Retain “From State assignees option) or assigned to no one (No assignments option), the workflow system does not record workflow history for that step.

- Assign from a list of participants.
With this option, you select roles. When a user assigns the workflow process to an asset, a list of users with the roles that you selected is displayed. The user selects one or more users from the list, and those users are assigned the asset when it is in the state that this step moves the asset to.
- Choose assignees when the step is taken.
You also select roles with this option. This option means that the person who takes this step (by completing the assignment) during the workflow process selects the user who is assigned the asset next from a list of users with the roles selected for the step.
- Whether the estimated time for the state that this step moves the asset to can be changed or not. (Called an Assignment Deadline.)
- Any step actions that should occur.
- Any step conditions that should be assessed.
- If the step can be taken by more than one user (assignee), whether all the assignees must take this step before the asset can move to the next state. (The **All assignees must vote** field.)
- If the step can result in a deadlock, what the deadlock action should be.
- If the process will be used for a workflow group, whether all the assets in the group must complete this step before any of the assets can move to the next state. (The **Step is group synchronized** field.)

There are example steps in all of the sample site workflow processes. On a system where the sample sites are installed, select **Workflow > Processes** and examine a process to learn about how the sample workflow steps are configured.

Planning Your Steps

When determining what your steps should be, consider the following questions:

- Every process needs a start step. Before you create a start step, you must first determine how the asset is created and then placed into workflow.
Does the first participant create the asset, place it into workflow, and then keep working on the asset? If so, configure the step so that it is assigned to the person taking the start step (choose the **Retain “From State” assignees** option) and remember to create a start menu item that assigns the workflow to the asset by default.
Or does a supervisor create the asset and then assign to it a participant? If so, configure the start step so that the person taking the start step has to select the assignees.
- What is the order of the states?

- How many paths do you need between each state? This answer determines how many steps you need.
- Do you need to create an iterative state? If so, configure the step that takes the asset back to that state so that it is assigned to the person taking the step (choose the **Retain “From State” assignees** option).
- For steps that move an asset from a state in which more than one person is working on that asset, must all the participants complete their assignment before the step can be taken? If so, configure the step to be an all-voting step.
- Does the asset have more than one possible path from a state? (That is, there needs to be more than one step with the same From State.) If yes, do any of the steps from the state require that all participants vote the same way before the asset can progress? If yes, try to design the process so only one of the steps that lead from that state require that all the participants vote the same way. If you have two all-voting steps from the same state, deadlocks can occur.
- For each step, should CS-Direct 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 e-mail message) should the step use?
- Do groups of related assets ever need to be worked on at the same time? If so, can you configure your steps to work appropriately for both a single asset and a group of assets? If not, create separate workflow processes for the workflow groups.
- For a workflow group, are there any states that all the assets in the group must enter at the same time? (For example, all the assets should be approved at the same time.) If so, configure that step to be a synchronize step. Note that it's best to have only one synchronize step per workflow process.
- What should happen to the asset if you decide not to publish it, after all? Should you have a cancel step? How many cancel steps do you need?

On your sketches of your workflow processes, right next to the box that represents a step, list the step actions, the roles who can take the step, the roles who should be notified when the step is taken, whether the step is an all-voting step, and whether the step is a synchronize step.

Determine the Function Privileges

When you create a function privilege, users are either allowed or not allowed to perform the function in the user interface when the asset is in the state specified by the privilege. You create one set of function privileges for each workflow process (if you are using this feature, that is).

The following table lists the CS-Direct functions that you can control access to in the user interface during a workflow process:

Function	Description
Abstain from Voting	Removes a user from the workflow process. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.

Function	Description
Approve Asset for Publishing	Marks an asset as approved for publishing. Appears as an option in the drop-down list in the action bar on an asset's Edit , Inspect , and Status forms.
Authorize	Allows users who are granted roles that permit this function to modify permissions to the asset.
Build	Builds a collection asset. Appears as an option in the drop-down list in the action bar on an asset's Edit , Inspect , and Status forms.
Checkout	When revision tracking is enabled for the asset type, this function checks out an asset to the user. Appears as the Check Out button at the top of an asset's Edit and Status forms.
Copy	Copies an asset. Appears as an option in the drop-down list in the action bar on an asset's "Inspect" form.
Delegate Assignment	Delegates an asset that was assigned to a user through a workflow process to another user (one who has the appropriate role for the asset while it is in that state). Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.
Delete	Deletes an asset. Appears as an icon in the action bar on an asset's Edit , Inspect , and Status forms. It also appears as an icon next to the asset in lists.
Edit	Displays the asset in the "Edit" form. Appears as an icon in the action bar on an asset's Edit , Inspect , and Status forms. It also appears as an icon next to the asset in lists.
Finish Assignment	Invokes the next step in a workflow process, which assigns the asset to the next participant. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.
Inspect	Allows users who are granted roles that permit this function to view fields that are otherwise restricted. Users without the "Inspect" permissions cannot view any asset-type-specific data, and instead see a generic page that contains only standard, unrestricted, fields (such as ID, name, and description).
Place Page	Places a page on the Site Plan tab. Appears in as an option in the right-mouse menu on the Site Plan tab.

Function	Description
Remove from Group	Removes an asset from a workflow group. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.
Remove from Workflow	Removes the asset from a workflow process. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.
Rollback	Reverts an asset back to one of the previous versions of the asset that are stored by the revision tracking system. Appears as the Rollback button at the top of an asset's "Edit" and "Status" forms.
Set Assignment Deadline	Sets a deadline for how long the asset should remain in the next state. A deadline set with this option overrides any deadlines set for the state in the workflow process. Appears as an option in the "Select Workflow" form and in the "Finish My Assignment" form if the workflow process is configured to allow someone to override the estimated time for a state.
Set Export Target Path/ Filename	Restricts users from filling in the Path and Filename fields on an asset's "New" or "Edit" forms.
Set Participants	Sets the participants for a workflow process. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form. Additionally, this function prompts a user to select participants when the workflow process is configured so that the participant finishing an assignment must select the next participant (person being assigned the asset).
Set Process Deadline	Sets a deadline for how long it should take a specific asset to go through the entire workflow process. Appears as an option in the "Select Workflow" form for assets and when a workflow group is created or edited if the workflow process is configured to allow someone to set a process deadline.
Share Assets	Shares the asset with another CM site. Appears as an option in the drop-down list in the action bar on an asset's "Edit," "Inspect," and "Status" forms when the asset type is enabled for more than one CM site.
Show Participants	Displays a list of the participants in the workflow process that is currently assigned to the asset. Appears as an option in the Workflow Commands drop-down list on the asset's "Status" form.

Function	Description
Show Status	Displays the “Status” form for an asset. Appears as an option in the drop-down list in the action bar on an asset’s “Edit,” “Inspect,” and “Status” forms.
Show Version	Lists information about each the versions of the asset that the revision tracking system is storing. Appears 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.

Planning Function Privileges

For each workflow process, determine which functions you want to restrict access to, during which states you want to restrict those functions, and which roles should or should not have access to those functions when assets that are assigned to them are in those states.

List any restrictions that you want to enforce next to the appropriate states on your sketches of your workflow processes.

Remember that in order for your function privileges to enforce the restrictions that you intend, the roles of the users specified for the state in the function privilege must match the roles of the users associated with those states in the workflow process itself. If the roles don’t match, the result can be a condition in which no one can move the asset out of a state because the users who are allowed to work with the asset by the privilege are not allowed to by the state.

Also, remember that you cannot create just one function privilege: if you create even one function privilege that allows or restricts access to a function, you must create function privileges that cover all the other possible conditions for your workflow process.

Implementing Simplified Access Control

Because function privileges are associated with workflow processes, you can restrict individual users’ access to specific functions only when an asset is participating in a workflow process.

What can you do if you do not want to design and implement workflow processes but you still want to control access to specific functions? Create a simplified workflow process with one state and use the **No assignments; control actions with function privs** assignment method.

When a step moves an asset into a state using the **No assignments** assignment method, the asset does not appear on anyone’s assignment list, it just stays in the state which means that any function privileges assigned to that state are enforced.

To implement access control in this way, follow these general steps:

1. Create a state. It needs a name and a description. It does not need a deadline or any timed actions.
2. Create a new workflow process. Select all the roles you want to enforce restrictions for and all the asset types you want to use this workflow process for.
3. Create one step—the start step that puts assets of those types into the state. For the step, select the **No assignments** option. Enable the step for the roles that you want to be able to create assets of this type and assign this workflow to.

4. Configure the function privileges that you want to enforce for assets in the state that you created.
5. Create start menu items for the asset types that automatically assign this workflow process. Be sure that the roles who can use the start menu item match the roles who are assigned to the start step in the workflow process.

Now when users select **New > 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.

Determine Additional Workflow Process Details

If you have been sketching your workflow processes, filling in all the details about actions, conditions, e-mail messages, states, steps, and function privileges, by now you have planned nearly the entire design of your workflow processes.

Although the main task when creating a process is to create the steps that link the states (which is how your business process is represented), you must also specify the following kinds of information for each workflow process:

- Its name and description. The name should be descriptive so that users select the correct workflow process for the correct asset types.
- Which sites can use the process.
- Which asset types can use the process.
- Which roles can participate. This is a superset of all the roles that are designated in the steps.
- Which role will serve as the administrator of the workflow. A workflow administrator can delegate assignments on behalf of other participants.
- Whether a process deadline can be set for assets that participate in this workflow process.
- What the delegate action is.
- What the steps are. For each step, you specify the information described in the section named “Determine the Steps,” on page 146.
- Whether any of the CS 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 “Determine the Function Privileges,” on page 148.

There are example workflow processes delivered with all of the sample sites. They are described in detail in the next section.

Sample Site Workflow Processes

Each sample site has a workflow process. Before creating a workflow process of your own, it is a good idea to examine the sample site workflow processes. There are five:

- **HelloArticle Process**, the sample workflow process for the **HelloArticle** asset type that the HelloAssetWorld sample site provides.

- **BF: Normal Article Process**, the sample workflow process for the **article** asset type that the Burlington Financial sample site provides.
- **GE Lighting Process**, the sample workflow process for the **product** asset type that the GE Lighting sample site provides. (Available only when you have CS-Direct Advantage.)
- **BF: Fund Category Process**, the sample workflow process for the **product parent** asset type that the extension to the Burlington Financial sample site provides. (Available only when you have Engage.)
- **BF: Segment Process**, the sample workflow process for the **segment** asset type that the extension to the Burlington Financial sample site provides. (Available only when you have Engage.)

The following sections describe the first three workflow processes: HelloArticle Process, BF: Normal Article Process, and GE Lighting Process.

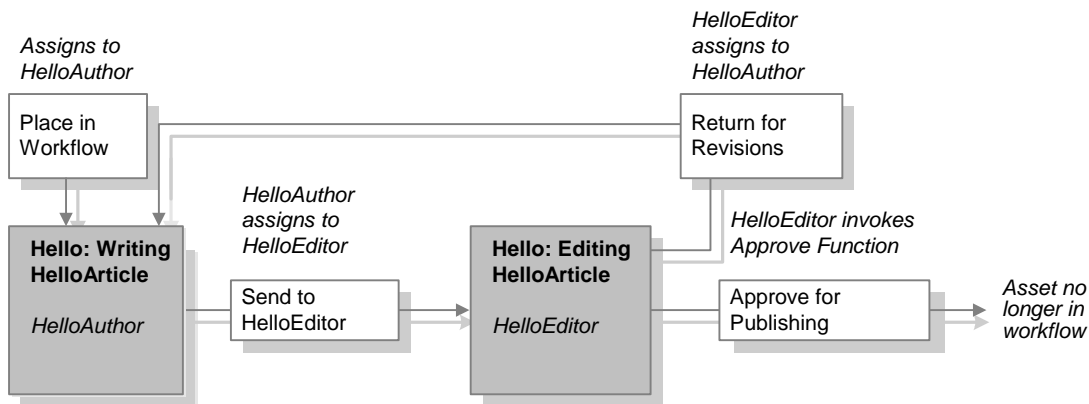
HelloArticle Process

At HelloAssetWorld, the authors, Joe and Moe, write HelloArticles. When their HelloArticles are ready to be reviewed, Joe and Moe send them to their editor, Flo.

Flo reads and edits the HelloArticle assets sent to her. If she decides that a HelloArticle asset needs more work, she sends it back to the author for revisions. If the HelloArticle asset is ready to be published, Flo approves it.

These work practices are reflected in the HelloArticle workflow process, which has two states and four steps, as follows:

When sketched as a flow chart, the HelloArticle workflow process looks like this:



1. The **Place in workflow step** puts the HelloArticle into the **Hello: Writing HelloArticle state**.

The start menu item named New HelloArticle is enabled for users who have the HelloAuthor role. This start menu item has the HelloArticle Process specified as the default workflow for assets of that type. When either Joe or Moe (the two users with the HelloAuthor role) selects **New > HelloArticle**, the HelloArticle Process is assigned to the asset, which invokes the first step in the process (Place in workflow).

The Place in workflow step is configured so that the HelloArticle asset is assigned to the person who creates the asset.

Roles: HelloAuthor

Assign to: “From State” assignees, which means that the person who creates the asset is assigned the asset.

Actions: None

2. When an author is done with his article, he finishes his assignment, which invokes the **Send to editor step**. This step moves the HelloArticle asset to the **Hello: Editing HelloArticle state**, and assigns it to Flo, the only user with the editor role.

Roles: This step is enabled for users with the HelloAuthor role only. This means that although editors and designers are allowed to create articles in order to troubleshoot and test the design, they cannot write articles that are meant to be published to the delivery system.

Assign to: HelloEditor

Actions: SendAssignmentEmail, which sends the default Assignment Message e-mail message to the editor participant.

3. When Flo, the editor, is done reviewing the HelloArticle asset that Joe or Moe assigned to her, she has two choices (steps that are enabled for users in the editor role) when she finishes her assignment:
 - a. The **Return for revisions step** sends the HelloArticle asset back to **Hello: Writing HelloArticle state** and assigns it back to the author who originally assigned it to her so that he can incorporate her comments.

Role: HelloEditor

Assign to: HelloAuthor

Action: SendRevisionNoticeEmail, which sends the Revisions Required Message e-mail to the original author participant when the editor takes the Return for revisions step.

- b. The **Approve for publishing step** invokes the Approve for publishing function, which approves the asset. The next time the publishing process runs, the HelloArticle asset is published (as long as all of its dependencies are met).

Role: HelloEditor

Assign to: No one.

Action: ApproveForPublish, which approves the HelloArticle for publishing.

The Approve for publishing step has no To state, which means that after this step is completed, the workflow process has ended for this asset.

This simple process has no step conditions or timed, deadlock, group deadlock, or delegate actions.

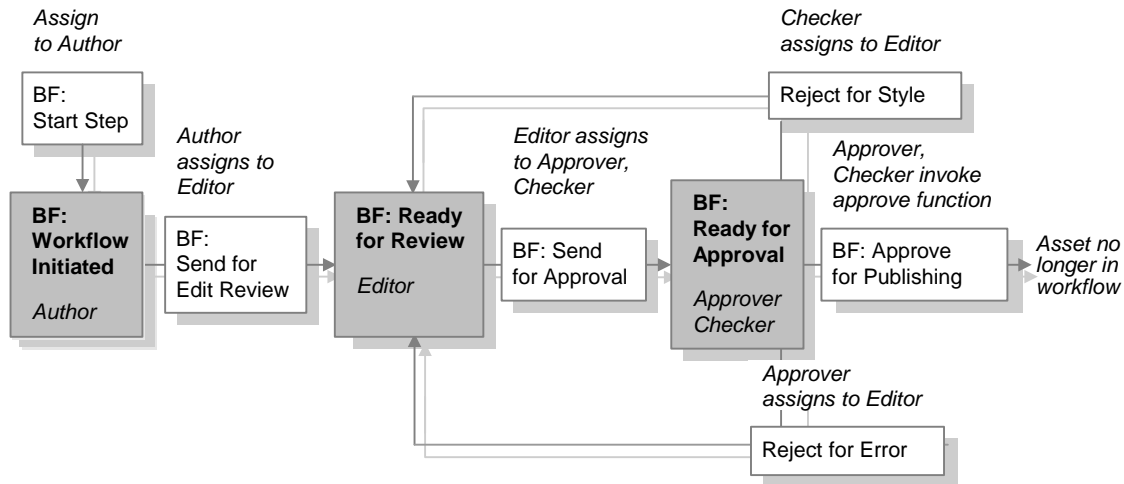
BF: Normal Article Process

At Burlington Financial, authors write articles and then send them to an editor. The editors not only review the articles, they actually modify and edit them. When they are finished with their input into the article, the editors send the article to a fact checker and an approver.

If either the fact-checker or the approver decide that there is some reason that the article cannot be published, they can reject the article, which sends it back to the editor. The editor then fixes the article and sends it back to the checker and approver.

Both the approver and the fact-checker must approve the asset before it is considered approved.

These work practices are reflected in the BF: Normal Article Process, which has six steps and three states, as follows:



1. The **BF:Start Step** step places the article asset into the **Workflow Initiated state**.

The Start Step step is enabled for users in the Author, Editor, Checker, and Approver roles. This means that any of the users in those roles can select the Normal Article Process from the **Workflow Process** drop-down list on the “Status” form for an article asset.

When a user selects the BF: Normal Article Process, CS-Direct invokes the BF:Start Step step that then starts the workflow process for the asset

The user who assigns the workflow to the asset must then select the participants. for each state from a list of users who have the appropriate roles for the process. The user from the author role is assigned the asset for the BF: Workflow Initiated state.

Roles: Author, Editor, Checker, Approver

Assign to: Author

Actions: Send Assignment Email, which sends the Assignment Message e-mail object to the author assigned the article by the BF:Start Step step.

Timed Actions: SendEmail, which sends the Assignment Due Message e-mail object one day before the deadline of the Workflow Initiated state.

2. When the author is finished writing the article asset, he or she finishes the assignment which invokes the **BF: Send for Edit Review step** that moves the article into the **BF: Ready for Review state**.

The article is now assigned to the editor who was selected when the workflow was first assigned to the article.

Roles: Author

Assign to: Editor

Actions: Send Assignment Email, which sends the Assignment Message e-mail object to the editor assigned the article by the BF: Send for Edit review step.

Timed Actions: SendEmail, which sends the Assignment Due Message e-mail object one day before the deadline of the BF: Ready for Review state.

3. The editor edits the article. When the editor is finished, he or she finishes the assignment, which invokes the **BF: Send for Approval step** that moves the article asset into the **BF: Ready for Approval state**.

The article is now assigned to the checker and the approver who were selected when the article was placed into workflow.

Roles: Editor

Assign to: Checker, Approver

Actions: Send Assignment Email, which sends the Assignment Message e-mail object to the checker and approver who are assigned the article by the BF: Send for Approval step.

Timed Actions: SendEmail, which sends the Assignment Due Message e-mail object one day before the deadline of the BF: Ready for Approval state.

4. The checker and the approver examine the article asset. When they are completed, they finish the assignment. However, because both of them have two possible steps, the “Finish My Assignment” form displays those steps as options. They must select the step (option) that represents their decision about the status of the article.

The approver can take either of the following steps:

- The **BF: Reject for Error step**, which moves the article back to the **BF: Ready for Review state** and assigns it the original editor.

Roles: Approver

Assign to: Editor

Actions: Send Rejection Email, which sends the Rejection Message e-mail object to the editor assigned the article by the BF: Reject for Error step.

Timed Actions: Now that the article is back in the Ready for Review state, a new review deadline is set and the SendEmail timed action sends the Assignment Due Message e-mail object one day before the new deadline for this state.

- The **BF: Approve for Publishing step**. This is an “all-voting” step that is enabled for both the checker role and the approver role, which means that the step will not be completed until both the checker and the approver choose this step for the article.

Roles: Checker, Approver

Assign to: No one.

Actions: ApproveForPublish, which marks the asset approved by the checker.

The checker can take either of the following steps:

- The **BF: Reject for Style step**, which moves the article back to the **BF: Ready for Review state** and assigns it the original editor.

Roles: Approver

Assign to: Editor

Actions: Send Rejection Email, which sends the Rejection Message e-mail object to the editor assigned the article by the Reject for Style step.

Timed Actions: Now that the article is back in the Ready for Review state, a new review deadline is set and the SendEmail timed action sends the Assignment Due Message e-mail object one day before the new deadline for this state.

- The Approve for Publishing step.

When both participants have taken the Approve for Publishing step, the article is approved for publishing. The Approve for Publishing step has no To State, which means that after the step is completed, the workflow is over for this asset.

Note that because neither the BF: Reject for Error step or the BF: Reject for Style step are all-voting, if either the approver or the checker rejects the article, it immediately returns to the editor.

This process has no step conditions, delegate actions, or deadlock actions.

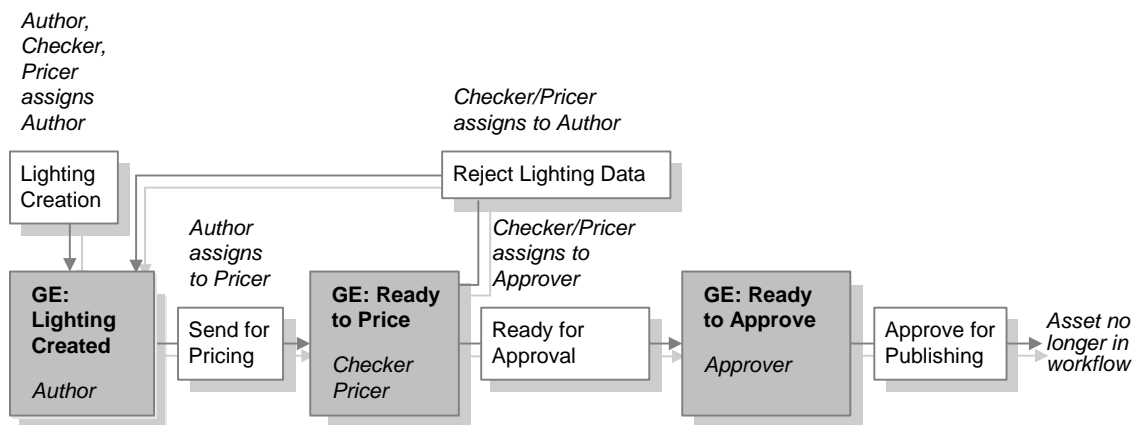
GE Lighting Process

At GE Lighting, authors create product assets but do not determine their prices. Users with the pricer role are responsible for pricing all products. Therefore, after authors have created a new product, they assign the product asset to a pricer so that it can obtain a price and to a checker to have all other data verified.

The pricer determines the price and the checker verifies the other data. Either one can send the asset back to the author for modifications or they can send it on to the approver.

The approver then approves the asset for publishing.

These work practices are reflected in the GE Lighting Process, which has five steps and three states, as follows:



1. The **Lighting Creation step** places the product asset into the **GE: Lighting Created** state and assigns it to an author.

The Send for Price step is enabled for users in the author, checker, or pricer roles. Any user in one of those roles can select the GE Lighting Process on a product asset's "Status" form. Selecting this process invokes the Lighting Creation step that then places the asset into the workflow process.

The user who assigns the workflow to the asset must also then select the participants for each state from a list of users who have the appropriate role for each state.

The author participant selected for the GE: Lighting Created state is assigned the asset.

Roles: Author, Checker, Pricer

Assign to: Author

Actions: SendAssignmentEmail

2. When the author is finished working on the product, he or she finishes the assignment. Finishing the assignment invokes the **Send for Review step** that moves the product asset into the **GE: Ready to Price state**.

The product asset is now assigned to the checker and the pricer who were selected when the product was first assigned the workflow.

Roles: Author

Assign to: Checker, Pricer

Actions: SendAssignmentEmail

3. The checker and the pricer examine the asset. The pricer assigns a price and the checker verifies the rest of the data. When they are finished with the asset, they finish the assignment.

Because there are two possible steps from the GE: Ready to Price state, the “Finish My Assignment” form displays two options:

- The **Reject Lighting Data step**, which moves the product asset back to the **GE: Lighting Created state** and assigns it to the original author.

Roles: Checker, Pricer

Assign to: Author

Actions: SendAssignmentEmail

- The **Send for Approval step**, which moves the product asset to the **GE: Ready to Approve state** and assigns it to the approver who was selected when the asset was first assigned this workflow process.

Roles: Checker, Pricer

Assign to: Approver

Actions: SendAssignmentEmail

Because the **Send for Approval** step is an all-voting step, both the checker and the pricer must take this step before the asset is moved to the GE: Ready to Approve state.

4. The approver examines and approves the product by finishing the assignment. Finishing the assignment invokes the Approve for Publishing step. The Approve for Publishing step has no To State, which means that after the step is completed, the workflow is over for this asset.

Roles: Marketer

Assign to: No one

Actions: ApproveForPublish, which marks the asset as approved for publishing to a specific destination.

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 e-mail objects, actions, and conditions or to schedule the timed action event, you need access to the **Admin** tab, which means that you need the GeneralAdmin role and the xceladmin ACL. To create workflow states and workflow processes, you need access to the **Workflow** tab, which means that you need the WorkflowAdmin role.

Overview

Before you create a workflow process, you must create the individual workflow components that you need for that process. Here are the general steps that you must take presented in the order you perform them:

1. Plan your workflow process by drawing it. See section “Planning Your Workflow Processes,” on page 137 for help with this step. Then refer to your sketch and notes throughout this section.
2. Create the roles that you need for your workflow processes. See Chapter 4, “Working with ACLs and Roles,” for help with this step. Be sure that any users who will participate in workflow have user profiles created for them. Otherwise, they will not receive e-mail messages from the workflow process.

Note

If you want the pool of users who are candidates to be participants in a workflow to include folks from all the sites that an asset is shared to, be sure to enable the cross-site assignments feature. See “Cross-Site Assignments and Participants,” on page 138 for details.

3. Create the e-mail objects that you need for your actions and enable the `xcelerate.emailnotification` property in the `futuretense_xcel.ini` file. See “Setting Up E-Mail Objects,” on page 160.
4. Create the step actions, timed actions, deadlock actions, group deadlock actions, and delegate actions that you need. See “Setting Up the Workflow Actions and Conditions,” on page 162.
5. Create your states. See “Setting Up the States,” on page 166.
6. Create your process. While creating your workflow process, you create the steps for that process. The steps link together the states so they occur in the proper order. Additionally, while creating your process, you configure any function privileges that you need. “Setting Up the Workflow Processes,” on page 168.
7. If your states have deadlines, be sure to configure the Timed Action Event so that the deadlines of assets are calculated regularly and the appropriate timed actions (if any) are invoked in a timely way. “Setting Up the Timed Action Event,” on page 164.
8. Test your workflow processes. See “Testing Your Workflow Process,” on page 177.

9. Set up your start menu shortcuts so that workflow processes are assigned automatically to assets when they are created. For help with start menu items, see “Creating Start Menu Items,” on page 108.

Setting Up E-Mail Objects

You create e-mail 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 e-mail messages you need, and then use the procedures in this section to create and edit them.

Your user account must give you access to the **Admin** tab in order for you to create e-mail objects.

Enabling the E-mail Feature

To ensure that your workflow process can successfully send e-mail messages, the following conditions must be true:

- In the `futuretense.ini` file, the properties on the **Email** tab must be configured to provide information about your e-mail server.
- In the `futuretense_xcel.ini` file, the `xcelerate.emailnotification` property must be set to `true`.

For information about the properties cited above and using the Property Editor, see the *Property Files Reference*.

- The workflow participants must have e-mail addresses specified in their user profiles. (For information about creating user profiles, see “Creating and Editing a User Profile,” on page 80.)

Creating E-Mail Objects

To create e-mail objects

1. In the Content Server interface, select the **Admin** tab and expand the **Email** item.
2. Under **Email**, double-click on **Add New**.

CS-Direct displays the “Add New Workflow Email” form:

Add New Workflow Email

The screenshot shows a web form titled "Add New Workflow Email". It contains four labeled input fields: "*Name:" with a single-line text box, "*Description:" with a single-line text box, "*Subject:" with a single-line text box, and "*Body:" with a multi-line text area. At the bottom of the form are two buttons: "Cancel" and "Add New Email".

3. In the “Add New Workflow Email” form, click in the **Name** field and enter a unique name of up to 36 characters. (This is the name that you use with the `emailname` variable when you use this e-mail object with an action).
4. In the **Description** field, enter a short, descriptive phrase of up to 36 characters.
5. In the **Subject** field, enter a phrase that describes the subject of the e-mail message. See “About E-Mail Objects,” on page 140 for a list of variables that you can use.
6. In the **Body** field, enter the text for the message. Once again, see “About E-Mail Objects,” on page 140 for a list of variables that you can use.
7. Click **Save**.

Editing E-Mail Objects

To edit e-mail objects

1. In the Content Server interface, select the **Admin** tab and expand the **Email** item.
2. Under **Email**, double-click on the e-mail object that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.
4. In the “Edit Workflow Email” form, make the necessary changes. See “About E-Mail Objects,” on page 140 for a list of variables that you can use.
5. Click **Save**.

Deleting E-Mail Objects

To delete e-mail objects

1. In the Content Server interface, select the **Admin** tab and expand the **Email** item.
2. Under **Email**, double-click on the e-mail object that you want to delete.

3. In the “Inspect” form, click the **Delete** icon.

A confirmation message appears.

4. Click **Delete**.

Setting Up the Workflow 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 e-mail message, you identify which e-mail object to send with the `emailname` variable.

Before you begin creating actions or step conditions, be sure that you have the elements and e-mail objects that you need. CS-Direct provides several default action elements and e-mail objects. Use the Content Server interface to examine the e-mail objects and use Content Server Explorer to examine the `ElementCatalog` table. Determine which elements you plan to use and then write down the entire name of those elements.

If you need additional e-mail messages, consult the previous section, “Setting Up E-Mail Objects,” on page 160 and create the e-mail messages that you need.

The following procedures describe how to create, edit, and delete the workflow actions—step, timed, delegate, deadlock, and group deadlock action—and step conditions.

Your user account must give you access to the **Admin** tab in order for you to create actions or conditions.

Creating Workflow Actions and Conditions

To create workflow actions and conditions

1. In the Content Server interface, select the **Admin** tab and expand the **Workflow Actions** item.
2. Under **Workflow Actions**, expand the kind of item that you want to create: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.
3. Under the item that you selected, double-click **Add New**.

CS-Direct displays the “Add New” form for that type of action or condition. For example, this is the “Add New Step Action” form:

Add New Workflow Step Action

*Name:	<input type="text"/>
*Description:	<input type="text"/>
*Element Name:	<input type="text"/>
Arguments:	<input type="text"/>

4. In the “Add New” form, click in the **Name** field and enter a unique name of up to 40 characters.
5. In the **Description** field, enter a short, descriptive phrase of up to 40 characters.
6. In the **Element Name** field, type the entire name of the element. For example, to use the default workflow element named **SendEmailToAssignees**, type: `OpenMarket/Xcelerate/Actions/Workflow/StepActions/SendEmailToAssignees`.
7. In the **Arguments** field, use the following convention to supply values for the arguments or variables that the element needs to function correctly:

`name=value`

For the `SendEmailToAssignees` element, for example, you must provide a value for the `emailname` variable (that is, the name of an e-mail object). For example:

`emailname=AssignmentDueReminder`

If an element takes more than one variable, separate the name/value pairs with the ampersand (&) character. For example:

`name1=value1&name2=value2`

8. Click **Add New 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 must create an additional Approve for Publish action for each publishing destination, or combination of publishing destinations.

The Approve for Publish step action uses the `OpenMarket/Xcelerate/Actions/Workflow/StepActions/ApproveForPublish` element which takes the `targets` variable. You can use this same element with as many additional approval step actions as you need.

For the default Approve for Publish step action, provide a value for the `targets` variable. Possible values for the `targets` variable are the names of any of the publishing destinations that have been created on this CS system.

For example:

`targets=serverX.`

To specify more than one publishing destination, separate each with a comma. For example:

`targets=serverX,serverY`

Note

If the name of a publishing destination that is referenced by the `targets` variable is changed, you must also change the value of the `targets` variable in your Approve for Publish steps.

Editing Workflow Actions and Conditions

To edit workflow actions and conditions

1. In the Content Server interface, select the **Admin** tab and expand the **Workflow Actions** item.
2. Under **Workflow Actions**, expand the kind of item that you want to edit: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.
3. Under the item that you selected, double-click the name of the item that you want to edit.
4. In the “Inspect” form, click the **Edit** icon.
5. In the **Edit Action** (or **Condition**) form, make your changes.
6. Click **Save**.

Deleting Workflow Actions and Conditions

To delete workflow actions and conditions

1. In the Content Server interface, select the **Admin** tab and expand the **Workflow Actions** item.
2. Under **Workflow Actions**, expand the kind of item that you want to delete: **Step Actions**, **Step Conditions**, **Timed Actions**, **Delegate Actions**, **Deadlock Actions**, or **Group Deadlock Actions**.
3. Under the item that you selected, double-click the name of the item that you want to delete.
4. In the “Inspect” form, click the **Delete** icon.
A confirmation message appears.
5. Click **Delete**.

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 for a CS system.

To set up timed action events

1. In the Content Server interface, select **Admin > Timed Action Event**.
2. In the “Workflow Timed Action Event” form, click **Edit**.

The “Edit Workflow Timed Action Event” form appears:

Edit Workflow Timed Action Event

Times: No existing workflow timed action event

Set Event Times:

Days of a week	Days of a month	Months	Hours	Minutes
Sunday	1	Jan	12AM	0
Monday	2	Feb	1AM	15
Tuesday	3	Mar	2AM	30
Wednesday	4	Apr	3AM	45
Thursday	5	May	4AM	
Friday	6	Jun	5AM	
Saturday	7	Jul	6AM	
	8	Aug	7AM	
	9	Sep	8AM	
	10	Oct	9AM	
	11	Nov	10AM	
	12	Dec	11AM	

Enabled: ☒ yes ☐ no

Cancel

Save

3. Set the schedule for how often the state deadlines should be calculated in terms of months, days, hours, and 15-minute increments.
4. Under **Enabled**, click **yes**.
5. Click **Save**.

A summary of the schedule is displayed.

CS-Direct 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,” on page 270.

Setting Up the States

When you create a workflow state, you specify a deadline, select a timed action, and configure when the timed action should run. Therefore, before you begin creating your workflow states, be sure that you have created the timed actions that you need.

To work with workflow states, your user name must be assigned the WorkflowAdmin role for the site that you are working with.

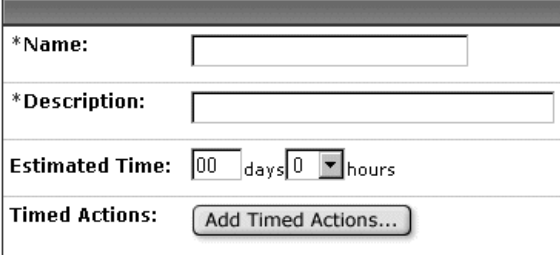
Creating Workflow States

To create workflow states

1. In the Content Server interface, select the **Workflow** tab and expand the **States** item.
2. Under **States**, double-click on **Add New**.

CS-Direct displays the “Add New Workflow State” form:

Add New Workflow State



3. In the “Add New” form, click in the **Name** field and enter a unique, meaningful name of up to 40 characters.
4. In the **Description** field, enter a short, descriptive phrase of up to 40 characters.
5. (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.
6. (Optional) Click the **Add Timed Actions** button and complete the following steps:

Add Timed Action to Workflow State: BF:Workflow Initiated

Name:	BF:FX									
Description:	BF:Workflow Initiated									
Estimated Time:	2days:0hours									
*Add Timed Action:	<table border="1"> <thead> <tr> <th>Action to Take</th> <th>Days</th> <th>Hours</th> <th>Offset</th> </tr> </thead> <tbody> <tr> <td>SendEmail</td> <td>00</td> <td>0</td> <td>Before Deadline After Deadline</td> </tr> </tbody> </table>	Action to Take	Days	Hours	Offset	SendEmail	00	0	Before Deadline After Deadline	
Action to Take	Days	Hours	Offset							
SendEmail	00	0	Before Deadline After Deadline							

Cancel

Add Timed Action

- a. In the **Action to Take** list, select a timed action.
 - b. In the **Offset** field, specify whether the action should be triggered before the deadline or after the deadline.
 - c. In the **Days** field and/or the **Hours** field, specify how many hours or days before or after the deadline (that you specified in step 5) that the action is to be triggered.
 - d. Click **Add Timed Action**.
 - e. Repeat this entire step for each timed action that you want to set up for this state.
7. Click **Add New State**.

Editing Workflow States

To edit workflow states

1. In the Content Server interface, select the **Workflow** tab and expand the **States** item.
2. Under **States**, double-click on the state you want to edit.
3. In the “Inspect” form, click the **Edit** icon.
4. In the “Edit” form, make the appropriate changes.
5. 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.
6. Click **Save**.

Deleting Workflow States

To delete workflow states

1. In the Content Server interface, select the **Workflow** tab and expand the **States** item.
2. Under **States**, double-click on the state you want to delete.
3. In the “Inspect” form, click the **Delete** icon.
A confirmation message appears.
4. Click **Delete**.

Setting Up the Workflow Processes

When you create a workflow process, you specify global process information and then you create steps, assigning step actions to them as needed. The steps in the process create the flow of the process by linking the states in a specific order.

Before you can begin creating a workflow process, you must have already created your step actions, step conditions (if necessary), and states.

To work with workflow processes, your user name must be assigned the Workflow Admin role for the site that you are working with.

Note

If your content providers will use workflow groups, be sure to enable the Workflow Groups tab for them. For information, see “Creating the Workflow Groups Tab,” on page 124.

Creating Workflow Processes

When you create a workflow process, you configure three categories of information: global process settings, steps, and function privileges.

Step A: Name and Set Global Settings for the Process

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on **Add New**.
The “Workflow Process: (new)” form appears:

Workflow Process: (new)

*Process Name:	<input type="text"/>
*Description:	<input type="text"/>
*Site:	<div>Any</div> <div>HelloAssetWorld</div> <div>BurlingtonFinancial</div> <div>GE Lighting</div>
*Asset Type:	<div>Any</div> <div>Article (Flex)</div> <div>Image (Flex)</div> <div>Recommendation</div> <div>Article</div>
*Roles:	<div>Analyst</div> <div>Approver</div> <div>Author</div> <div>Checker</div> <div>Designer</div>
*Start Step:	No process steps are currently defined. <input type="button" value="Add New Step"/>
Administration Roles:	<div>Any</div> <div>Analyst</div> <div>Approver</div> <div>Author</div> <div>Checker</div>
Process Deadline:	<input type="radio"/> allowed <input type="radio"/> not allowed
Delegate Actions:	No delegate actions are currently defined.
ID:	(new)

3. In the “New Workflow Process” form, click in the **Name** field and enter a unique name of up to 25 characters.
4. In the **Description** field, enter a short, descriptive phrase of up to 64 characters.
5. In the **Sites** list, select the sites that can use this workflow process.
6. In the **Asset Type** list, select the asset type(s) that this workflow process is for.
7. In the **Roles** list, select the roles that will participate.
8. In the **Administration Roles** list, select the roles that can act as the administrator for this workflow process when an asset is using it.
9. 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.
10. If you have any delegate actions configured, select the appropriate actions in the **Delegate Actions** list.
11. In the **Start Step** section, click **Add New Step**.

Step B: Create the Start Step

Add New Workflow Process Step

Process Name: Recommendation Process	
*Step Name: <input type="text"/>	
*States:	<div>From State</div> <div> none(Start of Workflow) BF:Fund Parent Created BF:Fund Parent Completed BF:Parent for Approval BF:Workflow Initiated </div> <div>To State</div> <div> none(End of Workflow) BF:Fund Parent Created BF:Fund Parent Completed BF:Parent for Approval BF:Workflow Initiated </div>
*Authorized Roles:	Analyst Approver Author Checker
*Assignment Method:	<input type="radio"/> Retain "From State" assignees <input type="radio"/> No assignments; control actions with function privs <input type="radio"/> Assign from list of participants <input type="radio"/> Choose assignees when step is taken
	(Select Roles) Analyst Author Checker Designer
Assignment Deadline:	<input type="radio"/> Can change <input checked="" type="radio"/> Use default
Step Actions:	ApproveForPublish SendAssignmentEmail SendRejectionEmail
Step Conditions:	ExampleStepCondition
Deadlock Actions:	SendDeadlockEmail
Voting:	<input type="checkbox"/> All assignees must vote
Workflow Groups:	<input type="checkbox"/> Step is group synchronized

Cancel

Save

1. In the “New Workflow Process Step” form, click in the **Step Name** field and enter a unique, meaningful name of up to 64 characters.
2. Configure the states as follows:
 - a. In the **From State** list, select **none (start of workflow)**.
 - b. In the **To State** list, select the name of the first state in the workflow process.
3. From the **Authorized Roles** list, select the roles that are allowed to take this step by assigning this workflow to an asset.
4. In the **Assignment Method** section, specify which roles will be assigned the asset by selecting one of the following options:

- **Retain “From Step” 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 of the last two options, select the appropriate roles from the list that is to the right of those options. (For definitions of these options, see “Determine the Steps,” on page 146.)

5. 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.
6. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
7. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
8. Click **Save**.

The start step is added and the “Steps for Workflow Process” form is displayed.

9. Click **Save**.

The process is saved and displayed in the “Inspect” form for the process. Note that this step appears as the **Start Step** in the form.

10. Continue to the next procedure.

Step C: Create the Subsequent Steps

Use the following procedure to create the rest of the subsequent steps, including the end step:

1. From the “Inspect” form of the workflow process, select the **Edit** icon.
2. Click **Add/Edit Steps** (a button at the bottom of the form).
3. In the “Steps for Process” form, click **Add Step**.
4. In the “Add New Workflow Process Step” form, click in the **Step Name** field and enter a unique, meaningful name of up to 64 characters.
5. Configure the states as follows:
 - a. In the **From State** list, select the name of the state that you selected as the **To State** for the previous step.
 - b. In the **To State** list, select the name of the next state in the workflow process. (Note that if you want to create an iterative step, select the same state in the **From State** and the **To State** lists.)
6. From the **Authorized Roles** list, select the roles that are allowed to take this step by finishing the assignment. The roles that you select from this list should either match or contain a subset of the roles that you selected for the **Assignment Method** in the workflow step that immediately precedes this workflow step.
7. In the **Assignment Method** section, specify which roles will be assigned the asset by selecting one of the following options:
 - **Retain “From Step” assignees**

If the **From State** and the **To State** are different, the first person who takes this step (finishes the assignment) is assigned the asset, even if the previous step assigned the asset to more than one participant.

If the **From State** and the **To State** are the same, every user who was assigned the asset by the last step is assigned the asset again with this step.

- **No assignments; control access with function privs**
- **Assign from list of participants**
- **Choose assignees when step is taken**

If you select either of the last two options, select the appropriate roles from the list that is to the right of those options. (For definitions of these options, see “Determine the Steps,” on page 146.)

8. In the **Assignment Deadline** section, determine whether the workflow administrator (or other user if you configure the appropriate function privileges) can override the Estimated Time (deadline) set for the state that this step moves the asset to.
9. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
10. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
11. If the step moves the asset from a state in which more than one participant was working on the asset and you want all participants to finish their assignments before the step can complete, select the **All Assignees must vote** option.
12. If there are any other steps in this process with the same **From State** and any of those steps are also all-voting steps (**All Assignees must vote** is selected), select a **Deadlock Action**. For information about deadlocks and avoiding them, see “Managing Deadlocks,” on page 130 and “About Delegate Actions,” on page 142.
13. If this workflow process will be used for workflow groups and you want all the assets to progress at the same time to the **To State** that you selected in step 4 of this procedure, select the **Step is group synchronized** option. FatWire recommends that you create only one synchronized step in the process.
14. Click **Save**.

The step is saved and the “Steps for Workflow Process” form is displayed.

15. Click **Save**.

The process is saved and displayed in the “Inspect” form for the process.

16. Repeat this procedure for each step—except the end step—that you want to create for this process.

To create the end step, continue with the next procedure.

To configure function privileges, continue to “Step E: (Optional) Configure Function Privileges,” on page 173.

Step D: (Optional) Create the End Step

An end step ends the workflow which means that the asset is no longer in a process and no longer has function privileges controlling access to it, if you are using function privileges. Use the following procedure to create an end step for the workflow process:

1. From the “Inspect” form of the workflow process, select the **Edit** icon.
2. In the “Edit Process” form, click **Add/Edit Steps** (a button at the bottom of the form).

3. In the “Steps for Process” form, click **Add Step**.
4. In the “New Workflow Process Step” form, click in the **Step Name** field and enter a unique, meaningful name of up to 64 characters.
5. Configure the states as follows:
 - a. In the **From State** list, select the name of the state that you selected as the **To State** for the previous step.
 - b. In the **To State** list, select none (**end of workflow**).
6. From the **Authorized Roles** list, select the roles who are allowed to take this step by finishing an assignment. The roles that you select from this list should either match or contain a subset of the roles that you selected for the **Assignment Method** of the workflow step that immediately precedes this workflow step in the process.
7. (Optional) In the **Step Actions** list, select one or more step actions that should be invoked when this step is taken.
8. (Optional) In the **Step Conditions** list, select a step condition, if appropriate.
9. If the step moves the asset from a state in which more than one participant was working on the asset and you want all participants to finish their assignments before the step can complete, select the **All Assignees must vote** option.
10. If there are any other steps in this process with the same **From State** and any of those steps are also all-voting steps (**All Assignees must vote** is selected), select a **Deadlock Action**. For information about deadlocks and avoiding them, see “Managing Deadlocks,” on page 130 and “About Delegate Actions,” on page 142.
11. If this workflow process will be used for workflow groups and you want this step to be performed on all the assets in the group at the same time, select the **Step is group synchronized** option. FatWire recommends that you create only one synchronized step in the process.
12. Click **Save**.

The start step is added and the “Steps for Workflow Process” form is displayed.
13. Click **Save**.

The process is saved and displayed in the “Inspect” form for the process.

If you do not need to configure function privileges, this workflow process is completed. If you do need to configure function privileges, continue to the next procedures.

Step E: (Optional) Configure Function Privileges

To set up function privileges

1. From the “Inspect” form of the workflow process, select the **Edit** icon.
2. In the “Edit” form, click **Add/Edit Function Privs** (a button at the bottom of the form).
3. In the “Workflow Process Functions” form, scroll down to the function that you want to set a privilege for.
4. Click the **New** button next to the function.

The “Add Privileged Function” form appears:

Add Privileged Functions

Process Name:	HelloArticle Process
Function:	Abstain From Voting
*State:	<div> Workflow Not Active Hello: Editing HelloArticle Hello: Writing HelloArticle </div>
*Role:	<div> HelloAuthor HelloEditor </div>
Allowed?	<input checked="" type="checkbox"/>

5. Select the appropriate state from the **State** list.
6. Select the roles that are allowed or not allowed to perform this function when an asset is in this state from the **Roles** list.
7. Do one of the following:
 - To allow users with those roles to perform the function, select the **Allowed** option.
 - To restrict users with those roles from performing the function, clear the **Allowed** option.
8. Click **Add New**.
The privilege is set and the “Workflow Process Functions” form appears again.
9. Repeat steps 3 through 8 for each function privilege that you need to configure. For more information about function privileges, see “Determine the Function Privileges,” on page 148.
10. After you have configured all of your function privileges, click **Save**.
The privileges are saved and the “Inspect Process” form appears again.
The workflow process is complete.

Editing Workflow Processes

To edit workflow processes

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.

4. To modify the name, description, or any of the other global settings, make your changes directly in the “Edit” process form.
5. To edit the steps, see “Editing Steps,” on page 175
6. To edit the function privileges, see “Editing Function Privileges,” on page 175.
7. When you are finished, click **Save**.

Editing Steps

To edit workflow steps

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.
4. In the “Edit Process” form, click **Add/Edit Steps** (a button at the bottom of the form).
5. In the “Steps for Workflow Process” form, click **Edit** next to the step that you want to change.
6. In the “Edit Workflow Process Step” form, make your changes and then click **Save**.
7. The workflow process is saved and the “Inspect” form appears.

Editing Function Privileges

To edit a function privilege, you must delete it and then re-create it.

To edit a function privilege

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.
4. In the “Edit Process” form, click **Add/Edit FunctionPrivs** (a button at the bottom of the form).
5. In the “Functions for Workflow Process” form, click **Remove** next to the function that you want to change.
6. Then click **New** next to the same function.
7. In the “Add Privileged Functions” form, create the function privilege and then click **Save**.
8. In the “Functions for Workflow Process” form, click **Save**.

The workflow process is saved and the “Inspect” form appears.

Copying Workflow Processes

You can copy a workflow process, which can save you some steps in configuring additional processes.

To copy workflow processes

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to copy.
3. In the “Inspect” form, click in the drop-down list on the action bar and select **Copy Process**.

The “Copy Workflow Process” form appears:

Copy Workflow Process: BF: Normal Article Process



4. In the **Process Name** field, enter the name of the process.
5. In the **Description** field, enter a short, descriptive phrase.
6. Click **Save**.
7. Edit the process as necessary. See the following procedures for help:
 - “Editing Workflow Processes,” on page 174.
 - “Editing Steps,” on page 175.
 - “Editing Function Privileges,” on page 175.

Deleting Workflow Processes

To delete workflow processes

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to delete.
3. In the “Inspect” form, click the **Delete** icon.

A confirmation message appears.
4. Click **Delete**.

Deleting Steps

To delete workflow steps

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.

4. In the “Edit Process” form, click **Add/Edit Steps** (a button at the bottom of the form).
5. In the “Steps for Workflow Process” form, click **Remove** next to the step that you want to delete.
6. Click **Save**.

The workflow process is saved and the “Inspect” form appears.

Deleting Function Privileges

To delete function privileges

1. In the Content Server interface, select the **Workflow** tab and expand the **Processes** item.
2. Under **Processes**, double-click on the workflow process that you want to edit.
3. In the “Inspect” form, click the **Edit** icon.
4. In the “Edit Process” form, click **Add/Edit FunctionPrivs** (a button at the bottom of the form).
5. In the “Functions for Workflow Process” form, click **Remove** next to the function that you want to change.
6. Click **Save**.

The workflow process is saved and the “Inspect” form appears.

Testing Your Workflow Process

Before you move your workflow process to the management system and implement it there, test it on your development system.

To test your workflow process

- Log in as a user with administrator access and configure start menu items that assign workflow processes to assets, if necessary.
- Log in as a user who has a role that has been authorized to take the start step of the workflow process. Create an asset, select the workflow process (only if the start menu item does not assign it), and then finish the assignment.
- Log in as the participant who has the assignment now. Finish the assignment and then log in as the next participant. Continue through the entire workflow in this manner.
- Verify that the e-mail messages that should be sent are being sent.
- Verify that your workflow sends the asset through the process correctly.

Moving Your Work

Typically you create and fine-tune a workflow process on a development system to ensure that it functions exactly as you need it to before you introduce that workflow process to the management system.

When your workflow is ready to be used by content providers, use the Initialize Mirror Target feature to move your workflow components to the management system.

For information about this feature, see “Migrating a Site from One System to Another System,” on page 264.

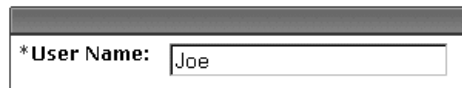
Clearing Workflow Assignments

People go on vacation, get reassigned to new work groups, and move on to different jobs. What should happen to their workflow assignments in these situations? One option for handling work assignments that cannot be finished by the original assignee is to delegate the assignments to someone else. However, even with the delegate feature, you, the administrator, might still be called on to clear assignments from some user’s assignment list.

To clear assignments for a user

1. In the Content Server interface, select the **Admin** tab and double-click on **Clear Assignments**.
2. In the “Search for Assignments” form, enter a user name. For example:

Search for Assignments



Show Assignments

3. Click **Show Assignments**.
CS-Direct displays a list of all the assets that are currently assigned to that user.
For example:

Clear Assignments

The following assets have been assigned to: **Joe**.

To clear an assignment, select the asset's checkbox and click Clear Assignments.

Type	Name	Description	Assigned to	Task Status	Clear
HelloArticle	movies	story about movies	Joe	active	<input type="checkbox"/>
HelloArticle	dogs	story about the DogsFromMars	Joe	active	<input type="checkbox"/>
HelloArticle	food	story about where to eat on planet Earth	Joe	active	<input type="checkbox"/>

Clear assignment comment:

Remove the asset from workflow if there are no other assignees for the asset:

☐ Yes

☒ No

Clear Assignments

- In the “Clear Assignments” form, select the **Clear** option next to each assignment that you want to clear.
- (Optional) Click in the **Clear assignment comment** field and enter a comment about why you are clearing these assignments. The text that you enter in this field appears in the **Action Taken** field in the **Workflow History** section on the “Status” form for the asset.
- Do one of the following:
 - If you want the asset to be removed from workflow if it is not assigned to anyone else, select **Yes**.
 - If you want the asset to remain on the assignment list of any other user who is also assigned the asset, select **No**.
- Click **Clear Assignments**.

CS-Direct displays a **Clear Assignments Report** that summarizes the changes.

Chapter 10

Replicating CM Sites

To speed up the deployment of online sites, Content Server provides Site Launcher. This new feature enables you to replicate a suitably-prepared CM site and, in the process, either share or copy its components to the new site. You can then modify the new site as necessary, and deploy it. This chapter provides an overview of Site Launcher, replication guidelines, requirements and options, and procedures for enabling and using Site Launcher.

This chapter contains the following sections:

- Site Launcher Overview
- Preparing for Replication
- Site Replication Steps
- Post-Replication Tasks and Guidelines

Site Launcher Overview

To minimize your effort in creating new sites, Content Server 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 department.

Site Launcher replicates source sites directly on their native Content Server system, re-using the existing database schema. The sites are replicated quickly and easily, without the need for coding. However, while replication itself is a quick and straightforward procedure, it does require preparation and followup on the part of the administrator. How much, depends on the content management needs.

The rest of this chapter provides the steps that you need to follow in order to successfully replicate sites for use in any content management model—1:1, 1:n, and x:n, described in “Content Management Models,” on page 28.

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 need to consider several recommendations regarding the nature of the site and its replicate. You must also make decisions about which components to copy or share, and finally ensure that the source site meets system requirements for replication. This section outlines our recommendations, your options, and the system requirements.

Ensuring the Source Site Meets Replication Requirements

Site Launcher can be used to replicate almost any CM site: small, large, functional, incomplete, independent of other sites, and overlapping other sites by the sharing of components.

However, to use Site Launcher most effectively, it is best for you to start with a site that is small, functional, and similar to the site that you need to have. In general, the source site should be a skeletal one, meaning that it has structure and design, but little content. These characteristics, beyond ensuring a tractable replicate, will help you conserve resources and minimize replication time.

In addition, make sure that the source site resides on the management system. (If you need to first create a replicable source site, follow the steps in “Site Configuration Steps,” on page 53.)

Planning the New Site

This section outlines your site replication options and the system requirements.

Copying vs. Sharing

When designating a source site, you must decide whether to copy or share certain site components, and therefore, determine from the start, how the new site will function in relation to its source site—as a duplicate or a subset; as an independent site or an overlapping site. (On a bigger scale, you will also need to determine how the new site will function in relation to other sites in your content management model.)

The table below shows that when the source site is being replicated, most if its components are shared by default. Other components are shared or copied at your discretion. Users are neither copied nor shared.

Source Site Component	Replication method
asset subtypes	Shared
asset types	Shared
assets	Either copied or shared
associations	Shared
CS-Desktop and CS-DocLink configuration	Copied
publishing destinations	Shared
roles	Shared
start menu items	Shared
templates (design assets)	Either copied or shared
tree tabs	Shared
users	Neither shared nor copied
workflow processes	Shared

Note that copying and sharing of assets is selective at the asset type level, but not at the asset level—when you select an asset type to be copied (or shared), *all* assets of that type are copied (or shared). The same holds for templates (design assets).

Concerning assets and templates, your specific tasks are to determine:

- Which asset types must have their assets copied or shared.
- Whether template assets must be copied or shared.
- Whether assets will be previewed on the new site as formatted content.

To determine whether assets and templates must be copied or shared, use the following guidelines:

- Sharing an asset does not require you to share the template that renders the asset. The template can be copied.

Whether you share or copy an asset depends on how the rendering logic in the template treats asset names. For example, template logic can be written in such a way as to assume either shared templates (constructed with explicit template names in the `render:satellitepage` and `render:callemlement` tags) or copied templates (constructed with a template name using `Variables.site` prepended to the name in the `render:satellitepage` and `callemlement` 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 assume that the asset type is either shared (explicit name) or copied (constructed name), and you need to designate the source site accordingly.

If you decide to copy template assets, make sure that the template assets do not contain hard-coded template names, and a template is not named in the “Default Values” list of any start menu item.

- When assets are shared to a new site by Site Launcher, their data is not changed. The single exception is Template assets. A Template asset maintains a SiteCatalog entry for each site to which it is shared. Therefore, when a template is shared to a new site, the template’s list of SiteCatalog entries is updated (the new site is added to the list of SiteCatalog entries).
- Because data in a shared asset is not changed, asset references of that asset also remain unchanged. Therefore, if assets with asset references are to be shared, the asset references must also be shared. For example, if you have Article assets that have references to ImageFile assets, you may
 - share both asset types,
 - copy both asset types, and
 - copy the Articles and share the ImageFiles.

However, you must not share the Article and copy the ImageFiles. Doing so will cause errors to be displayed when a copy of the site is launched.

- Start menus are shared among the source site and its replicates. Therefore, if a workflow process is set in the source site's start menu, that workflow will be set in the replicate sites. When users in the new site invoke the start menu item to create an asset, the workflow will take effect.

Your decisions in the planning stage are governed by the specifics of your installation as well as your own methods of managing the Content Server environment. The site that you create by replication must ultimately fit in to the content management model that is being (or has been) established on the management system. For example, if you have central sites and centrally managed sites, you need to determine how your new site will function in relation to them. For information about site modeling, see “Content Management Models,” on page 28.

Naming Assets

Assets that are copied to the new site are named algorithmically, by use of prefixes as follows:

- The prefix for the source is specified when you designate a site as a site launcher source site and is, by default, the name of the site.
- The prefix for the new site is specified in Site Launcher itself and is, by default, the name of the new site. When specifying a prefix, make sure it is short.
- If the source asset has a prefix, or the asset type requires a unique name, then a prefix will be used in the name of the new asset. If the source asset has a prefix, the new asset will have the new prefix in place of the old prefix.
- If the source asset does not have a prefix, then the prefix is prepended to the asset name to make the new asset name.

Planning Users

Given that users are not copied or shared to the new site during replication, you will have to manually add them to the new site.

If the users will be different from those on the source site, you will need to create new user accounts. In the process, consider the users' functions on the new site, the roles they will need, their access to start menu items, workflows, tree tabs, and so on.

Bear in mind that the original associations among components are preserved on the new site. If you redefine users' functions, you may have to redefine their associations with other components on the site. Accounting for new associations in the planning stages will help you to assess your post-replication workload and create a functional site that also meets your expectations.

Choosing the Replication Time

Because normal operations affect site replication, it is best to run Site Launcher when the source site is not being actively edited. Otherwise, the replicate site might not match the source site. Also, make sure that prior to replication, all assets in the source site are checked in to the database. Otherwise, if revision tracking is enabled and an asset to be shared is checked out, the sharing of the asset will fail.

Site Replication Steps

This section shows you how to enable and use Site Launcher. The procedure consists of three basic steps:

- I. Ensure that Replication Requirements Are Satisfied
- II. Enable the Source Site for Site Launcher, on page 186
- III. Replicate the Source Site, on page 187

The steps are given in detail in the rest of this chapter.

I. Ensure that Replication Requirements Are Satisfied

1. Prepare for replication by ensuring that source sites and planned sites meet the requirements. For more information, see “Preparing for Replication,” on page 182.
2. Go to section “II. Enable the Source Site for Site Launcher” to enable the source site.

II. Enable the Source Site for Site Launcher

In order to replicate a source site, you must first configure Site Launcher.

To configure Site Launcher

1. Make sure that you have completed the steps above (“I. Ensure that Replication Requirements Are Satisfied”).
2. When you complete the steps in this section, be prepared to follow up immediately with the steps in the next section, “III. Replicate the Source Site.” Immediate followup ensures that the condition of the source site has not changed.
3. In the **Admin** tab, expand the **Sites** node and double-click the site you want to replicate. The site you select is the source site.
4. In the site’s “Inspect” screen, click the link **Configure CS-Site Launcher for this site** (at the bottom of the screen).
5. In the screen “Configure CS-Site Launcher,” do the following:
 - a. If you want to copy assets, go to the **Asset prefix** field and either enter a prefix for the assets, or accept the default (the name of the source site).
 - b. In the “Enabled Asset Types” panel, locate the asset types whose assets you want to copy or share, and click their **Copy** or **Share** radio buttons.

Note

If you want to copy or share all assets of all types, click the **copy all** or **share all** button.

- c. Click the **Save** button to save your configuration options.

Configure CS-Site Launcher: CO

*** Asset prefix:**

All assets will be shared or copied to new site. Copied assets may be prefixed with the new site name if a unique name is required. All start menu items will be shared with new site. All CS-Desktop and CS-DocLink configuration information will be copied to the new site.

Enabled Asset Types:

Description	Asset Type	Copy	Share
		<input type="button" value="copy all"/>	<input type="button" value="share all"/>
Attribute Editor	AttrTypes	<input type="radio"/>	<input checked="" type="radio"/>
CSElement	CSElement	<input type="radio"/>	<input checked="" type="radio"/>
Co. Attribute	CO_Attribute	<input type="radio"/>	<input checked="" type="radio"/>
Co. Content Asset	CO_Content	<input checked="" type="radio"/>	<input type="radio"/>
Co. Definition	CO_Def	<input type="radio"/>	<input checked="" type="radio"/>
Co. Filter	CO_Filter	<input type="radio"/>	<input checked="" type="radio"/>
Co. Parent Asset	CO_Parent	<input type="radio"/>	<input checked="" type="radio"/>
Co. Parent Definition	CO_ParentDef	<input type="radio"/>	<input checked="" type="radio"/>
Link	Link	<input checked="" type="radio"/>	<input type="radio"/>
Page	Page	<input checked="" type="radio"/>	
Query	Query	<input type="radio"/>	<input checked="" type="radio"/>
Recommendation	AdvCols	<input type="radio"/>	<input checked="" type="radio"/>
SiteEntry	SiteEntry	<input type="radio"/>	<input checked="" type="radio"/>
StyleSheet	StyleSheet	<input type="radio"/>	<input checked="" type="radio"/>
Template	Template	<input type="radio"/>	<input checked="" type="radio"/>

Workflow Process: All workflow processes will be shared with new site.

Workflow

6. Go to section “III. Replicate the Source Site” to complete the site replication procedure.

III. Replicate the Source Site

Once you have enabled the source site for replication, you can use Site Launcher to create as many copies of the source site as you wish.

To replicate the source site

1. Make sure that you have completed the steps above (“I. Ensure that Replication Requirements Are Satisfied,” and “II. Enable the Source Site for Site Launcher”).
2. At the top of the “Inspect” form, double-click the **Launch copy** icon.
3. In the Site Launcher window, do the following:
 - a. In the **Name** field, enter the name for the new site.
 - b. In the **Description** field, enter the description for the new site.
 - c. In the “Publish Destinations” panel:
 - 1) Select the publish destinations you would like to be available for the new site.
 - 2) Initialize the destinations by selecting the checkboxes, as necessary.

- d. Click the **Add Site** button to copy the site, and add it to the “Sites” node in the “Admin” tab.

CS-Site Launcher: New Site from CO (Company Launchpad)

***Name:**

Name should be short. It is used as a prefix for names of assets.

***Description:**

Enabled Asset Types:

Description	Asset Type	Copy	Share
Attribute Editor	AttrTypes		✓
CSElement	CSElement		✓
Co. Attribute	CO_Attribute		✓
Co. Content Asset	CO_Content	✓	
Co. Definition	CO_Def		✓
Co. Filter	CO_Filter		✓
Co. Parent Asset	CO_Parent		✓
Co. Parent Definition	CO_ParentDef		✓
Link	Link	✓	
Page	Page	✓	
Query	Query		✓
Recommendation	AdvCols		✓
SiteEntry	SiteEntry		✓
StyleSheet	StyleSheet		✓
Template	Template		✓

Workflow Process:
CO: One-Step for Content

Publish Destinations:

Enable?	Initialize
<input type="checkbox"/> Destination 1 (static)	<input type="checkbox"/> Set publish start point using 'Home'.
<input type="checkbox"/> Destination 2 (dynamic)	<input type="checkbox"/> As a production site.
	<input type="checkbox"/> As a management site.

After the new site is added, Content Server displays a screen that indicates:

- Which assets were copied or shared. For each asset that is shared, Content Server adds a row in the `AssetPublication` table, indicating the site id (publication id) of the new site. For each asset that is copied, Content Server enters the copy into the table where the original asset is stored.
 - The number of start menu items, workflow processes, and publish destinations that were shared with the new site.
4. At this point, you need to add users and otherwise ensure that the new site is properly configured. For instructions and guidelines, see the next section, “Post-Replication Tasks and Guidelines.”

Post-Replication Tasks and Guidelines

When the new site is established, you need to complete its configuration by completing the following steps:

1. Add existing users to the new site, or establish new users, depending on how you planned the new site.
 - To add existing users, follow instructions in “Granting Users Access to a Site (Assigning Roles to Users),” on page 97.

- To establish new users, follow the guidelines in Chapter 3, “Site Configuration Guidelines.”
2. If workflow processes exist, make sure that the correct asset types and roles are associated with the processes through Start Menu items.
 3. Test the new site.
 4. If necessary, create a site administrator. Be sure to assign the xceladmin ACL to the user account and give the user the SiteAdmin role for the new site.

Part 3

System Configuration Procedures

This section describes the configuration options that affect all the sites on your CS system.

It contains the following chapters:

- Chapter 11, “Configuring the User Interfaces”
- Chapter 12, “Managing the CS Publishing System”
- Chapter 13, “Revision Tracking”
- Chapter 14, “Search Engines”

Chapter 11

Configuring the User Interfaces

There are several CS user interface features that must be configured before content providers can use them.

Some of these features are alternate interfaces to the Content Server interface: Content Server Desktop, Content Server DocLink, and the InSite Editor. Others change the Content Server interface: locale settings and Ektron's eWebEditPro; and one changes the behavior of the preview function.

This chapter contains the following sections:

- Setting the Locale for the Content Server Interface
- CS-Desktop
- CS-DocLink
- InSite Editor
- eWebEditPro
- Maintaining Separate Browser Sessions for Preview

Setting the Locale for the Content Server Interface

If your organization purchased and installed one of the CS language packs, you must configure a default locale for your CS system to use.

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

- The **Locale** item on the **Admin** tab sets the default locale for the Content Server interface. Individual users can override the default and use a different locale (if one is present) by setting a preference.
- The `cs.emailcharset` and `cs.emailcontenttype` properties in the `futuretense.ini` file determine the character set that is used in the e-mail messages that the workflow system sends. These properties—which must be set to a character set that can appropriately represent characters for all of the locales that your content providers can select—are usually set during installation.
- If your system is using the UTF-8 character set and you are using the Mirror to Server publishing delivery type, you must set the value for the `cs.mirrorrowsperpost` property in the `futuretense.ini` file to 4 or lower.
- If your system is using the UTF-8 character set and you are using the Burlington Financial sample site's article asset type, you must set the value of `xcelerate.body.length` property in `futuretense_xcel.ini` to 500.

System Default Locale for the CS Users

When a language pack is installed, the installer writes the language and country code for the locale of the language pack to the `LocaleMap` database table. For example, Canadian French is `fr_ca`. When there is more than one locale listed in the `LocaleMap` table, you, as the administrator, need to specify which should be the default locale.

You can override the default locale for individual users by setting it in their user profile. See “Creating and Editing a User Profile,” on page 80 for information about how to set a user's locale preference.

CS-Direct determines which language to use for an individual user through its universal session variable named `locale`. This variable contains the language and country code associated with the user who is logged in.

When a user logs in to CS, CS-Direct obtains the value for the `locale` variable as follows:

- If the user's user profile has a value for the locale user attribute, CS-Direct obtains and uses it.
- If the user profile does not specify a preference, CS-Direct 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 CS-Direct cannot match that user's browser locale, CS-Direct uses the default locale that is set for the system.

To set the system default locale

1. On the **Admin** tab, select **Locale**.

Locale Manager

Language	Locale	Default
  English (United States)	en_US	
  French (France)	fr_FR	

Add New

2. In the “Locale Manager” form, click the **Edit** button next to the locale that you want to select.
3. In the “Edit Locale” form, select the **Default** option.
4. Click **Save**.

System Locale Properties

The properties in the following list should have been configured during the installation of the language pack on your CS system. They are listed here for troubleshooting purposes. For example, if the workflow e-mails are displaying odd characters, perhaps the e-mail properties were not set correctly.

For instructions on how to start and use the Property Editor to verify or modify property values, see the *Property Files Reference*.

Property	Property File	Description
cs.contenttype	futuretense.ini	Determines the HTTP header character set for all Content Server pages, as well as other HTTP interaction (Content Server Explorer, CatalogMover, and so on). This value is set to <code>text/html ; charset=UTF-8</code> by default. This value must be set to an appropriate value for the online site that your CS system is delivering.
cs.emailcharset	futuretense.ini	Determines the character set used in the subject lines of e-mail messages sent by Content Server. (Typically these are workflow e-mails.)
cs.emailcontenttype	futuretense.ini	Determines the character set used in the body of e-mail messages sent by Content Server.
cs.mirrorrowsperpost	futuretense.ini	Specifies the number of table rows that can be mirrored during each HTTP POST while a mirror publish session is underway. If you are using UTF-8 and your database serves non-ASCII text, this value must be set to 4 or lower.

Property	Property File	Description
<code>xcelerate.charset</code>	<code>futuretense_xcel.ini</code>	Determines the HTTP header character set used for all CS-Direct 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 Content Server database as well as individual items such as categories and source codes can have one name only. This means that although much of the text on an individual CS-Direct form can be displayed in any of the languages that you have installed on your CS system, items such as field names and asset type names can be displayed in one language only.

Following is a list of items in CS that can have one name only, which means that they can be displayed in one language only:

- Asset type names
- Field names
- Asset names
- Categories
- Source codes
- Tree tab names
- Site names
- Names of workflow building blocks (actions, e-mail 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 CS system. However, there are several core columns/fields that CS-Direct creates for all asset types by default:

- `id`
- `name`
- `description`
- `status`
- `createdby`

- createddate
- updatedby
- updateddate
- startdate
- enddate
- subtype
- filename
- path
- template
- category

These column names are created in English, by default. When CS-Direct displays them as fields in the Content Server interface, it uses the language that is set as the system default language.

For descriptions of the default columns for asset types, see the “Data Design” section in the *Content Server Developer’s Guide*.

Locale and the Article Asset Type

If your site developers decided to use the sample site asset type named article, you must also modify the value for the `xcelerate.body.length` property in the `futuretense_xcel.ini` file. This property determines how many characters are stored in the `urlbody` column of the `Article` table.

Because `urlbody` is a URL column, the data entered in it is actually stored as a file outside of the Content Server 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 Content Server interface.

When this column holds non-ASCII text, the value for this property should be set to 500.

CS-Desktop

Content Server Desktop (CS-Desktop) enables Microsoft Word users to create, import, and edit Word documents as Content Server assets using the familiar Word user interface. They save these assets to, and open them from, the Content Server database.

Note

CS-Desktop supports the use of standard Word documents and templates. In Microsoft Word 2000, web page templates that enable Word users to create HTML pages using Word were introduced.

The Word web page templates are neither supported nor needed by CS-Desktop.

Overview

When CS-Desktop is configured for a system, users can log in to Content Server from a special toolbar in the Microsoft Word user interface.

You, the administrator, configure which asset types will be available in the Word interface. You enable the asset types for CS-Desktop and then determine which fields the Word users can enter data for. This information determines the appearance and the functionality of the Content Server toolbar when users create assets of those types in Word. Your content providers use the Content Server toolbar to insert field markers that mark the data for each field.

When someone saves the Word document as an asset, CS-Desktop sends that asset to Content Server. A temporary copy of the document is also saved locally on the user's hard drive.

The file conversion subsystem converts the binary form of the document into XML, which, in turn, is parsed into a form suitable for storing in the Content Server database. The asset is then managed as follows:

- The Word document is also stored with the asset. That document is retrieved when a user opens the asset in Word.
- When a visitor to your online site (on the delivery system) requests the asset, Content Server serves the asset, not the Word document.

Note that when you examine a CS-Desktop asset that in the Content Server interface, you can access its "Inspect" form, but you cannot edit the asset. When the source of the asset is Word, the asset can be edited only in Word.

Sources of Data for a Word Asset

The Microsoft Word document is the complete source for a word asset. There are several ways to enter data for an asset's field in the Word interface:

- A user enters text and then marks it with the appropriate field marker.
- Certain required fields such as **Name** and **Description** are hidden in Word. For these fields, the user selects the field name from the Content Server toolbar and then enters the data in a dialog box.

This kind of data is considered to be Word metadata.

In Word, you can select **File > Properties** and click the **Custom** tab to see these hidden fields and other metadata that results from storing the Word asset in the Content Server database. Do **not** modify this data.

- Obtain information from Word template .dot files. You can create Word templates for the Word assets that provide field/value pairs for specific fields. (You can also use Word templates to create a form for your content providers to use to enter information about their Word assets.)

Preserving the Formatting of the Word Data

You, the administrator, determine how the data from an individual field is managed by using the **Preserve formatting** option when you configure asset types for CS-Desktop.

When you select **Preserve formatting** for a field, CS-Direct converts the Word format to the corresponding HTML format and stores the HTML in the column that represents the field in the Content Server database. Paragraph breaks are converted to <p> tags, bold text is marked with pairs, tables are stored as HTML tables, and so on.

Therefore, it is best if you use the **Preserve formatting** option for fields that contain text only. Do not use the **Preserve formatting** option for fields that contain files (image or other types of files).

It is also best to keep text and images separate—that you do not allow the CS-Desktop users to save both text and images to the same blob attribute or upload field. For example, if you have an article asset type that contains an image with a caption, use one attribute or field for the caption and another attribute or field for the image file.

By keeping text and images in separate fields and controlling whether the formatting is preserved or not, your template writers can know what kind of data will be extracted from a particular field and can code their templates to display that data appropriately. If you decide to allow users to save both text and image to the same blob attribute or upload field, both you and your template developers should note the following:

Format Preserved	Word Content Marked to Be Saved in the Blob Attribute or Upload Field	How the Word Content Is Stored in the Database
Yes	text and image	The image is written to the directory identified by the <code>keyview.imgdir</code> property in <code>futuretense_xcel.ini</code> and the text is stored as HTML in the URL column; the HTML has an unmanaged link to the image in the <code>keyview.imgdir</code> directory. In such a case, you must implement a process that regularly mirrors the <code>keyview.imgdir</code> directory to the delivery system because the publishing system does not mirror this directory.
Yes	text only	The text is stored as HTML in the URL column.
Yes	image only	The image file is written to the URL column.
No	text and image	The text is stripped out and the image file is written to the URL column.

Format Preserved	Word Content Marked to Be Saved in the Blob Attribute or Upload Field	How the Word Content Is Stored in the Database
No	text only	The formatting of the text is stripped out and the text is written to the URL column.
No	image only	The image file is written to the URL column.

Configuring the Word Asset Types for CS-Desktop

The term “Word asset type” refers to any Content Server asset type configured to be created and edited in Microsoft Word. It is not a new asset type, but rather, an asset type already known to Content Server that you (or another administrator) have enabled for Word (CS-Desktop).

While there are no restrictions on which asset types might fall into this category, practical limits require that you restrict the qualifying asset types to those that have text content or are otherwise document-related.

To configure the asset types for CS-Desktop

- Enable the asset types that you want to make available through CS-Desktop.
- For each subtype for the enabled basic asset types and for each definition for the enabled flex asset types, specify which fields will appear on the Content Server toolbar within the Microsoft Word interface.
- For each field that you enable, determine whether Content Server should retain the formatting of the data that was entered in the Microsoft Word application. For fields that hold blobs, use the formatting option to determine how text and files are managed when they are written to that field.
- Create CS-Desktop start menu items for each of the Word asset types.

The following sections describe each step in detail.

Enabling Asset Types for CS-Desktop

You enable an asset type for CS-Desktop by adding it to those that are selected for CS-Desktop.

To enable asset types for CS-Desktop

1. In the Content Server interface, do one of the following:
 - Select **Admin**, expand the **Sites** option, and expand the site that you are configuring CS-Desktop for.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Select **CS-Desktop > Enable CS-Desktop**.

The “Enable Asset Types” form appears. It displays all the asset types that exist on this system that have not yet been enabled for CS-Desktop for this site:

Enable for CS-Desktop:

Name	Description
<input type="checkbox"/> <u>CSElement</u>	CSElement
<input type="checkbox"/> <u>Collection</u>	Collection
<input type="checkbox"/> <u>HelloArticle</u>	HelloArticle
<input type="checkbox"/> <u>HelloImage</u>	HelloImage
<input type="checkbox"/> <u>Page</u>	Page
<input type="checkbox"/> <u>Query</u>	Query
<input type="checkbox"/> <u>SiteEntry</u>	SiteEntry
<input type="checkbox"/> <u>Template</u>	Template

Cancel

Enable Asset Types

3. Select the asset types that you want to enable for this site.
4. Click **Enable Asset Types**.
5. Continue to step 4 in “Specifying Which Fields Are on the Content Server Toolbar,” on page 201.

Specifying Which Fields Are on the Content Server Toolbar

Before you begin this procedure, be sure that you have enabled the asset type that you want to configure for CS-Desktop. If you have not yet enabled the asset type, see “Enabling Asset Types for CS-Desktop,” on page 200.

To specify which fields of an asset type will be displayed on the CS toolbar in the MS Word interface

1. In the Content Server interface, do one of the following:
 - Select **Admin**, expand the **Sites** option, and expand the site you are configuring CS-Desktop for.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Expand the **CS-Desktop** option.
3. Expand the asset type that you want to configure.
4. Select the appropriate subtype (basic asset types) or definition (flex and flex parent asset types), and then click **Edit Configuration**.

CS-Direct displays a form similar to this one:

Edit Configuration of asset type HelloArticle, subtype food for CS-Desktop

Fieldname	Description	Enable?	Preserve formatting?
category	Category	Required	<input type="checkbox"/>
description	Description	<input type="checkbox"/>	<input type="checkbox"/>
Publist	Sites	Required	<input type="checkbox"/>
template	Template	<input type="checkbox"/>	<input type="checkbox"/>
headline	Headline	Required	<input type="checkbox"/>
urlbody	Body	Required	<input type="checkbox"/>
startdate	Start Date	<input type="checkbox"/>	<input type="checkbox"/>
Association-named		<input type="checkbox"/>	<input type="checkbox"/>
name	Name	Required	<input type="checkbox"/>
byline	Byline	Required	<input type="checkbox"/>
enddate	End Date	<input type="checkbox"/>	<input type="checkbox"/>
filename	Filename	<input type="checkbox"/>	<input type="checkbox"/>
path	Path	<input type="checkbox"/>	<input type="checkbox"/>

5. In the “Edit Configuration” form, select the **Enable** option for each field that you want to appear on the Content Server toolbar. As a general rule, select only text-entry fields. Note that any required fields are already selected.
6. (Optional) If you want the format of the text that is entered in a field through Word to be preserved when that data is stored in the Content Server database, select the **Preserve formatting** option for that field.
As a general rule, enable the **Preserve formatting** option only for fields that store text. Do not enable the **Preserve formatting** option for fields that hold uploaded files.
7. Click **Save**.
8. Repeat steps 4 through 7 for each subtype or definition for this asset type.
9. Repeat steps 3 through 8 for each asset type.

Creating CS-Desktop Start Menu Items

The subtype configuration determines which fields are available on the Content Server toolbar in the Microsoft Word interface, but your Word asset types do not become available in Word until you create start menu items for them.

To create CS-Desktop Start Menu items

1. In the Content Server interface, select the **Admin** tab.
2. Expand **Asset Types** and then expand the asset type that you are interested in.
3. Select **Start Menu > Add New Start Menu**.
4. Click in the **Name** field and enter the name of the item. This is the name that is displayed in the list of asset types that are available from the Content Server toolbar.
5. Click in the **Description** field and enter a short description for the start menu item.
6. In the **Type** field, select **CS-Desktop**.

Note

You cannot set default field values for assets whose source is Word. Ignore the **Default Values** section when creating start menu items for Word asset types.

7. In the **Sites** field, select which sites can use this start menu item. Use Ctrl + click to select more than one site.
8. In the **Roles** field, select all the roles that can have access to the start menu item. Use Ctrl + click to select more than one role.
9. (Optional) To configure workflow process details for the assets that are created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the “Select Workflow” form, select the appropriate workflow process from the drop-down list, and then click **Select Workflow**.
 - c. In the “Set Participants” form, select at least one user for each role that appears and then click **Set Participants**.
 - d. Click **Continue**.

The system saves the workflow information and displays the “Start Menu” form again.
10. In the “Start Menu” form, click **Save**.
11. Repeat steps 2 through 10 for each of the Word asset types.

User Accounts and CS-Desktop

Like the content providers who use the Content Server interface, the content providers who use the CS-Desktop application must have Content Server user accounts as well. User accounts for the individuals who use CS-Desktop must have the following ACLs:

- Browser
- ElementReader
- PageReader
- RemoteClient
- xceleditor
- UserReader
- Visitor (if you have Engage installed)

For information about creating user accounts, see “Editing a User,” on page 78, and “Required ACLs for Custom Users,” on page 312.

Give the CS-Desktop users the following information:

- Their user name/password combination
- The appropriate URL for logging in to your CS management system

They need both pieces of information to log in to Content Server through CS-Desktop. Your Content Server URL uses one of the following conventions:

All Other Application Servers, Including Sun ONE

`http://<server>:<port>/servlet`

where *<server>* is the name of the server that hosts your CS management system and *<port>* is the port number.

Installing the CS-Desktop Client Application

Each of the content providers who will use the CS-Desktop feature must run the `csdesktop.exe` executable file on their client systems.

To install the client application

1. On a PC that has a supported operating system and a supported version of Microsoft Word, set the encoding either to UTF-8 or to another type of encoding that supports an even larger character set than does UTF-8.

2. Open Internet Explorer and navigate to the following URL:

`http://yourserver:itsPortNumber/Xcelerate/DownloadPage.html`

where *yourserver* is the name of the server that hosts your CS management system and *itsPortNumber* is the port number.

3. On the **Download** page, select **CS-Desktop**, and then click **Download**. Note that if there are any language packs installed on your CS system, the **Download** page presents CS-Desktop options in the languages that are installed on the system.

4. In the **File Download** dialog box, click **Open**.

The Winzip application starts.

5. Click **I Agree**.

Winzip unzips the zip file.

6. In the Winzip window, double-click on `csdesktop.exe`.

The installation program begins.

7. Follow the instructions that appear on the screen.

Specifying Locale for the Client Application

The Download page is generated dynamically based on the language packs that are installed on your CS system. If your CS system has a language pack installed, the Download page presents a CS-Desktop option in that language and the `csdesktop.exe` file installs a version of the help file in that language.

The first time a CS-Desktop user logs in to Content Server, the Log In dialog box is in English. The user clicks the Select Language link to select the language that matches the version of CS-Desktop help file that he or she installed.

After a user selects a language, CS-Desktop copies the appropriate XML language file from Content Server to that user's hard drive. CS-Desktop uses the information in that language file for menu names and dialog box text until the user selects a different language.

Additionally, whenever the language file on the Content Server server is updated, be sure to remind your CS-Desktop users to update their language file on their hard disks by using the Select Language link again the next time they log in to Content Server.

Testing the CS-Desktop Configuration

To determine whether your CS-Desktop application is configured correctly, you must create and save an asset from the Microsoft Word interface and then examine it through CS-Direct. Complete the following steps:

1. Open Microsoft Word.
2. On the button bar, select **Content Server > Login**.
3. In the **Login** dialog box, enter the following information:
 - The name and password of a user who has the RemoteClient ACL.
 - The server URL, as follows:

All Other Application Servers, including Sun ONE:

```
http://<server>:<port>/servlet
```

where *<server>* is the name of the server that hosts your CS management system and *<Port>* is the port number.

4. Select the site, asset type, and subtype, as necessary, to set the field markers list.
5. Enter text in the Word document.
6. Select (highlight) the entire range of text.
7. Click the **Field Markers** button on the toolbar and select the marker for the field that you are entering text for.
8. Repeat steps 6 through 7 for all required asset fields that require you to enter text in the Word document.
9. Set values for the field markers with ellipses (...). These field markers represent hidden fields; that is, their values do not appear in the Word document, but are part of the asset definition, and are thus stored in the Content Server database. When you click a field marker with an ellipsis, a dialog box opens. Enter the appropriate data.
10. On the button bar, select **Content Server > Save to Content Server**.
11. Open Internet Explorer and log in to the Content Server interface as a user who has access to the type of asset that you just created.
12. Select the site that you created the asset for and then click **Search**.
13. Find the asset that you entered through CS-Desktop.
14. Click **Inspect**. (Do **not** select **Edit**.)
15. Examine the data listed for the fields and verify that the data that you marked in CS-Desktop matches the data displayed in the “Inspect” form.

Configuring Word Templates for the Word Assets

The field markers on the Content Server toolbar are Word bookmarks. Therefore, you can create .dot files for your Word assets, marking the appropriate areas of those templates with either the Content Server field markers or your own Word bookmarks.

Create a new Word document and mark placeholder text with the appropriate field markers, just as you would for an asset of that type (that is, be sure that you do not mark

text for the metadata fields such as Name and Description and so on). Then save the document as a template (a .dot file) rather than to Content Server.

CS-DocLink

Content Server DocLink (CS-DocLink) provides a drag-and-drop interface for uploading and downloading documents, graphics, or other files that are managed as flex assets by Content Server. This application presents the hierarchical data structure of the flex parents and flex assets in your Content Server database as folders and files in the Windows Explorer application.

Overview

CS-DocLink enables your content providers to use their third-party tools to create content for your online sites, and then save it to the Content Server database without having to log in to the Content Server interface.

CS-DocLink enables content providers to drag and drop Word files, Excel spreadsheets, graphic files, and so on—that is, single binary files—into the Content Server database by dropping each file onto the appropriate Microsoft Windows Explorer folder that represents the asset type or parent asset type that the document should be saved as.

After a content provider drops the file onto a folder, CS-DocLink displays a dialog box that prompts the user to enter any metadata that is required, depending on how you, the administrator, have configured your system. Examples of such metadata include name, description, and so on.

CS-DocLink differs from CS-Desktop for Word as follows:

- With CS-Desktop, content providers produce the content and tag the data in that document that should be inserted into various fields from within the Word interface. The tagging process creates a structured asset when the document is stored in the database as an asset.
- With CS-DocLink, content providers produce their content in their third-party applications but they do not use those applications to provide structure to the data.

They drop a binary file into the Windows Explorer interface, and then provide information about the file through the CS-DocLink dialog boxes that appear in response to the drop. The information required depends on the kind of asset type the file is going to be stored as, based on the place in the graphical representation of the database that the user dropped the file.

Configuring the CS-DocLink Asset Types

The term “CS-DocLink asset type” refers to a flex or flex parent asset type that is configured to be accepted by CS-DocLink. CS-DocLink asset types must be flex asset types that store an uploaded, binary file. That is, one (and **only** one) of the flex attributes that define the asset **must be** of type **blob**.

To configure the CS-DocLink asset types

- Verify that the flex or flex parent asset type is enabled for the site. Be sure that the asset type has **one** flex attribute of type **blob**.
- Enable the flex or flex parent asset type for CS-DocLink.

- For each definition for those enabled asset types, specify which fields will be available in CS-DocLink and specify which field is the upload field.
- Create CS-DocLink start menu items for each of the CS-DocLink asset types.

Enabling Asset Types for CS-DocLink

You enable an asset type for CS-DocLink by adding it to those that are selected for CS-DocLink.

To enable asset types for CS-DocLink

1. In the Content Server interface, do one of the following:
 - Select **Admin**, expand the **Sites** option, and expand the site that you are configuring CS-DocLink for.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Select **CS-DocLink > Enable CS-DocLink**.

The “Enable Asset Types” form appears. It displays all the asset types that exist on this system that have not yet been enabled for CS-DocLink for this site:

Enable for CS-DocLink:

Name	Description
<input type="checkbox"/> AArticles	Article (Flex)
<input type="checkbox"/> AttrTypes	Attribute Editor
<input type="checkbox"/> CSElement	CSElement
<input type="checkbox"/> Collection	Collection
<input type="checkbox"/> CAttributes	Content Attribute
<input type="checkbox"/> ContentTmpls	Content Definition
<input type="checkbox"/> ContentGroups	Content Parent
<input type="checkbox"/> CGroupTmpls	Content Parent Definition
<input type="checkbox"/> HFields	History Attribute
<input type="checkbox"/> HistoryVals	History Definition
<input type="checkbox"/> AImages	Image (Flex)
<input type="checkbox"/> Link	Link
<input type="checkbox"/> Page	Page
<input type="checkbox"/> PAttributes	Product Attribute
<input type="checkbox"/> ProductTmpls	Product Definition
<input type="checkbox"/> ProductGroups	Product Parent
<input type="checkbox"/> PGroupTmpls	Product Parent Definition
<input type="checkbox"/> Products	Products
<input type="checkbox"/> Promotions	Promotion
<input type="checkbox"/> Query	Query
<input type="checkbox"/> AdvCols	Recommendation
<input type="checkbox"/> Segments	Segment
<input type="checkbox"/> SiteEntry	SiteEntry
<input type="checkbox"/> Template	Template
<input type="checkbox"/> ScalarVals	Visitor Attribute

Cancel

Enable Asset Types

3. Select the asset types that you want to enable for this site.
4. Click **Enable Asset Types**.
The asset types appear on the **Admin** tab under the CS-DocLink node for this site.
5. Continue to step 3 in “Specifying Which Fields Are on the Content Server Toolbar,” on page 201.

Specifying Which Fields Are Available in CS-DocLink

Before you begin this procedure, be sure that the following conditions are true:

- You have enabled the asset type that you want to configure for CS-DocLink. If you have not yet enabled the asset type, see “Enabling Asset Types for CS-DocLink,” on page 207.
- The asset type that you want to configure has one attribute of type blob.

To specify which of an asset type’s fields will be available to content providers who are using CS-DocLink to create or edit assets of that type

1. In the Content Server interface, do one of the following:

- Select **Admin**, expand the **Sites** option, and expand the site that you are configuring CS-DocLink for.
 - If you have logged in to the site that you are configuring and you have access to the **SiteAdmin** tab, select the **SiteAdmin** tab.
2. Expand the **CS-DocLink** option.
 3. Under the CS-DocLink option, expand the asset type that you want to configure and then select the appropriate definition.
 4. Click **Edit Configuration**.

CS-Direct displays a form similar to this one:

Configuration of asset type PDF, subtype PDFTYPE for CS-DocLink

Fieldname	Description	Enable?	Accepts Documents?
description	Description	<input type="checkbox"/>	-
Publist	Sites	<input type="checkbox"/>	-
Relationships		<input type="checkbox"/>	-
template	Template	<input type="checkbox"/>	-
renderid		<input type="checkbox"/>	-
startdate	Start Date	<input type="checkbox"/>	-
name	Name	<input type="checkbox"/>	-
Attribute_PDFFile	PDFFile	<input type="checkbox"/>	-
			<input type="button" value="Select Document Types"/>
enddate	End Date	<input type="checkbox"/>	-
filename	Filename	<input type="checkbox"/>	-
ruleset		<input type="checkbox"/>	-
path	Path	<input type="checkbox"/>	-

5. In the “Edit Configuration” form, select the **Enable** option for each field that you want to be displayed in the CS-DocLink interface. An enabled field means that users can view and edit the data in that field.

As a general rule, enable the following kinds of fields:

- The blob attribute that will hold the file (just one)
- Text entry fields (**Description**, for example)

Do **not** enable the **Publist** field—assets cannot be shared with other sites through the CS-DocLink interface.

Note that users are prompted to fill in the data for required fields when they create an asset—whether or not you have selected the **Enable** option for those fields. Therefore, if you do not want users to be able to edit the data in a required field for existing assets of this type, do not enable the field.

6. Next to the upload field (flex attribute) that you want to specify as the field that documents are to be stored in, click the **Select Document Types** button.

Note

If your developers have created a flex filter for assets of this type, the flex attribute that you specify as the upload field must match the field that is designated as the input attribute for the flex filter.

For information about flex filters, see the *Content Server Developer's Guide* and talk to your developers.

CS-Direct displays a form similar to this one:

Configuration of asset type PDF, subtype PDFType for CS-DocLink

Fieldname	Document Types	Max file size
PDFFile	<div>Any</div> <div>CSS StyleSheet</div> <div>HTML page</div> <div>Image in GIF format</div> <div>Image in JPEG format</div> <div>MS Word</div>	<input type="text"/> bytes

7. In the **Document Types** field, select the types of documents that this upload field can store. Use Ctrl + click to select more than one.

The values displayed in the **Document Types** field come from the `description` column of the entries in the `MimeType` table. If the document type that you want to select does not appear in the list, use Content Server Explorer to add it to the `MimeType` table.

Note

When you select the **Any** option, it means that CS-DocLink accepts any type of file for assets of this type, whether or not there is an entry for it in the `MimeType` table.

However, when there is no entry for a file type in the `MimeType` table, a file of that type cannot be displayed correctly in the Content Server interface.

8. In the **Max file size** field, specify a size limit (if there is one) for the documents that can be stored in this upload field.
9. Click **Save**.
10. In the “Configuration” form, click **Save** again.
11. Repeat steps 3 through 10 for each definition for this asset type.
12. Repeat steps 3 through 11 for each asset type that you want to configure.

Creating CS-DocLink Start Menu Items

The CS-DocLink configuration determines which fields are available in the CS-DocLink interface, but your content providers cannot create assets of this type in CS-DocLink until you create start menu items for those asset types.

To create CS-DocLink Start Menu items

1. In the Content Server interface, select the **Admin** tab.
2. Expand **Asset Types** and then expand the asset type that you are interested in.
3. Select **Start Menu > Add New Start Menu**.
4. Click in the **Name** field and enter the name of the item.
5. Click in the **Description** field and enter a short description for the start menu item.
6. In the **Type** field, select **CS-DocLink**.

Note

You cannot set default field values for assets whose source is CS-DocLink in this version. Ignore the **Default Values** section when creating CS-DocLink start menu items.

7. In the **Sites** field, select which sites can use this start menu item. Use Ctrl + click to select more than one site.
8. In the **Roles** field, select all the roles that can have access to the start menu item. Use Ctrl + click to select more than one role.
9. (Optional) To configure workflow process details for the assets that are created with this start menu item, complete the following steps:
 - a. Click **Select Workflow**.
 - b. In the “Select Workflow” form, select the appropriate workflow process from the drop-down list and then click **Select Workflow**.
 - c. In the “Set Participants” form, select at least one user for each role that appears and then click **Set Participants**.
 - d. Click **Continue**.

The system saves the workflow information and displays the “Start Menu” form again.
10. In the “Start Menu” form, click **Save**.
11. Repeat steps 2 through 10 for each of the Word asset types.

Users and CS-DocLink

Like the content providers who use the Content Server interface, the content providers who use the CS-DocLink application must have Content Server user accounts as well. User accounts for the folks who use CS-DocLink must have the following ACLs:

- Browser
- ElementReader

- PageReader
- RemoteClient
- UserReader
- xceleeditor

For information about creating user accounts, see “Creating a New User,” on page 77.

Give the CS-DocLink users the following information:

- Their user name/password combination
- The appropriate URL for logging in to the Content Server database

They need both pieces of information to log in to Content Server through CS-DocLink. Your CS-DocLink URL uses one of the following conventions:

All Other Application Servers, including Sun ONE

`http://<server>:<port>/servlet/ContentServer?pagename=OpenMarket/CSCClient/InitialURLs`

where *<server>* is the name of the server that hosts your CS management system and *<Port>* is the port number.

Installing the CS-DocLink Client Application

Each of the content providers who will use the CS-DocLink feature must download the application to their client systems.

To install the client application

1. Open Internet Explorer and navigate to the following URL:
`http://yourserver:itsPortNumber/Xcelerate/DownloadPage.html`
where *yourserver* is the name of the server that hosts your CS management system and *itsPortNumber* is the port number.
2. On the **Download** page, select **CS-DocLink**, and then click **Download**.
3. In the **File Download** dialog box, click **Open**.
The Winzip application starts.
4. Click **I Agree**.
Winzip unzips the zip file.
5. In the Winzip window, double-click on `setup.exe`.
The installation program begins.
6. Follow the instructions that appear on the screen.

Testing the CS-DocLink Configuration

To determine whether your CS-DocLink application is configured correctly, you must create and save an asset from the CS-DocLink interface and then examine it through CS-Direct.

To test the CS-DocLink configuration

1. Open Microsoft Explorer.

2. Double-click on the CS-DocLink node.
3. In the **Login** dialog box, enter the following information:
 - The CS-DocLink URL. See “Users and CS-DocLink,” on page 211 for the appropriate URL for your system.
 - The username and password of a user who has the ACLs listed in “Users and CS-DocLink,” on page 211.
4. Drag a document from the file system to the file structure represented under the CS-DocLink node and drop it in the appropriate place.
5. If there are required fields that must be filled in, the **Properties** dialog box appears. Enter the required information and save the asset.
6. Open Internet Explorer and log in to the Content Server interface as a user who has access to the type of asset that you just created.
7. Select the site that you created the asset for and then click **Search**.
8. Find the asset that you created through CS-DocLink.
9. Click **Inspect**.
10. Examine the data listed for the fields and verify that the data that you specified in CS-DocLink matches the data displayed in the “Inspect” form.

InSite Editor

The InSite Editor is a CS-Direct feature that enables infrequent users or users who perform a limited role to find, edit, and submit content directly from the rendered (Preview) version of an asset, which means that they do not have to learn how to use the Content Server interface.

Instead, they enter a Content Server URL that is appropriate for your system: for example, a URL that displays the home page of the rendered site. If the template for that page is coded with the `INSITE.EDIT` tags, the InSite Editor control panel appears.

The control panel provides a subset of the CS-Direct features and functions. Your InSite Editor users interact with the CS system by using the control panel instead of the Content Server interface.

Enabling this feature requires four general steps:

- You, the administrator, set the `xcelerate.enableinsite` property in the `futuretense_xcel.ini` file to `true` on the CS management system.
- Your developers code templates that invoke the InSite Editor feature for the fields that you want content providers to be able to edit in this way. Note that these same templates do not display the InSite Editor control panel on the CS 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 CS users who need to use the InSite Editor.

Note

The InSite Editor cannot display assets that are created or edited with the CS-Desktop feature. That is, if an asset has a value in its `externaldoctype` column, the InSite Editor cannot display it.

Enabling the InSite Editor

To enable the InSite Editor

1. Start the Property Editor and open the `futuretense_xcel.ini` file.
2. On the **xcelerate** tab, select `xcelerate.enableinsite` and set its value to `true`.
3. (Optional) If you want to use Ektron's eWebEditPro HTML editor within the InSite Editor, verify that it is installed and confirm that the `xcelerate.ewebeditpro` property is set to `true`.

For information about eWebEditPro, see “eWebEditPro,” on page 215.

Note that to use eWebEditPro within the InSite Editor, your developers must use the `EWEBEDITPRO` parameter with the `INSITE.EDIT` tags on your templates. For more information, see the *Content Server Developer's Guide*.

4. Select **File > Save**.
5. Select **File > Exit**.

After you enable the InSite Editor, CS-Direct displays the InSite Editor control panel whenever a CS content provider uses a browser to navigate to an asset that is rendered by a template coded with the `INSITE.EDIT` tag.

For information about how to use the control panel, see the InSite Editor help file.

User Accounts and the InSite Editor

Like the content providers who use the Content Server interface, the content providers who use the InSite Editor control panel must have Content Server user accounts as well. User accounts for the content providers who use the InSite Editor must have the following ACLs:

- Browser
- ElementReader
- PageReader
- RemoteClient
- UserReader
- Visitor (if you are using Engage)
- xceleeditor

For information about creating user accounts, see “Creating a New User,” on page 77.

Be sure that you give the InSite Editor users the following information:

- Their user name/password combination

- The appropriate URL for your system; that is, the URL that serves the appropriate site page (that is, `SiteCatalog` page entry) for a session in which someone uses the InSite Editor

eWebEditPro

eWebEditPro is a third-party HTML editor from Ektron, Inc. that the CS-Direct and CS-Direct Advantage products support. Your developers can create basic assets whose text-entry fields use eWebEditPro as the input mechanism for the field. They can also create attribute editors for flex attributes that use eWebEditPro as the input mechanism.

Three versions of eWebEditPro are supported: v 3.0.0.7, v 4.0.0.14 (both used strictly as HTML editors), and eWebEditPro+XML. Only one version of eWebEditPro can be used by each Content Management installation. To obtain eWebEditPro or eWebEditPro+XML, contact your FatWire sales representative.

Configuring eWebEditPro

Installation and configuration instructions are included in the eWebEditPro release notes that accompany the version that you purchase through FatWire.

User Accounts and eWebEditPro

You do not need to add any extra ACLs to user accounts for CS users who work with eWebEditPro. If a user can log in to the Content Server interface, he or she can enter data in an asset field that has eWebEditPro as the input mechanism.

Maintaining Separate Browser Sessions for Preview

If the online site that you are serving from your CS delivery system requires visitors to log in, or it logs them in by default, your content providers need a separate browser session for preview if they must log in to the previewed site on the CS management system with a user name that is **different** from the one they are using in Content Server.

Without a new, separate browser session for preview, the Content Server session ends when the preview session begins, which means that content providers can lose any unsaved work.

To ensure that a preview session does not end the Content Server session, you can do the following:

- 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 Content Server user accounts.
- Configure your CS management system so that the web server opens a new browser session for the preview window without closing the current session in the Content Server window.

To configure your CS management system so that the web server opens a new browser session for the previewed asset, you configure properties in the

futuretense_xcel.ini file: `xcelerate.previewhost` and `xcelerate.previewervlet`.

You use these properties to point to the URL for the CS 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 CS, include it in the value for the `xcelerate.previewhost` property.
- If content providers are required to include the port number, specify a different port number for the property—the web server port number or the port number of a redirect request to the web server.
- Create an alias and use it only for the `xcelerate.previewhost` property.
- Use the `xcelerate.previewervlet` property to specify the `SatelliteServer` servlet rather than the `ContentServer` servlet.

Start the Property Editor, open the `futuretense_xcel.ini` file, select the **xcelerate** tab, and then do one or both of the following:

- Set the `xcelerate.previewhost` property as follows:
For most application servers, including Sun ONE:
`http://servername:port/servlet/`
- Change the value of the `xcelerate.previewervlet` property from `ContentServer` to `Satellite`

Chapter 12

Managing the CS Publishing System

You make content available to the visitors of your online site by moving it from your management system to your delivery system. Moving content from one CS system to another is called **publishing**.

Content Server provides two publishing APIs, called Export and Mirror. The Export API renders Content Server pages into files. The Mirror API copies rows from tables in one Content Server database to the corresponding tables in another Content Server database.

CS-Direct delivers a publishing system that uses these Content Server APIs. While the actual rendering of files or mirroring of rows is completed by the API, the CS-Direct publishing system includes an approval system that determines which content gets published, a schedule that determines when publishing occurs, and the ability to configure site-specific destinations. In addition, in dynamic CS systems, the publishing system communicates with the CacheManager on the destination server.

This chapter describes the CS-Direct publishing system, which is also used by the other CS content applications. It contains the following sections:

- Overview
- The Export to Disk Delivery Type
- The Mirror to Server Delivery Type
- The Export Assets to XML Delivery Type
- Configuring the Publishing Process for Your CS System
- Additional Publishing Procedures
- Troubleshooting

For information about the Content Server Export and Mirror APIs, see the *Content Server Developer's Tag Reference*.

Overview

The CS-Direct publishing system interacts with several underlying systems and APIs before, during, and after a publishing session.

Before CS-Direct can publish assets to a destination system, the following information must be determined:

- Which **delivery type** should CS-Direct use: Export to Disk, Export Assets to XML, or Mirror to Server?

As an administrator, you specify the delivery type for the destination when you configure the publishing destination.

- Where should CS-Direct publish to? That is, what is the destination for the publishing session?

As an administrator, you configure the **publishing destinations** for your system. When content providers mark their assets “approved,” they approve them for a specific destination.

- Which assets are ready to be published to the current destination?

The **approval** system manages this information. When an asset is ready to be published, a content provider marks it as “approved” for a specific destination. The publishing system checks with the approval system and then only the assets that are marked as approved are published, which protects the site from having any broken links. Connected parts of the site are held until they are all ready to be published.

- When should the assets be published?

As an administrator, you set up the **publishing schedule**. The publishing process runs as a batch process that you schedule to occur at regularly-occurring intervals. On an as-needed basis, you can also override the schedule and publish on demand.

During a publishing session, CS-Direct writes information about that session to log files. Because the publishing process runs in the background, you can use your browser to complete other administrative tasks while the session runs. And you can monitor the session through the **Publish Console**, which shows both the publishing history and the status of any publishing sessions that are currently running.

When the publishing session uses the Mirror to Server delivery type, the publishing system communicates and cooperates with the **CacheManager** on the delivery system. The CacheManager is a Content Server servlet that manages the page cache on a system. It ensures that pagelets or pages that refer to the assets that will be mirrored are cached. Then, after the publishing session concludes, CacheManager generates those pages again to display the updated content, and caches the new pages and pagelets.

Delivery Types

The term “delivery type” means the publishing method. There are three delivery types and they use either the Export API or the Mirror API. Which delivery type you use for your CS system depends on how you plan to deliver the content on your delivery system (statically or dynamically) or whether you are delivering your content to some other non-CS system:

- On a **static** CS delivery system, a web server delivers static files to your site visitors. You use the **Export to Disk** delivery type to create the files. During an export publishing process, CS-Direct instructs Content Server to render all approved assets according to their templates and then saves those files. You, the administrator, must then use FTP or another file transfer protocol to move those files to the web server that is your delivery system.
- On a **dynamic** CS delivery system, Content Server generates (renders) pages upon request. Content Server makes dynamic content decisions based on the request. You use the **Mirror to Server** delivery type to copy the rows for your approved assets from the Content Server database on the management system to the Content Server database on the delivery system.
- The **Export Assets to XML** delivery type is a data transformation method. Rather than creating pages that are ready to be displayed by a web server, it uses the Export API to create one XML file for each approved asset. You use this delivery type when your end goal is to publish assets to another (non-CS) delivery system or database and that system needs your content to be formatted in XML.

These sections describe each delivery type in detail:

- “The Export to Disk Delivery Type,” on page 228.
- “The Mirror to Server Delivery Type,” on page 238.
- “The Export Assets to XML Delivery Type,” on page 242.

Publishing Destinations

A publishing destination defines the location where your content is published to. A destination is a named object that contains the following information:

- A delivery type (that is, Export to Disk, Mirror to Server, or Export Assets to XML)
- A location
 - For Export to Disk and Export Assets to XML, the location is the **root file directory** that the resulting HTML or XML files are saved to after the publishing system converts the approved assets into files. You set this directory with the `cs.pgexportfolder` property in the `futuretense.ini` file.
 - For Mirror to Server, the destination is the **server name in URL format** (which includes the port number if it is anything other than 80) of the server that is supposed to deliver that content. You set the server name in the **Destination address** field of the “Destination” form.
- Various **publishing arguments**, depending on which delivery type you are using. For example, if you are using Export to Disk you could specify a URL prefix for your internal links (HREFs).
 - For information about Export to Disk publishing arguments, see “Publishing Arguments for Export to Disk,” on page 230.

- For information about Mirror to Server publishing arguments, see “Publishing Arguments for Mirror to Server,” on page 240.
- For information about Export Assets to XML publishing arguments, see “Publishing Arguments for Export Assets to XML,” on page 244.
- The CM sites whose assets can be published to the destination.

You can configure more than one publishing destination for your CS system. Depending on how your online sites are designed by the developers, you may need to publish both static and dynamic content. For this kind of dual implementation, consult with your developers to determine how to configure your publishing destinations.

In this document, the following terms are used when discussing publishing destinations and configuration:

- **Source**, which means the Content Server database that serves as the source for a publishing session. Because you can mirror assets from any CS system to any other CS system, the source is not always the CS management system.
- **Destination**, which means either the Content Server database that you are mirroring to or the directory that you are exporting to.

For information about creating publishing destinations, see “Step 2 (Export to Disk): Configure an Export Destination,” on page 247 and “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.

Publishing and the Approval Process

The publishing system and the approval system work very closely together. The approval system determines the answers to the following questions:

- Which assets are ready to be published (that is, are marked as approved)?
- Are there any assets that must accompany those approved assets or that must exist on the destination to ensure that all the links on the site will work correctly?

When an asset is ready to be published, a content provider marks it as approved for a specific destination. (For information about how you approve assets, see the chapter on publishing in the *Content Server User's Guide*.) CS-Direct then verifies that the asset is ready to be published by calculating the list of assets upon which approval depends.

For example, the Burlington Financial sample site has article assets with associated imagefiles. An approved article with an unapproved associated imagefile cannot be published until the imagefile is also approved.

It is typical to build an approval step into your workflow processes. For information about workflow, see Chapter 9, “Creating and Managing Workflow Processes.”

Approval Dependencies

The approval status of an asset is determined by its dependency relationships, which include the approval status of all asset items associated with a particular asset item, as well as the dependency relationships of those associated items.

How are approval dependencies calculated? That depends on the publishing method:

- For **Export to Disk**, the approval system renders the asset according to the template that is assigned to it when the asset is approved, or, if there is one, according to the default approval template for assets of that type. The tags in the template code set approval dependencies that determine the appropriate dependents for the approved

asset. The dependent assets must be in an appropriate approval state before the current asset can be published.

- For **Mirror to Server** or **Export Assets to XML**, the approval process examines the data relationships between asset types. Basic assets have associations. Flex assets have family relationships. Both of these relationships create approval dependencies for these publishing methods. For example, if you approve a flex asset, it will be held from a publishing session unless its parent assets are in an appropriate approval state.

When the approval system calculates approval dependencies, it further categorizes those dependencies by the following types:

- **Exists** – some version of the dependent asset must exist on the destination so that links to it will work.
- **Exact** – the version of the dependent asset must exactly match the version of it that was approved. That version must either already exist on the destination or be approved for the destination so that it can be published with the asset that is being evaluated. The version of an asset is based on the timestamp in the `updateddate` column for that asset in the asset's primary table.

An asset can be published only if it meets all specified dependencies: that is, all associated assets are also in the correct approval state (either approved or previously published, as necessary). If not, the approved asset is held by CS-Direct until its dependent assets have been approved.

Note

When someone deletes an asset, its status is changed to Void and it is automatically approved for any destination that it was ever published to. Then, during the next publishing sessions for those destinations, that asset is published and its status becomes Void on those destinations.

Held Assets

A **held** asset is one that cannot be published because its approval dependencies are not yet met. When its approval dependencies are met, it will be published.

An asset will be **held** from a publishing session when any of the following conditions are true:

- It is not approved.
- It was approved, but someone edited it again and it has not yet been re-approved.
- It has a dependent asset with an exact dependency that is not approved.
- It has a dependent asset with an exists dependency and that dependent asset has never been published.
- It has a dependent asset with an exact dependency, the dependent asset has been published, but it has since been edited and is not yet approved.
- It is an exact dependent of another asset that was previously approved, but has since been edited and is not yet re-approved.
- It is checked out (revision tracking).

The approval system calculates dependencies for basic assets differently when the delivery type is Export to Disk than when it is Mirror to Server and Export Assets to XML.

Calculating Dependencies for Export to Disk

When an asset is approved to a destination that uses **Export to Disk**, the approval system determines the compositional dependencies for that asset by evaluating the code in the template that is assigned to the asset. It performs a “test-render.”

- For **basic** assets, tags such as `ASSET.LOAD`, `RENDER.LOGDEP`, `ASSET.SITEPARENT`, and `RENDER.GETPAGEURL` mark dependencies between assets and the pages that they are rendered on.
- For **flex** assets, several tags in the `ASSETSET` family of tags, `RENDER.LOGDEP`, and the `RENDER.GETPAGEURL` tags mark dependencies between assets and the pages that they are rendered on.

When the delivery type is Export to Disk, the dependencies are determined as the code in the template is evaluated. If a default template has been assigned for 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 specified for the asset.

Your developers create default templates for asset types for the following reason: when Export to Disk actually publishes the asset, it does not necessarily use the template that is assigned to the asset—the code in another element could determine that a different template is used for that asset in certain cases. If this is the case for your online site, it is likely that the developers who created the templates also designed default templates for the approval system to use when calculating approvals.

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,” on page 272.

For more information about how dependencies are created between assets when the delivery type is Export to Disk, see the chapters on template design in the *Content Server Developer's Guide*.

Calculating Dependencies for Mirror to Server and Export Assets to XML

When an asset is approved to a destination that uses **Mirror to Server** or **Export Assets to XML**, the approval system calculates the dependencies for that asset based on the relationships to other assets that are described in the database. That is, approval is based on data dependencies.

Basic Asset Types

For basic assets, the approval system follows these dependency rules:

- Either an exists or an exact dependency on dependent assets referenced through named asset associations, depending on how your asset associations are designed
- An exists dependency on dependent assets referenced through unnamed asset associations
- For page assets, an exists dependency for page assets that are lower than it in the hierarchy of the site plan (which is reflected in the `SitePlanTree` table)
- An exists dependency on assets referenced by an embedded link

- An exact dependency on assets referenced as an embedded pagelet

The following table summarizes this information:

Relationship to Approved Asset	Exists	Exact
Named association	✓ by default, unless specifically configured to be exact	
Unnamed association	✓	
For page asset only: another page asset at a lower level in site plan	✓	
Embedded link	✓	
Embedded pagelet		✓

For information about embedded links and embedded pagelets, see the *Content Server Developer's Guide*.

CSElement and SiteEntry Assets

The root element for a SiteEntry asset is represented by a CSElement asset. There is an exists dependency on the CSElement asset that a SiteEntry asset refers to.

Flex Families

For the flex asset family member and the flex parent family member, the approval system follows these dependency rules:

- An exists dependency on its flex definition (or parent definition, for a parent asset)
- An exists dependency on its attributes

Note

The “exists” and “exact” dependencies are configurable. Which dependency is in effect depends on the value of the property `mwb.conservative.dependencies`, in the `gator.ini` property file. The default value is `false`, which sets “exists” dependencies. If “exact” dependencies are required, the value of `mwb.conservative.dependencies` must be set to `true`. If you change the value of the property, you must re-approve the assets that were affected by the change.

Note also that when `mwb.conservative.dependencies` 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.

- An exists dependency on its parents
- An exists dependency on its template

The following rules are used for flex definitions and flex parent definitions:

- An exists dependency on its parent definitions
- An exists dependency on its attributes

The following rules are used for flex attributes:

- An exists dependency on its attribute editor (if one is assigned).
- 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” dependent on that asset.

The following table summarizes this information:

Approved Asset	Dependency Type of Dependent Assets					
	Flex Parent Definition	Flex Parent	Flex Definition	Flex Attributes	Flex Filter	Attribute Editor
Flex Parent Definition	exists			exists		
Flex Definition	exists			exists		
Flex Parent	exists	exists		exists		
Flex Asset		exists	exists	exists		
Flex Attribute					exists	exists

Engage Visitor Data Assets

The approval system uses the following rules to calculate data dependencies for the Engage visitor data asset types:

- 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)

The following table summarizes this information:

Approved Asset	Dependency Type of Dependent Assets				
	History Definition	History Attribute	Visitor Attribute	Recommendation	Related Flex Asset
History definition		exact			
Segment	exists		exists		
Recommendation					

Approved Asset	Dependency Type of Dependent Assets				
	History Definition	History Attribute	Visitor Attribute	Recommendation	Related Flex Asset
Promotion				exists	
Flex Asset					exists

Approving Multiple Assets

The approval system provides a feature called **Approve Multiple Assets**. You use it to approve multiple assets in the following situations:

- You are upgrading from one version of CS to another and you have previously published assets that you want to be marked as approved.
- You add a new publishing destination and want to publish assets to it that you know are ready because they are approved for other destinations.
- The default approval template for an export destination is changed.

This feature is described in “Approving Multiple Assets,” on page 266.

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 Content Server database with the BulkLoader utility. Both BulkLoader and BulkApprover are described in the *Content Server Developer's Guide*.

The Publishing Schedule (Publish Events)

A publishing session (no matter what the delivery type) runs as a background, batch process called a **publish event**. As the administrator, you configure the schedule for the publishing events for your CS system.

You can schedule publishing events to occur daily, weekly, hourly, every 15 minutes, or in various combinations of these time increments.

Because each publishing session is a background event, there can be only one publishing session for the same destination at any given time. If a publishing session to a specific destination is still ongoing when the next event for that destination is scheduled, the second event attempts to run and then fails, reporting that a publishing session to that destination is already underway.

Note that you should 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 you publish from the management system to the delivery system. You should never run a publishing session from the development system to the management system while the management system is publishing to the delivery system.

Additionally, you can publish to more than one destination at the same time. You just set up events for the destinations and select the same time for them.

Be aware that the reverse is not true: that is, **you cannot have more than one source publish to the same destination**. If you do, there will be problems with data integrity as well as publishing errors.

Because a publishing event completes database transactions, the publishing feature must have a user account specified for it. This user is called the **batch user** and you use the `xcelerate.batchuser` and `xcelerate.batchpass` properties in the `futuretense_xcel.ini` file to identify the batch user account for your CS system.

Note

The batch user is not the same as the mirror user. For more information, see “Users and Mirror to Server,” on page 240.

What Happens During the Publishing Session?

When the publishing system begins publishing approved assets, CS-Direct does the following:

- Creates a publishing session by adding a row to the `PubSession` table and assigns the session a unique ID, called the `PubSession ID`.
- Runs a query to gather all the assets that are approved and are ready to be published to the destination that the session is publishing to.
- Locks the assets that are returned by the query, and also notifies the `CacheManager` about these assets, so that they cannot be edited while the publishing session is underway.
- Invokes the appropriate element for the delivery type (Export to Disk or Mirror to Server), 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 delivery type in use. For information about what the Export to Disk, Export Assets to XML, and Mirror to Server delivery types do with the list of approved assets, see the following sections:

- “The Export to Disk Delivery Type,” on page 228.
- “The Mirror to Server Delivery Type,” on page 238.
- “The Export Assets to XML Delivery Type,” on page 242.

When the publishing session concludes, the publishing system notifies the approval system of which assets were published and then it updates the page cache.

Note

The CS-Direct publishing system publishes assets that are held in the Content Server database.

If you are storing files directly on your web server or your site designers use links from assets to files that are not assets, you must implement an additional process to move those files to your web server and to update them when necessary.

Obtaining Information About a Publishing Session

CS-Direct writes information about each publishing session to log files and to several tables in the Content Server database.

Much of this information is displayed in the **Publishing Console** as the publishing history for a session. To display the **Publishing Console**, click the **Publish** button located in the top button bar in the main window. See also “Troubleshooting,” on page 278.

Note

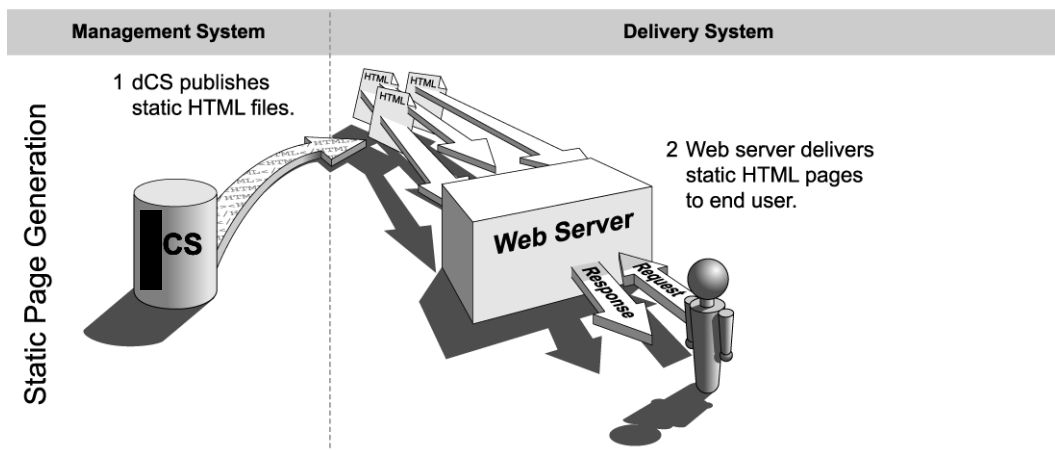
You must have the `xcelpublish` ACL to view the **Publish Console**.

The Export to Disk Delivery Type

In extremely simple terms, Content Server renders a page as follows:

1. A page name is submitted to Content Server (through HTTP or another protocol).
2. Content Server then locates the page in the SiteCatalog table, invokes the root element, and renders the page.

On a dynamic delivery system or when a user of the management system previews an asset, Content Server renders a page and posts it through HTTP to the visitor's or the user's browser. However, when the **Export** API is involved, Content Server **renders the page to a file**. You then move those files to the web server that is hosting your online site.

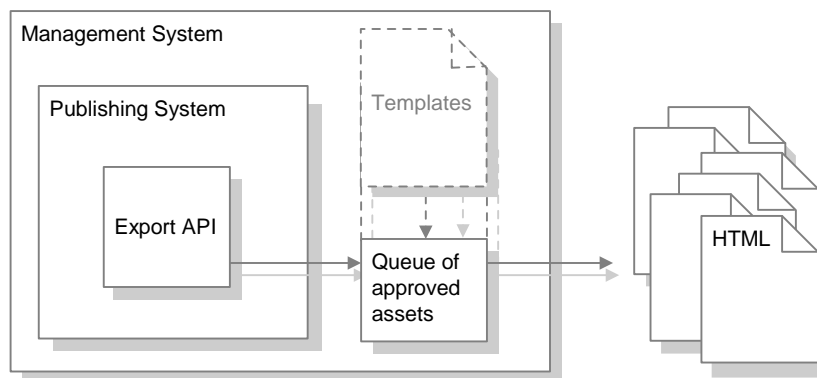


When you use the Export to Disk delivery type, typically you set up a quality assurance process to test the generated files before you move the files—either through FTP or another file transfer protocol—to the web server of your delivery system.

How Export to Disk Works

When you use the Export to Disk delivery type, the approval system, the publishing schedule, and your destination configurations all contribute to the process of exporting your approved assets into static files.

When an Export to Disk publishing session runs, this is what happens:



1. The publishing system notifies the CacheManager servlet that a session is about to begin for a specific destination. CacheManager clears all the pages that were previously exported to this destination from the page cache on the source system.

During an export publishing session, exported files are cached to the page cache. If publishing sessions occur more frequently than the cache is configured to be cleared (expiration time), there could be exported pages in the cache that have the same names as the newly exported pages. To ensure that old files are not used, the CacheManager servlet clears the page cache of any pages that were exported to during previous publishing sessions before a new publishing session starts.

2. CS-Direct creates an export queue and adds all references that can be published to it. A reference that can be published is one of the following:
 - An asset that has been designated as an **export starting point**. (See “Export Starting Points,” on page 237.)

There must be a starting point for the site and it must be approved or the export session cannot begin. Every site that is exported must have at least one export starting point designated for it.
 - Any previously exported page or pagelet (reference) whose assets have changed and been approved again since the last publishing session to this destination. (After a site has been published, the publishing system knows which references have been published. It reads the data in the `PublishedAssets` table and then adds any references whose assets have been changed and approved again for this destination to the export queue.)
 - Any asset whose template uses `RENDER.UNKNOWNDEPS` tag.

This tag alerts the approval system that the dependencies for the asset being rendered by the template cannot be calculated because they are unknown. It is typically used for queries. When the dependencies are unknown, the system cannot determine whether changed dependents would require the asset to be published. Therefore, the presence of this tag means that the asset must be published during every publishing session.
 - Any page or pagelet (reference) that has not yet been published and is connected to another page or pagelet through a `RENDER.GETPAGEURL` tag. (That is, it is referenced from another page and has not yet been published.)
 - Any page or pagelet (reference) identified by a `RENDER.SATELLITEPAGE` or `satellite.page` tag.
3. For the first export starting point, CS-Direct determines whether it has been approved:
 - If the asset that is the starting point has not been approved, the publishing session ends.
 - If the asset is approved, CS-Direct determines the page name of the template assigned to the asset.
4. CS-Direct renders the export starting point by passing the following information to Content Server:
 - 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.

- Values for publishing arguments that are explained later in this section (“Publishing Arguments for Export to Disk,” on page 230.)
5. Content Server invokes the root element of the page name, and begins rendering every approved asset that is connected to that export starting point as a reference. Content Server writes the resulting rendered pages to files rather than posting them to the browser.
 6. After each file is rendered, CS-Direct writes a message about that reference to the publish log for the session.
 7. If there is more than one export starting point, the process cycles back to step 3 in this description.
 8. When the publishing session is successful, CS-Direct concludes the session by writing information about the published references to the `PubKey` and `PublishedAssets` tables.
 9. CS-Direct 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.

Publishing Arguments for Export to Disk

When you configure publishing destinations that use the Export to Disk delivery type for your CS system, you must provide certain values that CS-Direct needs in order to export publish your assets correctly. You provide these values with **publishing arguments**.

Some export publishing arguments are used to name the files and some are used to generate the links (HREFs) within the files.

The following table lists the publishing arguments for Export to Disk:

Publishing Argument	Description
URLPREFIX	<p>Optional.</p> <p>A prefix that the Export process adds to the beginning of the URLs that are used as links (HREFs) within the exported files.</p> <p>Use this argument to 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 do not specify a value for this argument, URLs are relative.</p> <p>The names of the generated files do not include this prefix—only the links within the files use it.</p> <p>Because you should always test the HTML files before you move them to the delivery system, be sure that the name of the web alias on the testing system is the same as the name of the web alias on the delivery system.</p>

Publishing Argument	Description
DIR	<p>Optional.</p> <p>A subdirectory name for the files published to this export destination. It is created as a child directory of the base directory specified by the <code>cs.pgexportfolder</code> property.</p> <p>When you use this argument, the directory path for the publishing destination is as follows:</p> <p>value of <code>cs.pgexportfolder</code>/value of DIR</p> <p>Note that 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 use the DIR argument and you are using the URLPREFIX argument, too, be sure that the web root or the web alias represented by the URLPREFIX argument points to this directory.</p> <p>This argument is typically used to organize the contents of the base export directory when there are multiple export destinations for your sites.</p>
SUFFIX	<p>Optional.</p> <p>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.</p>
SIMPLENAME	<p>Optional.</p> <p>If set to <code>true</code>, this argument overrides the normal file-naming conventions. (For a description of the file-naming conventions, see “File and Directory Naming Conventions,” on page 232.)</p> <p>When set to <code>true</code>, this argument instructs CS-Direct to ignore the SiteCatalog page entry of the template that is used to render the asset and to use only the value entered in the asset’s Filename field for the name of the file. If the asset does not have a filename, CS-Direct uses the asset’s ID, instead.</p> <p>There is one exception to this rule: assets with upload fields—that is, an asset that is a blob— will always receive a standard (long) file name, even when this value is set to <code>true</code>.</p> <p>This argument must be used with caution, and, if your site design is such that individual assets are rendered by more than one template, you cannot use this argument. Without the SiteCatalog page entries used in the file name, file names cannot be guaranteed to be unique when an asset is rendered by more than one template.</p>
SIMPLEDIR	<p>Optional.</p> <p>If set to <code>true</code>, this argument overrides the normal directory-naming convention. (For a description of the file-naming conventions, see “File and Directory Naming Conventions,” on page 232.)</p> <p>When set to <code>true</code>, this argument instructs CS-Direct to put the rendered HTML file directly into the default export directory when there is no path information available from the parent asset. (Normally, CS-Direct uses the page asset’s ID when there is no path information.) Note that if there is a value for path, this argument is ignored.</p>

Publishing Argument	Description
OLDTEMPLATE	<p>Required in special cases.</p> <p>This argument instructs CS-Direct to use the rendering methodology that was standard in the 3.5 and earlier versions of the product.</p> <p>If the online site that you present on your delivery system was designed for the rendering model used in versions of the product before the 3.6 version and it has not yet been redesigned for the 3.6 rendering model, you must set this argument to <code>true</code>.</p> <p>Note that if you set this argument to <code>true</code>, you do not set an export starting point for your site. The export starting point is determined by the templates themselves.</p>
VERBOSE	<p>Optional.</p> <p>Activates error logging during the publishing process. When set to <code>true</code>, additional messages are written to the <code>PubMessage</code> table, rather than just errors.</p> <p>Be sure to use the <code>VERBOSE</code> publishing argument only for troubleshooting because it causes the publishing process to take longer than normal.</p>

File and Directory Naming Conventions

Exported files are named according to specific file-naming conventions and are saved to file directories that are named according to specific directory-naming conventions. This section describes those conventions and provides examples that you can use to determine how your files and directories will be named, based on the information that you provide for a destination.

CS-Direct needs several bits of information that it uses to name the files that Export to Disk generates from your approved assets and to determine where to store them. This section presents each of the variables that affect file and directory names and describes the file-naming conventions that CS-Direct follows.

Directory Names

This section describes the properties, publishing arguments, and fields that contribute to directory names and then presents the directory naming conventions and provides some examples.

Values Used in Directory Names

The following values are used to create the directory names for the exported files:

- `cs.pgexportfolder` property in the `futuretense.ini` file

This property sets the base directory for all exported files. It is a required property whose value applies to all publishing destinations that use the Export to Disk publishing method.

Files are exported either directly into this directory, or to subdirectories within this directory if you use the `DIR` argument (described, above, in “Publishing Arguments for Export to Disk,” on page 230).

The directory specified by this property, or by the combination of this property and the `DIR` argument is the **destination directory**.

If your system is running on either a Solaris or an AIX operating system, you can use symbolic links to map a local subdirectory to a remotely mounted file system.

Typically, this value is set to a testing location (file system) where you or others test the exported files rather than to a location directly on the delivery system.

- **DIR** publishing argument (see “Publishing Arguments for Export to Disk,” on page 230)
- **SIMPLEDIR** publishing argument (see “Publishing Arguments for Export to Disk,” on page 230)
- **Path** information

In general, the directory name for a file is based on the value set in the **Path** field of the asset’s parent page asset. If there is no value set for path, CS-Direct uses the ID of the parent page asset instead.

However, note the following about path information:

- The path information entered for a page asset is used for its children assets only. When a page asset is exported, it receives the path set for its parent page, not the one set for that page asset.
- For assets that are not page assets, if path information is entered in the **Path** field for that asset, the path is used instead of the path information provided by the parent page asset.
- Path information set for an asset for a specific destination on the “Specify Path/Filename *for destination*” form is used in place of any path information provided in the Path field on the asset’s “New” or “Edit” form or provided from the parent page asset when the asset is exported to that destination.

Directory Naming Conventions

Directory names for both page assets and other assets are created by concatenating the values from the items in the preceding list in the following order:

`value_of_cs.pgexportfolder/value_of_DIR/parent page’s path`

If an asset other than a page asset has path information of its own, CS-Direct uses that path rather than the path provided by the parent page asset. In such a case, the directory is created like this:

`value_of_cs.pgexportfolder/value_of_DIR/asset’s path`

If **no** path information is available from the parent page or the asset itself, the directory is created like this:

`value_of_cs.pgexportfolder/value_of_DIR/ID of parent page`

If **no** path information is available from the parent page or the asset, and **SIMPLEDIR** is set to **true**, the directory is created like this:

`value_of_cs.pgexportfolder/value_of_DIR`

The following table provides examples of how directory names are created for **page assets** during Export to Disk:

Page Asset	Value of cs.pgexport folder	Value of DIR	ID of Parent Page	Path of Parent Page	Directory for Exported File
Home	/export				/export
Home	/export	/Japan			/export/Japan
World News	/export		998877665		/export/Japan/998877665
World News	/export		998877665	/news/	/export/news
World News	/export	/Japan	998877665	/news/	/export/Japan/news

The following table provides examples of how directory names are created for **assets other than page assets** during Export to Disk:

Asset	Value of cs.pgexport folder	Value of DIR	Path from Asset	ID of Parent Page	Path of Parent Page	Directory for Exported File
OilPrice	/export			997766554		/export/997766554
OilPrice	/export	/Japan		997766554		/export/Japan/ 997766554
OilPrice	/export	/Japan		997766554	/news/	/export/Japan/news
OilPrice	/export		/energy	997766554	/news/	/export/energy
OilPrice	/export	/Japan	/energy	997766554		/export/Japan/energy
OilPrice	/export	/Japan	/energy	997766554	/news/	/export/Japan/energy

Paths for Links Within the Files

The 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 `URLPREFIX` argument is used to resolve URLs by specifying the location of those files on the delivery system. That is, URLs for links within the exported files are created like this:

`URLPREFIX/path_of_parent_page`

This means that the value for the `URLPREFIX` argument 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 `URLPREFIX` represents must point to that location. And if you are using the `DIR` argument to add a subdirectory to the destination directory specified by `cs.pgexportfolder`, be sure that the web root or the web alias represented by the `URLPREFIX` argument points to this directory.

File Names

This section describes the arguments and other items that contribute to file names and then presents the file naming conventions and provides some examples.

Values Used in File Names

The following values are used to create file names for the exported files:

- The `SiteCatalog` page entry of template used to render the asset
All template assets have `SiteCatalog` page entries. The name of the `SiteCatalog` page entry of the template that is used to render the asset is included in the name of the file generated for the asset (unless you use the `SIMPLENAME` publishing argument).
- Packed arguments passed in from root element of page entry
If any packed arguments are passed in from the root element of the template that is used to render the asset, the values of those arguments are also included in the name of the file generated for the asset.
For information about packed arguments and how they relate to URLs, see the *Content Server Developer's Guide*.
- **Filename** field
If there is a value set in an asset's **Filename** field, the name of the file uses the value from that field in the name of the exported file. If there is no value in the **Filename** field, CS-Direct uses the ID of the asset in the asset.
Note that 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 it is exported to that destination.
- `SUFFIX` publishing argument (see "Publishing Arguments for Export to Disk," on page 230)
- `SIMPLENAME` publishing argument (see "Publishing Arguments for Export to Disk," on page 230)

File Naming Conventions

File names for all assets are created by concatenating the values from the items in the preceding list in the following order:

```
pagename_packedargs(if any)_filename.SUFFIX
```

If there is **no filename** information provided, the file name is created like this:

```
pagename_packedargs(if any)_assetID.SUFFIX
```

If `SIMPLENAME` is set to `true`, the file name is created like this:

```
packedargs(if any)_filename.SUFFIX
```

If `SIMPLENAME` is set to `true` but there is no file name information provided, the file name is created like this:

```
packedargs(if any)_assetID.SUFFIX
```

Note

Be sure to use `SIMPLENAME` carefully. If your assets are rendered by more than one template, do not use `SIMPLENAME`.

Note the following syntax changes in the file names:

- Slash characters (/) in pagenames are converted to hyphens (-) in file names.
- Equals signs (=) in packed arguments are converted to hyphens (-) in file names.
- Ampersand characters (&) in packed arguments are converted to underscores (_) in file names.

The following table provides examples of how file names are created during Export to Disk:

Asset Name	Asset ID	File Name	Value of SUFFIX	SiteCatalog Pagename	Arguments Passed with Pagename	File Name of Exported File
Home	123			BF/Page/Home	cid=123 c=Page	BF-Page-Home_123.html
Home	123		.htm	BF/Page/Home	cid=123 c=Page	BF-Page-Home_123.htm
Home	123	home.html		BF/Page/Home	cid=123 c=Page	BF-Page-Home_home.html
Home	123	home.html	.htm	BF/Page/Home	cid=123 c=Page	BF-Page-Home_home.html
Home	123	home.htm		BF/Page/Home	cid=123 c=Page PACKEDARGS="topicword=oil"	BF-Page-Home_topicword_oil_home.htm

Note

CS-Direct needs both the asset's ID and the asset's type to determine whether there is a file name provided for it. If the identity of the asset's type is not provided, the file name cannot be used. In such a case, CS-Direct uses the asset's object ID in place of its file name.

Export Starting Points

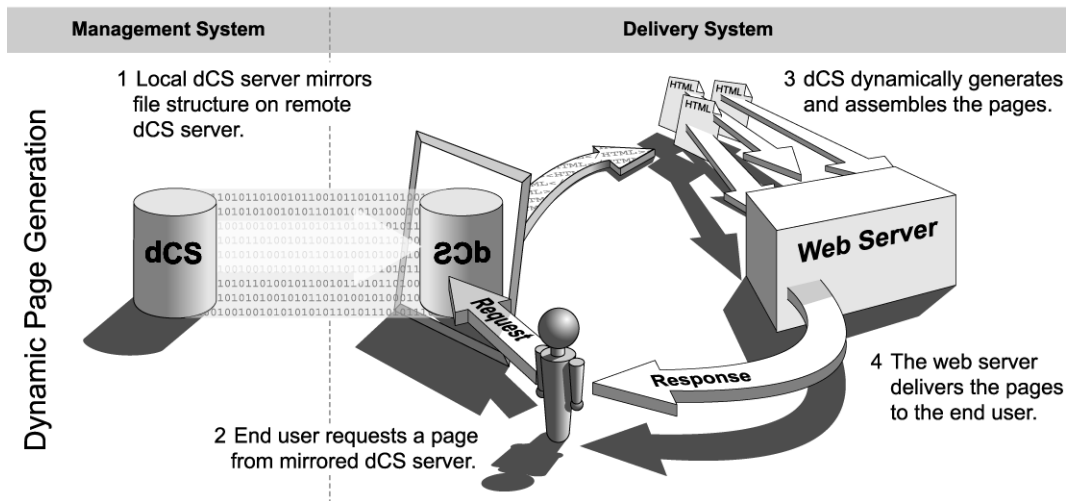
Export to Disk cannot begin rendering files unless it knows where to start. You tell it where to start by designating at least one page asset as an **export starting point** and specifying the template that should be used to render it. Content Server invokes the root element of the template's page name, and begins rendering every approved asset that is connected to that export starting point—assets connected through an association, a hyperlink, a navigation bar, a query, and so on.

There must be at least one export starting point for an exported site. Typically it is your home page. However, depending on how your online site is designed, you may need to designate more than one export starting point. Why? Because if there is a section on your site that isn't connected to the rest of the site, it won't be rendered. For example:

- If your online site has more than one top-level page asset, you need an export starting point for each one.
- If your site uses a combination of static pages and dynamic pages and there is a hard-coded static URL in an HREF on a dynamically-generated page, the asset that the URL points to should be designated as an export starting point.

The Mirror to Server Delivery Type

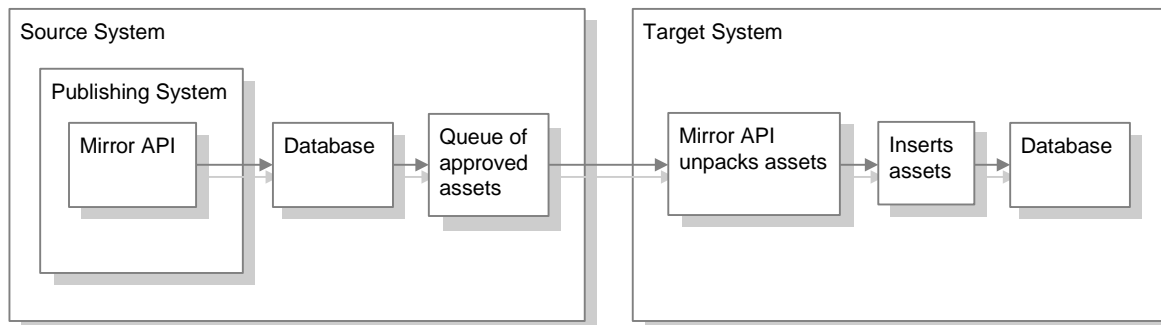
The CS-Direct Mirror to Server delivery type gathers information from the approval system, the publishing schedule, and the destination configurations, copies data to the destination, unpacks that data on the destination system, and then invokes the CacheManager servlet to refresh any pages that should be regenerated to take advantage of the new content.



Whether or not your CS delivery system is dynamic, you probably use the Mirror to Server delivery type to move data from your development systems to your management system. For information about that process, see “Troubleshooting,” on page 278.

How Mirror to Server Works

When a Mirror to Server publishing session runs, this is what occurs:



1. On the source system, Mirror to Server 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. CS-Direct 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 de-serializes the objects in the `_Publish` tables, and inserts the results in the appropriate tables.

For basic assets, each row in the queue is copied.

5. When the items in the mirror queue are unpacked, the publishing system sends messages that the mirror publish concluded successfully. The destination system responds as follows:
 - The newly-published assets are marked as changed on the destination system, which means that before they can be published from that system to another destination, they must be approved. Note that this feature is enabled by default. You can turn it off if you need to. See “Step 11 (Mirror to Server): Turn Off Asset Invalidation on the Delivery System,” on page 260 for details.
 - The `CacheManager` servlet on the destination regenerates the appropriate pages in the cache so that all pages that refer to the assets that were just published are updated. It also rebuilds any pages that have unknown compositional dependencies.
 - `CacheManager` then communicates a message about which pages must be refreshed to each Satellite Server identified by the `satellite.ini` file on the destination system. It communicates with the co-resident version and any remote instances that are identified as belonging to this CS 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 delivery type, which writes to the publish log after each asset is exported, Mirror to Server 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, CS-Direct 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. CS-Direct 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.

Users and Mirror to Server

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 delivery type 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 destination, you must create the mirror user account on the destination system that the destination represents and you must specify the identity of that user to the source system with the `cs.mirroruser` and `cs.mirrorpassword` properties in the `futuretense.ini` file.

Publishing Arguments for Mirror to Server

There are three publishing arguments for Mirror to Server.

Because properties in the `futuretense.ini` file are global for all publishing destinations, there are two publishing arguments for Mirror to Server that enable you to specify destination-specific mirror users, in case you have more than one delivery system that you are publishing to from the same source and you do not want to name the mirror user account identically on each destination system.

Those publishing arguments are as follows:

- `REMOTEUSER`, which specifies the mirror user account on the destination system when it is different than the one specified by the `cs.mirroruser` property in the `futuretense.ini` file on the source.
- `REMOTEPASS`, which specifies the password for the user account designated by the `REMOTEUSER` argument. Note that value of this argument is not stored in an encrypted form, which means that anyone with access to the database on the source can see the password.

The final Mirror to Server publishing argument is one that it shares with Export to Disk:

- `VERBOSE`, which activates error logging during the publishing process. When set to true, additional messages are written to the `PubMessage` table, rather than just error messages. Be sure to use the `VERBOSE` publishing argument only for troubleshooting because it causes the publishing process to take longer than normal.

CacheManager

The CacheManager is a Content Server servlet that maintains the page cache on any dynamic CS system, including the management system.

CacheManager is important to the publishing system because it locks the appropriate assets on the destination during a mirror publish and ensures the integrity of the page cache both before and after a mirror publishing session.

For more information about the CacheManager, see the *Content Server Developer's Guide*.

Configuring a Mirror Destination

The Mirror to Server delivery type copies information for approved assets from one CS 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 (e.g., assets types, associations, start menu items, etc.) will be mirrored to the target destination. Mirror destination configuration moves configuration data and rows from auxiliary tables—which are non-asset database tables that are used for the dynamic display of the assets—from one CS database to another.

When to Use Mirror Destination Configuration

You use the Mirror Destination Configuration feature in several situations:

- To set up any new mirror destination.
- To move configuration items that support asset types (start menu items, associations, and so on) from a development system to a management system or to a delivery system.
- To move workflow configuration data from a development system to a management system.
- If you or your developers add any additional sites, asset types, or auxiliary tables to your system.
- If you or your developers add any additional categories or subtypes for existing asset types.
- When you need to troubleshoot your configuration. If you can successfully initialize a mirror destination, that means that the source and the destination systems are communicating.

The Export Assets to XML Delivery Type

The CS-Direct **Export Assets to XML** delivery type is a hybrid between the Export to Disk and Mirror to Server delivery types. Assets are rendered into files, but this delivery type uses the Mirror to Server dependency calculation method rather than that of Export to Disk.

This delivery type differs from the other two in that it is really a data transformation method. While Export to Disk 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 delivery type.

Export to Disk creates static files that are to be delivered from a static CS delivery system. In contrast, Export Assets to XML creates **XML files** that are to be delivered to another **non-CS** system or database.

When an Export Assets to XML publishing session runs, this is what happens:

1. On the source system, the approval system provides a list of the assets approved for this destination to Export Assets to XML.
2. Export Assets to XML renders each asset in the list to an XML file. The output file describes the asset, stating all the values for all of the asset's fields.

When you use the Export Assets to XML delivery type, typically you set up a quality assurance process to test the generated files before you move them to the external, non-CS system.

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 GE Lighting sample site product asset:

```
<attribute name="Attribute_price">
    <decimal value="5.9"/>
</attribute>
```

Note that the XML output uses the column names rather than the field name of each field/attribute and that column names and field names can be different.

The following is a description of the XML file created by this publishing method, presented as a pseudo-dtd file

```
<!-- ASSET:
-- ASSET defines an asset object
-- an asset is made up of attributes
-->
```

```

<!ELEMENT ASSET (ATTRIBUTE)*>
<!ATTLIST ASSET TYPE CDATA #REQUIRED>
<!ATTLIST ASSET ID CDATA #IMPLIED>
<!ATTLIST ASSET SUBTYPE CDATA #IMPLIED>

<!ELEMENT ATTRIBUTE (STRING | DATE | INTEGER | DECIMAL | BINARY |
FILE | ASSETREFERENCE | ARRAY | STRUCT | LIST)>
<!ATTLIST ATTRIBUTE NAME CDATA #REQUIRED>

<!ELEMENT STRING>
<!ATTLIST STRING VALUE CDATA #REQUIRED>

<!ELEMENT DATE>
<!ATTLIST DATE VALUE CDATA #REQUIRED>

<!ELEMENT INTEGER>
<!ATTLIST INTEGER VALUE CDATA #REQUIRED>

<!ELEMENT DECIMAL>
<!ATTLIST DECIMAL VALUE CDATA #REQUIRED>

<!ELEMENT BINARY>
<!ATTLIST BINARY VALUE CDATA #REQUIRED>

<!ELEMENT FILE (CDATA) >
<!ATTLIST FILE NAME CDATA #REQUIRED>

<!ELEMENT FILE>
<!ATTLIST FILE NAME CDATA #REQUIRED>

<!ELEMENT ASSETREFERENCE>
<!ATTLIST ASSETREFERENCE TYPE CDATA #REQUIRED>
<!ATTLIST ASSETREFERENCE VALUE CDATA #REQUIRED>

<!ELEMENT ARRAY (STRING | DATE | INTEGER | DECIMAL | BINARY | FILE
| ASSETREFERENCE | ARRAY | STRUCT | LIST)+>

<!ELEMENT STRUCT (FIELD)+>

<!ELEMENT FIELD (STRING | DATE | INTEGER | DECIMAL | BINARY | FILE
| ASSETREFERENCE | ARRAY | STRUCT | LIST)+>
<!ATTLIST FIELD NAME CDATA #REQUIRED>

<!ELEMENT LIST (ROW)+>

<!ELEMENT ROW (COLUMN)+>

<!ELEMENT COLUMN ( STRING | INTEGER | DECIMAL | DATE |
ASSETREFERENCE)>
<!ATTLIST COLUMN NAME CDATA #REQUIRED>

```

Publishing Arguments for Export Assets to XML

There is one publishing argument for Export Assets to XML: `DIR`. It provides a subdirectory name for the files published to the destination. It is created as a child directory of the base directory specified by the `cs.pgexportfolder` property.

When you use this argument, the directory path for the publishing destination is as follows:

value of `cs.pgexportfolder`/value of **DIR**

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 delivery type does not use any information that is entered in an asset's **Path** or **Filename** field.

Configuring the Publishing Process for Your CS System

Configuring your system for publishing includes more than just creating destinations and the steps that you complete are different based on which delivery type you are using, Export to Disk or Mirror to Server.

The procedures in this section mention several properties in the `futuretense.ini`, `batch.ini`, and `futuretense_xcel.ini` files. In addition to the properties mentioned in this chapter, there are additional properties that you can use to fine-tune your system configuration. For a complete list of all properties, see the *Property Files Reference*.

Create the Batch User Account

There are certain steps that you must complete regardless of your delivery type. You must:

- Configure a **batch user** account for the publishing system on the source to use.
- Specify where the publish **logs** should be stored.

To create the batch user account

1. Log in to the Content Server interface on your source system.
2. On the **Admin** tab, select **Content Server Management Tools > Users**.
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
 - Visitor (if your installation includes Engage)
 - VisitorAdmin (if your installation includes Engage)
 - xceleeditor
 - xceladmin

If you need help with this step, see “Creating a New User,” on page 77.

4. Start the Property Editor and open the `futuretense_xcel.ini` file for the source system. (If you need help with this step, see the *Property Files Reference*.)
5. On the **Publishing** tab, set values for the following properties:
 - `xcelerate.batchhost`
Set this property to the host name of **web server** (not the application server or the load balancer) 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`
 - `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.
- 6. Save the settings by selecting **File > Save**.
- 7. Open the `batch.ini` file for the source system.
- 8. On the **Results** tab, set the `request.folder` property to point to the directory that you want to hold your publish log files. By default, this property is set to the `dispatcher` subdirectory in the application server installation directory.

Note

Be sure to set this property to a directory that can be written to. Otherwise, there will be no Publish History summary information displayed in the Publish Console for your publishing sessions.

9. Save the settings (**File > Save**) and then exit the Property Editor (**File > Exit**).
10. Stop and restart the application server.
11. Continue with the configuration procedure for the delivery type you are planning to use:
 - For Export to Disk, go to “Configuring Your System for Export to Disk Publishing,” on page 246 the next procedure in this section.
 - For Mirror to Server, go to “Configuring Your System for Mirror to Server Publishing,” on page 252.
 - For Export Assets to XML, go to “Configuring Your System for Export Assets to XML,” on page 261.

Configuring Your System for Export to Disk Publishing

The main steps when configuring a system to publish using the Export to Disk delivery type are these:

- Configure the export directory that stores the files.
- Create a publishing destination that points to the export directory.
- Map a URL prefix on the web server of the delivery system.
- Create an export starting point.
- Approve some test assets, including the export starting point.
- Publish the test assets and verify them.
- Set up the schedule.

Step 1 (Export to Disk): Specify the Base Export Directory

As mentioned previously in this chapter, the directory that CS-Direct exports files to is determined by the `cs.pgexportfolder` property in the `futuretense.ini` file. If you want to add subdirectories to this directory, you use the `DIR` publishing argument in the definition of the destination.

To set the base directory for the exported files

1. Start the Property Editor. If you need help with this step, see the *Property Files Reference*.

2. Open the `futuretense.ini` file for the source CS system.

3. On the **Export/Mirror** tab select the `cs.pgexportfolder` property and set the value to the file directory (location) where all files should be exported to.

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, use the `DIR` publishing argument when you configure those destinations.

For more information about export directories, see “File and Directory Naming Conventions,” on page 232.

4. Save the new value by selecting **File > Save**.
5. Select **File > Exit** to close the Property Editor.
6. Stop and restart the application server.

Note

Before you run an export publish, make sure that this destination directory exists and that it has enough space for the HTML pages that will be created there.

Step 2 (Export to Disk): Configure an Export Destination**To create the export destination**

1. Log in to the Content Server interface on the source CS system.
2. On the **Admin** tab, select **Publishing > Destinations**.
3. Select **Add New**.

The “Add New Destination” form appears.

Add New Destination

*Name:	<input type="text"/>
Delivery Type:	Export to Disk: Export Web files to disk ▾
Destination address:	<input type="text"/>
Arguments:	<input type="text"/>
*Sites:	<div> Burlington Financial GE Lighting Hello Asset World </div>

Cancel

Add New Destination

4. In the **Name** field, enter a unique name for the destination.
5. In the **Delivery Type** field, select **Export to Disk: Export Web Files to Disk**.
6. Leave the **Destination Address** empty. This field is for Mirror to Server only.
7. In the **Arguments** field, specify any arguments (URLPREFIX, DIR, SUFFIX, SIMPLEDIR, SIMPLENAME, OLDTEMPLATE or VERBOSE) that are appropriate for this destination, separated with the ampersand (&) character.

For descriptions of the publishing arguments for this delivery type, see “Publishing Arguments for Export to Disk,” on page 230.

Here are some examples:

```
URLPREFIX=/siteName&SUFFIX=htm
```

```
URLPREFIX=/siteName&DIR=/Japan&SUFFIX=htm
```

```
URLPREFIX=/siteName&DIR=/Japan&SUFFIX=htm&OLDTEMPLATE=true
```

8. In the **Sites** box, select the sites whose assets can be approved for and published to this destination.
9. Click **Add New Destination**.

The information about this destination is written to the PubTarget table.

10. Repeat steps 2 through 9 for each additional export destination that you need to configure for your source system.

Step 3 (Export to Disk): Map a URL Prefix for Your Web Server

To make your exported content available to the public at its final destination (your delivery system), you must configure your web server to map a URL path prefix to the place where the files will reside so that URLs can be resolved.

Refer to the vendor documentation for your web server for information about the appropriate procedure for mapping URL prefixes on your web server. Remember that the name you specify for this prefix must match the name that you specified with the

URLPREFIX argument for the destination. If you use the DIR argument, be sure that the web root or alias includes the DIR directory.

Step 4 (Export to Disk): Create the Export Starting Points

To create an export starting point

1. Using the Content Server interface on the source system, find the asset that you want to specify as an export starting point.
2. Open the asset in its “Status” form and scroll to the **Publish Destination** section.
3. Do one of the following:
 - If the asset is not yet approved for the destination you are currently configuring, select **Approve for Publish** from the drop-down list in the action bar and go to step 4.
 - If this asset is already approved for the destination you are currently configuring, go to step 6.
4. In the publish approval form, select the destination that you are approving it for and click **Approve**.

The approval system calculates dependencies for all the assets linked to this asset and displays the results.
5. When CS-Direct displays the approval results, select **Status** from the drop-down list in the action bar.
6. Scroll down to the “Publishing Destination” section of the form; next to Path/Filename, click the **Specify Path/Filename, Start points** link.
7. Configure the export starting point:
 - a. Select the **Yes** option.
 - b. Select a template.

Note

If the publishing argument OLDTEMPLATE is set to true for this destination, no templates will be displayed for you to choose from. With the old, pre-version 3.6 style of rendering, the template is determined by the asset itself and you cannot override the template assigned to the asset in an export starting point.

- c. (Optional) If you want to specify path information that is different for this destination than any path information provided elsewhere for this asset, enter the path in the Path field. For more information about path names, see “File and Directory Naming Conventions,” on page 232.
- d. (Optional) If you want to specify file name information that is different for this destination, enter it in the **Filename** field. A file name entered in this form for this destination overrides any file name information provided elsewhere for the asset.

- e. (Optional) If you want the directory name for the path created according to the simple naming convention described in the definition of the `SIMPLEDIR` publishing argument, select **Force specified path**. Selecting this check box is like setting the `SIMPLEDIR` publishing argument to `true`, but only for this asset rather than for all assets exported to this destination.
 - f. (Optional) If you want the file name created according to the simple naming convention described in the definition of the `SIMPLENAME` publishing argument, select **Force specified filename**. Selecting this check box is like setting the `SIMPLENAME` publishing argument to `true`, but only for this asset rather than for all assets exported to this destination.
8. Click **Save**.
 9. Repeat steps 1 through 8 for each top-level page asset in the site.

Step 5 (Export to Disk): Approve Your Assets

If you want to perform a simple test of your configuration, select the export starting point with the fewest number of dependent assets and approve each of those assets.

If you want to complete a test publish of your entire site, approve all the assets for the site. See “Approving Multiple Assets,” on page 266 for help with approving many assets at once.

Step 6 (Export to Disk): Publish and Test the Results

After you have approved assets that can be published to your destination, you can run a test publishing session:

1. In the Content Server interface on the source system, checkboxes will configure the selected asset types.
2. In the **Publish Console**, select your destination from the drop-down list and then click **Select Destination**.

CS-Direct displays information about the assets that are ready to be published to this destination. (If you have not yet created an export starting point for this destination, the form states this fact. You must create at least one export starting point before Export to Disk can begin.)

3. Click **Publish**.

The publishing system exports all the approved assets for this destination into files.

4. Click the link to the **Publish Console**.
5. In the **Publish Console**, scroll down to Publishing History and click the inspect icon next to the publishing session.

A session summary displays a list of all the files that were created during the export process.

6. Click the link to file that represents the top-level page in your site.

The system displays the file in a browser window.

7. Scroll down that page, looking for errors, and click all the links in the file to verify that they work. Test all the links to all the other files that were generated.

8. In addition, test the results by directly entering the URL of the rendered home page in your browser and navigating the entire site. Note that to do this you must first set up a web server root on the CS management system that points to the export directory.
9. When you are satisfied that the files are ready, use FTP or another file transfer protocol to move them to your CS delivery system.

Step 7 (Export to Disk): Set Up the Schedule

After you have ensured that your destination is configured correctly, that the file names and directory names are created according to your needs, and that your web server is delivering the files correctly, you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see “Scheduling Publish Events,” on page 269.
- Plan how you will move the generated files to your web servers, for both the testing area and the delivery system. For example, you can set up a regular FTP transfer that automatically moves files to a testing area after a publishing session completes.

Configuring Your System for Mirror to Server Publishing

The main steps when configuring a system to publish using the Mirror to Server delivery type are as follows:

- Identify the mirror user on the destination to the publishing system on the source.
- If there is a firewall separating the source and destination systems, identify the proxy server.
- On the source system, create a publishing destination that points to the destination system.
- Initialize the destination.
- Configure the destination.
- Approve some test assets.
- Publish the test assets and then verify the results.
- Set up the schedule.

Step 1 (Mirror to Server): Set Up the Destination System

To mirror publish assets from one CS system to another, the sites and asset types must be the same on both the source and the destination system.

When the delivery type is Mirror to Server, 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

The database properties on the CS system and the destination CS system, if not an exact match, must be compatible. Otherwise, your assets cannot be mirrored successfully.

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. This user must have the following ACLs:
 - Browser
 - ElementEditor
 - PageEditor
 - TableEditor
 - Visitor (if Engage is installed)
 - VisitorAdmin (if Engage is installed)
 - xceladmin
 - xceleeditor

Additionally, because the CacheManager uses the mirror user account 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, if ACLs are assigned to any page entries in the SiteCatalog table or database tables that are holding data that needs to be rendered, the mirror user account must be assigned those ACLs, too.

If the destination system has any of the sample sites installed, a user named “mirroruser” already exists. For security reasons, if you decide to use this user as your mirror user, be sure to **change the password** for this user. If you decide to create a different mirror user, be sure to delete the sample site mirroruser.

Note

If you plan to mirror publish from any source system to more than one destination system, use either of the following options for the username and password of the mirror users that you create on each destination:

- Make all username/password match exactly, because you can specify only one mirror user in the `futuretense.ini` file on your source system.
- Create a mirror user on the second destination with a different name from the one specified in `futuretense.ini` and then provide that information with the REMOTEUSER and REMOTEPASS publishing arguments.

If you need help with this step, see “Creating a New User,” on page 77.

Step 2 (Mirror to Server): Identify the Mirror User to the Source System

Next, you identify the name and password of the mirror user on the destination system to the source system by setting property values in the `futuretense.ini` file on the source system.

To identify the mirror user to the source system

1. Start the Property Editor and open the `futuretense.ini` file for the source CS system. If you need help with this step, see the *Property Files Reference*.
2. On the **Export/Mirror** tab specify values for the following properties:
 - `cs.mirroruser`
Set this property to the name of the user that you created on the destination system in the preceding procedure.
 - `cs.mirrorpassword`
Set this property to the password of the user that you created on the destination system in the preceding procedure.
3. Select **File > Save** to save the settings.
4. Do one of the following:
 - If there is a firewall separating your source system from the destination system, go to “Step 3 (Mirror to Server): Identify the Proxy Server (If There Is a Firewall),” on page 254.

- If there is not a firewall separating your source and destination systems, select **File > Exit**. Stop and restart the application server. Then go to “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.

Step 3 (Mirror to Server): Identify the Proxy Server (If There Is a Firewall)

If you have a firewall between your source and your destination, you must identify the proxy server to the source system. With the `futuretense.ini` file still open in the Property Editor, complete the following steps:

To identify the proxy server

1. On the **Export/Mirror** tab, specify values for these properties:
 - `cs.mirrorproxyserver`
Set this property to either the name or the IP address of the firewall’s proxy server.
 - `cs.mirrorproxyserverport`
Set this property to the port number of the firewall’s proxy server.
2. Select **File > Save** to save the values.
3. Select **File > Exit** to close the Property Editor.
4. Stop and restart the application server.

Step 4 (Mirror to Server): Create a Mirror Destination

To create a mirror destination

1. Log in to the Content Server interface on the source CS system.
2. On the **Admin** tab, select **Publishing > Destinations**.
3. Click **Add New Destination**.

The “Add New Destination” form appears.

Add New Destination

*Name:	<input type="text"/>
Delivery Type:	Mirror to Server: Copy database rows to remote dynamic server ▾
Destination address:	<input type="text" value="http://[targetserver]/servlet/"/>
Arguments:	<input type="text"/>
*Sites:	<div> Burlington Financial GE Lighting Hello Asset World </div>

Cancel

Add New Destination

4. In the **Name** field, enter a unique name for the destination.

5. In the **Delivery Type** field, select **Mirror to Server: Copy database rows to remote dynamic server**.
6. In the **Destination Address** field enter the URL containing the web and application server information for the destination.

Example: WebLogic, WebSphere, and Sun ONE Application Server

`http://your_Webserver_name/servlet/`

Note

A slash is required at the end of the URLs because these URLs are appended dynamically.

7. In the **Arguments** field, specify any appropriate arguments for this destination, separated by the ampersand (&) character. For more information, see “Publishing Arguments for Mirror to Server,” on page 240.
8. In the **Sites** box, select the sites whose assets can be approved for and published to this destination.
9. Click **Add New Destination**.

The information about this destination is written to the `Pubdestination` table.

10. 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 Step 5 (Mirror to Server): Initialize the Destination” for detailed instructions (begin at step 4 of that procedure).

If you need to create more mirror destinations, repeat steps 2 through 9 for each additional mirror destination that you need to create for your source system.

Step 5 (Mirror to Server): Initialize the Destination

You must initialize the destination system before you can publish to it. This creates the basic site information on the destination system. Specifically, the `Publication` and `PublicationTree` tables are updated with the site names and asset types published to the target.

To initialize the destination system

1. In the Content Server interface on the source system, select **Admin > Publishing > Destinations**.
2. Double-click on the destination that you want to initialize.
3. In the “Publish Destination” form, select the **Initialize Mirror Destination** link.

The **Initialize Mirror Destination** screen appears.

Initialize Mirror Destination: Destination 2 (dynamic)

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 CS-DocLink, flex filter assets, the ImageFile asset type, or if you are using this table for any of your custom asset types.
 - Any other auxiliary tables (such as lookup tables) for your asset types.

There are only five fields for tables on this form. If you have to enter more than five table names, complete this procedure for the first five tables, then repeat steps 3, 4, and 5 for the remaining tables.

6. Click **Mirror**.

If the initialization is successful, a verification message appears on the screen. If not, an error message appears.

Based on the sites that you selected in step 4, appropriate 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 in the **Publish Console**. Click the **Publishing** button from the main toolbar. In the **Publish Console**, select your mirror destination from the drop-down list and then click **Select Destination**. Click the **Create Site on Target** button. This creates the basic site information on the destination system.

Step 6 (Mirror to Server): Configure the Mirror Destination

Now you need to configure the data that will be on your target destination. This configuration will vary depending on the purpose of the target destination. For example, if the target destination is strictly a delivery machine, it makes sense to only include asset types when mirroring. Before proceeding with configuring the destination, you may find it helpful to refer to “Migrating a Site from One System to Another System,” on page 264.

1. In the Content Server interface on the source system, in the **Admin** tab, expand the **Sites** node. Then, double-click one of the sites whose data you want to publish on the target destination.
2. In **Publish Destinations** (near the bottom of the screen), click **Mirror site configuration for xxx**, where **xxx** is the target destination.

The **Mirror site configuration** screen is displayed.

Mirror site configuration for: » [BurlingtonFinancial](#)

Publish destination:	Destination 2 (dynamic)
Asset Types:	enabled in BurlingtonFinancial <div> <div>Content Attribute</div> <div>Content Definition</div> <div>Content Parent</div> <div>Content Parent Definition</div> <div>Drill Hierarchy</div> </div> <div> <input checked="" type="checkbox"/> Asset Subtypes for selected asset types <input checked="" type="checkbox"/> Asset Associations for selected asset types <input checked="" type="checkbox"/> Categories for selected asset types <input checked="" type="checkbox"/> CS-Desktop configuration for selected asset types <input checked="" type="checkbox"/> CS-DocLink configuration for selected asset types </div>
Start Menu Items:	available in BurlingtonFinancial <div> <div>Find Link</div> <div>Find Page</div> <div>Find PDF</div> <div>Find Product</div> <div>Find Product Attribute</div> </div>
Workflow Processes:	available in BurlingtonFinancial <div> <div>-none-</div> <div>BF: Fund Parent Process</div> <div>BF: Normal Article Process</div> <div>BF: Segment Process</div> </div> <div> <input checked="" type="checkbox"/> Workflow States used by selected workflow processes <input checked="" type="checkbox"/> All Workflow Actions <input checked="" type="checkbox"/> All Workflow Conditions </div>
Workflow Groups:	available in BurlingtonFinancial none
Tree Tabs:	available in BurlingtonFinancial <div> <div>Active List</div> <div>Query</div> <div>Design</div> <div>Marketing</div> <div>Content</div> </div>
Saved Searches:	available in BurlingtonFinancial <div> <div>-none-</div> </div>
Roles:	available in all sites <div> <div>Checker</div> <div>Designer</div> <div>Editor</div> <div>Expert</div> <div>GeneralAdmin</div> </div>

Cancel

Mirror

3. Configure the data that will be on your target destination:

- **Asset Types:** Select the asset types that you want to make available on the target destination.
If your asset types have subtypes, categories, or associations, select the appropriate options from the list in the **Asset Types** section. Select the CS-Desktop / CS-DocLink configuration checkboxes to configure the selected asset types for CS-Desktop and CS-DocLink.
- **Start Menu Items:** Select any start menu items that you or the developers designed for your content providers to use.
- **Workflow Processes:** If there are workflow processes for your asset types, select these processes and the appropriate workflow items.
- **Workflow Groups:** Select any workflow groups that you or the developers designed for your content providers to use.
- **Tree Tabs:** Select the tree tabs that you want to make available on the target destination.
- **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.

4. Click the **Mirror** button.

Based on which asset type configuration options you selected in step 3, appropriate rows from the `AssocNamed`, `AssocNamed_Subtypes`, and `Category` tables are copied to the destination.

If you selected **Start Menu Items** or **Saved Searches**, appropriate rows from the tables that implement those features are copied to the destination.

Based on which workflow configuration options you selected in step 3, appropriate rows from the workflow tables are copied to the destination.

5. Repeat steps 1 through 4 for each site whose assets you want to publish on the target destination.

Step 7 (Mirror to Server): Approve Your Assets

To truly test your published site, you must approve and publish all the assets for the site. See “Approving Multiple Assets,” on page 266 for help with approving many assets at once.

If you want to perform a simple test of your configuration, you could temporarily create a home page asset with just a few dependents. For information about approving individual assets, see the *Content Server User's Guide*.

Step 8 (Mirror to Server): Publish the Assets

After you have approved assets that can be published to your destination, you can run a test publishing session:

1. In the Content Server interface on the source system, click the **Publishing** button from the main toolbar.
2. In the **Publish Console**, select your mirror destination from the drop-down list and then click **Select Destination**.

CS-Direct 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 publish will not be successful. To initialize the destination, click the **Create Site on Target** button in the **Publish Console**. This creates the basic site information on the destination system.
4. Click **Publish**.

The publishing system mirrors all the approved assets for this destination to the Content Server database on the destination system.

Step 9 (Mirror to Server): Test the Results

To test the results, use your browser to navigate to the URL of the home page asset on the destination system and examine the site.

First, you must determine what that URL is. A Content Server URL is constructed by concatenating the following values:

- The name of the destination system's server and sometimes the port number as well.
- The CGI path, which it obtains from the `ft.cgiopath` property in the `futuretense.ini` file. For example, for WebLogic and other application servers with servlet architectures, this path is `/servlet/`.
- The string `ContentServer?pagename=`
- A page name from a SiteCatalog entry.
- Additional information that is passed in with the CS-Direct page criteria variables, `c`, `cid`, `tid`, and `p` (see the *Content Server Developer's Guide* for information about these variables).

For example, the URL for the Burlington Financial home page looks like this when it is rendered by the JumpStart Kit:

```
http://localhost:7001/servlet/
ContentServer?pagename=BurlingtonFinancial/Page/Home
```

Complete the following steps to determine the URL of your home page and then use it to test the site:

1. Start the Property Editor and open the `futuretense.ini` file for the destination system. (If you need help with this step, see the *Property Files Reference*.)
2. Select the **App Server** tab.
3. Select the `ft.cgiopath` property and write down the value.
4. Select the **Compatibility** tab.
5. Select the `ft.approot` property and write down the value.
6. Select **File > Exit** to close the Property Editor.
7. Open a text editor. Type the server name, a slash (/), and then the cgiopath. Precede the server name with the proper protocol—`http://` or `https://`—as in the following example:

WebLogic, WebSphere, and Sun ONE Application Server
`http://serverABC/servlet/`

8. At the end of the string, type a slash, and then add the following text:

`ContentServer?pagename=your_home_page`

Now the URL should look similar to the following examples:

Example for WebLogic, WebSphere, and Sun ONE:

`http://serverABC/servlet/ContentServer?pagename=ExampleSite/
Page/Home`

9. Open a web browser and enter the URL for the home page in the address field.
10. Scroll from the top to the bottom of the page and check for errors.
11. Click on all links to verify that they are working properly.

Note

FatWire recommends 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.

Step 10 (Mirror to Server): Set Up the Schedule

After you have ensured that your destination is configured correctly, you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see “Scheduling Publish Events,” on page 269.
- If you are using images that are not assets in the design of your site—that is, your site designers want to store all images on the web server rather than manage them as assets—plan how you will move the image files from the web server for the management system to the web server for the delivery system. For example, you can set up a regular FTP transfer.
- If you are using elements and SiteCatalog page entries that are not CSElement and SiteEntry assets, you must use the CatalogMover tool to mirror them to the destination system. For help with CatalogMover, see the *Content Server Developer's Guide*.
- If you are using the eWebEditPro HTML editor (from Ektron) and your content providers will use the upload feature on that toolbar to upload image files, you must set up an FTP or other process to move those uploaded image files from the management system to the delivery system. Because these files are not assets, they are not mirrored to the destination.

Step 11 (Mirror to Server): Turn Off Asset Invalidation on the Delivery System

By default, the publishing system is configured to mark an asset as changed on the destination system when you publish an asset from one system to another (source to destination). Then, the newly-published asset must be approved on the destination system before it can be published to another destination.

The default configuration is appropriate for development and management systems. However, when you are publishing to the delivery system, there is no need for the

publishing system to take the time to mark the change—assets are published to the delivery system but not from it.

Therefore, on the delivery system, turn off this publishing feature by completing the following steps:

1. Start the Property Editor and open the `futuretense_xcel.ini` file. If you need help with this step, see the *Property Files Reference*.
2. On the **Publishing** tab, select the `xcelerate.publishinvalidate` property.
3. Set the value to `false`.
4. Save the file and exit the Property Editor.

Configuring Your System for Export Assets to XML

The main steps when configuring for the Export Assets to XML delivery type are as follows:

- Configure the export directory that stores the files.
- Create a publishing destination that points to the export directory.
- Approve some test assets.
- Publish the test assets and test them.
- Set up the schedule.

Step 1 (Export to XML): Specify the Base Export Directory

As mentioned previously in this chapter, the directory that CS-Direct exports files to is determined by the `cs.pgexportfolder` property in the `futuretense.ini` file.

To set the base directory for the exported files, complete the following steps:

1. Start the Property Editor. If you need help with this step, see the *Property Files Reference*.
2. Open the `futuretense.ini` file for the source CS system.
3. On the **Export/Mirror** tab select the `cs.pgexportfolder` property and set the value to the file directory (location) where all files should be exported to.

This is a global setting for the system. If you have multiple destinations or are publishing with both the Export to Disk and Export Assets to XML delivery types and you want the files stored in separate places, use the `DIR` publishing argument to override the value of `cs.pgexportfolder` when you configure your Export Assets to XML destinations.

4. Save the new value by selecting **File > Save**.
5. Select **File > Exit** to close the Property Editor.
6. Stop and restart the application server.

Note

Before you run an Export to XML publish, make sure that this destination directory exists and that it has enough space for the XML files that will be created there.

Step 2 (Export to XML): Configure an Export Assets to XML Destination

Next, create the Export to XML destination by completing the following steps:

1. Log in to the Content Server interface on the source CS system.
2. On the **Admin** tab, select **Publishing > Destinations**.
3. Select **Add New**.

The “Add New Destination” form appears.

Add New Destination

*Name:	<input type="text"/>
Delivery Type:	Export Assets to Xml. Export XML files for each asset
Destination address:	<input type="text"/>
Arguments:	<input type="text"/>
*Sites:	<div> Burlington Financial GE Lighting Hello Asset World </div>

Cancel

Add New Destination

4. In the **Name** field, enter a unique name for the destination.
5. In the **Delivery Type** field, select **Export Assets to XML**.
6. Leave the **Destination Address** empty. This field is for Mirror to Server only.
7. (Optional) If you want to use a different directory than is specified by the `cs.pgexportfolder` property, click in the **Arguments** field and set an `DIR` argument.

For example:

`DIR=/export/XML`

8. In the **Sites** box, select the sites whose assets can be approved for and published to this destination.
9. Click **Add**.

The information about this destination is written to the `PubTarget` table.

10. Repeat steps 2 through 9 for each additional export destination that you need to configure for your source system.

Step 3 (Export to XML): Approve Your Assets

If you want to perform a simple test of your configuration, select an asset and approve it and any of its dependent assets.

If you want to complete a test publish of your entire site, approve all the assets for the site. See “Approving Multiple Assets,” on page 266 for help with approving many assets at once.

Step 4 (Export to XML): Publish and Test the Results

After you have approved assets that can be published to your destination, you can run a test publishing session:

1. In the Content Server interface on the source system, click the **Publishing** button from the main toolbar.
2. In the **Publish Console**, select your destination from the drop-down list and then click **Select Destination**.

CS-Direct displays information about the assets that are ready to be published to this destination.

3. Click **Publish**.

The publishing system exports all the approved assets for this destination into files.

4. In the **Publish Console**, click the inspect icon next to the session summary for this session.
5. In the session summary, which displays a list of all the resulting XML files, verify that the files are correct.

Step 5 (Export to XML): Set up the Schedule

After you have ensured that your destination is configured correctly, that the directory names are created according to your needs, you can finish configuring your publishing system by completing the following steps on the source system:

- Create scheduled publish events for the destination. For help with this step, see “Scheduling Publish Events,” on page 269.
- Plan how you will move the generated files to your external, non-CS systems. For example, you can set up a regular FTP transfer that automatically moves files to a testing area after a publishing session completes.

Additional Publishing Procedures

In addition to the configuration steps described in preceding sections of this chapter, administrators and developers also perform the following publishing procedures whenever necessary:

- Migrating a Site from One System to Another System
- Approving Multiple Assets
- Creating Destinations
- Editing Destinations
- Deleting Destinations
- Creating Export Starting Points
- Scheduling Publish Events
- Reading the Schedule Abbreviations
- Editing Publish Events
- Overriding the Schedule
- Assigning Approval or Preview Templates
- Monitoring Sessions in the Publishing Console
- Verifying Publishing Readiness
- Managing Publishing History Information
- Publishing All Approved Assets

Migrating a Site from One System to Another System

During the development phase for your sites, your site designers and developers develop sites and asset types with all their supporting configuration (subtypes, associations, start menu items, and so on), code templates, and assist the administrators in customizing the user interface, when necessary, by writing elements for workflow and so on. Your site designers and developers do this work on the development system.

When the site is ready for the content providers, you can use the **Mirror Destination Configuration** feature to migrate it to the management system and the delivery system.

Note

You can use this feature to move your configuration data only when **none** of the **sample sites** exist on the **destination** system.

Moving a Site from a Development System to a Management System

Complete the following procedure **once**:

1. Log in to the Content Server 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 Content Server interface.
5. Create a **Publishing Destination** for the destination system that uses the Mirror to Server delivery type.
For help with this step, see “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.
6. Initialize the mirror destination.
For help with this step, see “Step 5 (Mirror to Server): Initialize the Destination,” on page 255.
7. Configure the mirror destination.
For help with this step, see “Step 6 (Mirror to Server): Configure the Mirror Destination,” on page 257.
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),” on page 97.
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,” on page 266.
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 the *Content Server Developer's Guide*.

Moving a Site to a Delivery System

When you move a site to a delivery system, there is no need to mirror start menu items, workflow, saved searches, and so on.

Complete the following steps:

1. Log in to the Content Server 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 Content Server interface.
4. Create a **Publishing Destination** for the destination system that uses the Mirror to Server delivery type.
For help with this step, see “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.
5. Initialize the mirror destination.

For help with this step, see “Step 5 (Mirror to Server): Initialize the Destination,” on page 255.

6. Configure the mirror destination.

For help with this step, see “Step 6 (Mirror to Server): Configure the Mirror Destination,” on page 257.

Do **not** select start menu items, saved searches, or any of the workflow options.

7. If you created a flex family, log back in to the source system. Approve and then publish all of the data structure flex assets to the management (destination) system.

You can either approve each asset individually or use the **Approve Multiple Assets** feature on the **Admin** tab. For information about this feature, see the “Approving Multiple Assets,” on page 266.

8. Approve and then publish all of the rest of the appropriate assets to the destination (delivery) system: template, page, collection, query, CSElement, and SiteEntry assets, and so on.

9. If there are any elements or page entries in the `SiteCatalog` table that are not CSElement or SiteEntry assets, use CatalogMover to mirror those items to the management system. For help with CatalogMover, see the *Content Server Developer's Guide*.

10. If there are any images in the online site that are not assets, be sure to copy them to the delivery system.

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 Content Server database with the BulkLoader utility. Both BulkLoader and BulkApprover are described in the *Content Server Developer's Guide*.

To approve a group of assets, follow these steps:

1. In the Content Server interface, select **Admin > Publishing > Destinations** and then expand the destination that you want to approve assets for.
2. Under that destination, select the **Approve Multiple Assets** option.

The “Approve Assets” form appears:

Approve Assets for Publish to Dynamic

***Asset Types:**

- Article
- Article (Flex)
- Attribute Editor
- CSElement
- Collection
- Content Attribute
- Content Definition
- Content Parent

Sample queries:

- Select sample query, if needed-
- Assets of selected asset type(s).
- Assets of selected asset type(s) in site:GE Lighting
- 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

Cancel Approve for Publish

3. In the **Asset Types** section, from the list on the left, select the asset types that you want to approve.
4. From the **Sample Queries** list on the right, select a query. If the exact query that you need is not present, select the query that is most like the one that you want.
CS-Direct creates a SQL query based on the items that you selected and displays it in the **Query** box.
5. (Optional) If necessary, edit the SQL query.
6. Click in the **Approve in Batches of** field and enter a numeric value. The default is set to 500.

Note

When you click the **Approve for Publish** button, the approval system approves batches of assets. The number of assets in each batch is determined by value in the **Approve in Batches of** field. Because the approval process is not a background process, it is possible that the browser session could time out while the batch is being approved.

When using the **Approve Multiple Assets** feature, to ensure that the session does not time out, adjust the following settings:

- The `cs.timeout` property in the `futuretense.ini` file, which sets the browser session timeout value. Start by setting this value to 1800 (which means 30 minutes). (See the *Property Files Reference*.)
- The value you specify in the **Approve in Batches of** field. Start by using the default of 500 and lower it if necessary.

7. Specify the **Approve Previously Approved Assets** option as follows:
 - If you want CS-Direct to ignore all previous approvals and re-approve all selected assets, select **Yes**. This is the option to use if previous approvals may have become invalid for some change such as a change in default templates for that destination.

- If you want CS-Direct 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 one of the procedures in the following list, as appropriate:

- To create an export destination, see “Step 2 (Export to Disk): Configure an Export Destination,” on page 247.
- To create a mirror destination, see “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.
- To create an export assets destination, see “Step 2 (Export to XML): Configure an Export Assets to XML Destination,” on page 262.

Editing Destinations

To edit a destination:

1. Select **Admin > Publishing > Destinations**,
2. Double-click on the destination that you want to edit.

Inspect Edit Delete Approve Multiple Assets Set Default Templates	
Name:	Dynamic
Delivery Type:	Mirror to Server: Copy database rows to remote dynamic server Initialize Mirror Destination...
Destination address:	http://burst7/servlet/
Arguments:	REMOTEUSER=gayle&REMOTEPASS=1
Sites:	Burlington Financial GE Lighting
Publish Event:	No existing publish event Set Publish Event...
ID:	1033139336575

Force Publish Approved Assets

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 “Step 2 (Export to Disk): Configure an Export Destination,” on page 247.
- For a Mirror to Server destination, see “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.

- For an Export Assets to XML destination, see “Step 2 (Export to XML): Configure an Export Assets to XML Destination,” on page 262.

Deleting Destinations

To delete a destination, complete the following steps:

1. In the Content Server interface, select **Admin > Publishing > Destinations**.
2. Double-click on the destination that you want to delete.
3. Click the **Delete** icon.

CS-Direct displays a verification message.

Creating Export Starting Points

Export starting points are defined in “Export Starting Points,” on page 237. For information about creating them, see “Step 4 (Export to Disk): Create the Export Starting Points,” on page 249.

Scheduling Publish Events

You set up publishing events for all of your publishing destinations (destinations). Because publish events run as background processes and there is no publish queue, you can set up publish events for different destinations that run at the same time, if necessary.

Note the following:

- There can be only **one** publish event 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, complete the following steps:

1. In the Content Server interface, select **Admin > Publishing > Destinations**.
2. Double-click on the destination that you want to schedule a publish event for.
3. In the “Publish Destination” form, click the **Set Publish Event** link.

CS-Direct displays the “Publish Event” form:

Edit Publish Event for Destination: Dynamic

Name: Dynamic				
Times: No existing publish event				
Set Event Times:				
Days of a week	Days of a month	Months	Hours	Minutes
Sunday	1	Jan	12AM	0
Monday	2	Feb	1AM	15
Tuesday	3	Mar	2AM	30
Wednesday	4	Apr	3AM	45
Thursday	5	May	4AM	
Friday	6	Jun	5AM	
Saturday	7	Jul	6AM	
	8	Aug	7AM	
	9	Sep	8AM	
	10	Oct	9AM	
	11	Nov	10AM	
	12	Dec	11AM	
Enabled: <input checked="" type="radio"/> yes <input type="radio"/> no				

Cancel

Save

Now you select days, months, hours, and minutes to set a schedule.

Example Schedule

For example, if you want to set a schedule so that CS-Direct 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 **Days of a week** field, click on **Monday** and shift-click on **Saturday** to select all days of the week except Sunday.
2. Skip the **Days of a month** field, since you have already selected the days you want.
3. In the **Months** field, click on **January** and shift-click on **December** to select all months.
4. In the **Hours** field, click on **7am**, scroll down the list, control-click on **4pm**, and control-click again on **8pm**.
5. In the **Minutes** field, click on 0.
6. Under Enable, select **yes**.
7. Click the **Save** button.

A summary of the schedule now appears in the **Publish Event** section of the “Publish Destination” form.

Reading the Schedule Abbreviations

The abbreviations for scheduled events are displayed in several places in the Content Server interface. Here is the key:

hours:minutes:seconds weekdays/daysOf Month/months

This information is displayed as follows:

- Hours are displayed as numbers from 0 (midnight) through 23 (11 p.m.) and are separated from the minutes with a colon.

If the schedule describes more than one hour, they are displayed with commas separating them.

The previous example (see “Example Schedule,” on page 270), was for 7 a.m., 4 p.m., and 8 p.m. as the times to publish, which is displayed as **7,16,20:**

- Because you can set minutes in increments of 15, minutes are displayed as numbers 0, 15, 30, or 45 and are separated from the seconds with a colon.

If the schedule describes more than one minutes increment, they are listed with commas separating them.

Minutes are displayed as a **0**, which 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 ending with the slash (/) character.
 - If no days were selected, it means to run every day and that is displayed with the asterisk (*) character.
 - If more than one day is scheduled, they are displayed with commas separating them.

The example schedule includes all the days of the week except Sunday, which means 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, it means that they are all selected and it is displayed as the asterisk (*) character.
 - 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, which means a value for this item is appended to the display as an asterisk:

7,16,20:0:0 1,2,3,4,5,6/*/

- At the end of the string is 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, it is displayed as the asterisk (*) character.

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

To edit a publish event, complete the following steps:

1. In the Content Server interface, select **Admin > Publishing > Destinations**.

2. Double-click on the destination that you want to schedule a publish event for.
3. In the “Publish Destination” form, click the **Set Publish Event** link.
CS-Direct displays the “Publish Event” form. The text at the top of the form describes how the event is currently configured.
4. Edit the event as needed and click **Schedule**. To clear your selections, click **Cancel**.

Overriding the Schedule

If you want to start a publishing session immediately, complete the following steps:

1. In the Content Server interface, click the **Publishing** button in the toolbar.
2. In the **Publishing Console**, from the drop-down list, select the destination that you want to publish to.
3. Click the **Select Destination** button.

CS-Direct determines whether there are any assets that can be published to the destination and displays a summary form like this one:

The screenshot shows a summary form for the destination 'Mirrorto'. It includes the following text: 'Publish destination: Mirrorto', 'Destination: Mirrorto using Mirror to Server', 'Arguments:', '12 assets are held for publish.', and '1014 assets are ready for publish.'. At the bottom, there are two buttons: 'Cancel' and 'Publish'.

If there are assets that can be published to the destination, the **Publish** button is displayed. If there are no assets to publish, there is no **Publish** button.

4. Click **Publish**.

CS-Direct starts the publishing session.

Note that if a publishing session for this destination is already underway, the publish event that you just tried to run fails and CS-Direct 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

As mentioned earlier in this chapter (see “Calculating Dependencies for Export to Disk,” on page 222), when assets are approved for a publishing destination that uses the Export to Disk publishing method, the approval system examines the template assigned to the asset to determine its dependencies.

However, when Export to Disk actually publishes the asset, it does not necessarily use the template that is assigned to the asset. Why? Because the code in another element could determine that a different template is used for that asset in certain cases.

Consider the Burlington Financial sample site. An article asset from this sample site can be rendered by several different templates, depending on the context.

So when someone approves an article asset for the Burlington Financial sample site, which template should the approval process use to determine the dependencies for the article? The one that contains the most representative set of dependencies for all of the templates.

What if the template that contains the most representative set of dependencies is not the template that you want to assign to the asset? Set it as the Default Approval Template for assets of that type.

Note that when you assign default templates to assets that are published to mirror destinations, those templates are not used to calculate dependencies, but they are used when someone previews the asset from the “Status” form.

To set a default template, complete the following steps

1. In the Content Server interface, select **Admin > Publishing > Destinations**.
2. Expand the destination that you want to configure default templates for and then select the Set Default Templates item.
3. CS-Direct displays a “Default Templates” form.
4. Click **Edit**.

The system displays a form like this one:

Default templates for: Static

Asset Type	Subtype	Template
Articles	DrillHierarchyCT	-- None specified; use asset's template field --
	Story	-- None specified; use asset's template field --
AdvCols	(no subtype)	-- None specified; use asset's template field --
Article	Columnist	-- None specified; use asset's template field --
	Standard	-- None specified; use asset's template field --
Collection	Article	-- None specified; use asset's template field --
	Collection	-- None specified; use asset's template field --
	HelloArticle	-- None specified; use asset's template field --
HelloArticle	(no subtype)	-- None specified; use asset's template field --
ImageFile	Standard	-- None specified; use asset's template field --
Page	Standard	-- None specified; use asset's template field --
Products	Lighting	-- None specified; use asset's template field --
	PDFTYPE	-- None specified; use asset's template field --
	PTypeFund	-- None specified; use asset's template field --
Query	Article	-- None specified; use asset's template field --
	Collection	-- None specified; use asset's template field --

Cancel

Save

5. For each asset type that you need to configure, select the template that you want CS-Direct to use as the approval or preview template for assets of that type. If there are subtypes configured for an asset type, you can specify a default template for each subtype of that asset type.
6. Click the **Save** button.

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 Content Server interface:

Publish Console

Select Publish Destination

Publish destination

Static (using Export to Disk)

Select Destination













Running Publish Sessions

No Running Publish Sessions

Scheduled Publish Tasks

No Scheduled Publish Tasks

Publish History

	Destination	Publish End Time	Status	Published by
 	ExportGayle	Aug 29, 2002 4:47:43 PM	Done	gayle
 	ExportGayle	Aug 29, 2002 3:35:22 PM	Done	gayle
 	ExportGayle	Aug 29, 2002 2:49:47 PM	Done	gayle
 	ExportGayle	Aug 29, 2002 12:01:59 PM	Done	gayle
 	ExportGayle	Aug 28, 2002 4:24:55 PM	Done	gayle
 	ExportGayle	Aug 28, 2002 12:26:21 PM	Failed	gayle

The console displays a summary of all the publishing activity, including any sessions that are currently running, scheduled, or completed.

Verifying Publishing Readiness

To determine whether a publishing session is ready or whether there are any problems with unapproved assets, follow these steps:

1. In the Publish Console, select the destination from the drop-down list.
2. Click the **Select Destination** button.

A summary form appears. It lists any assets that need to be published but are not ready (they are held) and assets that can be published. For example:

Publish destination: Mirrorto

Destination: Mirrorto using Mirror to Server

Arguments:

[12 assets are held for publish.](#)
[1014 assets are ready for publish.](#)

Cancel

Publish




















3. To see the details, click on the link.

CS-Direct displays a list of the assets that are either approved or approved and still held for dependency reasons. For example:

Publish destination: GayleExp

Destination: GayleExp using Export to Disk
Arguments: URLPREFIX=/mysite

1164 references are ready for publishing.

Asset Name	Type	Template	Other Arguments
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685128066
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685128066
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685129142
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685129142
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685129804
 Sony-A40-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685129804
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685128066
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685128066
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685128818
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685128818
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685129804
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685129804
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968695082734
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968695082734
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968695082886
 US-A44-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968695082886
 Malaysia-A51-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685128066
 Malaysia-A51-2001Mar9	Article	BurlingtonFinancial/Article/FullText	p=968685128066
 Malaysia-A51-2001Mar9	Article	BurlingtonFinancial/Article/Full	p=968685128734

If you need to resolve an issue, click the link to the asset and make any changes that are necessary (and perhaps approve it):

- For information about why assets can be held, see “Held Assets,” on page 221.
- For procedures that describe how to work with assets, see the *Content Server User’s Guide*.

Managing Publishing History Information

Publishing sessions that are completed are listed in the **Publish History** section of the **Publish Console**, which is displayed when you click the **Publishing** button. Each publish session in the list has a status; Running, Done, or Failed.

Each item in the list has two icons:

- The inspect icon (the circled letter “i”), which displays a summary of that session. Examining the summary is the first step in troubleshooting a failed session.
- The delete icon (the trash can), deletes the session. When you click this icon, CS-Direct deletes the row from the PubSession table, the publishing log file for the session, and any messages for the session from the PubMessage table.

Note that the greater the number of publishing sessions listed in this section, the longer it takes CS-Direct 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,” on page 278.

About Session History

The publishing history log files are displayed in 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 delivery type:











- For Mirror to Server, a reference is an **asset**. Each asset that is published is listed by its asset type and its ID.
- For Export to Disk and Export Assets to XML, a reference is a **generated file**. Each file that is created is listed by its file name.

Here is an example of a session history for an Export to Disk publishing session:

Publish session: 1035311794199

Destination:	GayleExp using Export to Disk
Arguments:	URLPREFIX=/mysite
Published by:	gayle
Publish Date:	Oct 24, 2002 10:47:40 AM

Exported references:

Asset Name	Type	Template	Other Arguments
 Janus-Col2-2001Mar27	Article	BurlingtonFinancial/Article/Columnist	p=968685129956
 Janus-Col2-2001Mar27	Article	BurlingtonFinancial/Article/Full	p=968685129956
 Janus-Col2-2001Mar27	Article	BurlingtonFinancial/Article/FullText	p=968685129956
 Home	Page	BurlingtonFinancial/Page/Home	
 Columnists	Page	BurlingtonFinancial/Page/ColumnistFront	
 Columnists	Page	BurlingtonFinancial/Page/ColumnistFrontText	
 Contact Us	Page	BurlingtonFinancial/Page/AboutUsText	
 Contact Us	Page	BurlingtonFinancial/Page/Shell	
 Hot Topic	Query	BurlingtonFinancial/Query/HotTopicFront	p=968685129956&topicword=Funds
 Hot Topic	Query	BurlingtonFinancial/Query/HotTopicFront	p=968685129956&topicword=Stocks

Successful completion

Go to Publish Console

If there was a problem with the session, an error message is written to the PubMessage table and it is displayed as the history instead. If you see an error message in the history for a publishing session, see the section called “Troubleshooting,” on page 278 and attempt to resolve the problem.

Publishing All Approved Assets

The publishing system conducts incremental publishing sessions. That is, during any session, only those assets that have been approved since the last session for this destination are published.

However, you may find a situation in which you need to publish all the approved assets in your entire site. (For example, after data repair on the delivery system.) When this is the case, use the **Force Publish All Items Next Time** button in the “Edit Destination” form:

 Force Publish Approved Assets

When you click this button, it is the equivalent of changing the status of all approved assets to that of “never been published.” That means that the next publishing session to the destination will publish all approved assets, regardless of whether they have already been published and have not changed since they were published.

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:

- About Publishing System Error Messages
- Numeric Messages
- Other Indicators of System or Configuration Issues

About Publishing System Error Messages

The publishing system reports information about publishing sessions in the following ways:

- Displays a status for each session in the **Publish History** list in the **Publish Console**. If you see a status of “Failed,” “Not Found,” or “false,” there was a problem with the publishing session.
- Writes information about the assets that were published to the log file for the publishing session. When you click the inspect icon for a publishing session in the Publish Console, the history log file for that session is displayed.

The `request.folder` property in the `batch.ini` file defines where the publishing history log files are located. Each log file is named as follows:

`PubSession ID + Output.html`

For example, if the PubSession ID for the session you are interested in is 9876544, then the log file is named `9876544Output.html`, and it is located in the directory specified by the `request.folder` property.

- Writes error messages about the publishing session to the `PubMessage` table. These messages are displayed in the **Messages** text area when you examine the publishing history in the Publishing Console for a specific publishing session.

You can use the `PubSession ID` to look up messages in the `PubMessage` table about a specific publishing session. Note that when you set the `VERBOSE` publishing argument to true, 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.

Numeric Messages

This section contains descriptions and suggested corrective actions for the numeric error messages that the publishing system might report in any of the log files mentioned in the list above.

Note

The following list is not a complete list of all the error messages that your Content Server system can report. For a complete list of all error messages, see the error conditions appendix in the *Developer's Tag Reference*.

-2

This error indicates that either the mirror user name or password is not identified correctly. The publishing system on the source began the publishing process, but the destination system couldn't authenticate the user that is identified as the mirror user.

Corrective Action

Correct the mirror user information. See the procedure "Step 2 (Mirror to Server): Identify the Mirror User to the Source System," on page 253 for help.

-3

This error indicates that the mirror user does not have the correct permissions to save the assets to the Content Server database on the destination system. In other words, the destination system user identified to the source as the mirror user does not have all the ACLs it needs.

Corrective Action

Edit the mirror user's account and assign it the appropriate ACLs.

See the procedure "Step 1 (Mirror to Server): Set Up the Destination System," on page 252 for a list of the ACLs that the mirror user needs.

If you still cannot determine which ACL is missing from the mirror user's account, examine the `futuretense.txt` file on the destination system. There should be an entry that

describes which table the mirror process failed to update and the name of the table may help you determine which ACL is missing.

For example, if the mirror session is failing on a visitor table, it is likely that your mirror user does not have the Visitor or VisitorAdmin ACL assigned to it.

-103

This error indicates that one of the tables whose data is being mirror published does not exist on the destination system. This error typically occurs when there is an asset type on the source that does not yet exist on the destination. It is also possible that someone created a custom table to support a function that was custom designed for your site but that table has not yet been created on the destination system.

Corrective Action

Examine the `futuretense.txt` file on the **destination** system and look for the lines that list the -103 error. The text in the message should mention which table is missing.

Then use the appropriate tool to create the table on the destination system. That is, if it is a basic asset type, use AssetMaker. If it is a flex asset type, use Flex Family Maker. If it is a custom table, use CS-Explorer.

-611

This error message is one of the two generic mirror publishing error messages (the other is -12011). Typically -611 means that there was a problem when the publishing system tried to access the destination system and there should be other error messages reported, too.

Corrective Action

Examine the `futuretense.txt` 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.

Corrective Action

Log into the Content Server interface on the **source**, examine the definition of the destination, and determine that it is correct. For help, see “Step 4 (Mirror to Server): Create a Mirror Destination,” on page 254.

-12011

This error message is one of the two generic messages (along with -611) which mean that the mirror process failed. It is unlikely that this message will appear without other error messages being reported. Typically the other errors will give more information about what went wrong.

Corrective Action

Examine the messages in the `PubMessage` table and the `futuretense.txt` 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.

Corrective Action

Start by verifying that the web server and application server are running on the **destination** system. If they are not, start them. Verify that the two systems can connect to each other through the network (perhaps the network went off-line, for example).

-12045

This error indicates that the publishing system was able to start the session, but it then failed in some way.

Corrective Action

Examine the `futuretense.txt` file on the **destination** system and look for error and exception messages in the log.

-12046

This error indicates that there was a problem at the end of a publishing session when the publishing system began to clean up the temporary tables that it creates for each session. When the cleanup process began, the connection with the destination failed. That is, the source system could get no response from the destination when the publishing process tried to clean up the temporary tables.

Corrective Action

Determine whether the web server and application server on the **destination** system are running. If they are not, start them. Check for network connection problems, as well.

-12047

This error indicates that the publishing system was able to begin the cleanup process but it then failed in some way during the operation.

Corrective Action

Examine the `futuretense.txt` file on the destination and look for error exception messages and exceptions that describe the problem.

-13054

This error occurs for complex and flex assets only. In this case, the publishing system began publishing a complex or flex asset but the data did not reach the destination. It can mean that the data itself was corrupt or that the HTTP request connection failed.

Corrective Action

Check the `futuretense.txt` file on the source system. If the problem was the data itself, there will be additional messages in this log file that can help you determine which asset caused the problem. However, if the HTTP request was dropped before the destination system could respond, there will **not** be an additional message in the log.

-13055

This error occurs for complex and flex assets only. In this case, the data for the flex or complex asset was delivered, but the destination system could not save the asset to its Content Server database.

This message indicates that there is a problem with your data. For example:

- If you are attempting to publish from more than one source to the same destination, there can be ID collisions and other problems with data integrity that will cause this error.
- If you are using the AltaVista search engine on the source system, you see this problem if you have not yet configured the search engine on the destination or if you have not configured it correctly

Corrective Action

Examine the `futuretense.txt` 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.

The following messages appear when your AltaVista search engine is configured incorrectly:

- “-806 Search remove index failed” in the `futuretense.txt` file indicates that the search engine is not installed on the destination system.
- “-823 Native method mismatch” in the `futuretense.txt` file indicates that the `doc_converters` path is not included in `LD_LIBRARY_PATH` in the startup script. See the AltaVista release notes for configuration instructions.

Other Indicators of System or Configuration Issues

This section describes symptoms your CS system may exhibit when the publishing system discovers configuration errors or system problems.

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 “Create the Batch User Account,” on page 245 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 **web server** that hosts the source system, **not** the destination, **not** the application server if the application server and the web server are on different boxes, 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 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 and Export Assets to XML, you need enough disk space in your file system to store all the files that are being generated.
- For Mirror to Server, you need space equal to four times the size of the data that is being mirrored available on both the source and the destination system or the mirror operation will fail.

Note that the publishing process will also fail if the Java temp directory is not large enough.

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 `futuretense.txt` file on the **destination** system and look for messages like these:

- “Number of satellite servers must match number of usernames and passwords.”
This message indicates that there is something wrong with the values specified for the `cs.satellitehosts`, `cs.satelliteuser`, and `cs.satellitepassword` properties in the `futuretense.ini` file.
- “-100.FormPoster failed flushing URL.”
This message indicates that either a Satellite servlet wasn’t running or that the destination system couldn’t reach a Satellite servlet that it tried to reach:

Open the `futuretense.ini` file, examine the properties on the **Satellite** tab, and note the values set for the user names, passwords, and host names. Open the `satellite.ini` files

for the co-resident Satellite Server and your remote Satellite Server applications. Compare the values specified for the `cs.satellitehosts`, `cs.satelliteuser`, and `cs.satellitepassword` properties in the `futuretense.ini` file to the values set for the `username` and `password` properties in the `satellite.ini` files. **They must match.**

Remember that the order in which you specify host names in the `futuretense.ini` file must match the order in which you identify user names and passwords.

DB2 Systems, Troubles When Publishing Assets with Associations

When you notice that there are problems with publishing assets that have associations and your system uses a DB2 database, it is likely that the `LOCKLIST` parameter is not set correctly.

Corrective Action

The Content Server installation guide that describes DB2 installations recommends that this property be set to at least 1000. If you find that you are having difficulties publishing assets with associations, increase this value.

Chapter 13

Revision Tracking

Content Server provides revision tracking functionality that prevents a row in a table from being edited by more than one user at a time. When you enable revision tracking for a table, Content Server maintains multiple versions of a row in that table.

The CS-Direct application applies this revision tracking functionality to your asset types. You decide which asset types should be tracked and determine how many revisions to store. The content providers then check their assets in and out and can compare versions, if necessary.

The *Content Server User's Guide* describes how content providers work with assets when revision tracking is in use. This chapter describes how to enable revision tracking and how to manage versions of assets.

This chapter contains the following sections:

- Overview
- Enabling Revision Tracking
- Disabling Revision Tracking
- Unlocking Revisions
- Additional Revision Tracking Functions for Non-Asset Tables

Overview

Content Server provides revision tracking functionality through its revision tracking API. CS-Direct uses this API to provide additional revision tracking functionality for asset type tables.

When you enable revision tracking for an asset type, Content Server 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.

CS-Direct's 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, Content Server stores versions in a **tracker** table; upload data is stored in a storage directory that you specify, as shown in "The RTInfo Table," on page 287. Because past versions are stored (that is, a history exists), a user can **roll back** an asset to a previous version or examine the **differences** between two versions of the asset.

- Administrative or maintenance features.

An administrator can **delete past versions** of an asset or **clear the checkout** for an asset by overriding the check out on it and checking it back in.

When an asset type is being tracked, CS-Direct provides checkin, checkout, and other revision tracking features on the **New** and **Edit** forms for assets of those types.

Tracker Tables and Storage Directories

When you enable revision tracking for an asset type, Content Server creates a tracker table that stores revision information for the records in the source table. A tracker table has the same name as the main storage table for the asset type with `_t` appended to it. For example, the tracker table for the article asset type would be named `Article_t`. The tracker table for the attribute type asset type would be named `AttrTypes_t`.

For each record in a tracked asset type table, there are several rows in the corresponding tracker table that stores its version information. Tracker tables have two kinds of columns:

- Columns that store the system information that the revision tracking system needs to keep track of all the versions
- Columns that hold the IDs of text files that are stored in a storage directory

When a new version of an asset is checked in, Content Server creates a separate text file to hold the data in each of the asset type's upload (URL) fields and in any text fields that are configured to hold more than 64 characters. These files are stored in the storage directory that you specify when you enable revision tracking. 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 Content Server Explorer. Do **not** attempt to modify the information in any of the tracker tables with a database tool.

The RTInfo Table

While the tracker tables are kept hidden, the `RTInfo` table is visible through Content Server Explorer. This table holds information about which tables are being revision tracked. It has the following columns:

Column	Description
<code>tblname</code>	The name of a table that is being revision tracked. For asset types, this is the name of the main asset type table.
<code>versions</code>	The number of versions to store for each asset of this type.
<code>storage</code>	The path to the storage directory that holds the text files for each version of assets of this type. If, when you enable revision tracking, you do not specify a storage directory, Content Server creates a revision tracking subdirectory in the <code>defdir</code> directory for the source table.
<code>recordupdate</code>	A timestamp of the last time a version was stored in the tracker table.
<code>trackingupdate</code>	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 Checkout

When revision tracking is on for an asset type, Content Server provides both implicit (or automatic) and explicit (or manual) checkout/checkin functionality. When users create or edit assets of a type that is being revision tracked, they do not have to manually check out the asset: it is automatically assigned to them. Then, when they click **Save**, the asset is checked back in. This is known as implicit checkin/checkout.

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, each save creates an archived version. Depending on how many revisions the asset type is configured to store, that author might overwrite an older version that he or she really wanted to keep.

When a content provider manually (explicitly) checks out an asset, a version is not stored—no matter how many times he or she saves it—until it is manually checked back in.

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 **Content Server Management Tools** feature on the **Admin** tab to enable tracking for the table. Content Server then creates a corresponding tracker table to support the tracked table. It is named the same way as a tracker table for an asset type: `nameOfTable_t`. For example, if you enabled revision tracking for the Source table, the tracker table would be called `Source_t`.

When revision tracking is enabled for a non-asset table, you can use either the revision tracking features accessible from the menus in Content Server Explorer or the “Content Server Management Tools” node on the “Admin” tab to lock (check out) a row and then unlock the row (check it back in) when you are finished with it.

If you need to provide additional support outside of the Content Server Explorer tool for revision tracking of a non-asset table, your developers can code additional forms, using the Content Server revision tracking API and revision tracking XML or JSP tags. For information, see the *Content Server Developer's Tag Reference* and the *Content Server Java API Reference*.

Note

If you need to delete a non-asset table from the Content Server database and that table is being revision tracked, be sure to 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 need for a rollback of several versions

For example, an asset that consists of a small amount of ASCII data occupies so little space that a large number of revisions would take little disk space. However, each version of an asset that holds a large amount of binary data could occupy a significant amount of disk space. In the second case, you must strike the appropriate balance, storing the fewest number of versions necessary for rollback purposes.

Enabling Revision Tracking

You enable revision tracking differently for asset types than you do for tables that do not hold assets.

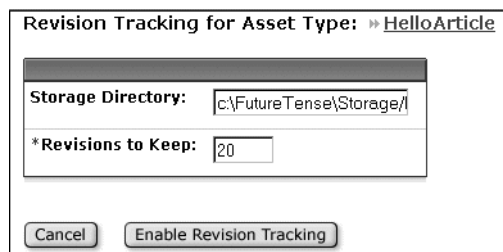
- For asset types, you use the nodes under the **Asset Type** item on the **Admin** tab.
- For non-asset tables, you use the **Revision Tracking** node under the **Content Server Management Tools** item on the **Admin** tab.

Enabling Revision Tracking for Asset Types

To enable revision tracking for asset types

1. In the Content Server interface, select the **Admin** tab and expand the **Asset Types** item.
2. Expand the asset type for which you want to enable revision tracking.
3. Select **Revision Tracking > Track**.

The “Track Asset Type” form appears.



The screenshot shows a web form titled "Revision Tracking for Asset Type: » HelloArticle". Inside the form, there is a "Storage Directory:" label followed by a text input field containing "c:\FutureTense\Storage\I". Below this is a "*Revisions to Keep:" label followed by a text input field containing "20". At the bottom of the form are two buttons: "Cancel" and "Enable Revision Tracking".

4. Click in the **Storage Directory** field and enter the path name to the directory where you want the supporting text files for versions of assets of this type to be stored. Use the full pathname and do not add a slash character after the directory name.
5. Click in the **Revisions to Keep** field and enter the number of revisions that you want the system to keep stored. Once this number of revisions is stored, the oldest version is deleted when a new revision is created. If you are using revision tracking solely for its record-locking feature and you do not need the ability to roll back to previous versions, you can set this field to 1.
6. 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 **Content Server Management Tools** to set revision tracking. Note that in older versions of the product, database tables were called “catalogs” and the **Content Server Management Tools** still use the old terminology in some places.

To enable revision tracking for non-asset tables

1. In the Content Server interface, select the **Admin** tab and then expand the **Content Server Management Tools**.
2. Double-click the **Revision Tracking** option.

3. In the “Revision Tracking” form, select **Track Tables** and click **OK**.
4. In the “Track Tables” form, click in the **Enter root storage directory** and enter the pathname to the directory where you want versions of rows to be stored. Use the full pathname and do not add a slash character after the directory name.
5. In the field **Enter number of revisions to keep** field, enter the number of revisions to keep in storage for rollback purposes.
6. Select the **Track?** option next to the name of the table for which you are enabling revision tracking.
7. Click **Track Catalogs**.

Editing Revision Tracking Settings

You must have the SiteGod 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 the SiteGod ACL, 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.

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 for 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

Increasing

If you want to increase the number of revisions to be stored for an asset type or table, simply increase the value in the **Revisions to Keep** field:

- For asset types, select **Admin** tab > **Asset Types** > the asset type you want to modify > **Revision Tracking** > **Set Revisions** and then increase the value.
- For non-asset tables, select **Admin** tab > **Content Server Management Tools** > **Revision Tracking** > **Set Table Revisions** and then increase the value.

Decreasing

Although you might need to decrease the number of versions while you are testing configuration settings on a CS development system or while you are fine-tuning the CS

management system, it is best if you do not decrease the number of versions being stored by Revision Tracking on a fully functioning management system.

If you decrease the value in the **Revisions to Keep** field to a number that is less than the number of revisions currently being stored for that asset type or table, the following occurs:

- The text files in the storage directory for the extra versions are orphaned.
- The rows in the tracker table for the extra versions are orphaned.

You can avoid creating orphan rows in the tracker table by deleting the extra versions (the oldest ones) before you decrease the number of revisions. However, you cannot avoid creating orphan text files in the storage directory and you should not attempt to delete them because it is very difficult to determine which ones to delete.

If you must decrease the number of revisions, complete the following steps (in this order):

1. Follow the steps in the procedure “Deleting Revisions.”
2. To decrease the number of revisions, do one of the following:
 - If you are decreasing the number of versions for an asset type, select **Admin > Asset Types > your asset type > Revision Tracking > Set Revisions** and decrease the number of versions stored for that asset type.
 - If you are decreasing the number of versions for a non-asset table, select **Admin > Content Server Administrator Tools > Revision Tracking > Set Catalog Revisions** and decrease the number of versions stored for that table.

Deleting Revisions

Under certain conditions, you might need to delete old versions for an asset type or table that is being revision tracked.

To delete revisions

1. Select the **Admin** tab and then expand **Content Server Administrator Tools**.
2. Double-click the **Revision Tracking** option.
3. In the “Revision Tracking” form, click in the **Enter Table Name** field and enter the name of the table.
4. Click in the **Enter Value for Key** field and enter the ID of the asset or the row whose versions you want to delete. (You can obtain the ID of an asset by inspecting it in the Content Server interface. You can obtain the ID of any row by examining the table with Content Server Explorer.)
5. Select **Delete Revisions** and then click **OK**.
6. In the “Delete Revisions” form, select the option next to the versions that you want to delete. You can use the **All** button to select all of the boxes and the **None** button to clear all of the boxes.
7. Click **Delete Revisions**.

Disabling Revision Tracking

When you disable revision tracking for an asset type or a non-asset table, the following occurs:

- The tracker table is inactivated, but not deleted. This means that the versions stored in it are orphaned, but not deleted. And, after revision tracking is disabled for a table, you can no longer delete versions by using the Content Server revision tracking forms.
- All links to the text files in the storage directory are broken, but the files themselves are not deleted. They, too, are orphaned.

If you later decide to enable revision tracking for that asset type or non-asset table, the old versions and text files are ignored. They remain orphaned.

Because these orphaned versions and text files take up disk space and there could be a large number of them stored on disk, it is best practice to complete the following tasks in the following order when you want to disable revision tracking:

1. Use the Content Server Management Tools revision tracking forms to delete all the versions stored for the table. See the procedure “Deleting Revisions,” on page 291.
2. Disable revision tracking for the table. See the procedures in this section.
3. Manually delete all the text files from the storage directory for that asset type or database table.


Disabling Revision Tracking for Asset Types

To disable revision tracking for asset types

1. In the Content Server interface, select the **Admin** tab and expand the **Asset Types** item.
2. Expand the asset type for which you want to disable revision tracking.
3. Select **Revision Tracking** and double-click **Untrack**.

A confirmation message appears.

Revision Tracking for Asset Type: » [HelloArticle](#)

 Are you sure that you want to Untrack?
If you select Untrack, revision tracking is cleared
and only the current version of each asset remains.

Cancel

Untrack

4. Click **Untrack**.

Disabling Revision Tracking for Non-Asset Tables

To disable revision tracking for non-asset tables

1. On the **Admin** tab, expand the **Content Server Management Tools** item.
2. Double-click the **Revision Tracking** option.
3. In the “Revision Tracking” form, select **Untrack Tables** and click **OK**.

4. In the “Untrack Tables” form screen appears, select the **Untrack?** option next to the table that you want to disable revision tracking for.
5. Click **Untrack Catalogs**.

Unlocking Revisions

Occasionally, user operations may leave an asset in an inappropriate locked state. Resolving these states is an administrative responsibility.

- For asset types, use the **Clear Checkouts** item on the **Admin** tab.
- For non-asset tables, use the **Content Server Management Tools** revision tracking forms.

Clearing Checkouts for Assets

Before you begin, find out the name of the user who has the asset locked. This information is listed on the “Inspect” form for the asset.

To clear checkouts for assets

1. In the Content Server interface, select the **Admin** tab and then double-click on **Clear Checkouts**.

The “Search for Checkouts” form appears:



The screenshot shows a web form titled "Search for Checkouts". Inside the form, there is a text input field with the label "*User Name:". Below the input field is a button labeled "Show Checkouts".

2. Enter the name of the user who has the asset locked and click **Show Checkouts**.
3. In the list of assets checked out to that user, select the **Clear** option for each asset for which you want to undo the checkout.
4. Click **Clear Checkouts**.

Unlocking Versions for Non-Asset Tables

To unlock rows for a non-asset table that you are revision tracking, you need to determine the object ID of the row that you want to unlock. You can use Content Server Explorer to examine the table and determine the object row.

When you know the object ID of the item, complete the following steps:

1. In the Content Server interface, select the **Admin** tab and then expand the **Content Server Management Tools** item.
2. Double-click the **Revision Tracking** option.
3. In the “Revision Tracking” form, enter the name of the table, select **Unlock Rows**.
4. In the “Unlock Rows” form, select the option next to the object ID for each row that you want to unlock. You can use the **All** button to check all of the boxes and the **None** button to clear all of the boxes.

5. When you have only the boxes checked for rows that you want to unlock, click **Unlock Records**.

Additional Revision Tracking Functions for Non-Asset Tables

The Content Server interface allows you to change the status of revisions in every way that a user normally does. This allows you to correct inappropriate states by administrative intervention when some error occurs.

The options all appear as radio buttons on the **Revision Tracking** screen:

- Lock
- Commit
- Release
- Rollback
- History
- Track Tables (see “Enabling Revision Tracking,” on page 289)
- Untrack Tables (see “Disabling Revision Tracking,” on page 292, above)
- Set Table Revisions (see “Editing Revision Tracking Settings,” on page 290)
- Delete Revisions (see “Deleting Revisions,” on page 291)
- Unlock Rows (see “Unlocking Revisions,” on page 293)

A summary of each of the other operations appears below.

Lock

The “Lock” form presents a list of rows that is restricted by any criteria that you specified on the “Revision Tracking” form:

- A red lock icon appears next to the rows that are locked by another user.
- A blue lock icon appears next to rows locked by you.

You can lock any asset by clicking the check box in the **Lock?** column and clicking the **Lock** button. The attempt fails if you do not have permission to lock the checked item. If it succeeds, the item is now checked out to you.

Commit

The “Commit” form presents a comment box and a list of rows checked out to you.

You can enter a comment and check the box in the **Commit** column for each item that you want to update. The check box in the **Keep Locked?** column specifies whether you want the item to be unlocked after it is committed (checked in).

The Commit action updates the revision to the version you have been editing.

Release

The “Release” form presents a list of rows that are checked out to you.

You can check the box in the **Release?** column for each item that you want to release. A release differs from a commit in that any changes that you have made to the asset since locking it are lost.

This form differs from the “Unlock Rows” form in that it only lists assets checked out to you.

Rollback

The “Rollback” form presents a list of rows and their versions. The list is restricted by any criteria that you entered on the “Revision Tracking” form.

Note

To roll back an asset to a previous version, use the buttons on the asset’s “Inspect” or “Status” form.

You can select any version for the rollback operation by clicking the radio button in the **Rollback?** column. A rollback creates a new version. It matches the version that you rolled back to but adds the comment “Version created by Rollback.”

History

The “History” form is similar to the “Rollback” form, except that it is informational only. For each row that it shows, it provides the date and revision information.

Chapter 14

Search Engines

The native search feature provided with CS-Direct for searching for assets in the Content Server interface is a SQL-based search that performs database searches for assets. For rich-text searches based on an index, you must use one of the supported search engine modules.

If your organization purchased a search engine module (Verity) to use with your CS systems, it was installed when your CS system was installed.

This chapter describes how to configure your asset types so that your assets are indexed correctly and can be found by your search engine. This chapter contains the following sections:

- Overview
- Configuring Asset Types for Your Search Engine
- Rebuilding an Index

Overview

As the administrator, setting up the indexes for asset types is simple: you select the **Searching** option on the **Admin** tab and select which of your asset types should be enabled.

The Searching Option

The **Searching** option on the **Admin** tab is active if only if a search engine module is installed. If a search engine module has been installed and configured, CS-Direct displays the following message when you select **Admin > Searching**:

Server Status

Content Server:	Configured for search engine
Search Engine:	Verity

 Select an asset type under "Searching" to manage its use of Alta Vista.

Search Engine Properties

Although your search engine module was configured when your CS system was installed, you might need to fine-tune its configuration. There are several search properties in the `futuretense.ini` file and the `futuretense_xcel.ini` file.

For information about these properties, see the *Property Files Reference*.

Asset Types and Search Engines

Different kinds of asset types are indexed differently by the search engine modules.

Default (Core) Asset Types

When you enable any of the core asset types for the **Searching** option, each of the input fields of the asset is indexed. That is, drop-down fields are not indexed.

The core asset types are made available as follows:

- CS-Direct delivers the template, query, collection, link, CSElement, SiteEntry, and page asset types.
- CS-Direct Advantage delivers the attribute editor asset type.
- Engage delivers the visitor attribute, history attribute, history definition, segment, recommendation, and promotion asset types.

AssetMaker Asset Types

Basic asset types that are created with AssetMaker are defined by an asset descriptor file. There are two kinds of fields in an asset type created by AssetMaker:

- Custom fields – these are the fields that are described in the asset descriptor file. For example, custom fields for the ImageFile asset type include Artist, Alt Text, Height, Width, and so on.

- Default fields – in addition to the custom fields specified in the asset descriptor file, AssetMaker asset types have the same default asset type fields that all asset types have: Name, Description, Status, Updated By, and so on.

When you enable an AssetMaker asset type for the **Searching** option, all of the default fields that are not drop-down fields are indexed. (For example, Category and Source are not indexed.) However, to specify which of the custom fields are indexed, your developers must customize certain elements for the asset type.

As the administrator, you specify which asset types can be found by the search engine, but it is the developers who determine which of the custom fields for assets of that type are indexed.

For information about designing basic asset types, see the “Data Design” section in the *Content Server Developer’s Guide*.

Flex and Flex Parent Assets

Flex and flex parent assets also have two kinds of fields:

- Flex attributes – the “custom” fields for a flex asset are flex attributes. Your developers create the flex attributes they need in order to define flex and flex parent assets.
- Default fields – in addition to the flex attributes that hold values for flex and flex parent assets, the asset types have the same default fields that all asset types have: Name, Description, Status, Updated By, and so on.

When you enable a flex or flex parent asset for the **Searching** option, all of their default fields that are not drop-down fields are indexed. However, to specify which of their fields—that is, flex attributes and their values—are indexed, your developers must use the **Search Engine** field on the flex attributes’ “New” and “Edit” forms to specify which attribute values should be indexed for the flex and flex parent assets that use them.

Other Flex Family Members

For all the other members in a flex family—flex attribute, flex definition, flex parent definition, flex filter—when you enable the **Searching** option, all of their fields that are not drop-down fields are indexed.

Location of Indexes

Indexes are stored in two different locations:

- The indexes for flex attribute values of flex assets are stored in the directory specified by the `mwb.searchdir` property in the `gator.ini` file.
- For all other asset types (including flex attributes when they are indexed as assets), indexes are stored in the directory specified by the `xcelerate.sePath` property in the `futuretense_xcel.ini` file.

Configuring Asset Types for Your Search Engine

To specify that assets of a specific type should be enabled for the search engine, you select the asset type and invoke the search engine to build the index for existing assets of that type. From that point on, each asset of that type is indexed when it is saved.

Enabling Asset Types for Search

To enable asset types for search

1. In the Content Server interface, select the **Admin** tab and then expand the **Searching** item.
2. Double-click on the asset type.

Note

Before continuing to the next step, examine the form and verify that this asset type has not been enabled for the search engine yet.

3. In the search form that is displayed, click **Enable Search Engine**.
A message lists the directory where the index for this asset type will be created. You can change the directory name if you want.
4. Click **Enable Search Engine**.

Note

Building the index can take a long time, depending on the number of assets of that type in the database.

5. Repeat steps 2 through 4 for each asset type that you want to enable.

Disabling Asset Types for Search

To disable an asset type

1. In the Content Server interface, select the **Admin** tab and then expand the **Searching** item.
2. Double-click on the asset type.
3. In the search form, click **Disable**.

Searches for assets of this type will now use the native SQL-based database search method rather than the rich-text search from the search engine.

Rebuilding an Index

If the index that the search engine uses for an asset type has been damaged or corrupted, you can rebuild it by completing the following steps:

To rebuild an index

1. In the Content Server interface, select **Admin tab > Searching**.
2. Under **Searching**, double-click the name of the asset type whose index you want to rebuild.

CS-Direct displays the following form:

Manage Search Engine Status for Asset Type: » [Article](#)

Asset Type:	Article
Status:	Alta Vista searching is enabled.
Index:	/local00/vail6/FutureTense/sedb/Article

3. Click the **Rebuild Index** button.
4. In the confirmation message, click **Rebuild Index** again.

The search engine module deletes the existing index and builds a new one. Note that this can take a long time, depending on how many assets exist of this type.

Appendixes

This part contains information about Content Server's system defaults and sample sites. It provides LDAP configuration procedures and describes the system behavior that results when user and site management operations are carried out when the LDAP options are installed.

This part contains the following appendixes:

- Appendix A, "System Defaults"
- Appendix B, "System Data: Content Server Database"
- Appendix C, "Sample Site Configurations"
- Appendix D, "Managing Users, Sites, and Roles in LDAP-Integrated CS Systems"

Appendix A

System Defaults

This appendix lists and describes the Content Server system defaults that you will be using routinely.

This appendix contains the following sections:

- ACLs
- System ACLs
- ACLs of Default Users
- Required ACLs for Custom Users
- System Roles
- System Asset Types
- Default Tree Tabs

ACLs

Content Server 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 Content Server's default system 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.

Table A-1 lists all the permissions that can be specified in an ACL.

Table A-1: Permissions Supported by Content Server

Permission	Bit Mask ¹	Action
Read	1	Read data from a table.
Retrieve	16	Retrieve the contents of a URL column, also known as an upload field. For information about URL columns, see the <i>Content Server Developer's Guide</i>
Write	2	Write information to a table.
Create	4	Create a table.
Delete	8	Delete information from a table.
Revision Tracking Audit	32	Access all the revision tracking information for the rows (records) in a tracked table.
Revision Tracking Admin	64	Assign or remove revision tracking on a table.

1. When an ACL is created, the bit mask numbers for each permission assigned to an ACL are added together and the totals are listed with the ACL in the `SystemACL` table.

Accessing ACLs

Content Server's 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 Content Server interface:
 1. From the **Admin** tab, expand the **Content Server Management Tools** node.
 2. Double-click the **ACLs** node.

3. Select **Modify** and click **OK**.

Caution

Never modify a default system ACL. And never modify the ACLs assigned to any of the system tables.

For descriptions of the system ACLs, see “System ACLs,” on page 308.

System ACLs

Table A-2 lists the system ACLs and their permissions. Each system ACL exists in order to control access to specific parts of the database tables, and subsequently, the product features that use those tables. Although several of the default ACLs have the same set of permissions, the ACLs are all necessary because they are assigned to different tables.

Table A-3 describes the functions of each ACL and how each ACL is used by Content Server and the CS content applications.

Table A-2: System ACLs and Their Permissions

ACL Name	Permissions						
	Read	Retrieve	Write	Create	Delete	Rev. Track Audit	Rev. Track Admin
Browser	x						
ContentEditor	x	x	x	x	x	x	
ElementEditor	x	x	x	x	x	x	
ElementReader	x						
PageEditor	x	x	x	x	x	x	
PageReader	x						
RemoteClient	x	x	x	x	x	x	x
SiteGod	x	x	x	x	x	x	x
TableEditor	x	x	x	x	x	x	
UserEditor	x	x	x	x	x	x	
UserReader	x						
Visitor	x	x	x	x	x	x	
VisitorAdmin	x	x	x	x	x	x	x
WSAdmin							
WSEditor							
WSUser							
xceladmin	x	x	x	x	x	x	x
xceleeditor	x	x	x	x	x	x	
xcelpublish	x	x	x	x	x	x	

Table A-3: System ACLs and Their Descriptions

ACL Name	Description
Browser	<p>Allows read-only access to the content in the Content Server database. It is assigned to most of the system default and sample site users.</p> <p>Content Server requires that all visitors to an online site that it manages have user accounts. For this reason, Content Server is delivered with a default user account, named DefaultReader, that it assigns to all non-authenticated visitors, that is, those who do not have a user account of their own.</p> <p>The Browser ACL is assigned to the DefaultReader user account, which gives non-authenticated visitors read-only access rights to the content in the Content Server database.</p>
ContentEditor	<p>Used for the sample Content Server portal site.</p> <p>This ACL is assigned to the tables that support the sample portal. Anyone who works with the portal sample needs this ACL.</p>
ElementEditor	<p>Allows users to write data to the ElementCatalog and SystemSQL tables.</p> <p>Site designers and anyone who creates templates, CSElement, and SiteEntry assets need this ACL.</p>
ElementReader	<p>Allows users to read data in the ElementCatalog and SystemSQL tables.</p> <p>CS-Direct users need this ACL so they can inspect the templates assigned to their assets.</p>
PageEditor	<p>Allows users to create page entries in the SiteCatalog table.</p> <p>Site designers and anyone who creates a template, CSElement, or SiteEntry asset need this ACL.</p>
PageReader	<p>Allows users to read page entries from the SiteCatalog table.</p> <p>CS-Direct users need this ACL so they can inspect the templates assigned to their assets.</p>
RemoteClient	<p>Grants users the ability to log in to the Content Server management system through a remote client like CS-Desktop.</p> <p>All CS-Desktop users need this ACL.</p>
SiteGod	<p>Enables complete access to all the tables in the Content Server database.</p> <p>At least one user of the management system, typically an administrator, must have the SiteGod ACL.</p>
TableEditor	<p>Allows users to create and delete tables in the Content Server database.</p> <p>Site designers who create database tables or who create new asset types (which causes new tables to be created) need this ACL.</p> <p>Administrators or anyone else who will use the Initialize Mirror Destination feature also needs this ACL.</p>
UserEditor	<p>Allows users to manage user accounts.</p> <p>Administrators need this ACL.</p>

Table A-3: System ACLs and Their Descriptions *(continued)*

ACL Name	Description
UserReader	Allows users to read information about user accounts, but not to change that information. CS-Direct users need this ACL so they can use workflow.
Visitor	Grants the holder of the ACL the ability to write data to the Content Server Engage tables that hold visitor data. Any visitor whose data you are collecting on the delivery system must have this ACL assigned to their user account.
VisitorAdmin	Grants users the ability to create visitor attributes, history attributes, history types, and recommendations. Any Engage user who needs to create any of those asset types needs this ACL.
WSUser	Assigned to <code>SiteCatalog</code> page entries for the Web Services feature. Grants users the ability to access Content Server through the Content Server web services.
WSEditor	Assigned to <code>SiteCatalog</code> page entries for the Web Services feature. Grants users the ability to access Content Server through the Content Server web services.
WSAdmin	Assigned to <code>SiteCatalog</code> page entries for the Web Services feature. Grants users the ability to access Content Server through the Content Server web services.
xceladmin	Grants users the ability to create user profiles, roles, sites, asset types, and so on—that is, to use all the functions on the Admin , Site Admin , and Workflow tabs. System, site, and workflow administrators need this ACL. Also, because the Admin tab has both administrative and site design functions, site designers also need this ACL.
xceleditor	Grants users the ability to log in to the CS content applications. The login request code verifies whether or not a user has the ACL. All users of the management system need this ACL.
xcelpublish	Grants users the ability to view the Publish Console.

ACLs of Default Users

Table A-4: Default Users and Their ACLs

User Name	ACLs	Description
fwadmin	Browser ElementEditor PageEditor PageReader RemoteClient TableEditor UserEditor UserReader Visitor VisitorAdmin xceladmin xceleeditor xcelpublish wsadmin wseditor wsuser	The basic administrator user that CS-Direct creates so that you can begin configuring your CS content applications. Do not delete this user unless another user with an identical ACL assignment already exists.
<i>Content Server</i> (the installation's user account)	Browser ContentEditor ElementEditor ElementReader PageEditor PageReader SiteGod TableEditor UserEditor UserReader	The 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.
DefaultReader	Browser Visitor	The default user name that Content Server assigns to non-authenticated site visitors on the delivery system.

Required ACLs for Custom Users

Table A-5: System ACLs Required by Users

User	Required ACLs
All users	Browser, Element Reader, PageReader, UserReader, xceleeditor
Workflow administrator Site administrator	xceladmin
Administrator	TableEditor, UserEditor, VisitorAdmin (if you have Engage installed), xceladmin
Site designer	ElementEditor, PageEditor, TableEditor, Visitor (if you have Engage installed), Visitor Admin (if you have Engage installed), xceladmin
Engage users	Visitor
CS-Desktop, CS-DocLink, and InSite Editor users	RemoteClient, Visitor (if your installation uses Engage)

System Roles

Table A-6: System Roles

Role	Description
GeneralAdmin	<p>A default system role.</p> <p>Required for users who need access to the Admin tab in the tree.</p> <p>Note that in addition to having the GeneralAdmin role for a site, a user must also have the xceladmin ACL in order to use any of the functions on the Admin tab.</p>
SiteAdmin	<p>A default system role.</p> <p>Required for users who need access to the Site Admin tab, which holds a subset of the features and functions that are on the Admin tab.</p> <p>Use the SiteAdmin role if you want certain users to manage, but not create, the users who can access a site.</p> <p>Note that in addition to having the SiteAdmin role for the site, a user must also have the xceladmin ACL in order to use any of the functions on the Site Admin tab.</p>
WorkflowAdmin	<p>A default system role.</p> <p>Required for users who need access to the Workflow tab in the tree.</p> <p>Note that in addition to having the WorkflowAdmin role for the site, a user must also have the xceladmin ACL in order to use any of the functions on the Workflow tab.</p>

System Asset Types

Table A-7 lists the default asset types. Unlike custom asset types, system asset types cannot be deleted.

Table A-7: System Asset Types

Asset Type	Description
Attribute Editor	The one core asset type in CS-Direct Advantage. An attribute editor specifies how data is entered for a flex attribute when that attribute is displayed on a “New” or “Edit” form for a flex asset or a flex parent asset. It is similar to a template asset. However, unlike a template asset, you use it to identify the code that you want CS-Direct Advantage to use when it displays an attribute in the Content Server interface—not when it displays the value of an attribute on your online site.
CSElement	Stores code (XML or JSP and Java) does not render assets. Typically, you use CSElements for common code that you want to call from more than one template (a banner perhaps). You also use CSElements to provide the queries that are needed to create DynamicList recommendations in Engage. Available in CS-Direct.
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. Available in CS-Direct.
History Attribute	Individual information types that you group together to create a vector of information that Engage treats as a single record. This vector of data is the history definition. For example, a history type called “purchases” can consist of the history attributes “SKU,” “itemname,” “quantity,” and “price.” Available in Engage.
History Definition	The vector of data in a History Attribute. This vector of data is the history definition. For example, a history type called “purchases” can consist of the history attributes “SKU,” “itemname,” “quantity,” and “price.” Available in Engage.
Link	Stores a URL to an external web site. You use this asset to embed an external link within another asset. Available in CS-Direct.
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 Content Server interface. Available in CS-Direct.
Promotion	Is a merchandising asset that offers some type of value or discount to your site visitors based on the flex assets (products, perhaps) that the visitor is buying and the segments that the visitor qualifies for. Available in Engage.

Table A-7: System Asset Types *(continued)*

Asset Type	Description
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 Content Server query tags or as a search engine query) that is identified in the “New” or “Edit” form. Available in CS-Direct.
Recommendation	This is like an advanced collection. It collects, assesses, and sorts flex assets (products or articles, perhaps) and then recommends the most appropriate ones for the current visitor, based on the segments that visitor belongs to. Available in Engage.
Segment	Assets that divide visitors into groups based on common characteristics (visitor attributes and history types). You build segments by determining which visitor data assets to base them on and then setting qualifying values for those criteria. For example, a segment could define people who live in Alaska and own fly fishing gear, or it could define people who bought a personal computer in the past six months, and so on. Available in Engage.
SiteEntry	Represents a Content Server 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 Content Server page. Available in CS-Direct.
Template	Stores code (XML or JSP and Java) that renders other assets into Content Server 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. Available in CS-Direct.
Visitor Attribute	Holds types of information that specify one characteristic only (scalar values). For example, you can create visitor attributes named “years of experience,” “job title,” or “number of children.” Available in Engage.

Default Tree Tabs

Table A-8 lists the default tabs in Content Server's tree. These tabs are critical to Content Server. All features which stem from Content Server can be accessed through these tabs; they are automatically created upon installation.

Table A-8: Default Tabs in Content Server

Tab	Description
Active List	The Active List is a tab which displays items that are in the process of being created or edited in Content Server.
Admin	Displays the administrative functions that affect all of the CM sites in the system. By default, only users with the default system role named GeneralAdmin have access to this tab.
Design	The design tab is a source for creating pages on your site. Some of these sources are: Templates, Product Definition, Content Definition, and other sources for the creating pages.
History	Displays the assets that you worked with during the current session. All users see this tab as soon as they create, inspect, edit, or copy their first asset.
Query	This tab is the means for which a user can query for certain types of articles and organize them in that fashion. This is accomplished using a SQL query.
Site Admin	Holds a subset of the administrative functions that apply only to the CM 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 CM sites, but who do not need to create new users or new sites.
Site Plan	A layout and overview of the site. This tab will show each site controlled by Content Server. It lists the "placed pages" and the "unplaced pages." The "placed pages" are pages which are created and have been integrated to the site. "Unplaced Pages" are pages which are finished but are not integrated into the live site.
Workflow	Has the workflow configuration functions. By default, only users with the Workflow Admin role have access to this tab.

Appendix B

System Data: Content Server Database

This appendix contains information about the dynamic tables in the Content Server database and how they grow. DBAs can use this information to determine how to size the Content Server database appropriately.

This appendix contains the following sections:

- Cache Management Tables (Content Server)
- Approval System Tables (CS-Direct)
- Publishing System Tables (CS-Direct)
- Workflow Tables (CS-Direct)
- Flex Asset Tables
- Visitor Tables (Engage)

This appendix also contains information about how to purge inactive data from the visitor tables.

Cache Management Tables (Content Server)

Content Server delivers the CacheManager, a page caching utility that manages both the Content Server page cache and the Satellite Server caches.

Because cached pages need to expire both when their freshness date expires and when an asset that the page refers to in some way is changed, the CacheManager keeps track of expiration times as well as the dependencies that exist between the pages and pagelets stored in the cache. It stores this information in the `SystemPageCache` and `SystemItemCache` tables.

Every CS system uses a CacheManager, which means that these tables grow dynamically on any system—development, management, or delivery. The cache-tracking tables grow at the following rate:

Table	Number of Rows
<code>SystemPageCache</code>	One row for every cached page. When a page expires, the row is removed.
<code>SystemItemCache</code>	One row for each asset, pagelet, or other item that is referenced by a cached page in the <code>SystemPageCache</code> table. For example, if a cached page was created from a page asset that has associations to three article assets, there would be four rows for that cached page.

Approval System Tables (CS-Direct)

The CS-Direct approval system keeps track of each asset that has been approved, the dependencies that approved assets have on other assets, and the targets for which assets are approved. It stores this information in the `ApprovedAssets` and `ApprovedAssetDeps` tables.

These tables have the potential to grow very large on a management system, but are not used on a delivery system. The approval system tables grow at the following rate:

Table	Number of Rows
<code>ApprovedAssets</code>	One row for each asset that has been approved, for each target destination. That is, if an asset has been approved for two destinations, that asset has two rows in this table.
<code>ApprovedAssetDeps</code>	One row for each asset dependency for each approved asset, for each target destination. For example, if an approved asset is dependent on four other assets, it has four rows in this table. If that same asset is approved to two destinations, it has one row each for each dependency for each destination.

Publishing System Tables (CS-Direct)

The CS-Direct 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 CS management system increases, so does the number of rows in these tables. These tables grow at the following rate:

Table	Number of Rows
<code>PubKey</code>	<ul style="list-style-type: none"> For mirror publishing: one row for every asset that is mirrored, for each target destination. For export publishing: one row for each page that is created during the export. That is, if 14 assets are rendered into 1 page, the table holds 1 row—not 14—for the entire group of assets.
<code>PublishedAssets</code>	<ul style="list-style-type: none"> For mirror publishing: one row for every asset that is mirrored, for each target destination. For export publishing: one row for each asset that is exported. To continue the preceding example, if 14 assets are rendered onto 1 page, the table holds 14 rows (one for each asset) for that page.

Workflow Tables (CS-Direct)

The workflow system keeps track of all the assets that are involved in a workflow process at any given time. It stores this information in the `Assignment` and `WorkflowObject` tables.

As the number of assets that are placed in a workflow process increases, so does the number of rows in these tables. These tables grow at the following rate:

Table	Number of Rows
<code>Assignment</code>	<p>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.</p> <p>These rows are not deleted when the workflow process is completed for the asset.</p>
<code>WorkflowObjects</code>	<p>One row for each asset in workflow.</p> <p>When an asset leaves workflow, the row is deleted.</p>

Basic Asset Tables (CS-Direct)

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 GE Lighting sample site asset type named `product` is `Products`.
- The `_AMap` tables for flex asset or flex parent asset types. (For example, `Products_AMap`.)
- The `_Group` tables for flex parent asset types. (For example, `ProductGroups_Group`.)
- The `_Mungo` table for the **flex asset** type. (For example, `Products_Mungo`.)
- The `_Mungo` table for the flex **parent** asset type. (For example, `ProductGroups_Mungo`.)
- The `Mungo_Blobs` table

These types of tables grow at the following rate:

Table	Number of Rows
<i>FlexAssetType</i> (for example: <code>Products</code>)	One row for every asset of this type.
<i>FlexAssetType_AMap</i> (for example, <code>Products_AMap</code>)	One row per attribute value—whether the attribute value is inherited or directly assigned—for each of the assets of that type.
<i>FlexAssetType_Group</i> (for example, <code>ProductGroups_Group</code>)	One row per ancestor relationship between flex parent asset and flex asset—includes rows for grandparent, great-grandparent, and so on, relationships.
<i>FlexAssetType_Mungo</i> (for example: <code>Products_Mungo</code>)	One row for every attribute value for each asset of this type. Note that for one FatWire customer, this equation resulted in 2.4 million rows and for another, this equation resulted in over 10 million rows.
<i>FlexParent_Mungo</i> (for example: <code>ProductGroups_Mungo</code>)	One row for every attribute value for every parent asset of this type.

Table	Number of Rows
Mungo_Blobs	One row for every attribute value saved for an attribute of type blob.

Visitor Tables (Engage)

Engage captures visitor information and stores it in the visitor data tables. These tables store information such as session IDs for visitors so that they can be linked with their previous sessions and values for the attributes that represent the data you are collecting.

As the number of visitors who visit your online site increases, so do the rows in these tables. These tables grow at the following rate:

Table	Number of Rows
scratch	One row for each visitor context session object that is created for a visitor. Visitor context session objects are things like promotion lists, segment lists, shopping carts, and so on. There are at least five rows added to this table for each visitor in each session.
VMVISITOR	One row for each visitor for each browser session. Engage creates a unique visitor ID for each visitor for each session.
VMVISITORALIAS	Most likely, one row for each visitor for each browser session. A row holds the name/value pair of an alias and the visitor ID (also listed in VMVISITOR) that marks a session.
VMVISITORSCALARVALUE	One row for each visitor attribute value (except for attributes of type binary) that is saved.
VMVISTORSCALARBLOB	One row for each visitor attribute value of type binary (also referred to as scalar objects) that is saved.
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.

Managing the Attribute Tables

Because the tables that hold attribute values can grow very quickly, you should purge inactive data from them regularly.

You use the following Engage XML object method or its JSP equivalent to delete inactive data from these tables:

```
<VDM.FLUSHINACTIVE STARTDATE="cutoffDate" />
```

Inactive visitor data is data marked with a visitor ID that is not connected through an alias to data that you consider current. You set a cutoff date (STARTDATE) for Engage to use. All visitor data recorded before that date is deleted from the previously listed visitor tables **unless** it is linked through an alias with data recorded after that date.

There are several ways to use the VDM.FLUSHINACTIVE tag. For example:

- You can create an administrative element that invokes the tag and prompts you to enter the cutoff date.
- You can provide it with an equation that calculates a cutoff date based on some parameter so that you can set it up to run as an automatic event at a regularly scheduled time. To set it up as an automatic event, use the Content Server APPEVENT tag (which functions like a `kron` job). For information about this tag, see the *Content Server Developer's Tag Reference*, and the "Coding Basics" section of the *Content Server Developer's Guide*.

Note that the value that you pass to the STARTDATE parameter must be in epoch time. You can use the Content Server DATE.CONVERT tag to obtain an epoch value for the date that you want to use. For example:

```
<DATE.CONVERT VARNAME="flushtime"
  YEAR="four digit year" MONTH="number in the range 1-12"
  DAY="number in the range 1-31"
  [HOURL="number in the range 0-11 where 0 is midnight" AMPM="am
    or pm" MINUTE="number in the range 0-59" TIMEZONE="timezone"]/>
<VDM.FLUSHINACTIVE STARTDATE="Variables.flushtime" />
```

For more information about these tags, see the *Content Server Developer's Tag Reference*.

Managing the Session Objects Table (scratch)

The `scratch` table can grow quickly because there are at least five session objects stored for each visitor in each session. Each object has a timestamp; you should purge old objects regularly, based on their timestamps.

You use the following Engage XML object method or its JSP equivalent to delete old session objects from the `scratch` table:

```
<SESSIONOBJECTS.FLUSH TIMESTAMP="cutoffTime" />
```

Because "session objects" includes carts, you must set the cutoff time to represent the point at which you consider a cart to be abandoned.

Note

This object method does not affect the VMSCALARBLOB table, which means that it does not delete carts that you have stored as a visitor attribute of type binary (a scalar object).

Note that the value that you pass to the STARTDATE parameter must be in epoch time. You can use the Content Server DATE.CONVERT tag to obtain an epoch value for the date that you want to use. For an example, see the code example for the VDM.FLUSHINACTIVE tag provided in the preceding section.

For more information about these tags, see the *Content Server Developer's Tag Reference*.

Deleting Unnecessary .class Files

Engage is a Java-based application and it generates Java .class files each time one of the following events occurs:

- A segment, recommendation, or promotion is created.
- A product is configured for a related items recommendation.
- A segment, recommendation, or promotion is calculated or invoked by Engage.

Typically, old .class files are deleted when a segment, recommendation, promotion, or product is updated and are then replaced with new .class files. However, if the segment, recommendation, product, or promotion is in use when an updated version is published, Engage cannot delete the old .class file because it is locked.

The old .class files can build up, filling up the disk and using memory. Therefore, depending on how much development work you are doing and how frequently you publish to the delivery system, you must manually delete these .class files at a regularly scheduled time.

Complete the following steps to delete old .class files:

1. Use the Property Editor to examine the `vis.genclasspath` property and note the directory name designated by that parameter. This is the directory where Engage stores .class files.
2. During a quiet time on your site, shut down and restart each instance of JRE runtime that is running. This process releases any old .class files that the JRE runtime has locked.
3. Using any file management tool, navigate to the directory that holds the .class files and delete the contents of this directory.

Engage regenerates any .class files that it needs when it needs them.

Appendix C

Sample Site Configurations

Content Server provides several sample sites that demonstrate the two sides of an online site: the online site itself, and the content management site on the management system, which the content providers interact with in order to supply content for the online site.

The sample sites vary in complexity and purpose:

- **Hello Asset World** is an extremely simple site that illustrates the primary coding and design concepts behind extracting and displaying basic assets.
- **Burlington Financial** is a complete, fully functioning financial services portal with a rich set of features. It uses the basic asset data model and illustrates most of the abilities of the CS-Direct product.
- **GE Lighting** is an online catalog of lighting products. It illustrates the power and flexibility of the flex asset model. When Engage is installed, GE Lighting also uses some of the Engage features.
- **Burlington Financial Extension** provides an extension to the Burlington Financial site that illustrates the Engage features, and adds two families of flex asset types to the data model.
- **CO** is a source site for demonstrating Site Launcher. The CO site exemplifies the type of site that is suitable for replication. The replicated site can then be modified and spun off as a new site.
- **Spark** is a sample site that illustrates the content providers' portal environment.

Note

For detailed information about the Spark site and the business users' portal environment, see the *Content Server User's Guide*, where the Spark sample site is used throughout to demonstrate content management procedures. (Note that while administrative portlets are packaged with the Spark sample site, their purpose is to help CS administrators manage the Spark sample site, if it is installed.)

In addition to providing code samples for your site designers, all the sample sites serve as examples of how to configure sites that run on a management system.

This appendix describes the configuration of the sample sites in the following sections:

- Sample Users and Their ACLs
- Sample Roles
- Burlington Financial
- GE Lighting
- Burlington Financial Extension (for Engage)
- CO

Sample Users and Their ACLs

Table C-1: Sample Site Users and their ACLs

User Name	ACLs	Description
Bobo	TableEditor, Visitor, VisitorAdmin, UserEditor, UserReader, xceladmin, Browser, xceleeditor, PageEditor, ElementEditor, RemoteClient, ElementReader	The administrator of the Hello Asset World sample site.
Coco	PageEditor, ElementEditor, Visitor, VisitorAdmin, TableEditor, UserReader, Browser, xceleeditor, RemoteClient	The developer/designer for the Hello Asset World sample site.
editor	PageEditor, ElementEditor, Visitor, UserReader, Browser, xceleeditor	A sample site user for the Burlington Financial sample site.
Flo	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	The editor for the Hello Asset World sample site.
Joe	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	One of the writers for the Hello Asset World sample site.
mirroruser	Browser, ElementEditor, PageEditor, TableEditor, UserReader, Visitor, VisitorAdmin, xceladmin, xceleeditor	An example mirror user that is installed for the sample sites so that the sample site assets can be published with the Mirror to Server delivery method.
Moe	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	One of the writers for the Hello Asset World sample site.
user_analyst	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	A sample site data analyst user who creates segments. This user is installed for the Engage extensions to the Burlington Financial sample site.
user_approver	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	Another user for the Burlington Financial and GE sample sites. This user has the approver role, which means this user has the ability to approve assets for publishing.

Table C-1: Sample Site Users and their ACLs *(continued)*

User Name	ACLs	Description
user_author	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	An author user for the Burlington Financial and GE sample sites.
user_checker	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	A user who does fact checking for the Burlington Financial sample site.
user_designer	PageEditor, ElementEditor, Visitor, VisitorAdmin, TableEditor, UserReader, Browser, xceleeditor, RemoteClient	A site designer user for the sample sites (Burlington Financial and GE Lighting).
user_editor	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	An editor user for the Burlington Financial and GE sample sites.
user_expert	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	A sample site user who is the supervisor of both user_marketer and user_analyst. This user is installed for the Engage extensions to the Burlington Financial sample site.
user_marketer	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	A sample site user who works on segments and funds. This user is installed for the Engage extensions to the Burlington Financial sample site.
user_pricer	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, RemoteClient	A sample site user who prices the products in the GE sample site.

Sample Roles

Table C-2: Default Sample Site Roles

Role	Description
Analyst	<p>Used by the Burlington Financial with Engage extensions sample site in the Segment workflow process to designate users who are allowed to create segments.</p> <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Approver	<p>Used by the Burlington Financial and GE Lighting sample sites in their workflow processes to designate the users who are allowed to approve assets for publishing.</p> <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Author	<p>Allows users to do the following:</p> <ul style="list-style-type: none"> • Burlington Financial – write articles as part of the Normal Article Process workflow. • GE Lighting – write flex articles, create products, and place products in workflow to obtain a price as part of the GE Lighting workflow process. <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Checker	<p>Allows users to do the following:</p> <ul style="list-style-type: none"> • Burlington Financial – perform fact checking for articles as part of the Normal Article Process workflow. • Burlington Financial with Engage extensions – verify funds for the Fund Process and segments for the Segments Process. • GE Lighting – verify the prices set for products in the GE Lighting Process. <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Designer	<p>Allows users to create site design assets by giving them access to the Design tab, a custom tab created for the Burlington Financial and GE Lighting sample sites, and to design and configure asset types by giving them access to the Admin tab.</p> <p>The designer role does not participate in any of the sample workflow processes.</p> <p>Grants access to the following tree tabs in the Burlington Financial and GE Lighting sample sites: Active List, Admin, Content, Design, Marketing, Product, Site Plan.</p>

Table C-2: Default Sample Site Roles *(continued)*

Role	Description
Editor	<p>Allows users to do the following:</p> <ul style="list-style-type: none"> • Burlington Financial – edit articles. • GE Lighting – review products and product prices. • Burlington Financial with Engage extensions—review funds and segments. <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Expert	<p>Used by the Burlington Financial with Engage extensions sample site in the Segment workflow process to designate the users who are allowed to create as well as approve segments.</p> <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
HelloAuthor	<p>Used by the Hello Asset World sample site to designate the users who are allowed to create HelloArticle assets (that is, write articles) and place them in workflow.</p> <p>Grants access to the following tabs: Active List, HelloContent, Site Plan.</p>
HelloDesigner	<p>Allows users to create site design assets by giving them access to the HelloDesign tab, a custom tab created for the Hello Asset World sample site, and to design and configure asset types by giving them access to the Admin tab.</p> <p>The HelloDesigner role does not participate in the sample workflow process for the Hello Asset World site.</p> <p>Grants access to the following tabs: Active List, Admin HelloContent, HelloDesign, Site Plan.</p>
HelloEditor	<p>Used by the Hello Asset World sample site to designate the users who edit the HelloArticles placed in workflow.</p> <p>Grants access to the following tabs: Active List, HelloContent, HelloDesign, Site Plan.</p>
Marketer	<p>Used by the Burlington Financial with Engage extensions sample site in both the workflow processes. Users with this role are allowed to create segments, create funds, and review funds.</p> <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>
Pricer	<p>Used by the GE Lighting sample site in the GE Lighting workflow process to designate the users who are allowed to price products.</p> <p>Grants access to the following tree tabs in the sample sites: Active List, Content, Marketing, Product, Site Plan.</p>

Hello Asset World

The Hello Asset World site is the simplest of the four sample sites. It has five users, six roles, two custom tabs, and one workflow process. For asset types, it uses only four of the system asset types and has only two custom asset types. The example users of this sample site provide articles and images for an online webzine called “Hello World.”

For a description of the site design of the Hello World webzine, see the *Content Server Developer's Guide*.

Users, ACLS, and Roles for Hello Asset World

The Hello Asset World site has five users:

- Coco, the site designer
- Bobo, the site administrator
- Flo, the editor
- Joe, an author
- Moe, another author

There are no custom ACLS—the user accounts are assigned only the system default ACLs.

Hello Asset World users are assigned one or more of the following five roles:

- HelloAuthor
- HelloDesigner
- HelloEditor
- GeneralAdmin
- WorkflowAdmin

The GeneralAdmin and WorkflowAdmin roles are system default roles, while the other three are custom roles for the users of the Hello Asset World site.

The following tables shows how the Hello Asset World user accounts and role assignments are configured:

Username	Password	ACLs	Roles
Coco	hello	Browser, ElementEditor, ElementReader, PageEditor, PageReader, TableEditor, UserReader, xceladmin, xceleeditor	HelloDesigner, GeneralAdmin (a system default role)
Bobo	hello	Browser, ElementReader, PageReader, UserEditor, UserReader, xceladmin, xceleeditor	GeneralAdmin, WorkflowAdmin (another system default role)
Flo	hello	Browser, ElementReader, PageReader, UserReader, xceleeditor	HelloEditor

Username	Password	ACLs	Roles
Joe	hello	Browser, ElementReader, PageReader, UserReader, xceleditor	HelloAuthor
Moe	hello	Browser, ElementReader, PageReader, UserReader, xceleditor	HelloAuthor

Asset Type Configuration for Hello Asset World

Asset type configuration for Hello Asset World is also quite simple. The site uses only four of the standard system default asset types—but adds an association to the page asset type—and provides two custom asset types (built with the AssetMaker feature).

These are the asset types that are used in Hello Asset World:

- HelloArticle (custom). It has one association field named Associated HelloImage. Moe and Joe, the Hello Asset World authors, create assets of this type. There are seven HelloArticle assets in the site.
- HelloImage (custom). Moe and Joe also create the HelloImages that illustrate their HelloArticles. There are eight HelloImage assets in the site.

Note

For information about the data structure of these asset types, use Content Server Explorer to examine their asset descriptor files and the HelloArticle and HelloImage database tables.

The Content Server Explorer tool is described in the *Content Server Developer's Guide*.

- Template (default). There are three template assets: one for pages, one for HelloArticles, and one for collections.
- Page (default). The Hello Asset World site adds an association field named HelloAssetWorld Collection to the page asset type. The site has one page asset.
- Collection (default). The site has one collection asset.
- Query (default). The site has one query asset.

Source and Category

Hello Asset World does not use categories but it does add one new source item. This source is also named HelloAssetWorld.

Start Menu Configuration for Hello Asset World

Even though Hello Asset World uses only 6 asset types, it has 16 start menu items. In addition to the New and the Search start menu items for the six asset types that are in use, there are New and Search start menu items for the CSElement and SiteEntry asset types in case Coco the site designer ever decides to use them.

The start menu items are also very simple: only one of them provides a default value for an asset created with it. The following table describes all the start menu items for Hello Asset World.

Name	Asset Type	Start Menu Type	Roles	Default Values
New CSElement, HelloAssetWorld	CSElement	New	HelloDesigner	
Find CSElement, HelloAssetWorld	CSElement	Search	HelloDesigner	
New Collection, HelloAssetWorld	collection	New	HelloEditor, HelloDesigner	
Find Collection, HelloAssetWorld	collection	Search	HelloEditor, HelloDesigner	
New HelloArticle	HelloArticle	New	HelloAuthor	workflow = HelloWorkflow
New HelloArticle for Testing	HelloArticle	New	HelloDesigner, HelloEditor	
Find HelloArticle	HelloArticle	Search	HelloAuthor, HelloEditor, HelloDesigner	
New HelloImage	HelloImage	New	HelloAuthor, HelloEditor, HelloDesigner	
Find HelloImage	HelloImage	Search	HelloAuthor, HelloEditor, HelloDesigner	
New Page, HelloAssetWorld	page	New	HelloDesigner	
Find Page, HelloAssetWorld	page	Search	HelloDesigner	
New Query, HelloAssetWorld	query	New	HelloDesigner	
Find Query, HelloAssetWorld	query	Search	HelloDesigner	
New SiteEntry, HelloAssetWorld	SiteEntry	New	HelloDesigner	
Find SiteEntry, HelloAssetWorld	SiteEntry	Search	HelloDesigner	
New Template, HelloAssetWorld	template	New	HelloDesigner	
Find Template, HelloAssetWorld	template	Search	HelloDesigner	

Tree Tab Configuration for Hello Asset World

Hello Asset World adds two custom tabs to the tree and adds the Hello Asset World roles for its users to the system default tabs.

Tab	Description	Roles	Users with Access
Admin (default)	Provides access to the administrative features	GeneralAdmin	Bobo, Coco
Active List (default)	Provides access to the user's active list	HelloAuthor, HelloDesigner, HelloEditor	Coco, Flo, Joe, Moe
Hello Content (custom)	Provides access to the HelloArticle and HelloImage assets	HelloAuthor, HelloEditor, HelloDesigner	Coco, Flo, Joe, Moe
Hello Design (custom)	Provides access to the template, CSElement, SiteEntry, page, collection, and query assets	HelloDesigner, HelloEditor	Coco
History (default)	Displays the assets that the user worked with during the current session.	All	All
Site Plan (default)	Displays a graphical representation of the online site	HelloAuthor, HelloDesigner, HelloEditor	Coco, Flo, Joe, Moe
Workflow (default)	Provides access to the workflow states and processes	WorkflowAdmin	Bobo

Workflow for Hello Asset World

Hello Asset World provides a sample workflow process named HelloArticle Process.

It has two states and four steps. For the most part, this workflow process was built with the default workflow building blocks. However, one additional e-mail object and one additional step action were created for this workflow process.

For a complete description of the Hello Asset World workflow process, see “HelloArticle Process,” on page 153.

Burlington Financial

The Burlington Financial sample site Burlington Financial is a fictitious financial news portal. As is the design of the online site, the configuration of the management system site is also more complex than that of Hello Asset World.

This site has seven users, eight roles, and one workflow process. For asset types, it uses all of the system default types and provides three custom asset types.

The example users of this sample site provide articles and images for an online financial news site and service called “Burlington Financial.” For a description of the site design of the Burlington Financial news site, see the *Content Server Developer’s Guide*.

Users, ACLS, and Roles for Burlington Financial

The Burlington Financial site has six users:

- user_approver, a manager who approves assets so that they can be published
- user_author, an author
- user_checker, a fact-checker
- user_designer, the site designer
- user_editor, an editor
- fwadmin, the site and system administrator (note that this is also a system default user)

There are no custom ACLS—the user accounts are assigned only the system default ACLs.

Burlington Financial users are assigned one or more of the following eight roles:

- Approver
- Author
- Checker
- Designer
- Editor
- GeneralAdmin
- SiteAdmin
- WorkflowAdmin

The GeneralAdmin, WorkflowAdmin, and SiteAdmin roles are system default roles, while the other five are custom roles for the users of the Burlington Financial site.

The following table shows how the Burlington Financial user accounts and role assignments are configured:

Username	Password	ACLS	Roles
user_author	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, xceleeditor	Author
user_approver	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, xceleeditor	Approver

Username	Password	ACLs	Roles
user_checker	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, xceleditor	Checker
user_designer	user	Browser, ElementEditor, PageEditor, RemoteClient, TableEditor, UserReader, xceleditor	Designer
user_editor	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, xceleditor	Editor
fwadmin	xceladmin	Browser, ElementEditor, TableEditor, PageEditor, RemoteClient, UserEditor, UserReader, xceladmin, xceleditor	GeneralAdmin, SiteAdmin, WorkflowAdmin

Asset Type Configuration for Burlington Financial

Asset type configuration for Burlington Financial is also more complex than that of Hello Asset World. The site uses all of the standard system default asset types, adds association fields to the page, collection, and query asset types, and provides two custom asset types built with the AssetMaker feature and one completely custom asset type built without AssetMaker (the article asset type). It also makes extensive use of categories and sources, and, for the article asset type, subtypes.

Note

For information about the data structure of the custom asset types, see the chapter on creating basic asset types in the *Content Server Developer's Guide* and then use Content Server Explorer to examine their asset descriptor files and their database tables.

The following table describes the asset type configuration for the Burlington Financial sample site:

Asset Type	Number of Assets	Association Fields	Subtype	Categories
Article (custom, non-AssetMaker type)	540	Main ImageFile, Popular Topics, Related Stories, Sidebar Bottom, Sidebar Top, Spot ImageFile, Spot ImageFile1, Teaser ImageFile, French Translation, German Translation, Spanish Translation	Standard, Columnist	Banking and Loans, Bonds, Companies, Consumer, Currency, Economy, Entertainment, French, Funds, General, German, Investing, Markets, News, Personal Finance, Portfolio, Rates and Bonds, Retirement, Services, Small Business, Spanish, Special, Sports, Stocks, Tax Guide, Technology, US Treasuries, World Business
CSElement (default)	42	none	none	none
Collection (default)	29	Query1, Query2, Query3	none	General, Money
ImageFile (custom)	105	none	none	General
Page (default)	21	HomeStoryGroups, PopularTopics, Recent Stories, Section Highlight, Sidebar Bottom, Sidebar Middle, Sidebar Top, Top Stories, WireFeed	none	Account Page, Content Page, Home, List Page, Section Front
Query (default)	28	none	none	Entertainment, General
SiteEntry (default)	9	none	none	none
StyleSheet (custom)	4	none	none	CSS HTML 4.0, XSL, General
Template(default)	47	none	none	Banner, Entertainment, General, Navigation, Page Template, Subpage Template

Sources

Burlington Financial uses the source feature to identify the news agency that provides an article when the article is not written by a Burlington Financial author. It adds the following 12 sources to the `Source` table:

- 123Jump
- AsiaPulse

- BurlingtonFinancial
- Efe
- FWNSelect
- ItarTass
- M2Comm
- NYPost
- TechWeb
- UPI
- WireFeed
- Xinhua

Start Menu Configuration for Burlington Financial

Burlington Financial provides 18 start menu items.

The start menu items are very simple: only one of the **New** items provides default values for an asset created with it and none of the **Search** items provide a default value to search by. The following table describes all the start menu items for Burlington Financial.

Name	Asset Type	Start Menu Type	Roles	Default Values
New Article	article	New	Approver, Author, Checker, Designer, Editor	
Find Article	article	Search	Approver, Author, Checker, Designer, Editor	
New CSElement	CSElement	New	Designer	
Find CSElement	CSElement	Search	Designer	
New Collection	collection	New	approver, author, checker, designer, editor	
Find Collection	collection	Search	Approver, Author, Checker, Designer, Editor	

Name	Asset Type	Start Menu Type	Roles	Default Values
New Columnist Article	article	New	Approver, Author, Checker, Designer, Editor	subtype = Columnist template = Columnist source = Burlington Financial
New ImageFile	image file	New	Approver, Author, Checker, Designer, Editor	
Search ImageFile	image file	Search	Approver, Author, Checker, Designer, Editor	
New Page	page	New	Designer	
Find Page	page	Search	Designer	
New Query	query	New	Designer	
Find Query	query	Search	Designer	
New SiteEntry	SiteEntry	New	Designer	
Find SiteEntry	SiteEntry	Search	Designer	
New Stylesheet	stylesheet	New	Author, Designer	
Find Stylesheet	stylesheet	Search	Author, Designer	
New Template	template	New	Designer	
Find Template	template	Search	Designer	

Tree Tab Configuration for Burlington Financial

Burlington Financial adds no custom tabs to the tree but it does add its roles to the system default tabs.

Tab	Description	Roles	Users with Access
Active List (default)	Provides access to the user's active list	Approver, Author, Checker, Designer, Editor	user_approver, user_author, user_checker, user_designer, user_editor
Admin (default)	Provides access to the administrative features	GeneralAdmin	fwadmin
History (default)	Displays the assets that the user worked with during the current session.	All	All
SiteAdmin (default)	Provides access to site administration functions	SiteAdmin	fwadmin
Site Plan (default)	Displays a graphical representation of the online site	Approver, Author, Checker, Designer, Editor	user_approver, user_author, user_checker, user_designer, user_editor
Workflow Admin (default)	Provides access to the workflow states and processes	WorkflowAdmin	fwadmin

Workflow for Burlington Financial

Burlington Financial provides a sample workflow process named Normal Article Process.

It has three states and six steps and uses four roles (author, approver, checker, and editor).

This workflow process was built with the default workflow building blocks only.

For a complete description of the this workflow process, see "BF: Normal Article Process," on page 154.

GE Lighting

The GE Lighting sample site is a complete example of a full-featured online catalog of lighting products. It uses the flex asset data model.

This site has 9 users, 10 roles, and 1 workflow process. For asset types, it uses all of the system default types and provides two flex families: the content family and the product family.

The example users of this sample site create products and articles about those products. For a description of the site design of the GE Lighting catalog, see the *Content Server Developer's Guide*.

Users, ACLS, and Roles for GE Lighting

The GE Lighting site has eight users:

- user_approver, a manager who approves assets so that they can be published
- user_author, an author
- user_checker, a fact checker
- user_designer, the site designer
- user_editor, an editor
- user_marketer, a marketing specialist
- user_pricer, a product specialist who prices the products in the catalog
- fwadmin, the site and system administrator (a system default user)
- editor, an editor who is allowed to perform all the roles on the site except the administrative roles (another system default user)

There are no custom ACLS—the user accounts are assigned only the system default ACLs.

GE Lighting users are assigned one or more of the following eight roles:

- Approver
- Author
- Checker
- Editor
- Designer
- GeneralAdmin
- Marketer
- Pricer
- SiteAdmin
- WorkflowAdmin

The GeneralAdmin, WorkflowAdmin, and SiteAdmin roles are system default roles, while the others are custom roles for the users of the GE Lighting sample site.

The following table shows how the GE Lighting user accounts and role assignments are configured:

Username	Password	ACLs	Roles
user_author	user	Browser, ElementReader, PageReader, UserReader, xceleditor	Author
user_approver	user	Browser, ElementReader, PageReader, UserReader, xceleditor	Approver
user_checker	user	Browser, ElementReader, PageReader, UserReader, xceleditor	Checker
user_designer	user	PageEditor, ElementEditor, Visitor, VistorAdmin, TableEditor, UserReader, Browser, xceleditor, RemoteClient	Designer
user_editor	user	Browser, ElementReader, PageReader, UserReader, xceleditor	Editor
user_marketer	user	Browser, ElementReader, PageReader, UserReader, xceleditor‘	Marketer, Designer
user_pricer	user	Browser, ElementReader, PageReader, UserReader, xceleditor	Pricer
fwadmin	xceladmin	TableEditor, UserEditor, UserReader, xceladmin, Browser, xceleditor, PageEditor, ElementEditor, RemoteClient	Designer, GeneralAdmin, SiteAdmin, WorkflowAdmin
editor	xceleditor	Browser, ElementReader, PageReader, UserReader, xceleditor	Approver, Author, Checker, Designer, Editor, Pricer, Marketer

Asset Type Configuration for GE Lighting

Asset type configuration for GE Lighting is much different than that of either Hello Asset World or Burlington Financial. This site uses the flex asset data model and provides two example flex families: the content family and the product family.

Because the GE Lighting site uses flex families for its content, it does not use associations or categories. It also does not use sources.

The content family includes the following asset types:

- content attribute
- content parent definition
- content parent
- content definition
- article (flex)
- image (flex)

There are four content attribute assets, one content definition asset, and three flex article assets.

The product family includes the following asset types:

- product attribute
- product parent definition
- product parent
- product definition
- product

There are 35 product attribute assets, 3 product parent definition assets, 31 product parent assets, 2 product definition assets, and 101 products.

Note

For information about the data structure of these asset types, see the chapter on creating flex asset types in the *Content Server Developer's Guide* and then use Content Server Explorer to examine the database tables for these asset types.

The following asset types are also enabled for use in the GE Lighting sample site

- attribute editor
- page
- template

There 2 attribute editors, 10 page assets, and 15 template assets.

If Engage is installed on your system, the GE Lighting sample site also provides 1 history attribute, 1 history definition, 2 visitor attributes, 1 segment, 1 promotion, and 3 recommendations.

Start Menu Configuration for GE Lighting

GE Lighting uses 42 start menu items. In addition to the **New** and **Search** start menu items for the 13 asset types that are in use, there are **New** and **Search** start menu items for the CSElement and SiteEntry asset types in case user_designer (the site designer) ever decides to use them.

The start menu items are very simple: none of the **New** items provide a default value for an asset created with it and none of the **Search** items provide a default value to search by. The following table describes all the start menu items used by the GE Lighting site.

Name	Asset Type	Start Menu Type	Roles
New Article (flex)	article (flex)	New	Author, Approver, Editor, Checker
Find Article (flex)	article (flex)	Search	Author, Approver, Editor, Checker
New Attribute Editor	attribute editor	New	Designer
Find Attribute Editor	attribute editor	Search	Designer
New CSElement	CSElement	New	Designer
Find CSElement	CSElement	Search	Designer
New Content Attribute	content attribute	New	Designer
Find Content Attribute	content attribute	Search	Designer
New Content Definition	content definition	New	Designer
Find Content Definition	content definition	Search	Designer
New Content Parent	content parent	New	Designer
Find Content Parent	content parent	Search	Designer
New Content Parent Definition	content parent definition	New	Designer
Find Content Parent Definition	content parent definition	Search	Designer
New History Attribute	history attribute	New	Designer
Find History Attribute	history attribute	Search	Designer
New History Definition	history definition	New	Designer
Find History Definition	history definition	Search	Designer
New Image (flex)	image (flex)	New	Author, Approver, Editor, Checker
Find Image (flex)	image (flex)	Search	Author, Approver, Editor, Checker
New Page	page	New	Designer

Name	Asset Type	Start Menu Type	Roles
Find Page	page	Search	Designer
New Product	product	New	Author, Approver, Checker, Designer, Editor, Pricer
Find Product	product	Search	Author, Approver, Checker, Designer, Editor, Pricer
New Product Attribute	product attribute	New	Designer
Find Product Attribute	product attribute	Search	Designer
New Product Definition	product definition	New	Designer
Find Product Definition	product definition	Search	Designer
New Product Parent	product parent	New	Designer
Find Product Parent	product parent	Search	Designer
New Product Parent Definition	product parent definition	New	Designer
Find Product Parent Definition	product parent definition	Search	Designer
New Promotion	promotion	New	Marketer, Designer
Find Promotion	promotion	Search	Marketer, Designer
New Recommendation	recommendation	New	Marketer, Designer
Find Recommendation	recommendation	Search	Marketer, Designer
New Segment	segment	New	Marketer, Designer
New Segment	segment	New	Marketer, Designer
New SiteEntry	SiteEntry	New	Designer
Find SiteEntry	SiteEntry	Search	Designer
New Template	template	New	Designer
Find Template	template	Search	Designer

Tree Tab Configuration

GE Lighting adds custom tabs to the tree and adds its roles to the system default tabs.

Tab	Description	Roles	Users with Access
Admin (default)	Provides access to the administrative features	GeneralAdmin	fwadmin

Tab	Description	Roles	Users with Access
Active List (default)	Provides access to the user's active list	Author, Approver, Checker, Designer, Editor, Pricer	all users
Content (custom)	Provides access to the images and articles on the GE Lighting site	Author, Approver, Checker, Editor, Pricer	user_approver, user_author, user_checker, user_editor, user_pricer, editor
Design (custom)	Provides access to the attribute editor, template, CSElement, SiteEntry, and the data design asset types.	Designer	user_designer, ladmin
History (default)	Displays the assets that the user worked with during the current session.	All	All
Marketing (default for Engage)	Provides access to the segment, promotion, and recommendation asset types.	Approver, Author, Checker, Designer, Editor, Pricer	user_approver, user_author, user_checker, user_designer, user_editor, user_pricer, editor
Product (custom)	Provides access to the products and product parents on the GE Lighting site	Author, Approver, Checker, Editor, Pricer	user_approver, user_author, user_checker, user_editor, user_pricer, editor
SiteAdmin	Provides access to site administration functions	SiteAdmin	fwadmin
Site Plan (default)	Displays a graphical representation of the online site	Author, Approver, Checker, Designer, Editor, Pricer	all users
Workflow Admin (default)	Provides access to the workflow states and processes	WorkflowAdmin	fwadmin

Workflow

GE Lighting provides a sample workflow process named GE Lighting Process.

It has three states and six steps. This workflow process was built with the default workflow building blocks only.

For a complete description of the this workflow process, see “GE Lighting Process,” on page 157.

Burlington Financial Extension (for Engage)

The Engage extension to the Burlington Financial sample site illustrates the Engage features. It creates a new section for the Burlington Financial site that provides information about mutual funds and uses the segment and recommendation asset types to market those funds to the Burlington Financial site visitors.

Users, ACLs, and Roles for Burlington Financial Extension

The Engage extension adds the following users to the Burlington Financial site:

- user_analyst, the market data analyst who creates determines the segments
- user_expert, the supervisor of both user_marketer and user_analyst
- user_marketer, the marketer who works on segments and funds

There are no custom ACLS—these users are assigned only the system default ACLs. However, because Engage adds two new system ACLs (Visitor and VisitorAdmin), those ACLs are assigned to these new users as well as all the existing Burlington Financial users.

There are also three new roles for those users:

- Analyst
- Expert
- Marketer

The following table shows how the user accounts and role assignments are configured for the extension to the Burlington Financial site:

Username	Password	ACLs	Roles
user_analyst	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleeditor	Analyst
user_approver	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleeditor	Approver
user_author	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleeditor	Author
user_checker	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleeditor	Checker

Username	Password	ACLs	Roles
user_designer	user	Browser, ElementEditor, PageEditor, RemoteClient, TableEditor, UserReader, Visitor, VisitorAdmin, xceleditor,	Designer
user_editor	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleditor	Editor
user_expert	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleditor	Expert
user_marketer	user	Browser, ElementReader, PageReader, RemoteClient, UserReader, Visitor, xceleditor	Marketer, Designer
fwadmin	xceladmin	Browser, ElementEditor, PageEditor, RemoteClient, TableEditor, UserEditor, UserReader, Visitor, VisitorAdmin, xceladmin, xceleditor	GeneralAdmin, SiteAdmin, WorkflowAdmin

Asset Type Configuration for Burlington Financial Extension

The Engage extension adds two flex asset families to the Burlington Financial sample site: the product family and the content family.

- The mutual funds that are featured on the site are products that are created with a product flex family.
- Some of the information that is displayed for those products is created by using the content flex family.

The members of the product family are as follows:

- product attribute
- product parent definition
- product parent (fund parent)
- product definition
- product (fund), a flex asset type
- PDF, another flex asset type

The site provides 28 product attributes, 10 product parent definitions, 2 product definitions (one for each of the flex asset types, product and PDF), 61 product parents (fund parents), 30 products (funds), and 1 PDF asset.

The members of the content family are as follows:

- content attribute
- content parent definition
- content parent
- content definition
- drill hierarchy, the flex asset type

There are two content attributes, one content definition, and nine drill hierarchy assets. There are no content parent definitions or content parents.

Note

For information about the data structure of these asset types, see the chapter on creating flex asset types in the *Content Server Developer's Guide* and then use Content Server Explorer to examine the database tables for these asset types.

The following Engage and CS-Direct Advantage asset types are also used for the site:

- history attribute
- history definition
- recommendation
- segment
- visitor attribute

The following two asset types are enabled for the site, in case the site designer or marketers ever decide to add them to the site:

- attribute editor
- promotion

Sources

The Engage extension to the Burlington Financial site adds no sources.

Start Menu Configuration for Burlington Financial Extension

The Engage extension to the Burlington Financial site provides an additional 36 start menu items.

The start menu items are very simple: none of the **New** items provide a default value for an asset created with it and none of the **Search** items provide a default value to search by. The following table describes the additional start menu items provided by Burlington Financial Extension.

Name	Asset Type	Start Menu Type	Roles
New Attribute Editor	attribute editor	New	Designer
Find Attribute Editor	attribute editor	Search	Designer
New Content Attribute	content attribute	New	Designer
Find Content Attribute	content attribute	Search	Designer
New Content Definition	content definition	New	Designer
Find Content Definition	content definition	Search	Designer
New Content Parent	content parent	New	Designer
Find Content Parent	content parent	Search	Designer
New Content Parent Definition	content parent definition	New	Designer
Find Content Parent Definition	content parent definition	Search	Designer
New Drill Hierarchy	drill hierarchy	New	Designer
Find Drill Hierarchy	drill hierarchy	Search	Designer
New History Attribute	history attribute	New	Designer
Find History Attribute	history attribute	Search	Designer
New History Definition	history definition	New	Designer
Find History Definition	History Definition	Search	Designer
New PDF	PDF	New	Analyst, Expert, Designer, Marketer

Name	Asset Type	Start Menu Type	Roles
Find PDF	PDF	Search	Analyst, Expert, Designer, Marketer
New Product	product	New	Analyst, Expert, Designer, Marketer
Find Product	product	Search	Analyst, Expert, Designer, Marketer
New Product Attribute	product attribute	New	Designer
Find Product Attribute	product attribute	Search	Designer
New Product Definition	product definition	New	Designer
Find Product Definition	product definition	Search	Designer
New Product Parent	product parent	New	Designer
Find Product Parent	product parent	Search	Designer
New Product Parent Definition	product parent definition	New	Designer
Find Product Parent Definition	product parent definition	Search	Designer
New Recommendation	recommendation	New	Analyst, Expert, Designer, Marketer
Find Recommendation	recommendation	Search	Analyst, Expert, Designer, Marketer
New Segment	segment	New	Analyst, Expert, Designer, Marketer
Find Segment	segment	Search	Analyst, Expert, Designer, Marketer
New Visitor Attribute	visitor attribute	New	Designer
Find Visitor Attribute	visitor attribute	Search	Designer

Tree Tab Configuration for Burlington Financial Extension

The Engage extension to the Burlington Financial site adds three custom tabs to the tree and adds its roles to the system default tabs.

Tab	Description	Roles Assigned	Users with Access
Active List (default)	Provides access to the user's active list	Approver, Author, Checker, Designer, Editor	user_approver, user_author, user_checker, user_designer, user_editor
Admin (default)	Provides access to the administrative features	GeneralAdmin	fwadmin
Content (custom)	For future use.	Analyst, Approver, Author, Checker, Designer, Editor	user_approver, user_author, user_checker, user_designer, user_editor
Design (custom)	Provides access to the site design, data design, and visitor data design asset types	Designer	user_designer
History (default)	Displays the assets that the user worked with during the current session.	All	All
Marketing (default with Engage)	Provides access to the promotion, recommendation, and segment asset types	Analyst, Designer, Expert, Marketer	user_analyst, user_designer, user_expert, user_marketer
SiteAdmin (default)	Provides access to site administration functions	SiteAdmin	fwadmin
Site Plan (default)	Displays a graphical representation of the online site	Approver, Author, Checker, Designer, Editor	user_approver, user_author, user_checker, user_designer, user_editor
Workflow Admin (default)	Provides access to the workflow states and processes	WorkflowAdmin	fwadmin

Workflow for Burlington Financial Extension

The Engage extension to the Burlington Financial site provides two sample workflow processes, Fund Parent Process and Segment Process:

- Fund Parent Process has three states, seven steps, and five roles.
- Segment Process has three states, five steps, and five roles.

These workflow processes are built with the default workflow building blocks only.

CO

The CO sample site contains many default types of asset types and uses a small amount of data to generate a simple website. This site demonstrates how to use flex assets in a simple manner, how to code in JSP, how to use the latest rendering model, and how to build a site that is designed for replication.

Users, ACLS, and Roles

Table C-3 shows how the CO user accounts and role assignments are configured.

Table C-3: CO: User Accounts and Roles

Username ¹	Password	ACLS ²	Roles ³
fwadmin ⁴	xceladmin	TableEditor, Visitor, VisitorAdmin, UserEditor, UserReader, xceladmin, Browser, xceleeditor, xcelpublish, PageEditor, ElementEditor, RemoteClient, WSUser, WSEditor, WSAdmin	Designer, WorkflowAdmin, GeneralAdmin (a system default role)
user_approver	user	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, xcelpublish, RemoteClient, WSUser, WSEditor	Approver
user_author	user	PageReader, ElementReader, Visitor, UserReader, Browser, xceleeditor, xcelpublish, RemoteClient, WSUser, WSEditor	Author
user_designer	user	PageEditor, ElementEditor, Visitor, VisitorAdmin, TableEditor, UserReader, Browser, xceleeditor, xcelpublish, RemoteClient, WSUser, WSEditor	Designer

1. For descriptions of the users, see Table C-1, on page 327.
2. There are no custom ACLs. The user accounts are assigned only the system default ACLs.
3. The GeneralAdmin and WorkflowAdmin roles are system default roles. All other roles are custom-defined.
4. The fwadmin user is a system default.

Asset Type Configuration

Fifteen asset types are used in the CO sample site. They are used to produce the basic web page that is generated by the sample site.

Table C-4 describes the asset type configuration for the CO sample site:

Table C-4: CO Asset Type Configuration

Asset Type	Number of Assets	Association Fields	Categories
Attribute Editor (default)	2	none	none
CSElement (default)	4	none	none
Link (default)	0	none	none
Page (default)	9	MainContent and HighlightContent for Page subtype "Corporate"; MinorContent for type "Standard"	Account Page, Content Page, Home, List Page, Section Front
Query (default)	0	none	Entertainment, General
Recommendation (default)	9	none	none
SiteEntry (default)	2	none	none
Template (default)	23	none	Banner, Entertainment, General, Navigation, Page Template, Subpage Template
Co. Attributes (custom)	15	none	none
Co. Content Asset (custom)	39	none	none
Co. Definitions (custom)	4	none	none
Co. Filter (custom)	1	none	none

Table C-4: CO Asset Type Configuration *(continued)*

Asset Type	Number of Assets	Association Fields	Categories
Co. Parent Asset (custom)	3	none	none
Co. Parent Definition (custom)	1	none	none
StyleSheet (custom)	0	none	CSS HTML 4.0, General, XSL

Start Menu Configuration

Below is the Start Menu Configuration for the CO-Company Launchpad sample site.

Table C-5: CO: Start Menu Configuration

Asset Type	Name	Start Menu Type	Roles
Attribute Editor	New Attribute Editor	New	Designer
	Find Attribute Editor	Search	Designer
CSElement	New CSElement	New	Designer
	Find CSElement	Search	Designer
Link	New Link	New	Checker, Editor, Author, Designer, Approver
	Find Link	Search	Checker, Editor, Author, Designer, Approver, SiteApprover, SiteAuthor
Page	New Page	New	Designer
	Find Page	Search	Author, Designer, Approver, SiteApprover, SiteAuthor
Query	New Query	New	Designer
	Find Query	Search	Designer
Recommendation	Find Recommendation	Search	Approver, Marketer, Author, Editor, SiteApprover, Expert, Analyst, Designer, Pricer, SiteAuthor, Checker

Table C-5: CO: Start Menu Configuration *(continued)*

Asset Type	Name	Start Menu Type	Roles
SiteEntry	New SiteEntry	New	Designer
	Find SiteEntry	Search	Designer
Template	New Template	New	Designer
	Find Template	Search	Designer
Co. Attrubute	New Co. Attribute	New	Designer
	Find Co. Attribute	Find	Designer
Co. Content Assets	New Co. Content	New	Author, Designer, Approver
	New Co. Document	New	Author, Designer, Approver
	New Co. Form	New	Author, Designer, Approver
	New Co. Graphic	New	Author, Designer, Approver
	New Co. Text Content	New	Author, Designer, Approver
	Find Co. Content	Search	Author, Designer, Approver
	CS-Desktop Co. Content	CS-Desktop	Author, Designer, Approver
	CS-DocLink Co. Content	CS-Desktop	Author, Designer, Approver
Co. Definitions	New Co. Definition	New	Designer
	Find Co. Definition	Find	Designer
Co. Filter	New Co. Filter	New	Designer
	Find Co. Filter	Find	Designer
Co. Parent Asset	New Co. Parent Asset	New	Designer
	Find Co. Parent Asset	Find	Designer
Co. Parent Definition	New Co. Parent Definition	New	Designer
	Find Co. Parent Definition	Find	Designer
StyleSheet	New Stylesheet	New	Designer
	Find Stylesheet	Search	Designer

Tree Tab Configuration

Below is an outline of the tree tab configuration in the CO sample site.

Table C-6: CO: Tree Tab Configuration

Tab	Description	Roles	Users with Access
Active List	Default tab. Provides access to the user's active list	Approver, HelloDesigner, Marketer, Author, Editor, SiteApprover, Expert, Analyst, Designer, Pricer, HelloAuthor, SiteAuthor, Checker, HelloEditor	fwadmin, user_approver, user_author, user_designer
Admin	Default tab. Provides access to the administrative features	GeneralAdmin	fwadmin
Content	Provides access to the content mamagment tab	Expert, Checker, Editor, Author, Pricer, Designer, Approver, Analyst	fwadmin, user_approver, user_author, user_designer
Design	Default tab. Provides access to the design tools	Designer	fwadmin, user_designer
Marketing	Default tab. Provides access to marketing tools	Approver, Marketer, Author, Editor, Expert, Analyst, Designer, Pricer, Checker	fwadmin, user_approver, user_author, user_designer
Query	Default tab. Provides access to useful search features	Designer	fwadmin, user_designer
SiteAdmin	Default tab. Provides access to site administration functions	Designer, WorkflowAdmin, GeneralAdmin	fwadmin, user_designer

Table C-6: CO: Tree Tab Configuration *(continued)*

Tab	Description	Roles	Users with Access
Site Plan	Default tab. Displays a graphical representation of the online site	Approver, HelloDesigner, Author, Editor, SiteApprover, Expert, Analyst, Designer, Pricer, HelloAuthor, SiteAuthor, Checker, HelloEditor	fwadmin, user_approver, user_author, user_designer
Workflow	Default tab. Provides access to the workflow states and processes	WorkflowAdmin	fwadmin

Workflow

The sample site CO comes with one predefined workflow. The workflow has one step from start to finish. It can be rejected or re-assigned, however, before it is published.

This workflow can be found under the “Workflow” tab, named “CO: One-Step for Content.”

Appendix D

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

This appendix is for Content Server administrators who are managing users, ACLs, sites and roles from the Content Server interface of an LDAP-integrated CS system and need to know how their operations affect both the CS database and the LDAP server. This appendix summarizes:

- Which operations are propagated to the LDAP server (given the type of LDAP server, the operation, and whether you are connected as user with read-only or read-write permissions)
- Which operations produce errors
- How to correct errors

This appendix contains the following sections:

- Overview
- User Management Operations
- Site and Role Management Operations

Overview

Content Server can be integrated with LDAP servers that use the LDAP-2 protocol or any other protocol (although, in the latter case, FatWire does not support write operations from the Content Server interface to the LDAP server).

This overview summarizes

- the types of LDAP schema that Content Server supports,
- operations that can be performed from the CS interface when each type of schema is deployed, and
- the outcome of the operations, given the permissions of the Content Server administrator to the LDAP server and the nature of the LDAP server.

LDAP Schema

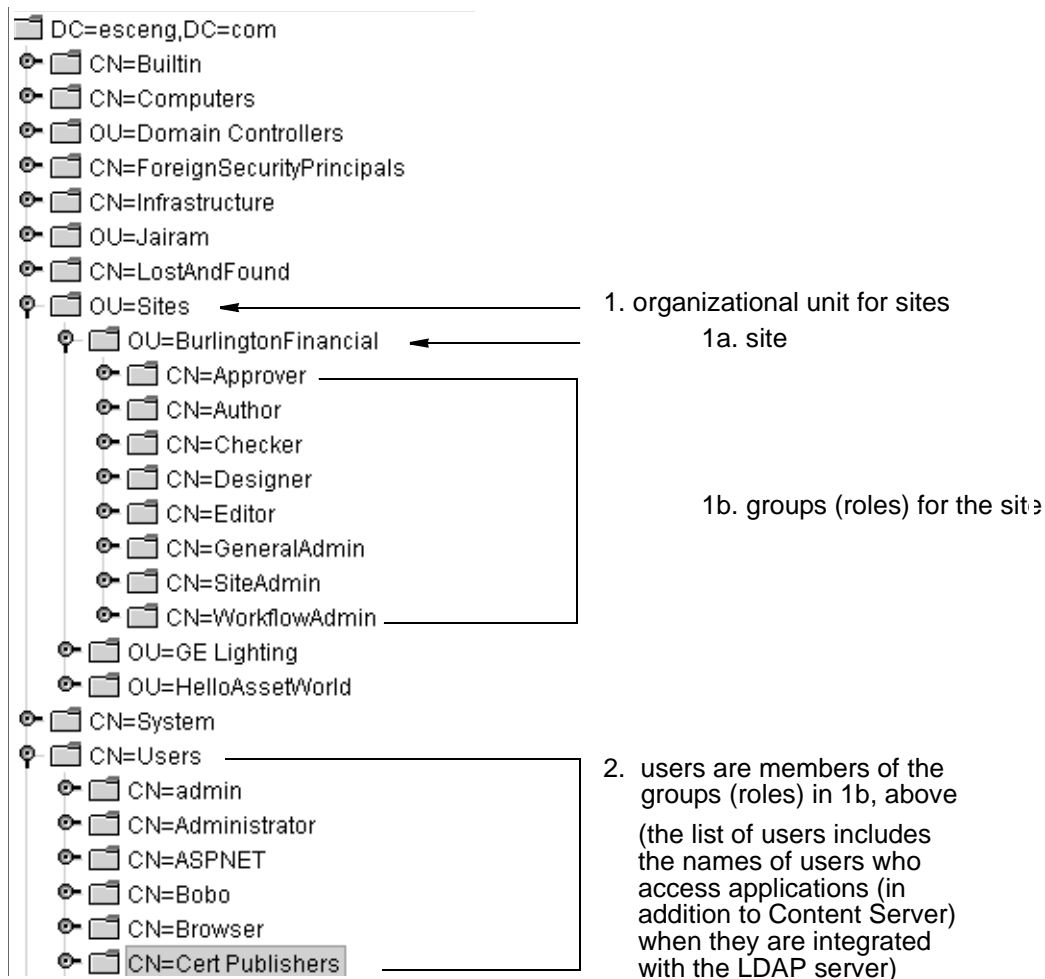
Four LDAP-schema scenarios are covered in this appendix: hierarchical- and flat-schema LDAP in both web-based and portal-based installations. Each schema supports selected operations and environments, as follows:

- Hierarchical-schema LDAP supports operations on sites and roles in **web-based CS installations**. This schema requires LDAP users to define a site organizational unit in the LDAP server, where they name each site and roles for that site. The roles must be subordinated to the site (for an example, see Figure D-1, “LDAP Hierarchies”).
- Flat-schema LDAP supports operations on sites and roles in **CS portal installations**. Flat-schema LDAP levels sites and roles so that no hierarchy exists among them.
- Both hierarchical- and flat-schema LDAP support operations on ACLs and users (user accounts, user profiles, and user attributes) in **web-based and portal installations**.

Table D-1 summarizes LDAP schema and the possible operations. Using the correct schema, the CS administrator can perform operations in the CS interface and have them propagate to the Content Server database, or the LDAP server, or both, depending also on the conditions that are outlined in “LDAP Connectivity for Site and Role Management,” on page 363.

Table D-1: LDAP Integration Scenarios

Type of Installation	Create/Modify/Delete Operations on ...	Requires	
		Flat-Schema LDAP	Hierarchical-Schema LDAP
CS-powered website	ACLs, Users, User Profiles, User Attributes	Table D-2	
	Sites		Table D-3
	Roles		Table D-3
CS-powered portal (with embedded LDAP)	ACLs, Users, User Profiles, User Attributes	Table D-2	
	Sites	Table D-4	
	Roles	Table D-4	

Figure D-1: LDAP Hierarchies

LDAP Connectivity for Site and Role Management

For operations on roles and sites to work as described in this appendix, LDAP connectivity must be enabled by setting the values of two properties in the `futuretense_xcel.ini` file:

```
xcelerate.usermanagerclass
xcelerate.rolemanagerclass
```

These properties allow you to manage sites and roles directly in the LDAP server (as shown in this appendix) or to bypass the LDAP server in order to manage sites and roles exclusively in the Content Server database.

- If the property values specify LDAP, then LDAP connectivity is supported such that sites and roles can be managed in the LDAP server (and in the Content Server database).

- For hierarchical-schema LDAP, only the `xcelerate.usermanagerclass` property must specify LDAP. The `xcelerate.rolemanagerclass` property uses the default value. For example:

```
xcelerate.usermanagerclass=com.openmarket.xcelerate.  
user.LDAPSchemaUserManager  
  
xcelerate.rolemanagerclass=com.openmarket.xcelerate.  
roles.RoleManager
```
- For flat-schema LDAP, both properties must specify LDAP. For example:

```
xcelerate.usermanagerclass=com.openmarket.xcelerate.  
user.FlatLDAPSchemaUserManager  
  
xcelerate.rolemanagerclass=com.openmarket.xcelerate.  
roles.FlatLDAPSchemaRoleManager
```
- If the property values do not specify LDAP (for example, `xcelerate.usermanagerclass=com.openmarket.xcelerate.user.usermanager`) then sites and roles can managed only in the Content Server database.

LDAP Users and Their Permissions to LDAP Servers

In an LDAP-integrated CS system, the administrator of the Content Server system may or may not be an administrative user of the LDAP server, depending on the value of the `jndi.connectAsUser` property in the `dir.ini` file. The value determines how the Content Server administrator is connected to the LDAP server, and therefore defines the LDAP user:

- If `jndi.connectAsUser` is set to `true`, then Content Server defines the LDAP user to be the same one that is logged in to Content Server and connects that user to the LDAP server.
- If `jndi.connectAsUser` is set to `false`, then Content Server defines the LDAP user to be the one that is specified in the `jndi.login` property (in `dir.ini`) and connects that user to the LDAP server.

For the connection to take place, the user and his permissions must also be defined in the LDAP server. If the user has read-only permissions to the LDAP server, he is not an administrative LDAP user. If the user has read and write permissions to the LDAP server, he is an administrative LDAP user (or simply, an LDAP administrator).

Note

For more information about LDAP-related properties, see the *Content Server Property Files Reference*.

LDAP-Integrated Operations

For the Content Server administrator to successfully perform an operation (such as creating an ACL in both the Content Server database and the LDAP server), it is critical for Content Server to be properly integrated with the LDAP server. Barring integration issues, the outcome of an operation depends on the following factors:

1. The LDAP user, as defined by the `jndi.connectAsUser` property (see the previous section).

2. The LDAP user's permissions, as defined in the LDAP server.

If the CS administrator is connected to the LDAP server as a user without administrative rights, his operations (such as deleting an ACL from the Content Server database) cannot be written to the LDAP server. The system responds by either writing the operations to the Content Server database, or not at all. The operations must then be repeated in the Content Server interface and performed manually in the LDAP server by an LDAP administrator.

3. The nature of the operation.

Certain operations that are performed in the Content Server interface (such as editing an ACL) are not written to the LDAP server, even when the Content Server administrator is connected with write permissions.

4. Whether the LDAP server supports the LDAP-2 protocol.

If the LDAP server does not support the LDAP-2 protocol, then FatWire does not support write operations from the Content Server interface to the LDAP server, and the result of an operation cannot always be predicted.

Tables D-2 through D-4 summarize the results of operations that can be performed by a Content Server administrator who is connected to the LDAP server as a user with read-only and read/write permissions.

User Management Operations

Table D-2, on page 366 applies to web and portal environments. Table D-2 summarizes system response to user management operations. The operations are performed by a Content Server administrator using the interfaces of three Content Server systems, each integrated with one of the following LDAP options:

- LDAP-2 server with read-only permissions for the LDAP user (defined on page 363)
- 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 operation are described on the pages that are noted in the left-hand column of Table D-2.

Note

Because ACLs, user accounts, user profiles, and user attributes directly relate to users (by defining them), the term “user management” in this appendix includes the management of ACLs, user accounts, user profiles, and user attributes.

Table D-2: System Response to User Management Operations in Content Server

Operation in Content Server	Result	System Response With:		
		LDAP-2: User has Read-Only Permissions ¹	LDAP-2: User has Write Permissions ²	Not LDAP-2: Write not Supported ³
Creating an ACL (p. 367) <i>ACL is created in:</i>		Returns error ⁴	LDAP + CS database	Unpredictable
Editing an ACL (p. 367) <i>ACL is modified in:</i>		CS database	CS database	CS database
Deleting an ACL (p. 367) <i>ACL is deleted in:</i>		Returns error	LDAP + CS database	Unpredictable
Assigning ACLs to Tables (p. 368) <i>ACL is assigned in:</i>		CS database	CS database	CS database
Assigning ACLs to Content Server Pages (Site Catalog Page Entries) (p. 368) <i>ACL is assigned in:</i>		CS database	CS database	CS database
Creating a User (Granting ACLs for Access Privileges) (p. 368) <i>User is created in:</i>		Returns error	LDAP	Unpredictable
Editing a User (p. 369) <i>User is modified in:</i>		Returns error	LDAP	Unpredictable
Deleting a User (p. 369) <i>User is deleted from:</i>		Returns error	LDAP	Unpredictable
Creating/Editing a User Profile (p. 370) <i>User profile is created/edited in:</i>		Returns error	LDAP	Unpredictable
Deleting a User Profile (p. 370) <i>User profile is deleted from:</i>		Returns error	LDAP	Unpredictable
Creating User Attributes (p. 371) <i>User profile is created in:</i>		Returns error	LDAP	Unpredictable
Editing/Deleting User Attributes (p. 371) <i>User profile is deleted from:</i>		Returns error	LDAP	Unpredictable

1. ACLs and users cannot be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user without write permissions.
2. ACLs and users can be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user with write permissions.
3. The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP servers, FatWire does not support write operations from the Content Server interface to the LDAP server.
4. “Returns error” means that the operation is not performed. The system returns an error.

Creating an ACL

Operation

The Content Server administrator attempts to create an ACL by using the “Content Server Management Tools” 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 Content Server database.
2. LDAP-2 Server with Write Permissions—The system creates the ACL in both the LDAP server and the Content Server 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 Content Server.

Editing an ACL

Operation

The Content Server administrator attempts to modify an ACL by using the “Content Server Management Tools” node on the “Admin” tab.

Note

The Content Server 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 Content Server 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).

Deleting an ACL

Operation

The Content Server administrator attempts to delete an ACL by using the “Content Server Management Tools” 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 Content Server database.

2. LDAP-2 Server with Write Permissions—The system deletes the ACL from the LDAP server and from the Content Server 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 Content Server.

Assigning ACLs to Tables

Operation

The Content Server administrator attempts to assign an ACL to system tables, using the “Content Server Management Tools” node on the “Admin” tab.

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).

Assigning ACLs to Content Server Pages (Site Catalog Page Entries)

Operation

The Content Server administrator attempts to assign an ACL to Content Server Pages by using the “Content Server Management Tools” node on the “Admin” tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system adds the selected ACLs for the specified Content Server 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).

Creating a User (Granting ACLs for Access Privileges)

Operation

The Content Server administrator attempts to create a user by using the “Content Server Management Tools” node on the “Admin” tab.

Note

Content Server displays only the ACLs that are stored in its database.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot add the user and the user's assigned ACLs. The LDAP administrator must use the LDAP server to manually add the user to the LDAP server and assign ACLs to the user.
2. LDAP-2 Server with Write Permissions—The system creates the user in the LDAP server and assigns ACLs to the user.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must use the LDAP server to manually add the user to the LDAP server and assign ACLs to the user.

Editing a User

Operation

The Content Server administrator attempts to modify the ACLs that are assigned to a user or to change the user's password by using the "Content Server Management Tools" node on the "Admin" tab.

Note

The "Modify user" operation allows you to change only the user password and the ACLs that are associated with the user.

The "Modify user attributes" operation allows you to change all attributes that are stored in the database or in the LDAP server.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot modify the user's password and/or ACLs. The LDAP administrator must manually modify the user in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system modifies the user's password and/or assigned ACLs in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually modify the user in the LDAP server.

Deleting a User

Operation

The Content Server administrator attempts to delete a user by using the "Content Server Management Tools" node on the "Admin" tab.

Before deleting a user, the administrator must delete the user profile. If a user profile exists, an error message is triggered. The behavior described below is expected when no user profile exists for the user being deleted.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot delete the user. The LDAP administrator must manually delete the user from LDAP server.
2. LDAP-2 Server with Write Permissions—The system deletes the user from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually delete the user from the LDAP server.

Creating/Editing a User Profile

Operation

The Content Server administrator attempts to create or modify a user profile by using the “User Profiles” node on the “Admin” tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot create/modify the user’s profile and/or Locale Preference. The LDAP administrator must manually create/modify the user profile in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system creates/modifies the user’s profile and/or locale preference in the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually create/modify the user profile in the LDAP server.

Deleting a User Profile

Operation

The Content Server administrator attempts to delete a user profile by using the “User Profiles” node on the “Admin” tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot delete the user’s profile. The LDAP administrator must manually delete the user profile in the LDAP server.
2. LDAP-2 Server with Write Permissions—The system deletes the user’s profile from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually delete the user profile from the LDAP server.

Creating User Attributes

Note

User attributes are stored in only one place—either in the database or in LDAP.

Operation

The Content Server administrator attempts to create new attributes for the user from the “Content Server Management Tools” node of the “Admin” tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot add the new attributes to the LDAP server. The LDAP administrator must manually add user attributes to the LDAP server.
2. LDAP-2 Server with Write Permissions—The system adds the user’s new attributes to the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually add user attributes to the LDAP server.

Editing/Deleting User Attributes

The Content Server administrator attempts to modify the user attributes from the “Content Server Management Tools” node of the “Admin” tab.

System Response

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message; it cannot modify or delete the user’s attributes from the LDAP server. The LDAP administrator must manually modify user attributes in the LDAP server and delete user attributes from the LDAP server.
2. LDAP-2 Server with Write Permissions—The system modifies the user attributes in the LDAP server and deletes them from the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The LDAP administrator must manually modify user attributes in the LDAP server, and manually delete user attributes from the LDAP server.

Site and Role Management Operations

This section covers operations that require hierarchical and flat schema LDAP (page 373).

Operations with Hierarchical-Schema LDAP Servers

Table D-3 applies to the web environment and summarizes system response to site and role management operations. The operations are performed by a Content Server administrator using the interfaces of three Content Server systems, each integrated with one of the following LDAP options:

- An LDAP-2 server with read-only permissions for the LDAP user (defined on page 363)
- An LDAP-2 server with write permissions for the LDAP user
- An LDAP server other than LDAP-2, in which case write operations are not supported

The results of each operation are described on the pages that are noted in the left-hand column of Table D-3.

Table D-3: System Response to Site and Role Management Operations with Hierarchical-Schema LDAP (Content Server Web Site)

Operation in Content Server	Result	System Response With Hierarchical-Schema LDAP:		
		LDAP-2: User has Read-Only Permissions ¹	LDAP-2: User has Write Permissions ²	Not LDAP-2: Write not Supported ³
Creating a Site (p. 374)	<i>Site is created in:</i>	CS database	CS database	CS database
Editing a Site (p. 374)	<i>Site is edited in:</i>	CS database	CS database	CS database
Deleting a Site (p. 375)	<i>Site is deleted in:</i>	CS database	CS database	CS database
Creating a Role (p. 377)	<i>Role is created in:</i>	CS database	CS database	CS database
Editing a Role (p. 377)	<i>Role is edited in:</i>	CS database	CS database	CS database
Deleting a Role (p. 378)	<i>Role is deleted in:</i>	CS database	CS database	CS database
Granting Users Access to Sites (p. 375)	<i>Access is granted in:</i>	Returns error ⁴	LDAP	Unpredictable
Removing Users' Access to Sites (p. 376)	<i>Access is removed from:</i>	neither the CS database nor the LDAP server	neither the CS database nor the LDAP server	neither the CS database nor the LDAP server

1. Sites and roles cannot be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user without write permissions.
2. Sites and roles can be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user with write permissions.
3. The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP servers, FatWire does not support write operations from the Content Server interface to the LDAP server.
4. "Returns error" means that the operation is not performed. The system returns an error.

Operations with Flat-Schema LDAP Servers

Table D-4 applies to the portal environment, and summarizes system response to site and role management operations. The operations are performed by a Content Server administrator using the interfaces of three Content Server systems, each integrated with one of the following LDAP options:

- An LDAP-2 server with read-only permissions for the LDAP user (defined on page 363)
- An LDAP-2 server with write permissions for the LDAP user
- An LDAP server other than LDAP-2, in which case write operations are not supported

The results of each operation are described on the pages that are noted in the left-hand column of Table D-4.

Table D-4: System Response to Site and Role Management Operations with Flat-Schema LDAP (Content Server Portals)

Operation in Content Server	Result	System Response with Flat-Schema LDAP:		
		LDAP-2: User has Read-Only Permissions ¹	LDAP-2: User has Write Permissions ²	Not LDAP-2: Write not Supported ³
Creating a Site (p. 374) <i>Site is created in:</i>		CS database	CS database	CS database
Editing a Site (p. 374) <i>Site is edited in:</i>		CS database	CS database	CS database
Deleting a Site (p. 375) <i>Site is deleted from:</i>		CS database	CS database	CS database
Creating a Role (p. 377) <i>Role is created in:</i>		Returns error ⁴	LDAP + CS database ⁵	Unpredictable
Editing a Role (p. 377) <i>Role is edited in:</i>		CS database	CS database	CS database
Deleting a Role (p. 378) <i>Role is deleted from:</i>		Returns error	LDAP + CS database	Unpredictable
Granting Users Access to Sites (p. 375) <i>Access is granted in:</i>		Returns error	LDAP	Unpredictable
Removing Users' Access to Sites (p. 376) <i>Access is removed from:</i>		Returns error	LDAP	Unpredictable

1. Sites and roles cannot be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user without write permissions.
2. Sites and roles can be stored in the LDAP server, as the Content Server administrator is connected to the LDAP server as a user with write permissions.
3. The LDAP server does not support the LDAP-2 protocol (such as Netscape Server). For non-compliant LDAP servers, FatWire does not support write operations from the Content Server interface to the LDAP server.
4. "Returns error" means that the operation is not performed. The system returns an error.
5. For a role to be created in both the LDAP server *and* the CS database, at least one site must exist in the LDAP server. If no sites exist, the role is created only in the CS database.

Creating a Site

Operation

The Content Server administrator attempts to create a site by using the “Sites” node on the “Admin” tab.

Note

If you are manually creating a site in an LDAP-integrated system, you must ensure that the database and the LDAP server are synchronized. Otherwise, sites will not be properly listed in the Content Server interface.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system creates the site in the Content Server 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).

Flat Schema

1. LDAP server with Read-Only Permissions—The system creates the site in the Content Server 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).

Editing a Site

Operation

The Content Server administrator attempts to edit the description of a site by using the “Site Edit” option on the “Admin” tab.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the site description in the Content Server database. (The site description is stored only in the Content Server 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).

Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the site description in the Content Server database. (The site description is stored only in the Content Server 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).

Deleting a Site

Operation

The Content Server administrator attempts to delete the description of the site by using the “Site Edit” node on the “Admin” tab.

Note

If you are manually deleting a site with LDAP integration, you must ensure that the database and the LDAP server are synchronized. Otherwise, sites will not be properly listed in the Content Server interface.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the site from the Content Server 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).

Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the site from the Content Server 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).

Granting Users Access to Sites

Operation

The Content Server administrator attempts to grant users access to a site by using the “Site > Users” nodes on the “Admin” tab.

Note

Content Server displays only the roles that are stored in its database.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system returns an error message because, in the LDAP server, it cannot assign any roles to the specified users of the selected site. The LDAP administrator must manually assign the roles.
2. LDAP-2 Server with Write Permissions—The system assigns roles, within the LDAP server, to the specified users of the selected site.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

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.

Removing Users' Access to Sites

Operation

The Content Server administrator attempts to remove user access to sites by using the “Site > Users” nodes on the “Admin” tab.

Note

Content Server displays only the sites that are stored in its interface.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The LDAP administrator must manually remove the user's permissions to the sites, directly in the LDAP server.
2. LDAP-2 Server with Write Permissions—Responds as above (scenario 1).
3. Not LDAP-2: Write not Supported—Behavior is unpredictable.

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.

Creating a Role

Operation

The Content Server administrator attempts to create a role by using the “Roles” node on the “Admin” tab.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system creates the role in the Content Server 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).

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 Content Server database. The LDAP administrator must create the role in the LDAP server. The same role must be re-created in the Content Server database.
2. LDAP-2 Server with Write Permissions—The system creates the role in the Content Server 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*, etc.).

Note

For a role to be created in both the LDAP server *and* the CS database, at least one site must exist in the LDAP server. If no sites exist, the role is created only in the CS database.

3. Not LDAP-2: Write not Supported—Behavior is unpredictable. An error message can be returned.

Editing a Role

Operation

The Content Server administrator attempts to modify the description of a role by using the “Roles” node on the “Admin” tab.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the description of the role in the Content Server database. (The role description is stored only in the Content Server 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).

Flat Schema

1. LDAP-2 Server with Read-Only Permissions—The system modifies the description of the role in the Content Server database. (The role description is stored only in the Content Server 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).

Deleting a Role

Operation

The Content Server administrator attempts to delete a role by using the “Roles” node on the “Admin” tab.

System Response

Hierarchical Schema

1. LDAP-2 Server with Read-Only Permissions—The system deletes the role from the Content Server 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).

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 Content Server 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 Content Server database.
2. LDAP-2 Server with Read-Only Permissions—The system deletes the role from both the Content Server database and the LDAP server.
3. Not LDAP-2: Write not Supported—Behavior is unpredictable. The system can return an error message.

Index of Procedures

Before creating a user	77
To assign ACLs to a page entry that is not associated with a SiteEntry or template asset	68
To assign ACLs to an existing table	67
To change the order of tree tabs	125
To clear assignments for a user	178
To clear checkouts for assets	293
To configure display of the tree and toggling rights	122
To configure Site Launcher	186
To configure the asset types for CS-Desktop	200
To configure the CS-DocLink asset types	206
To configure the number of items under a node in a tree tab	126
To copy workflow processes	176
To create a mirror destination	254
To create a new ACL	65
To create a new role	70
To create a new tree tab	122
To create a site	92
To create a start menu item for the New list	109
To create a start menu item for the Search list	112
To create a user	77
To create an export starting point	249
To create CS-Desktop Start Menu items	202
To create CS-DocLink Start Menu items	211
To create e-mail objects	160
To create or edit a user profile	80
To create the batch user account	245
To create the export destination	247

To create the Workflow Groups tab	124
To create workflow actions and conditions.	162
To create workflow states	166
To delete a custom ACL	66
To delete a role	71
To delete a site.	95
To delete a tree tab.	125
To delete a user from a site	99
To delete a user from the system.	79
To delete a user profile	81
To delete e-mail objects.	161
To delete function privileges.	177
To delete revisions.	291
To delete workflow actions and conditions.	164
To delete workflow processes	176
To delete workflow states	167
To delete workflow steps.	176
To disable an asset type.	301
To disable revision tracking for asset types.	292
To disable revision tracking for non-asset tables	292
To edit a custom ACL	66
To edit a function privilege	175
To edit a site description	95
To edit a tree tab	124
To edit a user	78
To edit e-mail objects	161
To edit the description of a role.	71
To edit workflow actions and conditions	164
To edit workflow processes.	174
To edit workflow states	167
To edit workflow steps	175
To enable asset types for a site	100
To enable asset types for CS-Desktop.	200
To enable asset types for CS-DocLink	207
To enable asset types for search	300
To enable revision tracking for asset types	289
To enable revision tracking for non-asset tables	289
To enable the InSite Editor	214
To follow up with post-creation procedures	78
To give a user access to a site	97
To identify the mirror user to the source system.	253
To identify the proxy server	254

To initialize the destination system	255
To install the client application	204
To install the client application	212
To log in to Content Server as the general administrator	39
To modify a user's attributes	82
To obtain site configuration information	94
To rebuild an index	301
To remove asset types from a site	102
To replicate the source site	187
To set access permissions from futuretense_xcel.ini	119
To set access permissions from the Content Server interface	117
To set the base directory for the exported files	247
To set the system default locale	194
To set up function privileges	173
To set up timed action events	164
To set up your destination system	252
To specify which fields of an asset type will be displayed on the CS toolbar in the MS Word interface	201
To specify which of an asset type's fields will be available to content providers who are using CS-DocLink to create or edit assets of that type	208
To test the CS-DocLink configuration	212
To test your workflow process	177
To view the list of users and roles currently defined for a given site, and to change role assignments	98

Index

A

- ACL (access control list)
 - adding 65
 - and roles 63, 65
 - assigned to system and sample site users 311
 - assigning to page 68
 - assigning to tables 67
 - bit mask numbers 306
 - creating 65
 - database tables 63
 - default system ACLs 64, 306
 - deleting 66
 - described 63
 - editing 66
 - for batch user 245
 - for mirror user 252, 279
 - list 66
 - page entries 64
 - permissions 306
 - restriction message 68
 - role in user management 62
 - SystemACL table 306
 - user accounts 63
- actions
 - creating 162
 - deadlock 131, 144
 - delegate 132, 142
 - deleting 164
 - editing 164
 - group deadlock 132, 144
 - step 129, 142
 - timed 128, 139, 141
- Active List tab 107
- admin
 - user name 311
- Admin tab 107
 - xcleadmin ACL 310
- administrators
 - roles they need 313
- aliases
 - table that stores aliases for visitors 321
- all-voting step 130
- analyst role 329
- APIs
 - Directory Services 74
 - Export 217
 - Mirror 217
- approval system 218
 - calculating dependencies 220, 222
 - interactions with publishing system 220
 - types of dependencies 221
- Approve Assets 266
- Approve for Publish step action
 - configuring 163
- Approve Multiple Assets 225, 266
 - vs. BulkApprover utility 225
- ApprovedAssetDeps table 230, 240
- ApprovedAssets table 230, 240
- Approver role 329
- approving assets
 - automatic upon delete 221
 - multiple assets 225
- Arguments field (publishing destination) 248, 262

- arguments, publishing
 - DIR 231, 233, 244
 - OLDTEMPLATE 232
 - REMOTEPASS 240
 - REMOTEUSER 240
 - SIMPLEDIR 231, 233
 - SIMPLENAME 231, 235
 - SUFFIX 231, 235
 - URLPREFIX 230
 - VERBOSE 232, 240
- asset invalidation 260
- asset types 286
 - Burlington Financial 336
 - Burlington Financial Extension 349
 - determining locale for 196
 - disabling for search engines 301
 - disabling for sites 102
 - disabling revision tracking for 292
 - enabling for Content Server Desktop 200
 - enabling for Content Server DocLink 207
 - enabling for search engines 300
 - enabling for sites 100
 - enabling revision tracking for 289
 - GE Lighting 343
 - Hello Asset World 332, 355
 - indexing differences 298
 - when missing from the destination 280
- AssetPublication table 239
- AssetPublishList table 239
- AssetRelationTree table 239
- assets
 - calculating dependencies 222
 - clearing checkouts (administrator) 293
 - held from being published 221
 - locked during publish 227
 - marked as changed on destination after being published 239
 - mirror queue and basic assets 238
 - mirror queue and flex assets 239
- assignment methods 146
- assignments
 - clearing 132, 178
 - cross-site 138
 - deadlines 134
 - delegating 132
 - described 128
- associations 120

- attributes, user
 - e-mail 80
 - locale 80
- Author role 329
- auxiliary tables
 - examples 241
 - for asset types 256

B

- base export directory 246
 - Export Assets to XML 261
 - Export to Disk 232
- basic assets
 - dependencies 222
 - mirror queue 238
- batch host 245
 - error when not configured correctly 283
- batch user
 - ACLS needed 245
 - creating 245
 - described 226
 - error when not configured correctly 282
- binary visitor attributes
 - where stored 321
- bit mask numbers for ACL permissions 306
- blob (binary large object)
 - file for when exported 231
- BlobServer
 - security 85
- Bobo user name 327
- Browser ACL 309
- BulkApprover utility 225
- Burlington Financial Extension 347
 - asset types 349
 - roles 347
 - start menu items 350
 - tree tabs 352
 - users 347
 - workflow 353
- Burlington Financial sample site 334
 - asset types 336
 - Normal Article Process workflow 154
 - sources 337
 - start menu items 338
 - tree tabs 340
 - users, ACLs, roles 335

C

CacheManager 241

- communicates with Satellite Server 239
- refresh the page cache after Mirror to Server publish session 239

catalogs. *See* database tables.

categories 120

CGI path 259

Checker role 329

checkins

- enabling with revision tracking 286
- explicit 287
- implicit 287

checkouts

- clearing 286
- clearing with administrative feature 293
- enabling with revision tracking 286
- explicit 287
- implicit 287

clearing

- checkouts 286
- deadlocks 131
- workflow assignments 178

Coco user name 327

conditions

- creating 162
- deleting 164
- described 143
- Example Step Condition 144
- planning 144

configuring

- Export Assets to XML 261
- Export to Disk 246
- function privileges 173
- Mirror to Server 252
- publishing 245
- security measures 86
- system default locale 194

Content Server

- database 317
- e-mail feature 160
- user 311

Content Server Desktop

- .dot files 205
- configuring 200
- Content Server toolbar 201
- creating start menu items 202
- described 198

installing the client application 204

RemoteClient ACL 309

required ACLs 312

testing configuration 205

users 203

Content Server Direct

- disabling certain forms from the delivery system 88

Content Server DocLink

- configuring 206
- creating start menu items 211
- described 206
- document types 210
- installing the client application 212
- required ACLs 312
- specify fields 208
- testing configuration 212
- users 211

Content Server Engage

- required ACLs 312
- Visitor ACL 310
- VisitorAdmin ACL 310

ContentEditor ACL 309

conventions

- directory names with Export to Disk 232
- Export Assets to XML file names 244
- Export to Disk file names 235

creating

- ACLs 65
- batch user 245
- e-mail objects 160
- end step 172
- export starting points 249
- publishing destination for Export to Disk 247
- publishing destination for Export to XML 262
- publishing destination for Mirror to Server 254
- roles 70
- search start menu items 112
- sites 92
- start menu items of type New 108
- tree tabs 122
- user profiles 80
- users 77
- workflow actions 162
- workflow conditions 162

- workflow process 168
- workflow start step 170
- workflow states 166
- workflow steps 171
- cross-site assignments 138
- CS-Desktop. *See* Content Server Desktop.
- CS-DocLink. *See* Content Server DocLink.

D

- database
 - tables that grow quickly 317
- database tables
 - ACLs 63, 67
 - auxiliary tables for Initialize Mirror 241
 - disabling revision tracking 292
 - enabling revision tracking 289
 - revision tracking 288
 - that grow quickly 317
 - unlocking revision tracked rows 293
 - VMVISITOR 321
 - VMVISITORALIAS 321
 - VMVISITORSCALARVALUE 321
 - VMVISTORSCALARBLOB 321
 - VMz 321
- deadlines
 - assignment 134
 - process 134
- deadlock actions 144
 - planning 144
 - Send Deadlock Email 144
- deadlocks
 - actions 131
 - defined 130
 - group deadlocks 132
 - preventing 131
 - resolving 131
- default tree tabs 107
- DefaultReader user 309, 311
 - and security 85
- delegate actions 142
 - planning 142
- deleting
 - ACLs 66
 - e-mail objects 161
 - function privileges 177
 - roles 71
 - sites 95
 - steps 176
 - tree tabs 125
 - users 79
 - workflow actions 164
 - workflow conditions 164
 - workflow processes 176
 - workflow states 167
- delivery system
 - moving a site to 265
 - security goals 86
- Delivery Type field (publish destination) 248, 262
- delivery types 219
- dependencies, approval
 - basic assets 222
 - CSElement and SiteEntry assets 223
 - exact 221
 - exists 221
 - flex families 223
 - visitor data assets 224
- dependencies, compositional
 - unknown 229
- Designer role 329
- Destination Address (publish destination) 255
- destination directory
 - Export Assets to XML 261
 - Export to Disk 232
- destinations, publishing 219
 - error message when incorrect 280
- development system
 - security goals 86
- DIR publishing argument 231, 233, 244
- directories
 - base export directory, Export to Disk 232
 - root storage for revision tracking 289
- directory names
 - conventions for Export to Disk 250
 - examples for Export to Disk 234
- Directory Services API 74
- due date . *See* deadlines.

E

- editing
 - ACLs 66
 - e-mail objects 161
 - functions privileges 175
 - publish destinations 268

- publish events 271
- roles 71
- sites 95
- tree tabs 124
- user profiles 80
- users 78
- workflow actions 164
- workflow conditions 164
- workflow processes 174
- workflow states 167
- workflow steps 175
- Editor role 330
- editor user name 327
- ElementEditor ACL 309
- ElementReader ACL 309
- elements
 - workflow 139
- e-mail
 - enabling the Content Server e-mail feature 160
- e-mail objects 139
 - Assignment Due Reminder 140
 - Assignment Message 140
 - creating 160
 - deleting 161
 - described 140
 - editing 161
 - planning 141
- emailname workflow variable 139
- end step
 - creating 172
- error messages
 - publishing 278
- events
 - APPEVENT 322
 - publish 225
 - timed action event 164
- eWebEditPro
 - configuring 215
 - users 215
- exact dependency 221
- exists dependency 221
- Expert role 330
- explicit checkins and checkouts 287
- Export API 217
- Export Assets to XML publishing 219
 - configuring 261
 - creating a publishing destination 262

- file names 244
 - specifying the base export directory 261
 - testing the results 263
- export starting points 229
 - creating 249
 - defined 237
 - how many do you need? 237
 - when OLDTEMPLATE is set to true 249
- Export to Disk publishing 219
 - calculating dependencies 222
 - configuring 246
 - creating a publishing destination 247
 - creating export starting points 249
 - described 228
 - directory-naming conventions 232
 - export directory 232
 - export queue 229
 - file names 232
 - publishing arguments 230
 - specifying the base export directory 246
 - testing the results 250

F

- file names, Export Assets to XML 244
- file names, Export to Disk
 - blobs 231
 - conventions 235, 250
 - examples 236
- filename field 235
- flex assets
 - creating the mirror queue for flex assets 239
 - dependencies 223
- Flo user name 327
- flushing
 - inactive data 322
 - session objects (scratch) table 322
- Force Specified Filename field (export starting point) 250
- Force Specified Path field (export starting point) 250
- From State 129, 146
- function privileges
 - configuring 173
 - defined 148
 - deleting 177
 - described 133
 - editing 175

- planning 151
- futuretense.txt file 279

G

- GE Lighting sample site 341
 - asset types 343
 - roles 341
 - start menu items 344
 - tree tabs 345
 - users 341
 - workflow process 157
- general error message 280
- GeneralAdmin role 313
- group deadlock actions 144
 - planning 145
 - Send Deadlock Email 144
- groups, workflow 132
 - synchronize steps 132
 - tab 124

H

- held assets 221
- Hello Asset World 331
 - asset types 332
 - HelloArticle workflow process 153
 - roles 331
 - start menu items 332
 - tree tabs 334
 - users 331
- HelloArticle workflow process 153
- HelloAuthor role 330
- HelloDesigner role 330
- HelloEditor role 330
- history
 - publishing 278
- History tab 107, 316, 334, 340, 346, 352
- history types
 - where stored 321

I

- IDs
 - visitor 321
- implicit checkins and checkouts 287
- inactive visitor data 322
- indexes

- how created for asset types 298
- rebuilding for asset types 301
- Initialize Mirror Destination 241
 - moving a site from development to management 264
 - moving a site to the delivery system 265
 - setting up a mirror destination 255
- InSite Editor
 - configuring 213
 - required ACLs 312
 - users 214
- installing
 - Content Server Desktop client 204
 - Content Server DocLink client 212

J

- Joe user name 327

L

- locale
 - asset types 196
 - Content Server Desktop 204
 - Content Server interface 194
 - properties 195
 - single-language restrictions 196
 - specifying system default 194
- locked assets, clearing 293
- log files
 - futuretense.txt 279
 - publish history 278

M

- management system
 - moving a site to 264
 - security goals 86
- Marketer role 330
- Marketing tab 107
- migrating
 - a site from one system to another 264
- mimetypes 120, 210
- Mirror API 217
- mirror queue
 - for basic assets 238
 - for flex assets 239
 - unpacking the items in the queue 239

- Mirror to Server publishing 219
 - calculating dependencies 222
 - configuring 252
 - creating a publishing destination 254
 - described 238
 - identifying the mirror user 253
 - initializing a target destination 255
 - mirror queue 238
 - proxy servers 254
 - publishing arguments 240
 - setting up the target system 252
 - testing the results 259
- mirror user
 - creating 252
 - described 240
 - error messages for 279
 - identifying 253
- mirroruser user name 327
- missing table publishing error 280
- Moe user name 327
- moving
 - a site from one system to another 264

N

- Normal Article Process 154

O

- OLDTEMPLATE argument 232

P

- page cache
 - refreshed after Mirror to Server 239
- PageEditor ACL 309
- PageReader ACL 309
- participants
 - cross-site 138
 - described 128
 - determining how selected 138
- paths
 - for links within exported files 234
 - precedence of path information for export 233
- permissions
 - ACL 306
- planning

- deadlock actions 144
- delegate actions 142
- e-mail objects 141
- function privileges 151
- group deadlock actions 145
- roles 137
- step actions 143
- step conditions 144
- timed actions 142
- workflow processes 137
- workflow states 145
- workflow steps 146
- Preserve formatting option 199
- preview
 - separate browser sessions for 215
- Pricer role 330
- process, workflow. *See* workflow.
- properties
 - asset marked as not approved on destination after publish 260
 - security 86
- proxy servers
 - and Mirror to Server 254
- PubKey table 230
- Publish Console 275
 - examining publishing history 276
- publish events 225
 - creating 269
 - editing 271
 - example 270
 - overriding 272
 - reading the schedule abbreviations 270
- publish point. *See* export starting point.
- published references
 - Export to Disk 277
 - Mirror to Server 277
- PublishedAssets table 229, 230
- publishing
 - See also* publishing arguments, publishing destinations, publishing schedule, publishing session, and publish events.
 - all approved assets, regardless of publish status 278
 - configuring a system to publish 245
 - defined 217
 - dependencies 220
 - directory path for exported files 231
 - directory path for exported XML files 244

- enable error logging with VERBOSE 232
- error messages 278
- Export Assets to XML destinations 262
- export destinations 247
- export directory for Export Assets to XML 261
- export directory for Export to Disk 246
- history 276
- history log files 278
- mapping URL prefixes 248
- messages 278
- Mirror to Server destinations 254
- setting mirror usernames and passwords 253
- testing sites 250, 263
- troubleshooting 278
- publishing arguments
 - Export to Disk 230
 - Mirror to Server 240
- publishing destinations
 - approving multiple assets 266
 - creating for Export to Disk 247
 - creating for Export to XML 262
 - creating for Mirror to Server 254
 - described 219
 - editing 268
 - initializing 255
- publishing errors
 - assets with associations and DB2 284
 - cannot contact the destination 281
 - cleanup failed 281
 - complex or flex asset could not be delivered 281
 - complex or flex asset could not be saved on destination 282
 - incorrect ACLs for mirror user 279
 - incorrect username or password for mirror user 279
 - mirror destination is incorrect 280
 - missing table on destination 280
 - pages not refreshed after publish 283
 - session does not end 282
 - session failed 281
 - system stops 283
- publishing method. *See* delivery type.
- publishing schedule 225, 269
 - See also* publish events.
 - interpreting 270
- publishing sessions

- amount of disk space needed for 283
- described 227
- displayed in Publish Console 275
- publishing system
 - described 218
 - export queue 229
 - firewalls 254
 - interactions with approval system 220
 - mirror queue 238
 - testing results for Mirror to Server 259
- PubMessage table 278
- PubSession table 279
- PubSessionID
 - in publish history files 278
 - in PubSession table 279
- purging
 - inactive data 322
 - session objects (scratch) table 322

Q

- queries
 - for approved assets 227
- queues
 - export publish queue 229
 - mirror publish queue 238

R

- references
 - published references 277
- RemoteClient ACL 309
- REMOTEPASS publishing argument 240
- REMOTEUSER publishing argument 240
- resolving
 - deadlocks 131
 - links 230
- revision tracking
 - and non-asset tables 288
 - changing the number of versions to store 290
 - defined 285
 - disabling for asset types 292
 - disabling for non-asset database tables 292
 - editing settings 290
 - enabling for asset types 289
 - enabling for non-asset database tables 289
 - implicit and explicit checkouts and

- checkins 288
- root storage directory 290
- unlocking assets 293
- unlocking rows in non-asset tables 293
- roles
 - and users for sites 98
 - assigning to users 97
 - Burlington Financial 335
 - Burlington Financial Extension 347
 - creating 70
 - deleting 71
 - determining for workflow 137
 - editing 71
 - GE Lighting 341
 - Hello Asset World 331
 - needed by administrators 313
 - relationship with ACLs 63, 65
 - required to configure workflow 136
 - start menu items 106
- rollback
 - defined 286

S

- sample sites
 - workflow processes 152
- Satellite Server
 - symptoms when not configured
 - correctly 283
- scalar objects 321
- scheduling
 - publish events 225
 - timed action event 164
- search
 - start menu items 112
- search engines 298
 - disabling asset types for 301
 - enabling asset types for 300
 - indexing differences between asset types 298
 - rebuilding indexes for asset types 301
- Searching option 298
- security
 - access control for assets 151
 - ACLs 84
 - BlobServer 85
 - cc.security property 84
 - changing default passwords 87

- configuring 86
- DefaultReader 85
- disabling CS-Direct forms on delivery system 88
- function privileges and assets 133
- goals for various systems 86
- properties 86
- testing 89
- URLs 88
- separate browser sessions for preview 215
- setting field values
 - start menu items 111
- SIMPLEDIR publishing argument 231, 233
- SIMPLENAME publishing argument 231, 235
- simplified access control 151
- single-language restrictions 196
- Site Admin tab 45, 107
 - xceladmin ACL 310
- site designers
 - ACLs they need 312
- Site Plan tab 107
- SiteCatalog table
 - ACLs 64
 - file names for exported assets 235
- SiteGod ACL 309
- sites
 - creating 92
 - deleting 95
 - disabling asset types for 102
 - editing 95
 - enabling asset types for 100
 - roles 69
- sources 120
 - Burlington Financial 337
 - defined 220
- Specify Path/Filename link 249
- start menu items
 - Burlington Financial 338
 - Burlington Financial Extension 350
 - Content Server Desktop 202
 - Content Server DocLink 211
 - creating for the New list 108
 - defined 106
 - field values set by 106
 - for Search list 112
 - GE Lighting 344
 - Hello Asset World 332
 - setting field values with 111

- start step
 - See also* steps.
 - creating 170
 - described 129
- states
 - creating 166
 - deleting 167
 - described 128
 - editing 167
 - From State 129, 146
 - planning 145
 - To State 129, 146
- step actions 142
 - Approve for Publish 143
 - configuring Approve for Publish 163
 - planning 143
 - Send Assignment Email 143
 - Send Rejection Email 143
- step conditions
 - creating 162
 - described 143
 - Example Step Condition 144
 - planning 144
- steps
 - actions 129, 139, 142
 - all-voting step 130
 - conditions 129
 - creating 171
 - deleting 176
 - described 129
 - editing 175
 - end step 172
 - planning 146
 - start step 129
 - start steps 170
 - synchronize step 132
- SUFFIX publishing argument 231, 235
- SystemACL table 306
- SystemInfo table 63

T

- TableEditor ACL 309
- tables
 - VMVISITOR 321
 - VMVISITORALIAS 321
 - VMVISITORSCALARVALUE 321
 - VMVISTORSCALARBLOB 321

- VMz 321
- tabs, tree. *See* tree tabs.
- target. *See* publishing destinations.
- targets
 - variable 139, 163
- testing
 - security measures 89
- The 124
- timed action event
 - scheduling 164
- timed actions 139
 - described 141
 - examples 128
 - planning 142
 - Send Email 141
- To State 129, 146
- tree tabs
 - Burlington Financial 340
 - Burlington Financial Extension 352
 - creating 122
 - default 107
 - defined 107
 - deleting 125
 - editing 124
 - example of custom tab 108
 - GE Lighting 345
 - Hello Asset World 334
 - roles 69
 - sort order of 125
 - workflow groups 124
- troubleshooting
 - locale problems 195
 - publishing 278

U

- unknown dependencies
 - Export to Disk 229
- URLPREFIX argument 230
- URLs
 - determining so you can test a site 259
 - in HREFs of exported pages 234
 - mapping URLs for specific servlets 88
- user attributes 80
 - adding 82
- user management
 - and sites 22
 - overview of concepts 62

- user profiles 80
 - creating 80
- user_anaylst 327
- user_approver 327
- user_author 328
- user_checker 328
- user_designer 328
- user_editor 328
- user_expert 328
- user_marketer 328
- user_pricer 328
- UserEditor ACL 309
- UserReader ACL 310
- users
 - and ACLs 63
 - and user profiles 80
 - assigning roles to users 97
 - batch user for publishing 245
 - Burlington Financial 335
 - Burlington Financial Extension 347
 - changing default users for security purposes 87
 - creating 77
 - deleting 79
 - editing 78
 - GE Lighting 341
 - granting users access to sites 97
 - Hello Asset World 331
 - list for a site 98
 - mirror user 252
 - role in user management 62

V

- variables
 - emailname 139
 - for workflow e-mail objects 140
 - targets 139, 163
- VERBOSE publishing argument 232, 240
- version number 286
- vis.genclasspath property 323
- Visitor ACL 310
- visitor attributes
 - binary 321
 - where stored 321
- visitor data
 - inactive 322
- VisitorAdmin ACL 310

- visitors
 - IDs 321
- VMVISITOR table 321
- VMVISITORALIAS table 321
- VMVISITORSCALARVALUE table 321
- VMVISTORSCALARBLOB table 321
- VMz tables 321
- voting
 - described 129

W

- web server alias 230
- Word
 - templates for Content Server Desktop 205
- workflow
 - See also*, actions, conditions, deadlocks, states, steps, timed action event, and workflow processes.
 - actions 128, 129
 - assignment 128
 - conditions 129
 - configuring 159
 - deadlock 130
 - delegate actions 132
 - described 128
 - elements 139
 - e-mail objects 139
 - e-mail variables 140
 - ending 136
 - function privileges 133
 - group deadlock action 132
 - groups 132
 - moving your work 178
 - participants 128
 - placing an asset in workflow 133
 - planning 137
 - process 128
 - required roles to configure 136
 - roles 69
 - sample site processes 153
 - set with a start menu item 106
 - states 128
 - steps 129
 - testing your process 177
 - variables 139
 - voting 129
- Workflow Groups tab

- creating 124
- workflow process
 - creating 168
 - deadlines 134
 - defined 128
 - deleting 176
 - editing 174

- Workflow tab 107
 - xceladmin ACL 310

X

- xceladmin ACL 310
- xceeditor ACL 310