

Getting Started with Content Server
10g Release 3 (10.1.3.3.0)

March 2007

Getting Started with Content Server, 10g Release 3 (10.1.3.3.0)

Copyright © 2007, Oracle. All rights reserved.

Contributing Authors: Sandra Christiansen, Ron van de Crommert

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

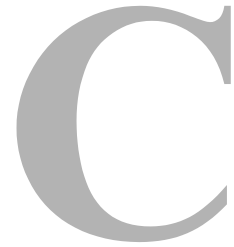
U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents



Chapter 1: About This Guide

About This Guide	1-1
------------------------	-----

Chapter 2: Content Management

Overview	2-1
About Content Management	2-2
Universal Content Management Architecture	2-4
Repository	2-5
Core Services	2-5
Library Services	2-5
Security	2-6
Personalization	2-6
Workflow	2-7
Conversion	2-7
User Input	2-7
Search	2-8
Administration	2-8
Application Modules	2-8
Web Content Management	2-9
Document Management	2-10
Records Management	2-12
Digital Asset Management	2-13
Integration	2-14
Integration Methods	2-14
Content Integration Suite	2-15
Content Portlet Suite	2-15
Content Portlet Suite for WebLogic	2-15
Content Portlet Suite for WebSphere	2-16
Content Portlet Suite for Plumtree	2-16
Content Portlet Suite for Sun ONE	2-16

Chapter 3: Content Server Overview

Overview	3-1
About Content Server.....	3-2
Hardware and Software Requirements	3-2
Basic Architecture	3-3
Content Server.....	3-5
Web Server	3-5
Content Information Database.....	3-6
Search Collection.....	3-6
Native File Repository (or “Vault”).....	3-7
Web-Viewable File Repository (“Web Layout”).....	3-7
Web Browser.....	3-8
Main Interfaces.....	3-8
Master Servers and Proxied Servers.....	3-11
Search Options	3-12
File Conversion	3-13
Component Architecture	3-13
Clustering.....	3-14
Integration Options.....	3-15
Portals and Portlets	3-17

Chapter 4: Documentation

Overview	4-1
Online Technical Newsgroup.....	4-1
Documentation.....	4-1
Online Help System.....	4-2
PDF Files.....	4-2

Chapter 5: Content Server Security

Overview	5-1
Users	5-3
User Types	5-3
Authorization Types.....	5-4
Predefined User Logins.....	5-6
Defining and Managing Users.....	5-6
Security Groups.....	5-6

Using Security Groups	5-7
Predefined Security Groups	5-7
Defining and Managing Security Groups	5-7
Permissions	5-8
Roles	5-8
Assigning Roles	5-9
Predefined Roles	5-11
Accounts	5-12
Using Accounts	5-12
Accounts Scenario	5-13
Accounts and Permissions	5-14
Account Hierarchy	5-15
Special Account Classifications	5-16
Assigning Accounts	5-16
Administrators and Sub-Administrators	5-16
Using External Security	5-18
Self-Registration	5-19
Setting Up Security	5-20
Security Recommendations	5-20
File System Access to the Content Server Directory Structure	5-21
Network Access	5-21
Database Access	5-22
Physical Access	5-22
Chapter 6: Starting and Stopping a Content Server	
Overview	6-1
Microsoft Windows	6-1
Startup Options	6-2
Starting a Content Server	6-2
Stopping a Content Server	6-4
Restarting a Content Server	6-5
UNIX	6-5
Starting a Content Server	6-6
Stopping a Content Server	6-7
Restarting a Content Server	6-7
Chapter 7: Workflow	
Overview	7-1

Workflows	7-1
Workflow Types	7-2
Workflow Steps	7-3
Jumps	7-3
Tokens	7-4
Aliases	7-4
Templates	7-4

Chapter 8: Working With Content

Overview	8-1
Grouping of Content	8-2
Content Information (Metadata)	8-3
Content Identification	8-4
Content Item Identifiers	8-4
File Naming	8-6
Maximum File Sizes	8-7
Content Storage	8-8
Storage of Content Files	8-8
Storage of Metadata	8-9
Storage of Full Text	8-10
Content Profiles	8-10
Content Subscriptions	8-11
File Subscriptions	8-11
Criteria Subscriptions	8-12
Checking In Files	8-12
Checking Out Files	8-14
Updating Content	8-16
Searching for Content	8-17
Navigating to Content	8-17
Searching Content Information (Metadata)	8-17
Metadata Search Operators	8-18
Wildcards	8-20
Performing a Full-Text Search	8-21
Full-Text Search Operators	8-21
Using Repository Manager to Look Up Content	8-22

Chapter 9: Administering Content Server

Overview	9-1
--------------------	-----

Administration Tools	9-2
Reviewing the System Settings	9-3
Admin Server	9-4
System Properties	9-5
User Admin	9-6
Batchloader	9-6
Repository Manager	9-8
Archiver	9-9
Exporting Content	9-10
Importing Content	9-10
Transferring Content	9-11
Local Transfer	9-12
Pull Transfer	9-12
Push Transfer	9-12
Replicating Content	9-12
Content Server Analyzer	9-13
Environment Packager	9-14
Log Files	9-14
Backup Strategy	9-15
Using Backups	9-16
Backup/Recovery Methods	9-16
Disaster Recovery	9-18

Chapter 10: Customization and Personalization

Overview	10-1
Using Interface Layouts and Skins	10-2
Providing Localized User Interfaces	10-4
Customizing the Portal Page	10-5
Personalizing the Interface	10-6
Creating the Library Hierarchy	10-6
Using Custom Components	10-7

Glossary

Index

ABOUT THIS GUIDE

ABOUT THIS GUIDE





This *Getting Started With Content Server* guide aims to provide an overview of Content Server, its capabilities, and its place in the spectrum of Oracle solutions. More in-depth information is available in other product documentation and online help.



Important: The information in this document applies to 10gR3 releases of Content Server. Some of the information may not apply to earlier releases.

Symbols

The following symbols are used throughout this document:

Symbols	Description
	This is a note. It is used to bring special attention to information.
	This is a technical tip. It is used to identify information that can be used to make your tasks easier.
	This is an important notice. It is used to identify a required step or required information.
	This is a caution. It is used to identify information that might cause loss of data or serious system problems.

CONTENT MANAGEMENT

OVERVIEW

Oracle provides a single, unified architecture that allows organizations to deploy web content management, document management, records management and digital asset management applications on one platform. The Oracle content management architecture is scalable and flexible enough to support long-term content management strategies, while the single-architecture approach ensures all the components of the system are truly interchangeable, extensible and complementary.

This section covers the following topics:

- ❖ [About Content Management](#) (page 2-2)
- ❖ [Universal Content Management Architecture](#) (page 2-4)
- ❖ [Repository](#) (page 2-5)
- ❖ [Core Services](#) (page 2-5)
- ❖ [Application Modules](#) (page 2-8)
- ❖ [Integration](#) (page 2-14)

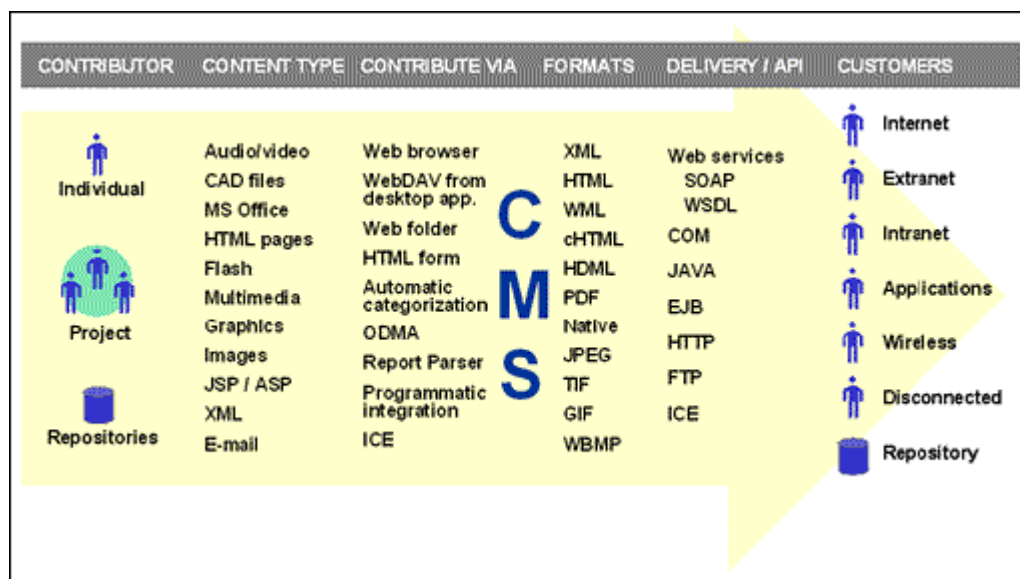
See also:

- [Chapter 3 \(Content Server Overview\)](#)

ABOUT CONTENT MANAGEMENT

The key to a successful content management implementation is unlocking the value of content by making it as easy as possible for it to be consumed. This means that any piece of content must be available to any consumer, no matter what their method of access. The implications of this simple statement are broad. It means that any piece of content, web or business, created in any application in any format, must be managed in any format, but made available for consumption on any device via the native application or a browser or wireless viewer via any standard delivery API. Figure 2-1 illustrates this “any-to-any” content management concept.

Figure 2-1 “Any-to-any” content management



As you can see from this graphical representation, this means that any contributor, group of contributors, or other repository must be able to contribute any content type via a variety of contribution methods chosen by the contributor (not mandated by an IT department or system limitations) into a managed environment (the content management system). That content’s value is then unlocked by allowing any consumer to access that content in any format, via any standard delivery method, via any access device (internet, extranet, intranet, another application, a wireless device, a disconnected business user via a laptop, or another repository).

The basic purpose of content management is to establish value around information. Information only has value when the right person can access the right information/content at the right time in a format they can use. Therefore, content management is all about *consuming* the content or information and not as much about *creating* the content or simply *storing* it.

In the content continuum as shown in Figure 2-1, we see a variety of contributors on the left side of the diagram and a number of consumers on the right. Content management is all about getting the content from the contributors to the consumers. At the center, we see the content management server (CMS) or repository, which provides a managed environment for storage, version control, retrieval, and conversion of content.

Once the content is in the managed environment, there are several choices to get it out, but the content must first be converted to a format that fits the desired delivery. The native file is stored in Content Server and can be converted to XML, HTML, cHTML, WML, or a variety of other formats depending on the application needs. Over 390 file formats can be converted to one or more of these formats.

Files may be converted from their native format to various other formats to enable delivery to a variety of devices. The native file is the single authoritative source; it can be modified once and output to several devices.

Once transformed to the appropriate format(s), the content can be delivered using any of a number of methods: SOAP, WebDAV, COM, Java, etc. Content Server complies with industry standards, providing the flexibility to choose which delivery/API mechanism makes the most sense for any particular situation.

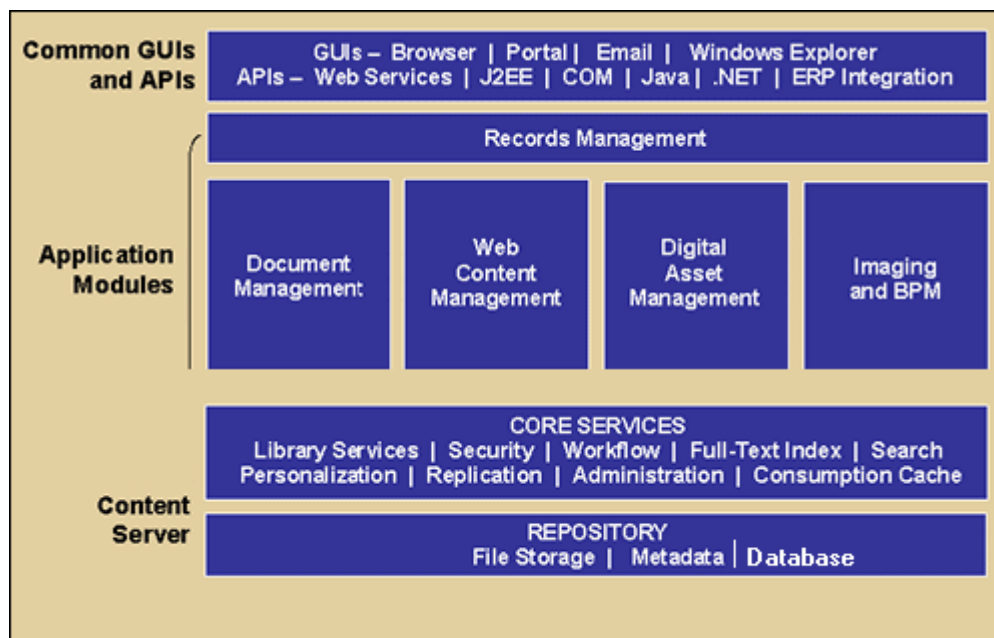
Finally, the content is delivered to the consumer via a website, another application, or a handheld device. The content may also be synchronized for offline use or delivered to another repository.

There is no one right way to move content from creator to consumer. Therefore, Oracle offers a wide variety of ways to match virtually any business need.

UNIVERSAL CONTENT MANAGEMENT ARCHITECTURE

shows the architecture of Oracle content management.

Figure 2-2 Content Management architecture



The foundation of Content Server begins with a completely web-based [repository](#) (see page 2-5), where all content and its associated metadata is stored for management, reuse, and access.

While stored in the repository, all types of content—ranging from e-mail, discussions, documents, reports, spreadsheets and records to images, multimedia or other digital formats—receive the same set of fundamental [core services](#) (see page 2-5).

The repository and core services are offered by [Content Server](#) (see chapter 3).

Built on top of the core services are five key [application modules](#) (see page 2-8). All modules draw from the same set of core services but then offer additional services specific to the type of content being managed.

Common GUIs and APIs enable [integration](#) of Content Server functionality with enterprise applications such as application servers, catalog solutions, personalization applications, and enterprise portals, as well as client-side software (see page 2-14).

REPOSITORY

The foundation of Oracle Universal Content Management begins with a completely web-based repository or database repository, where all content, regardless of content type, is stored for management, reuse and access.

The content files are stored on a file system or database in a [native file repository](#) and a [web-viewable file repository](#), and the [content information \(metadata\)](#) in a [database](#).



Note: See [Chapter 3 \(Content Server Overview\)](#) for more information about the native file repository, web-viewable file repository, and the content information database.

CORE SERVICES

There are a number of basic services that any content management product must provide regardless of the business problem being solved:

- ❖ [Library Services](#) (page 2-5)
- ❖ [Security](#) (page 2-6)
- ❖ [Personalization](#) (page 2-6)
- ❖ [Workflow](#) (page 2-7)
- ❖ [Conversion](#) (page 2-7)
- ❖ [User Input](#) (page 2-7)
- ❖ [Search](#) (page 2-8)
- ❖ [Administration](#) (page 2-8)

All these core services are offered by [Content Server](#) (see chapter 3).

Library Services

The core library services include the following:

- A robust metadata model to provide organizations with a method to better organize and add additional structure to managed content.
- Check-in/out functionality to ensure that only one person at a time can make changes to content.

- Revision control to allow organizations to easily track new revisions and provide users with the ability to roll back to previous versions.
- Subscription capabilities to enable users to subscribe and receive automatic e-mail notifications when items change or when new content is added to the system.
- Release and expiration schedules to

See chapter 8 for more information about working with content.

Security

Role and account-based security models enable administrators and content authors to define which users may access which content, allowing organizations to flexibly secure different levels of content while providing speed and performance for the end-user.

Rule-based security uses a complex set of rules and custom logic to secure sensitive information, including both web content and standard business documents, and to determine the appropriate authorizations and permissions.

In addition, Content Server may authenticate against a centralized security model such as LDAP, Active Directory, NDS, and Netegrity. Security may be simple or complex depending on the application needs.

See chapter 5 for more information about Product_Name security.

Personalization

Basic personalization is included in Content Server. More involved personalization can be created on this foundation to provide customers with personalization tailored to their specifications. Personalization may be:

- ❖ Security-defined, where the user's role defines how they view the content.
- ❖ User-defined, where the user selects preferences to define their personalized experience.
- ❖ Administrator-defined, where the administrator predefines what the end-users experience will be.
- ❖ Rules-based, where rules are created to make personalization decisions based on the users' metadata information (for example, a user's department defines the look and feel so that someone in Sales has a different view than someone in R&D).

See chapter 10 for more information about customization and personalization.

Workflow

Workflows are useful in the process of reviewing and approving content before it is released and published to the website. They specify how content is routed and who needs to review and approve it. Workflows also integrate to external business process for data look-up, translation, and other external needs.

See chapter 7 for more information about workflows.

Conversion

Content managed by Content Server can be converted from its native format to a number of other formats that can be viewed in industry-standard web browsers, including HTML, XML, WML, cHTML, and PDF. Several [modules](#) are available for Content Server that provide conversion capability.

User Input

It needs to be easy for users to interface with the content repository. Oracle supports a variety of methods, including:

- ❖ **Web browser**—Users may check content in and out through a check-in page in a standard web browser.
- ❖ **WebDAV** (Web-based Distributed Authoring and Versioning) provides the ability for a contributor to drag and drop content into the repository through the Windows Explorer interface. Basic functionality is available through the server side installation with additional features found in the client installation.
- ❖ **Direct linkage** between client applications and a Oracle content server instance.
- ❖ **E-mail integration**, which allows direct contribution from Microsoft Outlook, including e-mail messages and attachments. Users may also attach content from Content Server to an e-mail message or include a link to the content in the e-mail message.
- ❖ **ODMA** (Open Document Management API), which allows direct contribution from ODMA-compliant applications such as Word, WordPerfect, Framemaker, and PowerPoint. ODMA functionality may be connected to multiple servers to retrieve and store information from/to a choice of instances.

Search

Content items can be indexed immediately upon check-in, resulting in quick and easy retrieval with full-text and metadata searches. A variety of search solutions can be used with Content Server:

- ❖ Content database (default, out-of-the-box configuration)
- ❖ Verity search engine (with Verity Integration, sold separately)
- ❖ FAST search engine (with FAST Integration, sold separately)

An optional Content Server feature is available which extends the search capability to support searches for content across multiple content server instances (“enterprise search”).

See page 3-12 for more information about the available search options.

Administration

Administration is accomplished through web-based applets. If the administrator has appropriate security clearance through firewalls and other security measures, Content Server may be administered from anywhere in the world accessible through one web-based user interface. Content Server also offers multi-site management and clustering capability.

Administration tasks that can be performed include configuration review and modification, user administration, repository management, server management, content archiving and transferring, audit trails and system reports, and back-up and recovery.

Administration services such as archiving and removal, audit trails and system reports, and back-up and recovery capabilities are available for all content management applications, using one web-based user interface.

See chapter 9 for more information about administering Oracle content servers.

APPLICATION MODULES

Content Server provides features to manage all phases of the content life cycle: from creation and approval to publishing, retrieval, searching, expiration, and archival or disposition. In addition, five key application modules are available that further extend the capabilities of Content Server. All modules draw from the same set of [core services](#) (see page 2-5), but offer additional functionality specific to the type of content being managed:

- ❖ **Web content management** (see page 2-9)
This application module provides additional services necessary for creating, managing, and publishing web content, and providing an infrastructure to support any number of websites. The requirements in this area include site design, publishing, templates, in-context editing, previews, site migration, publishing workflows, and deployment.
- ❖ **Document management** (see page 2-10)
This application module provides additional services necessary for capturing, securing, and sharing digital and paper-based documents and reports. The requirements in this area include PDF conversion, automatic input, scanning, categorization, document merging, watermarking, CAD drawing, report distribution, and XML integration.
- ❖ **Records management** (see page 2-12)
This application module offers either a DoD 5015.2-certified solution that provides additional services specific to creating, declaring, classifying, retaining, and destroying records based on active content or a corporate retention/disposition product. The requirements in this area include archiving and retrieval, disposition, certifications, and release/expiration management.
- ❖ **Digital Asset management** (see page 2-13)
This application module provides additional services focused on the unique needs of managing and providing access to rich media and digital assets. The requirements in this area include compression, audio and video indexing, streaming video handling, thumbnailing, and image conversion.

Web Content Management

Web Content Management is a suite of functionality built upon the [Content Server](#) foundation, offering additional functionality necessary to create, manage and publish websites and portal content. Web Content Management offers flexibility for website developers and designers. The applications and tools that you decide to use within the suite may vary depending on how you would like to build your Web sites as well as the skill-set of your site designers. The desired architecture of your final website, its audience (consumers), and the functionality you would like to offer your content contributors will also impact your selection.

Oracle's core products for web content management include the following:

- ❖ **Content Server**
[Content Server](#) is a flexible, secure, centralized, web-based repository that manages

all phases of the content life cycle, from creation and approval to publishing, searching, expiration, and archival. It provides all [core services](#) such as library services (check-in, check-out, version control, subscriptions, release and expiration schedules), security, personalization, workflow, conversion, user input, search, and administration.

❖ **Site Studio**

Site Studio is the application most often used for designing websites. Site Studio offers site developers and designers a built-in methodology for building websites as well as a customizable library to easily reuse custom code and fragments—enabling them to create and deploy robust websites quickly. Additionally, Site Studio enables companies to design and develop websites for dynamic contribution and viewing, as well as publish these sites as static representations built and delivered with HTML code on standard web servers.

❖ **Dynamic Converter**

Dynamic Converter is the application used for on-demand, ad-hoc conversion of native business documents to your information portal, Internet, intranet and extranet sites. Dynamic Converter is often used in conjunction with Site Studio to build robust websites built on form-based as well as native content contribution. With conversion capability for more than 225 file formats (including XML) to HTML, JPEG GIF, and wireless formats, Dynamic Converter brings a new level of simplicity and productivity to providing web-viewable business content. Criteria-based rules determine the appropriate templates used to deliver consistent look and feel for corporate business content.

❖ **Content Publisher**

Content Publisher is the application used when site developers want more granular control of the presentation layer of a website. With Content Publisher, site developers can design and develop their own website creation and contribution methodologies for sites. This application lends itself well for converting and publishing volumes of well formatted and web-viewable content while allowing web developers to use another framework for site design, such as a portal or commerce server.

Document Management

Document Management is a bundle of applications built upon the [Content Server](#) foundation, offering additional functionality necessary to effectively and efficiently capture, secure, share, and distribute digital and paper-based documents and reports. With Oracle's document management products, organizations can effectively and efficiently capture, secure, share, and distribute digital and paper-based documents and reports. They

can save money and create operational efficiency by streamlining communications, automating routine tasks and lowering costs related to the printing, shipping and storage of business documents. Additionally, Oracle can help reduce risk and lower costs associated with a variety of regulatory and legal compliance processes.

Oracle's core products for document management include the following:

❖ **Content Server**

[Content Server](#) is a flexible, secure, centralized, web-based repository that manages all phases of the content life cycle, from creation and approval to publishing, searching, expiration, and archival. It provides all [core services](#) such as library services (check-in, check-out, version control, subscriptions, release and expiration schedules), security, personalization, workflow, conversion, user input, search, and administration.

❖ **PDF Converter**

PDF Converter enables the automatic publishing of native business content to the web-viewable PDF format. Upon check-in of new business content, a PDF rendition of the native format is generated and stored in the repository. Content may be viewed without the need for the native application.

❖ **Dynamic Converter**

Dynamic Converter is the application used for on-demand, ad-hoc conversion of native business documents to your information portal, Internet, intranet and extranet sites. Dynamic Converter is often used in conjunction with Site Studio to build robust websites built on form-based as well as native content contribution. With conversion capability for more than 225 file formats (including XML) to HTML, JPEG GIF, and wireless formats, Dynamic Converter brings a new level of simplicity and productivity to providing web-viewable business content. Criteria-based rules determine the appropriate templates used to deliver consistent look and feel for corporate business content.

❖ **Desktop**

This is software that is installed on a user's desktop to allow direct linkage between client applications and a Content Server instance. It also provides integration with Microsoft Outlook, which allows direct contribution from Microsoft Outlook, including e-mail messages and attachments. Users may also attach content from Oracle to an e-mail message or include a link to the content in the e-mail message.

Records Management

Records Management is are applications built upon the [Content Server](#) foundation, offering additional functionality specific for creating, declaring, classifying, retaining and destroying records. It can also serve as a complementary technology solution to any existing records management program.

Oracle's core products for records management include the following:

❖ **Content Server**

[Content Server](#) is a flexible, secure, centralized, web-based repository that manages all phases of the content life cycle, from creation and approval to publishing, searching, expiration, and archival. It manages the content [repository](#), and provides all [core services](#) such as library services (check-in, check-out, version control, subscriptions, release and expiration schedules), security, personalization, workflow, conversion, user input, search, and administration.

❖ **Records Manager**

Records Manager enables enterprises to manage the retention and disposition of records and non-record content—regardless of their source or format—using a single, consistent, manageable infrastructure. Records and non-record content managed by Records Manager are assigned retention schedules and disposition rules that control their lifecycle. The items and their associated metadata are stored in file plans, which are hierarchies with categories that define disposition instructions for records and non-record content. Access to the items is controlled by rights assigned to users by a records administrator. The items can be accessed, reviewed, retained, or destroyed in an easy and efficient manner, by authorized people according to the requirements of your organization

❖ **Physical Content Manager**

Physical Content Manager extends Record Manager's retention and disposition capabilities to include physical records and non-record content that are not stored in the content server in electronic form. The same file plan and retention schedules are used for both electronic (internal) and physical (external) records and non-record content.

Physical Content Manager also keeps track of the storage locations and retention schedules of the physical records and content. The application provides space management, circulation services, barcode file processing, label creating and printing, and retention management.

❖ Oracle Desktop

This is software that is installed on a user's desktop to allow direct linkage between client applications and a Content Server instance. It also provides integration with Microsoft Outlook, which allows direct contribution from Microsoft Outlook, including e-mail messages and attachments. Users may also attach content from Oracle to an e-mail message or include a link to the content in the e-mail message.

Digital Asset Management

Digital Asset Management is a suite of applications built upon the [Content Server](#) foundation, offering additional functionality to effectively manage rich media files. With Digital Asset Management, normal business users can find, view and use digital assets quickly and easily. Additionally, when the system provides secured web-based access, users across an organization can quickly and easily access, share, reuse and modify these digital assets.

Oracle's core products for digital asset management include the following:

❖ Content Server

[Content Server](#) is a flexible, secure, centralized, web-based repository that manages all phases of the content life cycle, from creation and approval to publishing, searching, expiration, and archival. It manages the content [repository](#), and provides all [core services](#) such as library services (check-in, check-out, version control, subscriptions, release and expiration schedules), security, personalization, workflow, conversion, user input, search, and administration.

❖ Image Manager

Image Manager enables users to quickly find, group, and download images of various sizes and resolutions. For example, an organization's logo may need to be available in a variety of sizes for advertisements, web pages, and presentation. At check-in, the image is automatically converted into the defined formats and sizes. Users can then search for the image using standard metadata, group renditions into a content basket, and download a single compressed file of the image renditions they need.

❖ Oracle Video Manager

Video Manager enables users to quickly find, group, and download videos of various sizes and resolutions. For example, a company training video may need to be available in a variety of sizes for streaming on an intranet, presenting to an audience, or copying to tape. At check-in, the video is automatically converted into the defined formats and sizes. Users can then search for the video using standard metadata, group renditions into a content basket, and download a single compressed file of the video renditions they need.

INTEGRATION

Oracle provides easy, flexible [methods for integrating](#) its content server functionality with enterprise applications such as application servers, catalog solutions, personalization applications, and enterprise portals, as well as client-side software.

In addition, a reliable and scalable [integration infrastructure](#) is available for integrations with Java 2 Enterprise Edition (J2EE) applications, as well as a [portlet suite](#) which enables you to integrate functionality into various portal servers.

Content Server's integration and scalability options are standards-based and allow for current and future expansion plans—regardless of whether the system is used as a stand-alone implementation or as part of a larger content management infrastructure to support multiple enterprise applications.

Integration Methods

The Content Server n-tier, web-based open architecture enables organizations to integrate content and hundreds of documented services (such as checking in content, performing a search, returning search results, or approving an item in workflow) using standard integration methods such as:

- ❖ Web services
- ❖ COM (Component Object Model)
- ❖ HTTP (HyperText Transfer Protocol)
- ❖ XML (Xtensible Markup Language)
- ❖ ICE (Information & Content Exchange)
- ❖ SOAP (Simple Object Access Protocol)
- ❖ Scripting languages
- ❖ J2EE (Java 2 Platform, Enterprise Edition)
- ❖ JSP tag library (Java Server Page)

See page 3-15 for more information about the integration options.

Content Integration Suite

The Content Integration Suite (CIS) provides a reliable and scalable integration infrastructure for integrations with enterprise applications. CIS represents an integration point for any J2EE application. It consists of the Content Server command layer (EJBs, IdcServer Bean, Commands, and Adapter), Content Server JavaBean, and the Oracle JSP tag libraries.

The Content Integration Suite provides the best architecture for combining enterprise content management with your existing business infrastructure. With high-performance, powerful integration capabilities that are easy to integrate, Oracle enables rapid development and a faster return on your investment.

Content Portlet Suite

Content Portlet Suite is built on top of the Content Integration Suite and offers a number of pre-built reference portlets for the [BEA WebLogic](#), [IBM WebSphere](#), [Plumtree](#), and [Sun ONE](#) portal servers. Content Portal Suite provides access to content stored in the content server, enabling users to update, search, and view portal content in a way that is efficient and easy to use.

Oracle provides the ability to manage the content creation and distribution process through the use of portlets. These portlets can be enabled for different users based upon the user's roles and permissions within the organization. Depending on the permission level, the user may be allowed to browse or search content, contribute a new content item and view the progress of their content through workflow.

Integrating your portal and Content Server with Content Portal Suite provides an easier way to keep the portal up to date—resulting in greater utilization, lower maintenance costs and a larger your return on your portal investment.

Content Portlet Suite for WebLogic

Content Portlet Suite (CPS) for WebLogic provides a reliable and scalable integration with BEA WebLogic Portlet Server. This suite leverages the [Content Integration Suite](#) (CIS) as the foundation layer for integration with WebLogic. The suit provides eight functional reference portlets (Library, Authenticated Library, Basic Search, Authenticated Search, Saved Search, Contribution, Workflow Queue, and Metadata Administration) which can be used immediately, or as examples of how to implement WebLogic portlets with Content Server and CIS.

Content Portlet Suite for WebSphere

Content Portlet Suite (CPS) for WebSphere provides a reliable and scalable integration with IBM WebSphere Portlet Server. This suite leverages the [Content Integration Suite \(CIS\)](#) as the foundation layer for integration with WebSphere. The suite provides eight functional reference portlets (Library, Authenticated Library, Basic Search, Authenticated Search, Saved Search, Contribution, Workflow Queue, and Metadata Administration) which can be used immediately, or as examples of how to implement WebSphere portlets with CIS.

Content Portlet Suite for Plumtree

Content Portlet Suite (CPS) for Plumtree provides a reliable and scalable integration with Plumtree Portlet Server. This suite leverages the [Content Integration Suite \(CIS\)](#) as the foundation layer for integration with Plumtree. The suite provides eight functional reference portlets (Library, Authenticated Library, Basic Search, Authenticated Search, Saved Search, Contribution, Workflow Queue, and Metadata Administration) which can be used immediately, or as examples of how to implement Plumtree portlets with CIS.

Content Portlet Suite for Sun ONE

Content Portlet Suite (CPS) for Sun ONE provides a reliable and scalable integration with the Sun ONE Integration Server. This suite leverages the [Content Integration Suite \(CIS\)](#) as the foundation layer for integration with Sun ONE. The suite provides eight functional reference portlets (Library, Authenticated Library, Basic Search, Authenticated Search, Saved Search, Contribution, Workflow Queue, and Metadata Administration) which can be used immediately, or as examples of how to implement Sun ONE portlets with CIS.

CONTENT SERVER OVERVIEW

OVERVIEW

This section covers the following topics:

- ❖ [About Content Server](#) (page 3-2)
- ❖ [Hardware and Software Requirements](#) (page 3-2)
- ❖ [Basic Architecture](#) (page 3-3)
- ❖ [Main Interfaces](#) (page 3-8)
- ❖ [Master Servers and Proxied Servers](#) (page 3-11)
- ❖ [Search Options](#) (page 3-12)
- ❖ [File Conversion](#) (page 3-13)
- ❖ [Component Architecture](#) (page 3-13)
- ❖ [Clustering](#) (page 3-14)
- ❖ [Integration Options](#) (page 3-15)
- ❖ [Portals and Portlets](#) (page 3-17)

ABOUT CONTENT SERVER

Content Server is the foundation for a variety of Oracle content management products (see chapter 2). It provides a flexible, secure, centralized, web-based repository that manages all phases of the content life cycle: from creation and approval to publishing, searching, expiration, and archival or disposition. Every contributor throughout the organization can easily contribute content from native desktop applications, efficiently manage business content via rich library services, and securely access that content anywhere using a web browser.

All content, regardless of content type, is stored in the web repository or database for management, reuse and access. While stored in the repository, all types of content—ranging from e-mail, discussions, documents, reports, spreadsheets and records to images, multimedia or other digital formats—receive the same set of fundamental [core services](#).

HARDWARE AND SOFTWARE REQUIREMENTS



Important: Due to the technical nature of browsers, databases, web servers, and operating systems, Oracle cannot warrant compatibility with all versions and features of third-party products.

Content Server

Content Server runs on industry-standard computers on a variety of operating systems:

- ❖ Microsoft Windows 2000 Server or Windows Server 2003
- ❖ UNIX (Sun Solaris, HP-UX, IBM AIX)
- ❖ Linux (Red Hat, SuSe)

Web Server

For the Content Server software to function properly, web server software need to be installed. The supported web servers include:

- ❖ Microsoft's Internet Information Services (IIS)
- ❖ Apache
- ❖ Sun Java System Web Server
- ❖ IBM HTTP Server (IBM AIX only)

Database

For the Content Server software to function properly, database software need to be installed. The supported databases include:

- ❖ Oracle
- ❖ Microsoft SQL Server
- ❖ Sybase
- ❖ DB2
- ❖ PostgreSQL

Java Virtual Machine

Content Server requires a Java Virtual Machine (JVM) to execute server-side Java programs on the web server. For new installations, a JVM can be installed automatically with the Content Server software. The default JVM that is installed depends on the operating system. If you want, you may point to an existing, compatible JVM on the computer that should be used for Content Server. For updates, you may be allowed to use a JVM already on the computer (depending on the current configuration).



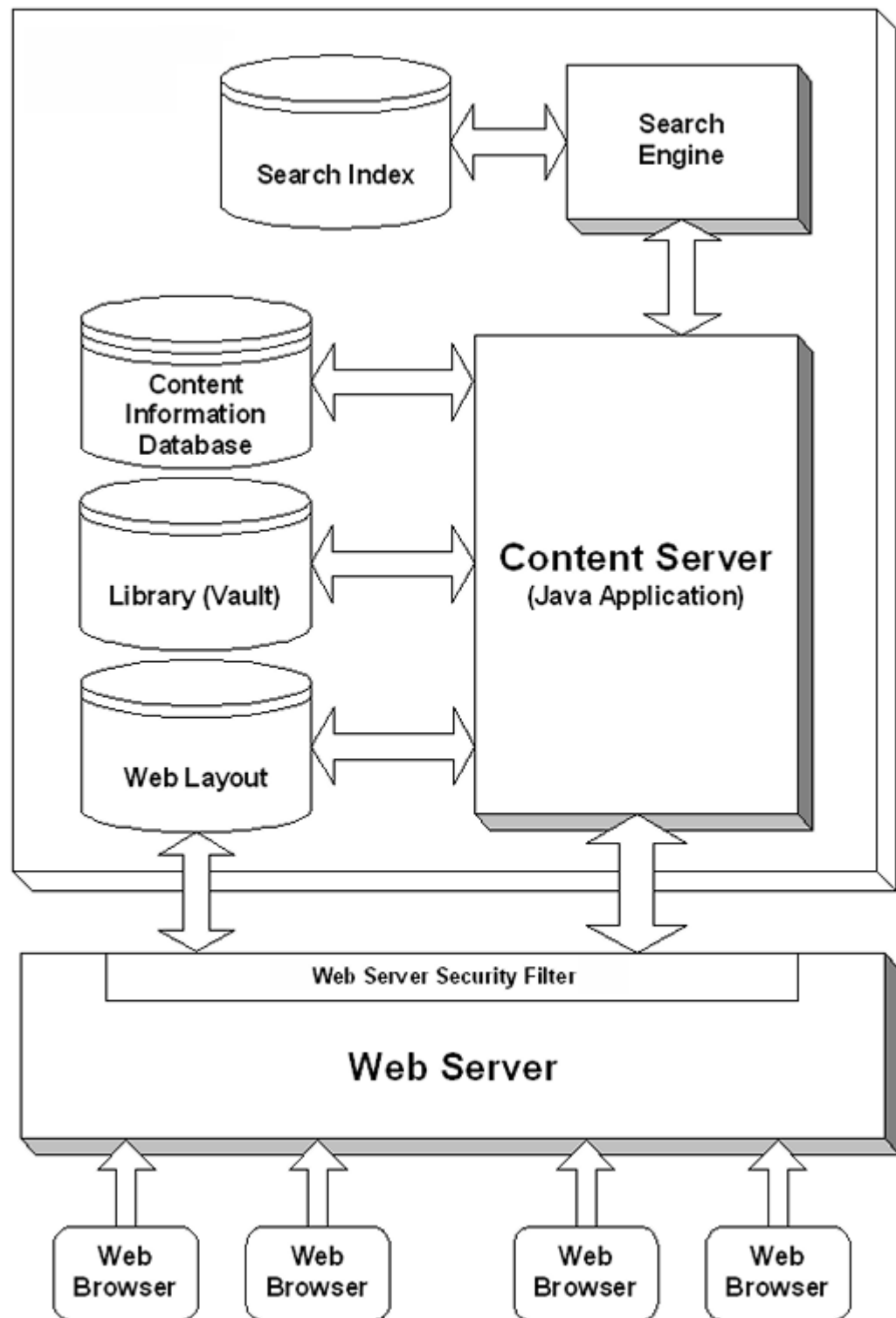
Note: For more information about the supported operating systems, web servers, databases, and JVMs, as well as specific details on supported versions, refer to the installation guides and release notes that are provided with the Content Server software.

BASIC ARCHITECTURE

The Content Server system is based on an open architecture. It uses standard web browsers and communicates through HTTP, SOAP, and JavaBeans. Figure 3-1 on page 3-4 shows the basic architecture of Content Server, which consists of the following main parts:

- ❖ [Content Server](#) (page 3-5)
- ❖ [Web Server](#) (page 3-5)
- ❖ [Content Information Database](#) (page 3-6)
- ❖ [Search Collection](#) (page 3-6)
- ❖ [Native File Repository \(or “Vault”\)](#) (page 3-7)
- ❖ [Web-Viewable File Repository \(“Web Layout”\)](#) (page 3-7)
- ❖ [Web Browser](#) (page 3-8)

Figure 3-1 Basic Content Server architecture



Content Server

The content server is the core component of the Oracle content management products. It is an advanced Java application that runs in the background (optionally as a service) and controls all content management tasks. The content server maintains a repository of content, which stores content in both their [native](#) and [web-viewable](#) file formats. In addition, it maintains a [database](#) with metadata information about the content, which is used for administrative, searching, and version control purposes.

Web Server

The web server functions as the main interface between the content server and the “outside world.” The internal communication between the web server and content server uses a direct socket connection from a plug-in in the web server to the content server. The protocol for the communication is derived and enhanced from the CGI (Common Gateway Interface) standard. A dedicated security filter runs continually on the web server. This filter handles all interaction with web browsers and other applications, and also checks user credentials.

If the web server receives a request from a browser, it sends the appropriate commands to the content server. The content server then processes these commands and returns the results to the web server for delivery to the browser.

The web server also plays an important role in [security](#). The security filter verifies for every request that the user making the request has the proper authorization. This prevents users from accessing content or performing actions beyond their authority. For example, if a user performs a search, the security filter first checks the user credentials and passes those on to the content server along with the search query itself. The content server will then only look for items that the user is allowed to access and returns the results to the web sever.

All file and information retrieval is done through the web server (or rather, the security filter on it). The web server will generally pass a request from an external browser or application to the content server, which then retrieves the required data and provides it to the web server. There is one exception to this. The web server obtains web-viewable files by directly accessing the [web-viewable file repository](#) (the “Web Layout”; see page 3-7), without going through the content server. It will, however, always first verify the security authorization of the user who requested the file before actually retrieving it.

Content Information Database

A database stores the metadata of all revisions of all the content managed by Content Server. Various [databases](#) are supported (see page 3-3). The metadata provides information about the content (for example, title, author, check-in date, and revision). It is used to describe, find, and access the content. As such, it is not unlike catalog cards in a library, where the information on the cards is used to keep track of the actual books, enabling librarians and visitors to locate them.

In addition to the metadata, the database also stores a wide variety of other data, including the user profiles (with the logon and security information for each individual user), workflow definitions, subscriptions, and historical data. Depending on the [search](#) solution used, the full text of content may also be stored in the database.



Note: No actual content item files are stored in the database, just their [file sizes](#), their [metadata](#) and, possibly, their extracted [full text](#) (depending on the search solution used)

Search Collection

Content Server provides functionality to enable users to find content managed by the content server. By default, the content database is used to provide search capability. Metadata searching is offered by all supported [databases](#) (see page 3-3), and some databases may be reconfigured to provide full-text searching as well.



Note: See the Content Server installation guides for details on modifying the database configuration to support full-text searching and indexing.

You can also use an external search engine in place of the content database to provide full-text search functionality to the content server. Integration kits are available for the following external search solutions:

- ❖ Verity K2 (www.autonomy.com)
- ❖ FAST InStream (www.fastsearch.com)



Tech Tip: If you upgraded to Content Server from an earlier release with Verity K2 4.5.1, the Verity search engine may continue to be used unless you modify the configuration.

If the web server receives a search request from an external web browser or application, it passes it on to the content server, which sends a query to the search engine. The search engine executes the query and returns its results to the content server, which processes and formats the results, and forwards them to the web server for delivery.

The search index contains the metadata and references to the full text of the latest revisions of all the content managed by Content Server. Since only the latest revisions of content are indexed, search queries always relate to the most up-to-date content versions.

See [Searching for Content](#) (page 8-17) for further details on finding content.

Native File Repository (or “Vault”)

The “Vault” is a file repository or database (for Oracle Content DB) where all revisions of checked-in files are stored in their native format—that is, the file format they were originally created in (for example Microsoft Word). The default filesystem location is a directory called “vault” under the installation directory of the content server instance, but you can also use a shared network location.

If content is checked into Content Server, the associated file is placed in the native file repository. This repository has a number of subdirectories, which represent the defined content types in the application. Each checked-in file is placed in the subdirectory that corresponds to its content type.

The files in the native file repository are named after their system-internal identifier (dID). See [File Naming](#) (page 8-6) for more information.

The native file repository enables users to access the original file of any checked-in content item and reuse it. This is particularly useful if they want to edit an existing file and check it in as a new revision.



Caution: Under no circumstance should you try and manipulate files in the native file repository outside the Content Server environment (for example, using Windows Explorer). This may seriously jeopardize system integrity.

Web-Viewable File Repository (“Web Layout”)

The “Web Layout” is a file repository (or database for Oracle Content Server DB) where all revisions of checked-in files are stored in their web-viewable format (for example, PDF). The default location is a directory called “weblayout” under the installation directory of the content server instance, but you can also use a shared network location.

If no conversion module is running or if the native file format cannot be converted, the content file in the web-viewable file repository is an exact copy of the one in the [native file repository](#) (although their [file names](#) are different). Clients then need the native application to view the files (for example, Microsoft Word).

If content is checked into Content Server and converted to a web-viewable format, the web-viewable file is placed in the web-viewable file repository (or database for Oracle Content Server DB). This repository has a file structure that includes subdirectories for each defined content type in the application. Every web-viewable file is placed in the subdirectory that corresponds to its content type.

The files in the web-viewable file repository are named after their content IDs. See [File Naming](#) (page 8-6) for more information.



Caution: Under no circumstance should you try and manipulate files in the web-viewable file library outside the Content Server environment (for example, using Windows Explorer). This may seriously jeopardize system integrity.

Web Browser

Standard web browser technology is used to access the content managed by Content Server. Several industry-standard web browsers are supported, including Internet Explorer and Firefox. Users access Content Server through a web browser on their own computer. They interface with a [web server](#), which handles all communication with the [Content Server](#) core.



Note: For specific details on supported web browsers and their versions, refer to the Content Server installation guides.

MAIN INTERFACES

The Content Server core (see [Basic Architecture](#) on page 3-3) interfaces with several other components of the content server system to deliver full content management functionality, as illustrated in Figure 3-2 on page 3-10.

Files may be checked into Content Server either manually (one by one) or automatically (in large volumes, using [Batchloader](#)). Files are stored in their native format, and may also be converted to a different format upon check-in (through [Inbound Refinery](#) and a number of optional conversion modules). In addition, thumbnails may be generated for the files that are checked in. The converted files may also be checked into Content Server.

Once content is checked in, it is stored in a number of places (see [Content Storage](#) on page 8-8). Other Content Server modules have access to the content items and can manipulate them. For example, Content Publisher can draw content from the content server and use it to automatically create and stage a content-driven website. This website

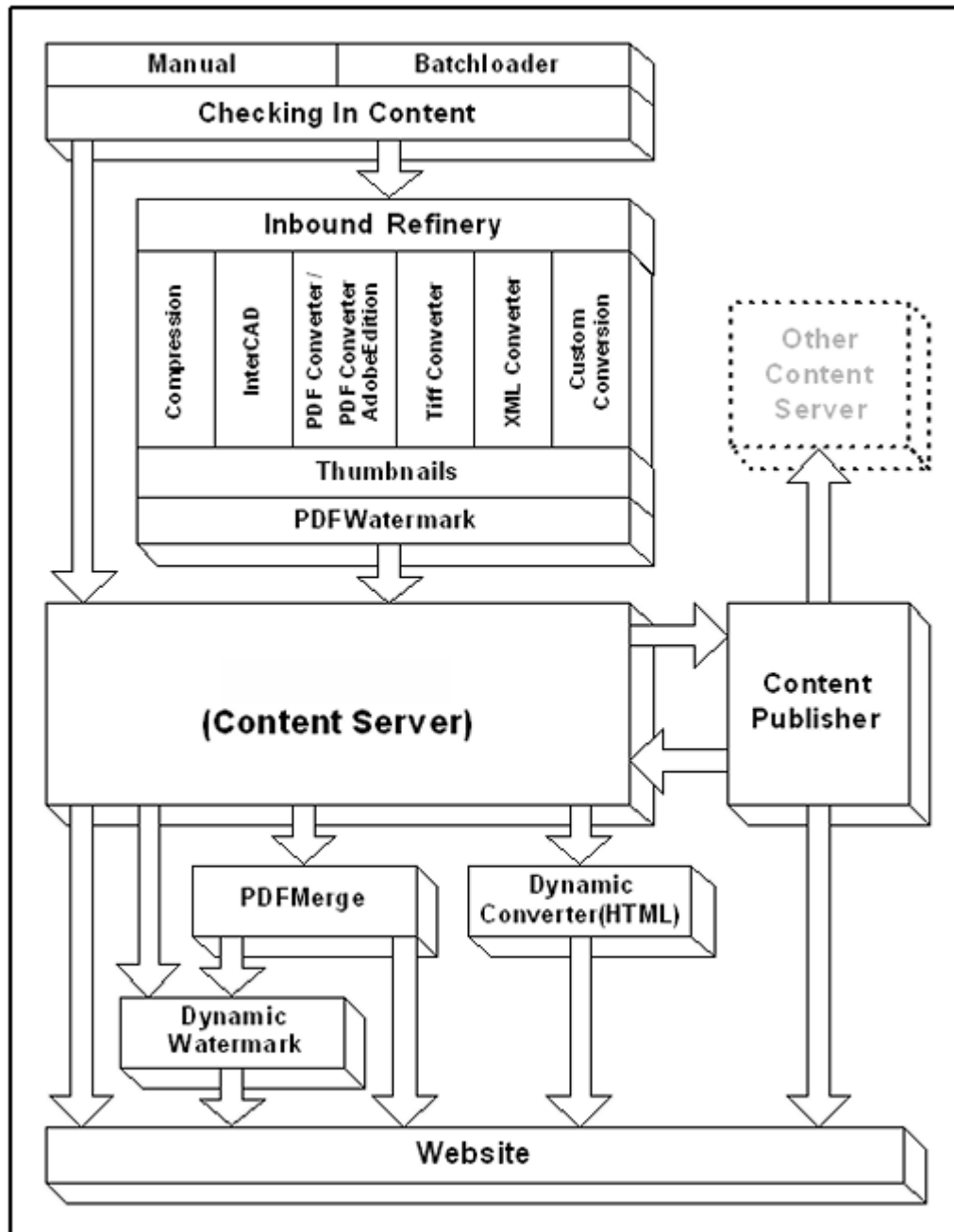
is either checked back into the content server, or published to a file system (for example, a web server) or another content server.

Checked-in content can be manipulated as it is delivered to a website for viewing through a standard web browser. For example, content items can be converted to HTML for easy viewing in a web browser, or checked-in or generated PDF files may be processed to include a watermark or merged into a single file.



Note: For further details on the Batchloader utility, refer to [Chapter 9 \(Administering Content Server\)](#).

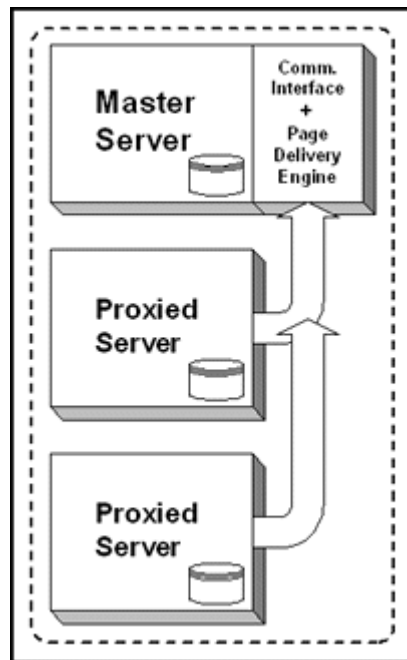
Figure 3-2 Main interactions within the Content Server system



MASTER SERVERS AND PROXIED SERVERS

Content Server supports multiple content servers running on the same computer. These can all be masters, which means they are fully separate from each other, but one server can also be designated as the “master” and one or more others as the “proxied server(s).” Proxied servers are separate content server instances which have their own content [database](#), [web-viewable file repository](#) (“Web Layout”), and [native file repository](#) (“Vault”), but share the main communication interface and page delivery engine with the master server (see Figure 3-3).

Figure 3-3 Master server and proxied servers



Master/proxied server configurations can be useful in situations where you want to create distributed content management environments:

- ❖ You want to set up a Content Server environment with distributed user management. You may, for example, want to set up a separate content server for each department in a company, each with its own user definitions and setup, but with a shared security model.
- ❖ You want to set up a Content Server environment with distributed management of corporate functions, while maintaining centralized user management. You may, for example, want to set up multiple content servers, each serving a specific corporate function, such as document management, website management, and customer

relationship management. If users are defined on the master server, they will automatically also have access to the other proxied servers (within the bounds of their privileges). It does not matter which server they use to log in.

- ❖ You want to set up a Content Server environment with distributed search capability. You may, for example, want to set up a number of content servers as logically separate entities, but still be able to search the content databases of all content servers with just one search query. This can be accomplished by creating a master server and a number of proxied servers, in combination with the optional Enterprise Search add-on.
- ❖ You want to set up a Content Server environment with distributed content management. This will typically be the case if the application needs to manage huge numbers of files (more than one million). You then need to split up the content management system into a master server and one or more proxied servers, each taking care of part of the volume.
- ❖ A master and proxied server can also be useful if you want Content Server's search index to handle documents in both Western European languages (for example, English, French, and German) and an Asian language (for example, Japanese). Since there can be only one search engine locale per instance, you need multiple instances to achieve this. You could, for example, set up a master server to handle the Western European documents, and a proxied server to handle the Asian-language documents.

SEARCH OPTIONS

By default, new installations of Content Server 10gR3 are set up to use the [database](#) to provide search capabilities. The out-of-the-box database configuration provides metadata-only searching, but you can modify the default configuration for some of the [supported databases](#) to enable full-text searching. For further details refer to the Content Server installation guides.

You can also use an external search engine in place of the content database to provide full-text search functionality to the content server. Integration kits may be available for the following external search solutions:

- ❖ Verity K2 (www.autonomy.com)
- ❖ FAST InStream (www.fastsearch.com)



Tech Tip: If you upgraded to Content Server 10gR3 from an earlier release with Verity K2 4.5.1, the Verity search engine will continue to be used unless you modify the configuration.

FILE CONVERSION

When files are checked into the content server, they are stored in their native format (see [Native File Repository](#) on page 3-7). This is the file format they were originally created in (for example, Microsoft Word or Excel). In addition, they may be converted to a different format, which may also be stored in the content server. Native files are typically converted to create [web-viewable files](#), which can be viewed using standard web browsers (for example, PDF or HTML).

Conversions may take place at the input side of the content server or at the output side. Figure 3-2 on page 3-10 illustrates the various conversion options. A number of optional add-on modules (filters) are available for Content Server to perform the conversions. They will typically run in the background and come into action as the need arises. The two main optional conversion modules are:

- ❖ [Inbound Refinery \(With Conversion Filters\)](#)
- ❖ [Dynamic Converter](#)

Some file formats cannot be converted, or the system administrator may have configured the system to “pass through” certain types of documents without conversion. For example, a compressed zip file cannot be converted to a web-viewable format. In these cases, a copy of the native file is stored in the [web-viewable file repository](#) (although the [file names](#) will be different), and users need to have the native application installed on their computer to be able to view the file.

COMPONENT ARCHITECTURE

Content Server is highly functional “out-of-the-box,” but there are a number of ways to tailor your content management system to your site requirements. At the heart of this are components, which are modular programs that are designed to interact with the content server at runtime. [Custom components](#) enable you to alter the look-and-feel and behavior of Content Server without affecting the core functionality of the software.

Component Wizard is a development tool that automates the process of creating custom components. You can use Component Wizard to create new components, modify existing components, and package components for use on other content server instances.



Note: See page 10-7 for more information about using components. For more information about component development, refer to the *Working with Components* guide and other Content Server documents for developers.

CLUSTERING

A server cluster is a group of independent servers managed as a single system that can be used as a multipurpose platform for database management, file and intranet data sharing, and general business applications.

Content Server can be a stand-alone system, or it can be modified and integrated in unlimited configurations to alleviate problems of maintenance, provide load balancing, and improve scalability. In a cluster configuration where multiple servers share a Content Server instance, all of the servers also use a common content server file system, [database](#), and [search collection](#). A load balancer can be used to provide high availability and scalability for consumption.

A cluster uses existing cluster infrastructure to provide enterprise-level software that seamlessly integrates into most commercially available cluster server platforms. A standard clustered content server is set up to simultaneously share file systems and databases. A shared database is one that is always highly available. A shared file system is one in which all nodes read and write to the same file system. All nodes also mount or map the shared file system identically.

A clustered Content consists of the installed Content Server product and multiple servers in a cluster environment. The clustered servers share a single content server dataset that includes directories for the [database](#), [search index](#), [native-file repository](#) (“Vault”) and [web-viewable file repository](#) (“Web Layout”).

Cluster systems are used for many reasons, but the greatest demand for cluster technology is to improve the availability and scalability of mission-critical data, applications and resources. Using clusters ensures that these key corporate components always remain available, even during maintenance upgrades, system failures, or backups. In general, the cluster is designed to prevent a single point of failure. Thus, server clusters preserve client access to data, applications and resources during both failures and planned outages. Server clusters make it possible to share a computing load over several systems without either the users or system administrators needing to know that more than one system is involved.



Note: For more information about clustering, as well as configuration instructions, refer to the *Clustering Concepts Guide* guide and the *Clustering Configuration* guides.

INTEGRATION OPTIONS

Oracle provides easy, flexible methods for integrating its content server with enterprise applications such as application servers, catalog solutions, personalization applications, and enterprise portals, as well as client-side software.

The content server not only serves as a content management solution for content-centric websites, but also provides a scalable content management infrastructure that supports multiple enterprise applications in many diverse environments and platforms. The Oracle integration solutions enable other enterprise applications to access content managed by Content Server, and provides these applications with critical content management capabilities such as full-text and metadata searching, library services, workflow, subscription notifications, and content conversion capabilities via a wide array of integration methods.

In general, these integration methodologies serve to translate or pass methods and associated parameters with the goal of executing content server services. The various content server services are the “window” for accessing the content and content management functions within Content Server.

Content Server’s integration and scalability options are standards-based and allow for current and future expansion plans—regardless of whether the system is used as a stand-alone implementation or as part of a larger content management infrastructure to support multiple enterprise applications.

The available application integration methods include the following:

- ❖ [Java API \(IdcCommand\)](#)
- ❖ [Component Object Model \(COM\)](#)
- ❖ [Java Server Page \(JSP\)](#)
- ❖ [Common Object Request Broker Architecture \(CORBA\)](#)
- ❖ [Open Document Management API \(ODMA\)](#)
- ❖ [Simple Object Access Protocol \(SOAP\)](#)
- ❖ [Web Distributed Authoring and Versioning \(WebDAV\)](#)

Java API (IdcCommand)

You can use the Java API to gain access to the content and content management functions within Content Server. The IdcCommand Java Command Utility is a stand-alone Java application that enables users to execute content server services.

Component Object Model (COM)

You can use a COM interface to integrate Content Server with Microsoft environments and applications. Oracle provides an ActiveX control and an OCX component as interface options to gain access to the content and content management functions within Content Server.

Java Server Page (JSP)

The Content Server core functionality can be accessed from a Java Server Page (JSP) running in Content Server, from a JSP through the Content Server JavaBean, or from a JSP through the Content Server Enterprise JavaBean (EJB) deployed on your J2EE application server. In addition, the JSP tag libraries are available to simplify JSP coding and to easily access content server services.

Common Object Request Broker Architecture (CORBA)

You can use the CORBA API to gain access to the content and content management functions within Content Server by implementing RMI over IIOP as the transport protocol and referencing the Oracle Enterprise JavaBean.

Open Document Management API (ODMA)

You can use the Oracle ODMA-based plug-in to gain access to the content and content management functions within Content Server (for ODMA-compliant desktop applications).

Simple Object Access Protocol (SOAP)

You can use a SOAP interface to access the content and content management functions within Content Server and to deploy your content management capabilities as a web service. The SOAP protocol integrates .NET servers, J2EE application servers, or other systems with XML-based interfaces.

Web Distributed Authoring and Versioning (WebDAV)

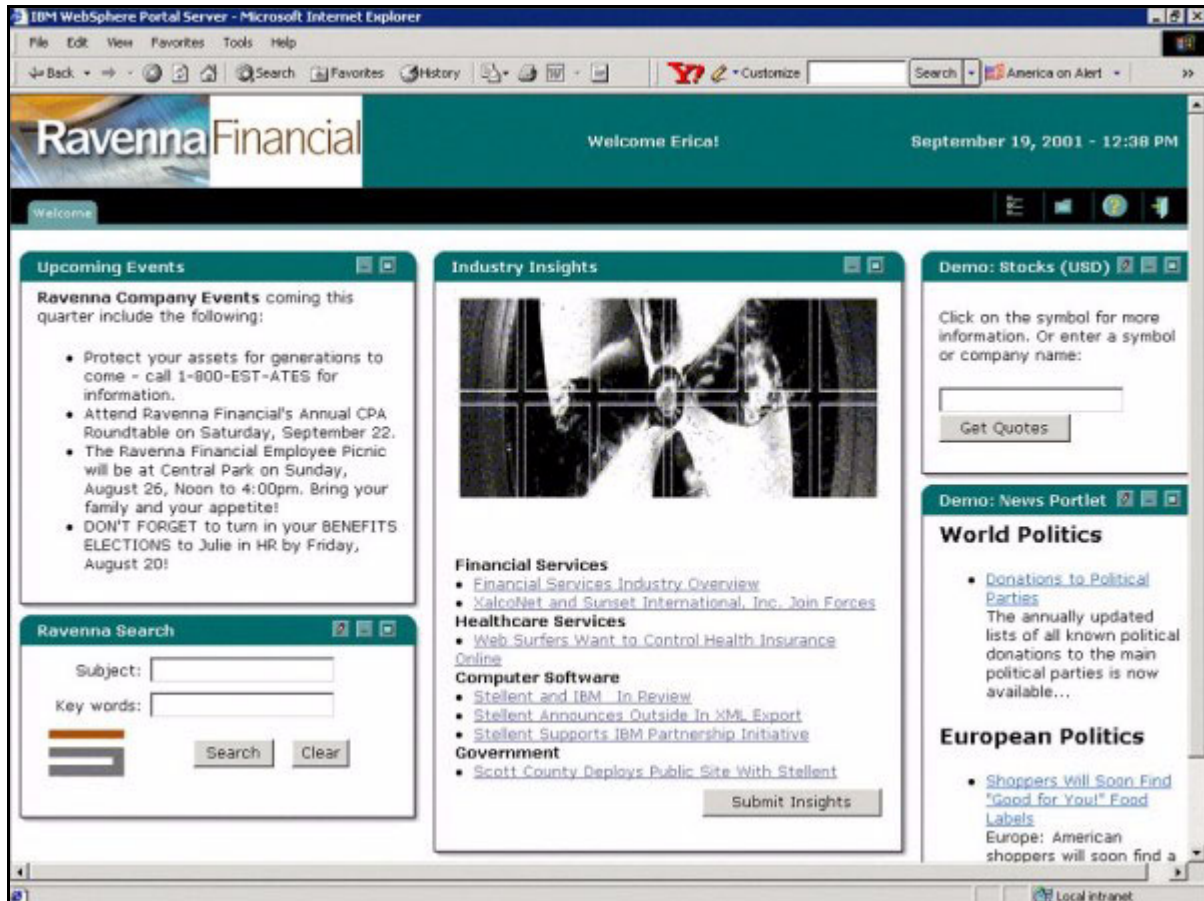
You can use a WebDAV-enabled desktop or business application to connect directly to the Content Server system and avoid the need of additional client-side software.

PORTALS AND PORTLETS

Organizations increasingly use portals, which provide secure, personalized points of access to the organization's resources and information. Portlets (also called "portal applications") are dynamic components that end users see within their portal pages. They are basically small "windows" within the portal that serve as content channels. A portlet could, for example, show the latest headlines or provide personalized search capability.

Figure 3-4 shows an example of a website with a number of individual portlets, each providing different, personalized pieces of information.

Figure 3-4 Portlets in a website



Content Server has the capability to interface with any J2EE-compliant application server and portal servers to populate portlets with content managed by Content Server. Oracle provides the ability to manage the content creation and distribution process through the

use of portlets. Users can update, search, and view portal content in a way that is efficient and easy to use. These portlets can be enabled for different users based upon the user's roles and permissions within the organization. Depending on the permission level, the user may be allowed to browse or search content, contribute a new content item and view the progress of their content through workflow.

To achieve portlet integration, Content Portlet Suit (CPS) uses Content Integration Suite (CIS) to abstract the portlets from the details of communication with the Content Server. The Portlet API facade abstracts the common operations within portlet containers so our framework will work on a variety of platforms using the same handler code. Portlet Actions are mapped to a custom MVC framework that then uses CIS to perform the desired task.

There are currently pre-built Content Server [portlet suites](#) available for the following portal servers:

- ❖ BEA WebLogic Portlet Server
- ❖ IBM WebSphere Portlet Server
- ❖ Plumtree Portlet Server
- ❖ Sun ONE Integration Server

DOCUMENTATION

OVERVIEW

This section covers the following topics:

- ❖ [Online Technical Newsgroup](#) (page 4-1)
- ❖ [Documentation](#) (page 4-1)

ONLINE TECHNICAL NEWSGROUP

The ‘intradoc_users’ newsgroup, hosted by Yahoo!Groups, is free of charge to anyone who is interested in customization and usage of Oracle solutions. To subscribe to this user-supported group, go to its homepage at http://groups.yahoo.com/group/intradoc_users, and follow the directions.

DOCUMENTATION

Content Server software is shipped with a comprehensive set of electronic documentation, which is available in two formats:

- ❖ [Online Help System](#) (page 4-2)
- ❖ [PDF Files](#) (page 4-2)



Note: The Content Server installation guides are also provided in printed form.



Tech Tip: In addition to the product documentation that is provided with the Content Server software, there is also *The Definitive Guide to Content Server Development* (ISBN 1-59059-684-6) by Brian “Bex” Huff. This book, which is available from Amazon.com and Apress.com, provides an in-depth discussion of the Content Server core service-oriented architecture and details how the flexible, component-driven model makes it easy to customize the appearance and behavior of the content server.

Online Help System

The Content Server software is installed with a full online help system, which you can call up from the Content Server user interface. You will then either see the system administration help or the user help, depending on where you are in the software.

You can also open the system administration help system (which includes the user help information) from the file system by starting the file `[Install_Dir]\weblayout\help\wh_start.htm` (where `[Install_Dir]` is the installation directory of the content server instance).



Note: There is a section in the help system called About This Help System, which provides an overview of all information included in the help system.



Note: The help system that is initially installed with the Content Server software does not include the help files for the Content Server [add-ons](#). Instead, it contains placeholders for the add-ons, which are replaced with the “real” help files upon installation of the add-on files.

PDF Files

All documents in the Content Server documentation set are available as PDF files.

Before installation of the Content Server software, these files are available on the software distribution media (typically a separate CD labeled ‘Documentation’).

After installation of the Content Server software, the PDF files are also available on the computer that the software was installed on. You can call up the PDF version of any current guide in the online help by clicking the PDF icon in the top-right corner of each help page of that guide.

You can also start a special documentation menu from the file system, which provides convenient navigation to all the documents in the documentation set. If you installed the Content Server software under Microsoft Windows, there is a shortcut to this file in

Start—Programs—Content Server—*[Instance_Name]*—Utilities—Documentation.
Under UNIX, navigate to the directory *[Install_Dir]/weblayout/help/ documentation* and start the file *menu.pdf*.



Note: To view and print the documentation files, you need the Adobe Acrobat Reader. This is a free utility, which can be downloaded from the Internet at <http://www.adobe.com/products/acrobat/readstep2.html>.



Note: The documentation root directory contains a file called *tips_en.pdf*, which provides useful information about using the PDF documentation set. This file can also be accessed from the navigation menu.

CONTENT SERVER SECURITY

OVERVIEW

Not all information is intended for everyone in an organization. Certain content may be confidential or simply irrelevant to particular groups of people. Content Server has extensive security features which effectively protect content from unauthorized viewing or manipulation. In addition to using Content Server's own security model, you can also configure the system to integrate into external security sources such as LDAP, or Active Directory.

Security in Content Server is set up in such a way that it is transparent to the users. This means that users only see content they have access to. If they try to search for files they do not have access to, they will not see them or even know they exist.



Note: This section aims to provide an overview of the basic security architecture and options of Content Server. It is not intended to be an exhaustive security setup guide. More in-depth information on security in Content Server can be found in the *Managing Security and User Access* guide.

Security in Content Server is made up of five main components:

❖ **Users**

Users are the logins made up of a user name and password. They are assigned one or more role(s) and/or account(s). For more information on users refer to page 5-3.

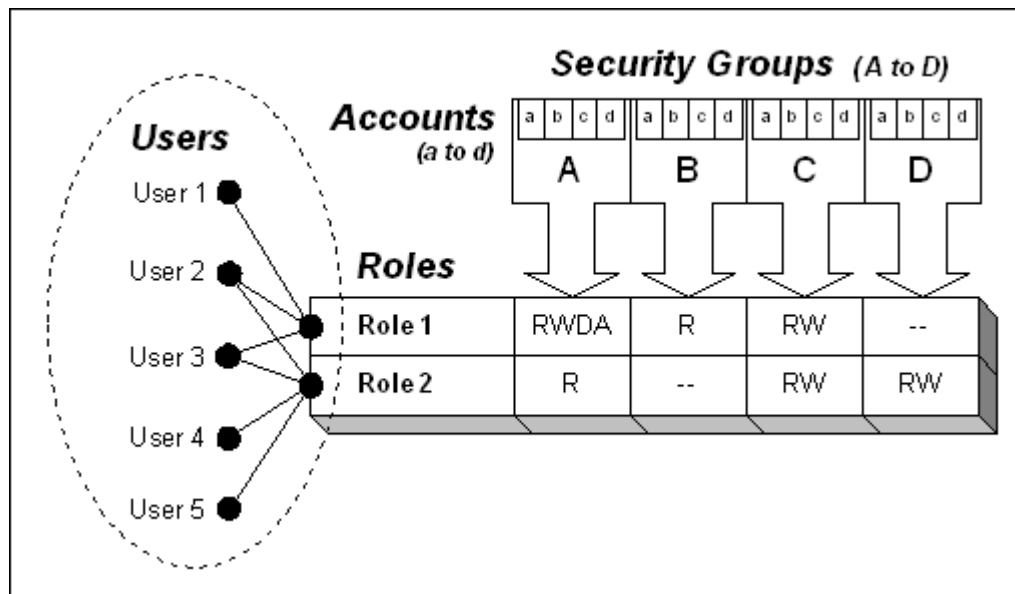
❖ **Security Groups**

Security groups are groupings of content that you control access to via roles. For more information on security groups refer to page 5-6.

- ❖ **Permissions**
Permissions define the actions that users can perform on content in security groups. For more information on permissions refer to page 5-8.
- ❖ **Roles**
Roles define the permissions (read, write, delete, and admin) that users have to specific security groups and accounts. Each user is assigned one or more roles. For more information on roles refer to page 5-8.
- ❖ **Accounts**
Accounts are security elements which, unlike security groups, are used to group users (rather than content). Content can be assigned to a particular account upon check-in. Users can only access the content assigned to an account if they have the appropriate privileges to that account. For more information on accounts refer to page 5-12.

Figure 5-1 shows the relationship between these components.

Figure 5-1 Relationship between users, security groups, accounts, and roles



Note: For more information on Content Server security refer to the *Managing Security and User Access* guide.

USERS

This section covers the following topics:

- ❖ [User Types](#) (page 5-3)
- ❖ [Authorization Types](#) (page 5-4)
- ❖ [Predefined User Logins](#) (page 5-6)
- ❖ [Defining and Managing Users](#) (page 5-6)

User Types

Users identify themselves to the content server by their login name and password, which define what they can do in the content server and what they have access to. The login names and password are stored in Content Server and/or an external security database.

The Content Server system distinguishes between four types of users:

- ❖ [Consumers](#)
- ❖ [Contributors](#)
- ❖ [Administrators](#)
- ❖ [Sub-Administrators](#)



Important: Users are not formally “assigned” to any of these four types—in other words, nowhere in the software is there a selection list of some sort that enables you to identify a user as, say, a ‘consumer.’ The type of user is determined by the privileges a user is given. For example, a contributor is anyone who has read and write (RW) [permission](#) to certain security groups (but no admin permission), whereas [\(sub-\)administrators](#) have one or more administration privileges.

Consumers

Consumers are users who access the content server through their web browser to search, find, and view files. They can only access files which they are permitted to see, based on their login credentials. Consumers cannot check new content into the content server. Typically, the majority of users are consumers. A consumer does not need a user name and password to access files, unless a security configuration has been set up.

Contributors

Contributors are users who—like consumers—can find, view, and print files, but can also check new content into the content server or revise existing content. To safeguard the integrity of the files, contributors require a user name and password to check files in and out of the system.

Administrators

Administrators are individuals who set up, maintain, and modify the configuration of the content management system and its user logins. Administrators can also assign sub-administrators to help them manage the system. To safeguard the integrity of the system, administrators require a user name and password to access the system. For details on administrators refer to [Administrators and Sub-Administrators](#) (page 5-16).

Sub-Administrators

Enterprise administrators can set up sub-level administrators to perform a subset of administrative tasks within applications, specific departments, or security groups. These sub-administrators maintain a portion of a content management system and its user logins. For details on sub-administrators refer to [Administrators and Sub-Administrators](#) (page 5-16).

Authorization Types

Users belong to a certain authorization type, which defines how they are authenticated by the system. There are three authorization types:

- ❖ [Local Users](#)
- ❖ [Global Users](#)
- ❖ [External Users](#)

The authorization type of a user is specified when the user is added to the system (using the User Admin utility).

Local Users

Local users are defined by an administrator or sub-administrator within a content server. The credentials of local users may extend to multiple content servers (for example, master and proxied servers). Administrators assign these users one or more roles, which provide them with access to security groups.

Local users can change their password, full name, e-mail address, and user type. The names of local users are included in the author drop-down lists in the content server user interface.

This type of user is recommended if there is a maximum of 1,000 users. If there are more, performance issues may become a problem, and global users should be used.

Global Users

Global users are recommended for enterprise situations with more than 1,000 users. They are lightly-managed users whose credentials extend to multiple content servers. The validation of global users is always performed dynamically. The user profiles are created and stored on a master server, and retrieved by proxied servers. The content server roles are set on the master server, which provides access to security groups across multiple content server instances.

Global user credentials are not published to the web server security filter, so the master server always validates the credentials by querying the database tables. Because of this, the master server must be set up to log in as a global user.

Global users can only log in if the master server is running. They can change their full name, e-mail address, and user type. The names of global users do not appear in the author drop-down lists in the content server user interface.

External Users

External users are users who are automatically registered in the system but are not manually set up by an administrator. These users might use a Microsoft login or some other type of provider login (for example, LDAP or Active Directory). Generally, these are users in a trusted domain to whom you grant access. These users cannot set their own password. Their passwords are owned by the Microsoft network domain or other type of provider.

External users can only log in if the content server is running. They cannot change their password. The names of external users do not appear in the author drop-down lists of the content server, but they can participate in workflows and use subscriptions.

External users are defined outside the Content Server system and authenticated through external security. External users who are automatically registered in the system but are not manually set up by an administrator might use a Microsoft login or some other type of provider (such as LDAP) login.

Generally, these are users in a trusted domain to whom you grant access and do not manage through Content Server. Their password is owned by the Microsoft network domain or other type of provider.

Predefined User Logins

The Content Server software is shipped with the following predefined user logins:

❖ **sysadmin**

This login is the system administrator and is assigned the ‘admin’ role (see [Roles](#) on page 5-8). The default password is “idc.” The ‘sysadmin’ user login is a local user, and cannot be deleted from the system.

❖ **user1**

This login is assigned the ‘contributor’ role (see [Roles](#) on page 5-8). The default password is “idc.” The ‘user1’ user login is a local user, and should be deleted from the system before a site goes into production.



Important: It is extremely important that you change the default passwords before a site goes into production.

Defining and Managing Users

Users are defined and managed using User Admin, which is one of the tools accessed from the Administration page. You need administrator privileges to start the User Admin administration tool.



Note: For further details on User Admin and step-by-step procedures, refer to the *Managing Security and User Access* guide.

SECURITY GROUPS

This section covers the following topics:

- ❖ [Using Security Groups](#) (page 5-7)
- ❖ [Predefined Security Groups](#) (page 5-7)
- ❖ [Defining and Managing Security Groups](#) (page 5-7)

Using Security Groups

A security group is a set of content items grouped together under a unique name. All documents from the Human Resources department could, for example, be grouped in a security group called “HRDocs.” Every file that is checked into the system is assigned to a security group. Security groups are used to control access to content. Access to security groups is controlled by permissions, which are assigned to roles, which are assigned to users.

Make sure that security groups have logical names. Also, keep their number to a minimum to provide optimum system performance and minimize administrative maintenance.

Predefined Security Groups

The Content Server software is shipped with the following predefined security groups:

- ❖ **Public**
Any user can view documents in the Public group without logging in.
- ❖ **Secure**
System files are stored in the Secure group and are available only to the system administrator.

Defining and Managing Security Groups

Security groups are defined and managed using User Admin, which is one of the tools accessed from the Administration page. You need administrator privileges to start the User Admin administration tool.



Note: For further details on User Admin and step-by-step procedures, refer to the *Managing Security and User Access* guide.

PERMISSIONS

Permissions define the actions that users can perform on content in security groups. There are four different permission types:

- ❖ **Read (R)**
This permission type allows users to view files in a security group.
- ❖ **Write (W)**
This permission type allows users to view, check in, check out, and get a copy of files in a security group.
- ❖ **Delete (D)**
This permission type allows users to view, check in, check out, get a copy of, and delete files in a security group.
- ❖ **Admin (A)**
This permission type allows users to view, check in, check out, get a copy of, and delete files in a security group. In addition, if users also have Workflow rights (see [Administrators and Sub-Administrators](#) on page 5-16), they can start or edit a workflow in a security group.

The permissions that users have to access the files in a security group is the combination of the permissions they have in their roles.

Privileges are assigned to users using User Admin, which is one of the tools accessed from the Administration page (Security—Permissions By Group or Permissions By Role). You need administrator privileges to start the User Admin administration tool.

ROLES

A role is a set of permissions (Read, Write, Delete, Admin) for one or more security groups. Roles are assigned to users, and as such define what content users have access to.

This section covers the following topics:

- ❖ [Assigning Roles](#) (page 5-9)
- ❖ [Predefined Roles](#) (page 5-11)

Assigning Roles

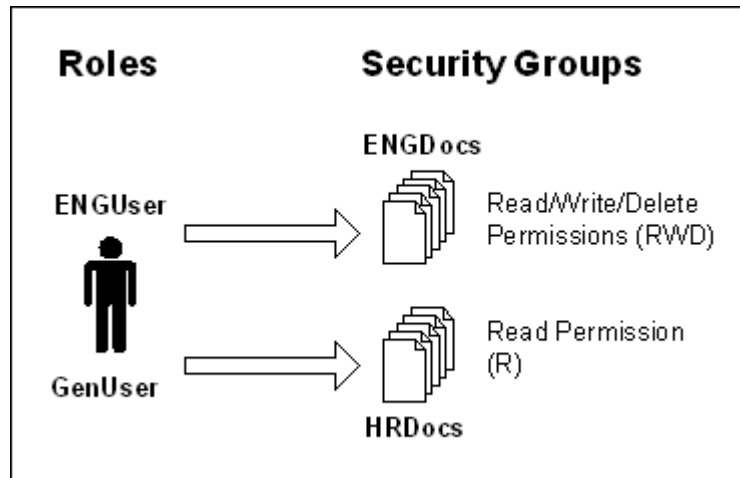
You can assign roles to users in different ways:

- ❖ [Assigning Multiple Roles to One User](#) (page 5-9)
- ❖ [Assigning Multiple Roles to One Security Group](#) (page 5-9)
- ❖ [Assigning Multiple Security Groups to One Role](#) (page 5-10)

Assigning Multiple Roles to One User

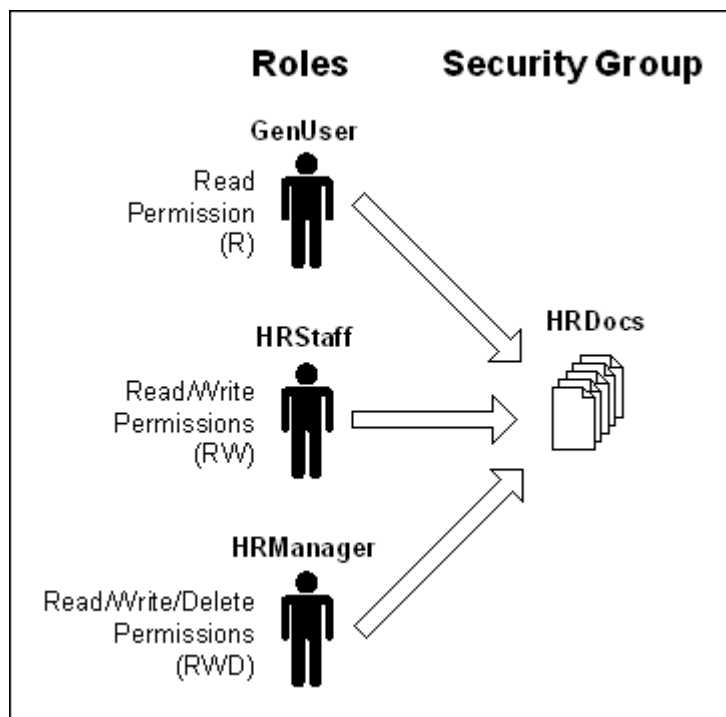
Users can be assigned more than one role. After all, they may have different “identities” when accessing different types of content. For example, engineers might be granted permission to read, write, and delete engineering documents, but only view documents from the Human Resources department (as anyone else in the company). They would then have two roles: one in their identity as engineers (“ENGUser”), and one as company employees in general (“GenUser”), as illustrated in Figure 5-2.

Figure 5-2 Assigning multiple roles to one user



Assigning Multiple Roles to One Security Group

Security groups can be assigned more than one role. After all, there may be different ways to access the content in the security group, depending on the identity and position of the user. For example, documents from the Human Resources department (grouped in the security group “HRDdocs”) may be viewed by anyone in the company (“GenUser”), but they may only be modified by HR staff and deleted by the HR manager, as illustrated in Figure 5-3.

Figure 5-3 Assigning multiple roles a one security group

Assigning Multiple Security Groups to One Role

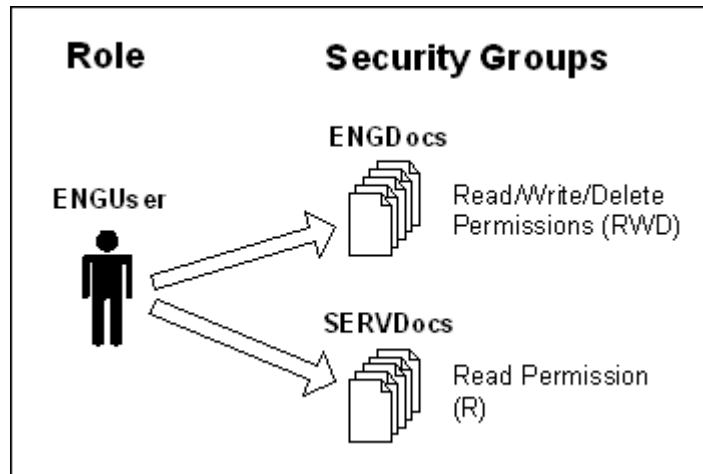
Roles can be assigned more than one security group. After all, a role may imply that users need to access different types of content and need to do different things with them. For example, an engineer with role “ENGUser” might be granted permission to read, write, and delete engineering documents, but he may also need to view documents from the Services department (grouped in the security group “SERVDocs”), as illustrated in Figure 5-4.



Note: Remember that access to content is controlled through security groups. Access to security groups is controlled by permissions, which are assigned to roles, which are assigned to users.

Roles are defined and managed using User Admin, which is one of the tools accessed from the Administration page (Security—Permissions By Role). You need administrator privileges to start the User Admin administration tool.

Figure 5-4 Assigning multiple security groups to a role



Predefined Roles

The Content Server software is shipped with the following predefined roles:

- ❖ **Guest**
The predefined Guest role allows the viewing of documents in the 'Public' security group. This role cannot be deleted from the system.
- ❖ **Contributor**
The predefined Contributor role allows viewing of documents in the 'Public' security group, and checking documents into and out of that security group.
- ❖ **Admin**
The predefined Admin role is typically assigned to a user who is the security group administrator. This role cannot be deleted from the system.
- ❖ **Sysmanager**
The predefined Sysmanager role grants privilege to access the Admin Server administration tool (see [Admin Server](#) on page 9-4).



Note: Undefined users are automatically assigned the "guest" role.

ACCOUNTS

This section covers the following topics:

- ❖ [Using Accounts](#) (page 5-12)
- ❖ [Accounts Scenario](#) (page 5-13)
- ❖ [Accounts and Permissions](#) (page 5-14)
- ❖ [Account Hierarchy](#) (page 5-15)
- ❖ [Special Account Classifications](#) (page 5-16)
- ❖ [Assigning Accounts](#) (page 5-16)



Note: For more details about accounts, refer to the *Managing Security and User Access* guide.

Using Accounts

Accounts enable you to obtain greater flexibility and granularity in your security structure than security groups alone.

[Security groups](#) use the “library” model, where you have broad access control. A library is organized by books, reference material, and periodicals—arranged by information and not by user. Accounts use the “Bank” model, where you can have thousands of accounts but limited users have right to them. Similarly, accounts are centered more around users than information. Accounts basically provide a way to “split up” content security groups into smaller categories which are only accessible to certain groups of users. Content can be assigned to a particular account upon check-in. Users can only access the content assigned to an account if they have the appropriate privileges to that account.



Tech Tip: The accounts functionality in Content Server is not enabled by default. You need set a system variable to turn it on. For details refer to the *Managing Security and User Access* guide.



Caution: Once you turn on accounts and use them, you cannot turn them off without losing data. Do not turn on accounts unless you are absolutely certain you want to use them. Contact Oracle’s [support](#) organization if you require help with enabling accounts.

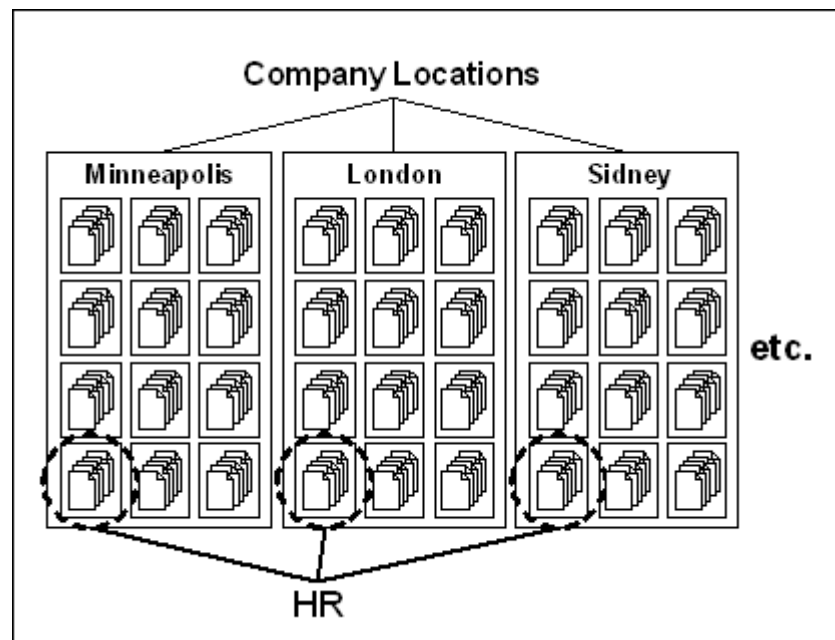
Accounts Scenario

In general, use accounts when you require more than 50 security groups. Consider the following situation:

- ❖ A company has 10 locations worldwide, including Minneapolis, London, and Sidney.
- ❖ Each location has 12 departments, including Sales, Marketing, Human Resources (HR), and Research & Development (R&D).

Some content may be seen by Human Resources at any location as shown in Figure 5-5.

Figure 5-5 Using accounts (1)

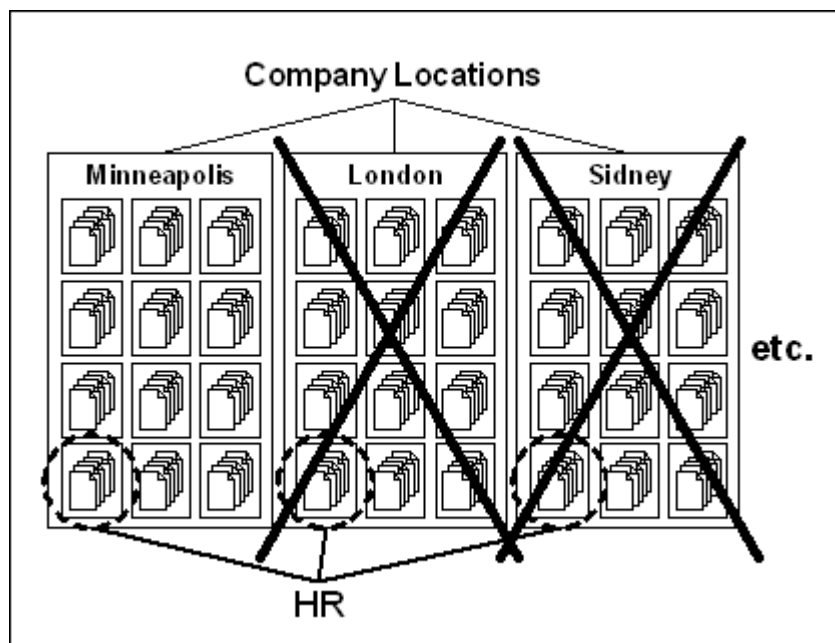


Some content may only be seen by HR in the Minneapolis location. This means HR individuals from other locations should not be able to see the HR content in Minneapolis as shown in Figure 5-6.

In this example, in order to accomplish the required security using security groups alone, each department within each location is a separate security group. This is a total of 120 security groups (10 locations times 12 departments), which greatly exceeds the recommended limit. You can use accounts to reduce the number of security groups.

If accounts are used in this example, there would be 12 security groups (one for each department) and 10 accounts (one for each location). This enables you to narrow down the access privileges taking the security group and location into account.

Figure 5-6 Using accounts (2)

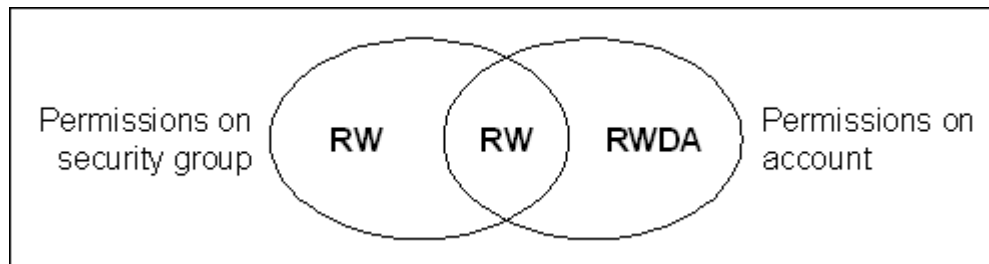


Accounts and Permissions

If accounts are used, the account becomes the primary permission to satisfy before roles. The permissions become the intersection of the permissions on the account and security group—in other words, the lowest common permissions. This is illustrated in Figure 5-7.

A user with read/write (RW) permission in the security group and read/write/delete/admin (RWDA) permission to the account will have read/write (RW) permission to the content in that security group matching their account.

Figure 5-7 Combining permissions on security group and account



Roles are ignored if the account does not permit access to a content item. If, for example, a role has read (R) permission to a particular security group, but there are no content items in the corresponding account of the security group, then the role is ignored.

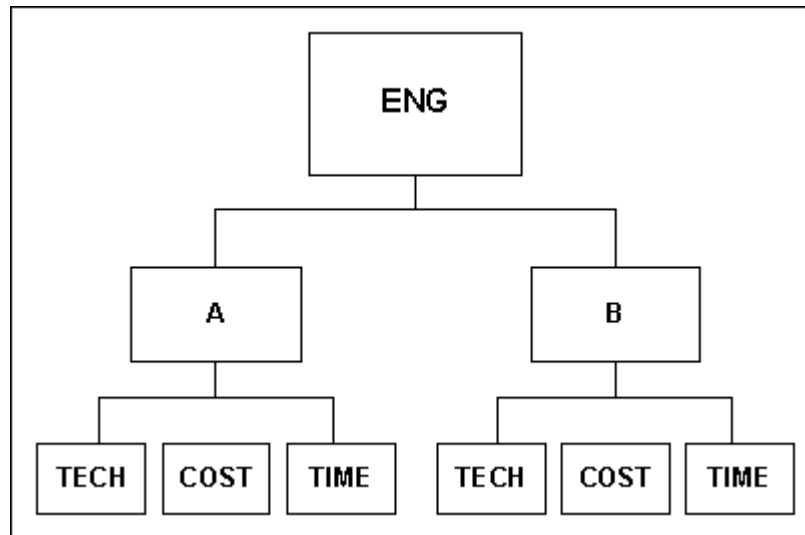


Note: For performance reasons, do not use more than 50 security groups if you enable accounts. Also, make sure that the security groups have relatively short names.

Account Hierarchy

Accounts can be set up in a hierarchical structure, which enables you to give some users access to entire branches of the structure, while limiting permissions for other users by assigning them accounts at a lower level in the structure. Figure 5-8 shows a typical hierarchical account structure:

Figure 5-8 Account hierarchy



In this example, anyone with access to “A” will also have access to all three subordinate accounts (“Tech,” “Cost,” and “Time”).

If you use slashes (/) in account names, the logical account hierarchy is reflected in the physical directory structure on the file system—in other words, subdirectories are created for each new hierarchy level. Each of the subdirectories stores the content assigned to the corresponding account. In the example in Figure 5-8, the account names include “ENG”, “ENG/A”, “ENG/A/TECH”, “ENG/A/COST”, etc. This means there would be a subdirectory called ‘A’ under ‘ENG’, and separate subdirectories for the content associated with each project activity. The account “ENG/A/TECH” only grants access to the lowest-level account “Tech” of “Project A.” The account “ENG/A” grants access to “Project A” and all three accounts below it (“Tech,” “Cost,” and “Time”). The account “ENG” grants access to all accounts.



Important: The account prefix does not have to include slashes. For example, if you have accounts called *abc*, *abc_docs*, and *abcdefg*, all users who have access to the *abc* account will have access to the other two accounts as well. The content is then not stored on the file system using an account-based file directory structure.

Special Account Classifications

There are two special account classifications:

❖ **[documents without accounts]**

This is an easy way to define what permission an individual has on all content checked in without an account. By default, users have read/write/delete/admin (RWDA) permissions to documents without accounts. The permissions you have is the intersection between the permissions given to the “documents without accounts” account and the security groups your role give you permission to.

❖ **[all accounts]**

This is an easy way to define what permission an individual has on all content checked in with an account. The permissions you have is the intersection between the permissions given to the “all accounts” account and the security groups your role give you permission to.

Assigning Accounts

Account [permissions](#) are assigned to users similar to the way [role](#) permissions are assigned, through the User Admin tool. An account can also be assigned to each content item. To access a content item that has an account assigned to it, the user must have the appropriate permission to the account.



Note: For further details on User Admin and step-by-step procedures, refer to the *Managing Security and User Access* guide.

ADMINISTRATORS AND SUB-ADMINISTRATORS

Administrators are individuals who set up, maintain, and modify the configuration of the Content Server system and its user logins. Enterprise administrators can set up sub-level administrators to perform a subset of administrative tasks (for example, within individual departments). These sub-administrators maintain a portion of the content server system and its user logins.

To safeguard the integrity of the system, sub-administrators require a user name and password to access the part of the system they are allowed to.

Sub-administrators need to have Admin permission in the roles they should use to perform administrative tasks. Sub-administrator rights are useless without Admin permission associated with at least one security group.



Caution: A sub-administrator can grant the sub-administrator privileges they have to other users. It is therefore recommended that you are very careful about giving users sub-administrator privileges. If you do not control the sub-administrator privileges, they might spread around in the organization, which may compromise system security. Sub-administrators with UserAdmin rights can assign their own roles to other users. This is why they should have multiple roles. Otherwise, they can only assign their sub-administrator role to the users they administer (which means these users would have sub-administrator privileges as well).

Administrator Rights

The administrator can give certain rights to sub-administrators which allow them to work with one or more administration tools in Content Server. Rights are assigned to sub-administrators using User Admin, which is one of the tools accessed from the Administration page (Security—Permissions By Role). The following rights can be granted:

❖ **UserAdmin**

This right allows sub-administrators to work with the User Admin administration tool. They can add, edit, and delete users whose roles and accounts are a subset of the roles and accounts of the sub-administrator.

❖ **WebLayout**

This right allows sub-administrators to work with the Web Layout administration tool. They can add, edit, and delete web pages for the security groups and accounts that they have Admin permission to.

❖ **RepMan**

This right allows sub-administrators to work with the Repository Manager administration tool. They can view, update, approve, and delete files that they have Admin permission to.

❖ **Workflow**

This right allows sub-administrators to work with the Workflow administration tool. They can add, edit, and delete workflows which are in a security group that they have Admin permission to.

USING EXTERNAL SECURITY

Content Server security can be **internal**, which means you set up user security within Content Server using the User Admin application. You assign each user one or more **roles** (see page 5-8), which in turn are assigned specific **permissions** (see page 5-8) to security groups. If you have enabled **accounts** (see page 5-12), you can assign each user specific permissions to certain accounts, which then limits the permissions they might otherwise have through their assigned roles. User logins, passwords, and permissions are created and stored in the content server database.

Content Server security can also be **external**, which means that, rather than setting up users and assigning roles and accounts in the User Admin application, Content Server is integrated with an external user base (such as Active Directory, or other LDAP server). When users log in to the content server, they are assigned roles and accounts based on their user attributes defined in the external system.

User logins, passwords, and permissions can be derived from one of the following external user bases:

- ❖ **Active Directory**—User information is stored in a Microsoft Active Directory user base.
- ❖ **LDAP**—User information is stored in an LDAP-compliant user base, such as iPlanet.
- ❖ **Active Directory with LDAP Provider**—User information is stored in a Microsoft Active Directory user base, which the content server accesses using an LDAP provider. This may be the preferred approach if your network includes the following:
 - Content server access through non-Internet Explorer client browsers
 - Content server installed on a UNIX operating system
 - Firewall between the content server and user base computers

A custom Active Directory LDAP component, which is available in the extras directory, is required for this type of security integration.



Tech Tip: A Oracle content management system can combine authentication methods. For example, you can define some local users in the content server, allow some users to log in using their Microsoft domain identity, and grant other users content server access based on their Active Directory or LDAP credentials.

Users who are authenticated through external security (Active Directory, or LDAP) are considered *external* content server users. The first time external users log into the content server, they are added to the database, and administrators can view their information

through [Repository Manager](#) (see page 9-8). However, external users are not included in user lists, such as the Author field on a check-in page.

By default, external security integrations map only a limited set of user information (user name, password, roles, and accounts) from the external user base to the content server. If you are using Active Directory or LDAP integration, additional user information (such as e-mail address or user locale) can be mapped from a content server administration page. More sophisticated mapping techniques, such as mapping a domain prefix to a specific account name, can be provided through use of the Proxy Credentials Extension component. This component is available in the extras directory.



Note: For details on integrating Content Server with external user bases refer to the *Managing Security and User Access* guide.

SELF-REGISTRATION

Content Server can be set up to support self-registration, which enables users to create their own content server logins (user name and password). Self-registered users are created as [global users](#) (see page 5-4).

Self-registration can be combined with other user log-in strategies (for example, local users using the standard Login button, and external users using an integrated Windows login).



Note: For details on setting up self-registration refer to the *Managing Security and User Access* guide.

SETTING UP SECURITY



Important: It is essential that the security model is properly planned and designed. Only then can adequate security of the Content Server system be implemented.

Setting up the security model typically consists of the following steps:

1. Determine the number of [users](#) (see page 5-3).
2. Determine whether the [authorization types](#) of the users (local, global, or external) (see page 5-4).
3. Determine how the users will be authenticated: in Content Server or by an [external system](#) (Active Directory, or LDAP) (see page 5-18).
4. Determine the required security granularity. This is the basis for deciding on your [security groups](#), [roles](#), and [accounts](#) (see pages 5-6, 5-8, and 5-12, respectively).
5. Set up the [security groups](#) (see page 5-6).
6. Define the [roles](#) (see page 5-8)
7. Set up the [permissions](#) for the security groups and roles (see page 5-8).
8. Assign the [user logins](#) (see page 5-3).
9. Define [accounts](#), if required (see page 5-12).

The *Managing Security and User Access* guide provides more detailed and in-depth information about setting up security in Content Server, including practical and background information.

SECURITY RECOMMENDATIONS

The following is a series of recommendations for improving overall security on a Content Server instance. It is recommended that you use four types of security to completely secure Content Server:

- ❖ [File System Access to the Content Server Directory Structure](#) (page 5-21)
- ❖ [Network Access](#) (page 5-21)
- ❖ [Database Access](#) (page 5-22)
- ❖ [Physical Access](#) (page 5-22)



Note: For more details on setting up Oracle security refer to the *Managing Security and User Access* guide.

File System Access to the Content Server Directory Structure

Secure the file system to allow access to only those operating system accounts that require access.

Read Access

- ❖ Grant read access to the system administrator to check on log files.
- ❖ Grant read access to those who need to perform regular backups and periodic disaster recovery backups.
- ❖ Grant read access to the account that runs the web server to access and deliver files from the content server web site to the user's browser. The website includes the files stored under the *weblayout* directory, the *data* directory, and the *idcplg* directory. Depending on the web server and security integration used, this may be a single operating system account. The web server does not need access to other directories.

Write Access

- ❖ Grant write access to the system administrator to install new software and perform customizations.
- ❖ Grant write access to the content server and optionally Inbound Refinery and its add-ons (all the same account).

There should not be a need to grant any other account access to the Oracle directory structure (unless you are running some other process to access data directly).

Network Access

Configure the network to only allow access to the Oracle directory structure through the Content Server application. If file sharing is required to allow an optional refinery connection on another computer access to the Oracle directories, then that sharing should be set up to allow only Inbound Refinery access to the directories. The *data* directory and the *config* directory contain user names and passwords. These directories should not be shared on the network.

For extra security, it is recommended that transmissions to and from the web server be encrypted using SSL.

Database Access

Content Server uses a single database account to access data stored in the database. The database user name and password should be chosen so that they are hard to break and should be updated periodically.

Physical Access

Keep the server that is running Content Server in a locked room, and make sure only authorized persons have access to it.

STARTING AND STOPPING A CONTENT SERVER

OVERVIEW

The way to start and stop a Content Server instance depends on the platform that the software is running on:

- ❖ [Microsoft Windows](#) (page 6-1)
- ❖ [UNIX](#) (page 6-5)

MICROSOFT WINDOWS

This section covers the following topics:

- ❖ [Startup Options](#) (page 6-2)
- ❖ [Starting a Content Server](#) (page 6-2)
- ❖ [Stopping a Content Server](#) (page 6-4)
- ❖ [Restarting a Content Server](#) (page 6-5)

Startup Options

There are two options in Microsoft Windows to start a content server instance:

❖ **Automatically starting Content Server as a service**

If Content Server runs as a Windows service, the content server software can start automatically when Windows starts. This enables you to launch a content server instance without logging into Windows.

❖ **Manually starting Content Server as an application**

You can also [start](#) Content Server manually, using either Admin Server or the command line (see below).

You initially make the choice between starting Content Server automatically (as a service) or manually (as an application) when you install the Content Server software. Regardless of what you choose, a service called “IDC Content Service [*Server Name*]” is set up for each content server instance. Depending on what you chose during setup, its startup type is set to either ‘Automatic’ or ‘Manual’. If you want to change this later, you need to change the startup type by choosing Start—Settings—Control Panel—Administrative Tools—Services.



Note: Another service is always set up in addition to the service called ‘IDC Content Service [*Server Name*]’. This second service is called ‘IDC Content Admin Service’, and it enables you to administer a content server remotely using HTML pages.

Starting a Content Server

You can start a content server running on Microsoft Windows in either of two ways:

- ❖ [Using Admin Server](#)
- ❖ [Using the command line](#)

Admin Server

If the IDC Content Admin Service is running, you can also start a content server from remotely within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page and click **Admin Server**.
3. All available servers are listed. To start a content server, click its ► symbol. (If this symbol is grayed out, it means the server is already running.)



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.

Command Line

You can also start a content server and admin server from the command line using the Launcher, if required with execution flags (parameters). This may be useful for tracing or troubleshooting purposes.



Note: The Launcher is a native C++ application that is provided with Content Server. It allows a Java program to start as a Windows service. For more information about the Launcher and its usage (including the available executable flags or parameters), refer to the *Idc Command Reference Guide*.

Starting the Content Server

To start the content server from the command line using the Launcher, complete the following steps:

1. Open a command-line window (console), and go the following directory:

```
[CS_Dir]\bin\
```

2. Start the following executable file to start the content server:

```
IdcServer.exe
```

The content server will start. The console remains open and shows status information. If the instance was already running, an error message is displayed and the console window remains open.



Tech Tip: If you want debug information to be displayed in the console, start the content server with the following executable flags (parameters):

```
IdcServer.exe -console -debug
```



Tech Tip: You can use the *intradoc.cfg* configuration file in the same directory as *IdcServer.exe* to control the way the content server is started. For more information, refer to the *Idc Command Reference Guide*.

Starting the Admin Server

To start the admin server from the command line using the Launcher, complete the following steps:

1. Open a command-line window (console), and go the following directory:

```
[CS_Dir]\admin\bin\
```

2. Start the following executable file to start the admin server:

```
IdcAdmin.exe
```

The admin server will start. The console remains open and shows status information. If the instance was already running, an error message is displayed and the console window remains open.



Tech Tip: If you want debug information to be displayed in the console, start the admin server with the following executable flags (parameters):

```
IdcAdmin.exe -console -debug
```



Tech Tip: You can use the *intradoc.cfg* configuration file in the same directory as *IdcAdmin.exe* to control the way the admin server is started. For more information, refer to the *Idc Command Reference Guide*.


Stopping a Content Server

If Content Server is running as a **Windows service**, you can stop the service by choosing Start—Settings—Control Panel—Administrative Tools—Services. Select the appropriate service, right-click, and choose **Stop** from the popup menu.

If Content Server is running as an **application**, you can stop the server by simply closing the Content Server window on the server.

Admin Server

If the IDC Content Admin Service is running, you can also stop a content server remotely from within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page and click **Admin Server**.
3. All available servers are listed. To stop a content server, click its  symbol. (If this symbol is grayed out, it means the server is already stopped.)



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.

Restarting a Content Server




Note: You may need to restart the content server if a new component is enabled or if configuration settings or configuration file entries have been changed.

If Content Server is running as a **Windows service**, you can restart the service by choosing Start—Settings—Control Panel—Administrative Tools—Services. Select the appropriate service, right-click, and choose **Restart** from the popup menu.

If Content Server is running as a desktop application, you can restart (stop and start) the server by simply closing the Content Server window and starting it again (see [Startup Options](#) on page 6-2).

Admin Server

If the IDC Content Admin service is running, you can also restart a content server from within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page, and click **Admin Server**.
3. All available servers are listed. To restart a content server, click its  symbol.



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.

UNIX

This section covers the following topics:

- ❖ [Starting a Content Server](#) (page 6-6)
- ❖ [Stopping a Content Server](#) (page 6-7)
- ❖ [Restarting a Content Server](#) (page 6-7)

Starting a Content Server

You can start Content Server running on UNIX in either of two ways:

- ❖ [Using UNIX commands](#)
- ❖ [Using Admin Server](#)

UNIX Commands

UNIX commands enable you to launch a content server without logging into the system and administer content servers from a remote system:

- ❖ **idcserver_start**
This command starts a content server in the background. It is available from the */etc* subdirectory of the Content Server installation directory.
- ❖ **idcserver_query**
This command checks the status of a Content Server to determine if it is running. It is available from the */etc* subdirectory of the Content Server installation directory.
- ❖ **idcserver_ctrl**
This is a script file suitable for use as an rc file, which contains startup instructions for launching Content Server automatically each time the system starts up. It is available from the */etc* subdirectory of the Content Server installation directory.
- ❖ **idcadmin_start**
This command starts Content Server Admin Server, which then enables remote, HTML-based administration of content server instances on that system. It is available from the *admin/etc* subdirectory of the Content Server installation directory.

Admin Server

If Admin Server is running, you can also start a content server from remotely within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page and click **Admin Server**.
3. All available servers are listed. To start a content server, click its ► symbol. (If this symbol is grayed out, it means the server is already running.)



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.


Stopping a Content Server

The UNIX command `idcserver_stop` enables you to stop a content server that was started with `idcserver_start`. It is available from the `/etc` subdirectory of the Content Server installation directory.

The UNIX command `idcadmin_stop` enables you to stop Content Server Admin Server. It is available from the `admin/etc` subdirectory of the Content Server installation directory.

Admin Server

If Admin Server is running, you can also stop a content server from within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page and click **Admin Server**.
3. All available servers are listed. To stop a content server, click its  symbol. (If this symbol is grayed out, it means the server is already stopped.)



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.

Restarting a Content Server



Note: You may need to restart the content server if a new component is enabled or if configuration settings or configuration file entries have been changed.

UNIX Commands

The UNIX command `idcserver_restart` enables you to stop a content server that was started with `idcserver_start` and start it again. It is available from the `/etc` subdirectory of the Content Server installation directory.

Admin Server

If the Content Server Admin Server is running, you can also restart a content server from within the Content Server software:

1. Make sure you are logged in with sufficient administrator privileges.
2. Go to the Administration page, and click **Admin Server**.

3. All available servers are listed. To restart a content server, click its  symbol.



Note: You need to have the ‘sysmanager’ role to access a content server through Admin Server.

WORKFLOW

OVERVIEW

This section covers the following topics:

- ❖ [Workflows](#) (page 7-1)

WORKFLOWS

Workflows are useful in the process of reviewing and approving content before it is released and published to the website. They specify how content is routed and who needs to review and approve it. Users are notified by e-mail when they have a file to review.

Workflows are defined and managed using Workflow Admin, which is one of the tools accessed from the content server's administration page. Only persons with administrator or sub-administrator privileges can create workflows. Defined workflows can be turned on and off. This means that workflows can be temporarily disabled, if required.

The following topics are important with regard to workflows:

- ❖ [Workflow Types](#) (page 7-2)
- ❖ [Workflow Steps](#) (page 7-3)
- ❖ [Jumps](#) (page 7-3)
- ❖ [Tokens](#) (page 7-4)
- ❖ [Templates](#) (page 7-4)
- ❖ (page 7-5)



Note: The Content Management menu contains a link called Active Workflows, which displays all workflows that are currently enabled.



Note: The My Oracle menu contains a link called My Workflow Assignments, which displays a list of content items that the user has been assigned to act on.

Workflow Types

There are three types of workflows:

- ❖ [Basic Workflows](#) (page 7-2)
- ❖ [Criteria Workflows](#) (page 7-2)
- ❖ [Sub-Workflows](#) (page 7-3)

Basic Workflows

In a **basic** workflow, files are specifically identified in the workflow, along with the contributors, reviewers, and steps. This type of workflow requires an administrator to initiate the process, and is best suited for groups of content items that need to go through a workflow together or individual content items with unique workflow requirements.

The administrator or a sub-administrator selects one or more specific files for entry into the workflow (using content IDs), and defines the workflow steps and the reviewers for each step.

Criteria Workflows

In a **criteria** workflow, files automatically go into a workflow if the values entered in the metadata fields upon check-in meet certain criteria. Criteria workflows are useful for individual content items that are approved by the same reviewers on a regular basis (newsletter articles, for example).

The administrator or a sub-administrator selects the entry criteria for the workflow, consisting of a security group and a value for one content information (metadata) field, and defines the workflow steps and the reviewers for each step.

For example, if strategic reports must always be reviewed and approved by key individuals before being released, a criteria workflow could be set up for this content type and security group. If a strategic report is then checked into the content server, a workflow is automatically initiated to start the approval process.

Sub-Workflows

A sub-workflow is a workflow that does not have an initial contribution step. Sub-workflows are useful for splitting large, complex workflows into more manageable pieces. A file can enter a sub-workflow only through a jump from a criteria workflow.

Workflow Steps

A workflow consists of one or more steps, and multiple users can be assigned to review the content at each step. There are four types of steps:

- ❖ A **contribution** step is the initial step of a basic workflow. Contributors are defined when the workflow is created.
- ❖ An **auto-contribution** step is the initial step of a criteria workflow. There are no predefined users involved in this step.
- ❖ In a **reviewer** step, the assigned users can only approve or reject the file. Editing is not allowed.
- ❖ In a **reviewer/contributor** step, users can edit the file, if necessary, and then approve or reject it.

If there is more than one user assigned to a step, it is possible to specify how many of them need to approve the content before it moves to the next step.

All persons involved in a workflow are notified about any actions they need to perform for each step. This is done entirely through e-mail. E-mail messages can also be sent to content authors and other users to inform them of the status of the workflow.

Jumps

Jumps enable you to customize workflows for your system, your content, and your users. Jumps are created using Idoc Script, which is Content Server's proprietary scripting language.

Typical uses of jumps include:

- ❖ Specifying more than one metadata field as the criteria for entering a workflow,
- ❖ Taking action on content automatically after a certain amount of time has passed, and
- ❖ Defining different paths for files to move through the same workflow depending on metadata and user criteria.

Tokens

A token is a piece of Idoc Script that defines variable users in a workflow. Tokens can be used for any of the following:

- ❖ Specify a variable user, such as the original author or the author's supervisor,
- ❖ Include users and aliases in workflow jumps, and
- ❖ Define users through conditional statements.

Aliases

An alias is a single name assigned to a group of named users. Aliases make it easier to manage users in workflows.

Suppose, for example, that a group of area managers needs to review content in various workflows. Instead of adding all individual area managers one at a time to every workflow they are involved with, you can create an alias called 'Area Managers,' which contains the names of all area managers. This group designation can then be reused in all workflows. Similarly, if all department managers need to be notified if certain documents have changed, then an alias could be used to easily subscribe the group of department managers to the documents.

Aliases also allow roles to be linked to positions rather than named users. For example, an alias could be created called 'Marketing Manager,' which is linked to, say, John Fisher. The marketing manager is involved with content review in a number of workflows, and is also subscribed to a number of documents. If John Fisher leaves the company and no alias was defined, you would need to go through every single workflow and subscription, and change the name. With an alias called 'Marketing Manager,' however, all you need to do is change the alias once, and all workflows and subscriptions are automatically updated to reflect the new name.



Note: Aliases are defined and managed using User Admin, which is one of the tools accessed from the Administration page.

Templates

There are two types of workflows-related templates:

- ❖ [Workflow Templates](#)
- ❖ [Script Templates](#)

Workflow Templates

Workflow templates are a quick way to reuse workflows that you have already created. Each workflow template is an outline for a basic workflow, criteria workflow, or sub-workflow that is stored in the Workflow Admin tool. A workflow template is not tied to a security group, and it cannot include jumps.

For example, if the first and last step of several workflows need to be the same, you could save these steps as a workflow template, and then use the template as the starting point for creating the individual workflows.

Script Templates

Script templates are a quick way to reuse jumps that you have already created. Each script template is a piece of Idoc Script stored in the Workflow Admin tool.

For example, if you have several workflow steps that require approval within one week, you could save the jump script for this as a template, and then reuse the script for creating the individual steps.

WORKING WITH CONTENT

OVERVIEW

This section covers the following topics:

- ❖ [Grouping of Content](#) (page 8-2)
- ❖ [Content Information \(Metadata\)](#) (page 8-3)
- ❖ [Content Identification](#) (page 8-4)
- ❖ [Content Storage](#) (page 8-8)
- ❖ [Content Profiles](#) (page 8-10)
- ❖ [Content Subscriptions](#) (page 8-11)
- ❖ [Checking In Files](#) (page 8-12)
- ❖ [Checking Out Files](#) (page 8-14)
- ❖ [Updating Content](#) (page 8-16)
- ❖ [Searching for Content](#) (page 8-17)

GROUPING OF CONTENT

Content Server manages business content, which is all information required to operate a business. All this content needs to be managed, accessed, and made available efficiently and securely. It should be easy to contribute new content and find existing content. In addition, content should be adequately protected, which means only the right people should be able to access it. After all, not all content is relevant to everyone or intended for all eyes in the organization.

One of the ways Content Server manages content is by grouping it. This can be done in a number of ways:

❖ **Content types**

When checking in content, contributors assign the content to a certain type, for example “Financial Statement” or “Sales Report.” This enables clustering of similar content into functional groups. The content types are fully user-definable, which means they can be geared to any business situation. The content types in a sales-oriented environment could, for example, include “Sales Presentation,” “Marketing Report,” and “Forecast,” whereas a research company might use content types such as “Technical Specification,” “Test Report,” and “Installation Guide.”



Note: Content types are defined and managed using Configuration Manager, which is one of the tools accessed from the Administration page.

❖ **Security groups**

When checking in content, contributors also assign the content to a security group. Security groups are sets of content grouped under a unique name. They are used to restrict access to the content. (Please note that security groups in a Content Server context are not groups of people as in Microsoft Windows, but rather groups of content.) Security groups could, for example, be based on departments (for example, “Sales” or “Marketing”) or location (“Boston,” “Seattle”). There are two predefined security groups: ‘Public’ and ‘Secure’. Content checked in with the ‘Public’ security group may be viewed by all consumers. The ‘Secure’ security group is typically used by the administrator to restrict access to content.



Note: Security groups are defined and managed using User Admin (Security—Permissions By Group), which is one of the tools accessed from the Administration page. For further details on the use of security groups, refer to [Chapter 5 \(Content Server Security\)](#).

CONTENT INFORMATION (METADATA)

Content Server uses “metadata” to manage content. Metadata basically is information about the content. It is used to describe, find, and access the content. As such, metadata is not unlike catalog cards in a library, where the information on the cards is used to keep track of the actual books, enabling librarians and visitors to locate them.

Content Server maintains a set of metadata for every file checked into the system. The metadata is stored in the content information database, which contains the metadata of all revisions of all the content checked into the content server.

Content contributors provide metadata when they check a file into the content server. Some content information fields are required (for example, Content ID, Type, Title, Author, and Security Group), whereas others are optional (for example, Comments, Expiration Date). Some of the fields may have default values (for example, Release Date, which defaults to the current date and time).



Note: The content ID needs to be unique for every file checked in. You can set up the system in such a way that content IDs are generated automatically.

The Content Server software is shipped with a set of predefined, standard content information fields (metadata fields). In addition to these predefined fields, you can also create new fields to enhance functionality and accommodate site-specific design requirements. Depending on the application, custom-defined fields such as “Part Number,” “Project Status,” and “Customer Name” could be useful. An application’s metadata is derived from how a company does business and organizes information.

The metadata is used to organize the content and make it easier for users to find content. It is possible to search for content based on matching criteria for metadata (for example, all content of which the Type field has the value “Sales Report”).



Note: Content information fields are defined and managed using Configuration Manager, which is one of the tools accessed from the Administration page.



Note: There is an optional add-on called Content Categorizer, which at document check-in automatically and intelligently suggests specified metadata values that are derived from the content of the document. This enables text-based content to be categorized so that users can easily find them when they need to.

CONTENT IDENTIFICATION

This section covers the following topics:

- ❖ [Content Item Identifiers](#) (page 8-4)
- ❖ [File Naming](#) (page 8-6)
- ❖ [Maximum File Sizes](#) (page 8-7)

Content Item Identifiers

dID

Each content item that is checked into the content server is assigned a unique identifier which is stored in a metadata field called **dID**. This metadata attribute is internal to Content Server, and is used to reference the content item. A new dID is assigned to *every* new item that is checked in, even if that item is a new revision of an existing content item. This allows the content server to reference each specific revision of a content item.

dIDs are assigned incrementally, starting with 1—that is, the first item that is checked in is assigned the dID 1, the second one 2, the third one 3, etc. If an item is deleted, its dID is not reused. This means that “gaps” in the dID sequence may occur over time.



Important: Once a dID has been assigned to an item, it cannot be changed.



Tech Tip: You can see the dID of a content item only in the [Repository Manager](#) utility (in the ID column on the Content tab). See page 9-8 for more information about Repository Manager.

Content ID (dDocName)

A new content item is also assigned a **content ID**, which is stored in a metadata field called **dDocName**. Contrary to dID, the content ID does *not* change with each new content item revision, which means that the content ID can be used to refer to the latest revision of a content item. This is very useful when creating “[persistent](#)” [hyperlinks or URLs](#) to items in the content server (see page 8-6).



Important: Once a content ID has been assigned to an item, it cannot be changed.



Tech Tip: Content IDs are assigned upon check-in, and it is one of the mandatory metadata fields. The system may be set up to allow users to provide their own content IDs with each check-in. This enables them to create “meaningful” content IDs for every item they check in. Alternatively, the system may be set up to generate content IDs automatically as items are checked in. This will be a numerical sequence that is incremented by one with each new check-in, preceded by a user-defined prefix if required. This method does not produce “meaningful” content IDs, but does have the advantage that all content IDs are formatted in a uniform manner (to fit the corporate standard) and there is a clear sequential order, for example DOC_00001, DOC_00002, DOC_0003, etc. You change this setting using the [System Properties](#) utility (Options tab). See page 9-5 for more information about System Properties.

Content Title (dDocTitle)

A content item is also assigned a **content title**, which is stored in a metadata field called **dDocTitle**. This field provides a descriptive name of the content item, which can be used in metadata searches. Content titles are assigned upon check-in, and it is one of the mandatory metadata fields.



Note: You can modify the content title of an item, either by editing the item’s metadata attributes or by providing a new title when checking in a new item revision.

The table below shows examples of how checked-in content items are identified:

Item	dID	Content ID	Content Title
Item 1, Rev. 1	1	DOC_001	Title of Item 1
Item 2, Rev. 1	2	DOC_002	Title of Item 2
Item 3, Rev. 1	3	DOC_003	Title of Item 3
Item 1, Rev. 2	4	DOC_001	Title of Item 1
Item 1, Rev. 3	5	DOC_001	Updated Title of Item 1
Item 2, Rev. 2	6	DOC_002	Updated Title of Item 2

File Naming

The names of the files in the [native file repository](#) (or “Vault”) are not identical to their original file names. They are given a system-internal file name, which is a unique [sequential number](#) (dID) based on the order in which they were checked in (see page 8-4). The original file extension is left untouched. For example, if a Microsoft Word file is the 25th file to be checked into the content server, then it will be called *25.doc* in the native file repository. If the next file to be checked in is a PowerPoint file, its file name in the native file repository will be *26.ppt*.

The files in the [web-viewable file repository](#) (or “Web Layout”) are named after their content IDs. Consider, for example, a new Microsoft Word file with the content ID ‘DOC_001’, which is checked into the content server and converted to a web-viewable PDF file. It will then be called *DOC_001.pdf* in the web-viewable file repository. This file name will never change for the latest revision of the file. If a new revision is created (by checking out the file and checking it in again), the web-viewable file of the new revision will still be called *DOC_001.pdf*. Old revisions are indicated by adding a tilde (~) and the revision number to the file name, for example *DOC_001~1.pdf*, *DOC_001~2.pdf*, etc. This is very useful when referring to content. It enables the use of “persistent URLs,” which means that URLs (web addresses) to content always refer to the latest version of that content, regardless of any new revisions that were checked in since the original reference was made.

The table below shows examples of how native files and web-viewable files are named:

Item	Native File Name	Web-Viewable File Name
Item 1, Rev. 1	Document.doc	DOC_001.pdf [upon initial check-in] DOC_001~1.pdf [upon check-in of rev. 2]
Item 2, Rev. 1	Presentation.ppt	DOC_002.pdf [upon initial check-in] DOC_002~1.pdf [upon check-in of rev. 2]
Item 3, Rev. 1	Spreadsheet.xls	DOC_003.pdf
Item 1, Rev. 2	Document.doc	DOC_001.pdf [upon initial check-in] DOC_001~2.pdf [upon check-in of rev. 3]
Item 1, Rev. 3	Document.doc	DOC_001.pdf
Item 2, Rev. 2	Presentation.ppt	DOC_002.pdf

Maximum File Sizes

The maximum file size of content items depends mostly on your file system's capabilities. In addition, the storage capability of your database plays a role. No actual content files are stored in the database, just their file sizes in bytes. This information is stored in the `dFileSize` column of the Documents table, which means that the file size is limited by the maximum capacity of this database field.

Content Server—like all Java programs—uses a 64-bit integer for file sizes and offsets, so your maximum theoretical size for content items is 1,8446,744,073,709,551,615 bytes or 16,384 petabytes. In practice, you would want to ensure that nothing in your system breaks as you go past 31-bit, 32-bit, 41-bit, and 42-bit integers—that is, 2 gigabytes, 4 gigabytes, 2 terabytes, and 4 terabytes. If you can contribute a 4 terabyte file, you should be good to go all the way up to 16 petabytes.

Database Capacity

These are the maximum capacities for the `dFileSize` field for various databases that can be used with Content Server:

SQL Server:	<code>bigint 2⁶³ - 1 = 8 petabytes</code>
Oracle:	<code>int 10³⁷ = ?</code>
Sybase:	<code>numeric(19) 2¹⁹ = 8.8 exabytes</code>
DB2:	<code>BIGINT 2⁶⁴ = 16 exabytes</code>
PostgreSQL:	<code>int8 2⁶⁴ = 16 exabytes</code>

File Systems

These are the maximum file sizes that can be handled by various file systems:

NTFS (Windows):	16 terabytes
FAT32 (Windows):	4 gigabytes
UFS (Solaris):	1 terabyte
ext2 (Linux):	2 terabytes
HFS (HP-UX):	128 gigabytes

JFS or VxFS (HP-UX):	2 terabytes
JFS (IBM AIX):	64 gigabytes
JFS2 (IBM AIX):	16 terabytes

CONTENT STORAGE

This section covers the following topics:

- ❖ [Storage of Content Files](#) (page 8-8)
- ❖ [Storage of Metadata](#) (page 8-9)
- ❖ [Storage of Full Text](#) (page 8-10)

Storage of Content Files

Content files are stored on a file system, which may be on the same computer that the Content Server software is running on, or it may be on a shared network location.

If a content item is checked into Content Server, its corresponding file is stored in two places on the file system: the original file is stored in the [native file repository](#) (or “Vault”) and another file rendition in the [web-viewable file repository](#) (or “Web Layout”). This file rendition will typically be a converted file that can be viewed in a web browser (for example, PDF or HTML).



Note: If no conversion module is running or if the native file format cannot be converted, the content file in the web-viewable file repository is an exact copy of the one in the native file repository (although their [file names](#) are different).



Note: No actual content item files are stored in the database, just their [file sizes](#), their [metadata](#) and, possibly, their extracted [full text](#) (depending on the search solution used).

Storage of Native Files

The default location for the **native file repository** is `[CS_Install_Dir]/vault`, and content items are stored in subdirectories that represent the defined [security groups](#). This means that a native content file could, for example, be stored as follows (on a Windows platform):

```
C:\Stellent\idcm1\vault\adacct\9456.doc
```

Storage of Web-Viewable Files

The default location for the **web-viewable file repository** is `[CS_Install_Dir]/weblayout/groups`, and content items are stored in subdirectories that represent the defined **security groups** (below the predefined ‘public’ and ‘secure’ security groups). This means that a web-viewable content file could, for example, be stored as follows (on a Windows platform):

```
C:\Stellent\idcm1\weblayout\groups\public\documents\adacct\ id_0435.pdf
```



Note: The names of the files in the native and web-viewable file repositories are not identical and they are not the same as the native file name. See below for more information.



Tech Tip: If the content server is set up to use **accounts**, subdirectories are added to the web-viewable file path to reflect the accounts that apply to the content item. See the *Managing Security and User Access* guide for more information.



Caution: Under no circumstance should you try and manipulate files in the native and web-viewable file repositories outside the Content Server environment (for example, using Windows Explorer). This may seriously jeopardize system integrity.

Storage of Metadata

Information about the content (“metadata”) is stored in the **database** that is used with the content server.

If the database is used only for **metadata searching**, then all metadata of all revisions is stored in the database once. Content Server then uses the release state to identify the latest revision of a content item.

If the database is also used for **full-text searching**, then two copies of the standard metadata are maintained in the database: one for the latest revisions and one for all older revisions. Please note that xComments and any custom metadata fields are stored only once.



Note: If an external search engine is used, then both metadata and full-text searching are handled by that search engine. However, you can configure the system so that metadata searching is handled by the database and full-text searching by the external search engine.

Storage of Full Text

The location of the full text of content items depends on the full-text search solution that is used with the content server:

- ❖ **Database full-text searching**—The full text of the content items is extracted from the file upon check-in (by Content Server) and stored in the [database](#). Only the text of the latest revisions are maintained.
- ❖ **External full-text search engine**—The full text of the content items is extracted from the file upon check-in (by the external search engine) and stored in an external search index. Only the text of the latest revisions are maintained.

CONTENT PROFILES

The metadata fields displayed on the content check-in, update, info, and search pages often make these pages complex to use. A myriad of metadata fields may be displayed, and many are unused by most end-users. Administrators can use content profiles to configure these pages so end-users see only metadata fields directly related to specific content types or security groups. Content profiles are used as a type of filter for what information will be displayed.

A complete content profile contains a set of one or more rules that are used to control the display of metadata fields on the check-in and search pages. For example, a profile's rules can determine the user type and, depending on the content type being checked in, ensure that only specific, relevant metadata fields are displayed. All other fields can be hidden.

The criteria for the rules can be based on user attributes, document attributes, or a combination of the two. The rules determine whether fields are editable, required, hidden, excluded, or read-only based on their criteria. Administrators can create multiple content profiles and all are available to the end-user. For each profile, the end-user has a distinct check-in page and search page available. Although all profiles are visible to all users, each user can configure their user interface to hide or display links to specified profiles.

These are two possible scenarios for using content profiles:

- ❖ **Customized check-in and search metadata fields:**

A typical company using content server to manage its internal knowledge database will generally have users contributing many types of content items, ranging from memos and press releases to technical specifications and design documents. A very diverse user base can result in the creation of many extra metadata fields. A side-effect of this is complex check-in, update, info, and search pages with many metadata fields

which may not be relevant to all users.

This complexity causes end-user confusion. For example, the customer relations manager sees fields such as Project Name and Expected Release Date that are irrelevant to him/her while an engineer finds that fields such as Customer ID and Last Contact Date are unnecessary. Yet, the engineer feels that Project Name and Expected Release Date should be required. Profiles tailored to the needs of each company department would alleviate these kinds of frustrations.

❖ **“Black-hole” check-ins:**

A company may need certain documents to be checked in so the contributors are unable to retrieve them (such as résumés). In this case, a profile can be created that automatically restricts accessibility to the checked-in document. For example, a rule in a “Resume” profile could hide the Security Group metadata field on the content check-in page but upon submit, the value is set to Human Resources by default.



Note: For further details on content profiles, including information on setting them up, refer to the *Managing Repository Content* guide.

CONTENT SUBSCRIPTIONS

Subscriptions are automatic e-mail notifications from the system to users alerting them that a specific file has been revised. Subscriptions are useful if a user wants to be kept up to date about a file without constantly having to look for the file manually and check its latest revision.

There are two types of subscriptions:

- ❖ [File Subscriptions](#) (page 8-11)
- ❖ [Criteria Subscriptions](#) (page 8-12)



Note: You can display a list of all files you are subscribed to by going to your User Profile page and clicking the Subscriptions for ‘[user name]’ link.

File Subscriptions

File subscriptions allow users themselves to select specific files that they want to be kept up to date on. The users are sent an automatic e-mail notification anytime a new revision is checked in for the file they subscribed to.

Users can subscribe to individual files by searching for the file(s) they want to subscribe to, and then clicking the Subscribe button on the content information page for each file.

(If you subscribe to a file, that button is renamed ‘Unsubscribe’ to allow you to cancel the subscription.)

Criteria Subscriptions

With criteria subscriptions, an administrator decides for users what files they are subscribing to. In other words, an administrator “forces” a subscription onto users. Administrators can subscribe specific users to specific groups of files that meet certain criteria. For example, they can set up a subscription that subscribes all users with the ‘Sales’ role to all files of content type ‘Monthly Sales Report.’ These users will then receive an e-mail notification every time any content of that type is updated. Similarly, all users within a particular department could be subscribed to the departmental organization chart.



Note: Criteria subscriptions are defined and managed using Repository Manager, which is one of the tools accessed from the Administration page.

CHECKING IN FILES

Files are checked into the content management system, which means the files themselves are copied to the native file repository (“library”), and their metadata is stored in the content information database. If an optional conversion module is running, a web-viewable file may also be created and stored in the Web Layout. If no conversion module is running, the native file is stored in the Web Layout (in addition to the library).

You can only check in files if you are properly logged into the system (or authenticated) and have been granted the appropriate privileges.

Files are checked in using the Content Check In Form, which is opened by clicking New Check In in the user interface (this option may be under a Content Management menu). Figure 8-1 shows the out-of-the-box content check-in form.

Figure 8-1 Example of content check-in form

The content check-in form typically contains a number of fields that must be completed before a file can be checked in (indicated in red). In addition, there are fields that may be completed, but do not need to be for the file to be checked in. The fields are the content information fields (also called ‘metadata fields’). They provide information about the content, which enables the system to keep track of the files and helps users find files. There are a number of predefined fields, but custom fields can also be added to suit specific applications.

Some fields have default values (for example, Revision and Release Date), which can be overridden by entering new values. Also, some fields require you to choose from a list of predefined options (for example, Type, Author, and Security Group).

After providing all the content information, you can check the file into the system by clicking the Check In button.



Note: Content Server is highly customizable, so the default look and feel and functionality of a content check-in page can be completely altered to suit virtually any business need.



Note: There is a special Content Server component that enables WebDAV, or Web-Based Distributed Authoring and Versioning. This enables you to remotely author and manage Content Server content using clients that support the WebDAV protocol. For example, you can use Microsoft Windows Explorer to check content into the Content Server repository rather than using Content Server's web browser interface, or you can check in content directly by saving a document in Microsoft Word.

CHECKING OUT FILES

If a file is checked out, a copy of the file in its native format is made available for further processing (for example, editing). You can only check out files if you are properly logged into the system (or authenticated) and have been granted the appropriate privileges.

Only one person at a time can check out a file. However, others can continue to view the released file. Depending on how the system is configured, you may only be able to check out files that you authored yourself (which means the value of the 'Author' field matches your user login).

To check out the file, [search](#) for the content item you want to check out, and do either of the following:


- ❖ On the content information page or search results page, choose the Check Out option from the Actions dropdown list (see Figure 8-2).
- ❖ On the search results page, you can also click the Action icon () and choose the Check Out option from the popup menu (see Figure 8-3).

Figure 8-2 Checking out a content item from its content information page

Content Information

Content ID : tips
Revision: 1
Type: ADENG - Acme Engineering Department
Title: Tips For Using the Documentation
Author: sysadmin
Comments:
Security Group: Public
Checked Out By:
Status: Done
Formats: Application/pdf

Actions: Select an action
 Select an action
 Check Out
 Update
 Check In Similar
 Send link by e-mail
 Subscribe

Links

Web Location: <http://scstest7/stellent/groups/public/documents/adeng/tips.pdf>
Native File: [tips_en.pdf](#)

Revision History

Revision	Release Date	Expiration Date	Status	Actions
[1]	12/3/04 11:16 AM	None	Done	Delete

Figure 8-3 Checking out a content item from its content information page

Search Results Found 1 item matching the query.

Actions: Select an action

ID	Title	Date	Author	Actions
tips	Tips For Using the Document...	12/3/04		Content Information Check Out Check In Similar Send link by e-mail

The native file is now presented as a hyperlink for downloading. At the same time, the file is blocked and cannot be checked out by other users (they can still view the latest released version). If a file is checked out, the Check Out options are replaced with Undo Check Out options, which enable you to cancel the checked-out status of the content item. In that case, the system restores the situation as it was before the file was checked out, and the revision number is left untouched.



Note: There is a special Content Server component that enables WebDAV, or Web-Based Distributed Authoring and Versioning. This enables you to remotely author and manage Content Server content using clients that support the WebDAV protocol. For example, you can use Microsoft Windows Explorer to check out content from the Content Server repository rather than using Content Server's web browser interface, or you can check out content directly by opening a document in Microsoft Word.

Displaying a List of Checked-Out Content

It is possible to display a list of content that is currently checked out. There are two options:

❖ **Display all content in the system that is currently checked out**

If you click the Checked-Out Content link under Content Management, a list is displayed with all the content in the system that is currently checked out (by all users). You can check in files from this list, but only the ones you checked out yourself (unless you have administrator rights).

❖ **Display content checked out by the current user**

If you click the My Checked-Out Content link under My Stellent, a list is displayed with the content in the system that has been checked out by the current user. You can check in files from this list.


UPDATING CONTENT

Updating content can be either of two things:

❖ **Creating a new revision of a file already checked into the system**

To create a new revision of a checked-in file, [check out](#) the file, edit it, and [check it in](#) again. A new revision is then created, with the revision number incremented by one. The system keeps all revisions of a file, which means you can retrieve an older version of a file if you need to. Search queries, however, always relate to the most current revision.

❖ **Editing the content information of a file already checked into the system**

It is also possible to edit the content information (“metadata”) of a file without checking it out. Each item in a search result page has an info icon () , which links to the content information page for that item. Here you can review and, if necessary, edit the content information (for example, to correct a typo). If you submit the changes, the revision number is not incremented by one.

SEARCHING FOR CONTENT

There are various ways to retrieve information from the content server:

- ❖ [Navigating to Content](#) (page 8-17)
- ❖ [Searching Content Information \(Metadata\)](#) (page 8-17)
- ❖ [Performing a Full-Text Search](#) (page 8-21)
- ❖ [Using Repository Manager to Look Up Content](#) (page 8-22)

Navigating to Content

The out-of-the-box interface layout has a menu called Browse Content, which contains one or more links that represent hierarchical structures of hyperlinks. These enable you to easily navigate (“drill down”) to files. The lower you get in the hierarchy, the more specific the content will get (for example, Division—Department—Document Type — Year—[File]). Going through the hierarchy, you will only see hyperlinks that you are entitled to see. For example, a ‘Strategic Reports’ link might be defined somewhere in the library hierarchy. If you do not have the privileges to access this category of files, then the hierarchy that you browse through will not have a link called ‘Strategic Reports’, and you will not even know of its existence.

The tree-like content hierarchy is unique to each application. It is defined by the system administrator using the Web Layout Editor (see [Creating the Library Hierarchy](#) on page 10-6).

Searching Content Information (Metadata)

By default, a newly installed instance of Content Server is set up for metadata searching (and not [full-text searching](#)). Metadata searching is similar to finding a book in a library by searching for its author, title, or subject. When you search by metadata, you specify as much information as you know about a file or a group of files. For example, if you want to find all files written by your supervisor for your department that were released on or after 1/1/2006, you would specify the following on the search page:

- ❖ Author: *supervisor’s user name*
- ❖ Department: *department name*
- ❖ Release Date From: *1/1/2006*

The query then only finds content that meets all three criteria.



Note: When searching for metadata, case sensitivity and wild card options will vary depending on how your system administrator has configured the content server. Consult with your system administrator for your specific configuration.

Metadata Search Operators

Most content information fields are preceded by a pull-down menu, which enables you to specify the match type (in other words, how exactly the field value should be interpreted in the search query). Depending on the [search solution](#) used, there are a number of match types which may include any of the following:

❖ Substring

This operator finds all content items with the specified string anywhere in the metadata field. This has the same effect as placing a wildcard before and after the search term. This is the most commonly used operator and it is the default operator.

Example: If you type *form* in the Title field, the search will find all files with words such as *forms*, *performance*, and *reform* in their title.

❖ Not Substring

This operator finds all content items that do not have the specified string anywhere in the metadata field.

Example: If you type *form* in the Title field, the search will find all files that do not have words such as *forms*, *performance*, and *reform* in their title.

❖ Matches

This operator finds all content items with the exact specified value in the metadata field.

Example: If you type *Tax Form* in the Title field, the search will find all the files with the exact title of *Tax Form*.

❖ Not Matches

This operator finds all content items that do not have the exact specified value in the metadata field.

Example: If you type *Tax Form* in the Title field, the search will find all the files that do not have *Tax Form* as their exact title.



Note: The Matches and Not Matches search operators are case-sensitive. This means that “Tax Form” as the search query does not find “Tax form” or “tax form.”

❖ **Contains** (*Verity only*)

This operator finds all content items with the specified whole word or phrase in the metadata field. No wildcard is placed before or after the specified value.

Example: If you type *form* in the Title field, the search will find all files with the word *form* in their title, but will not find the words *performance* or *reform*.

❖ **Has Prefix** (*Verity only*)

This operator finds all content items with the specified word or phrase at the beginning of the metadata field. This has the same effect as placing a wildcard after the search term when using the Contains operator.

Example: If you type *form* in the Title field, the search will find all files with titles that begin with the word *form*, including *forms*, *forming*, etc.

❖ **Starts**

This operator finds all content items with the specified value at the beginning of the metadata field. This has the same effect as placing a wildcard after the search term when using the Matches operator.

Example: If you type *form* in the Title field, the search will find all files with titles that begin with the word *form*, including *forms*, *forming*, etc.

❖ **Ends**

This operator finds all content items with the specified value at the end of the metadata field. This has the same effect as placing a wildcard before the search term when using the Matches operator.

Example: If you type *form* in the Title field, the search will find all files with titles that end with the word *form*, including *form*, *perform*, *misinform*, etc.

❖ **Has Word** (*Verity zoned searches only*)

This operator finds all content items with the specified word in the metadata field. No wildcard is placed before or after the specified value.

Example: If you type *form* in the Title field, the search will find all files with the word *form* in their title, but will not find the words *performance* or *reform*.

❖ **Has Word Prefix** (*Verity zoned searches only*)

This operator finds all content items with the specified word in the metadata field. No wildcard is placed before or after the specified value.

Example: If you type *form* in the Title field, the search will find all files with the word *form* in their title, but will not find the words *performance* or *reform*.

❖ **Not Has Word Prefix** (*Verity zoned searches only*)

This operator finds all content items that do not contain the word in the metadata field. No wildcard is placed before or after the specified value.

Example: If you type *form* in the Title field, the search will find all files that do not have the word *form* in their title.

Wildcards

A wildcard substitutes for unknown or unpredictable characters in the search term. You can use wildcards in conjunction with certain [operators for metadata searches](#), and also in the QuickSearch field. The available wildcards and their interpretation vary depending on the [search solution](#) used:

❖ ***** (**asterisk**) (*Verity only*)

This wildcard stands for zero or more alphanumeric characters. For example

- *form** matches *form* and *formula*
- **orm* matches *form* and *reform*
- **form** matches *form*, *formula*, *reform*, and *performance*

❖ **?** (**question mark**) (*Verity only*)

This wildcard stands for one (and only one) alphanumeric character. For example

- *form?* matches *forms* and *forml*, but not *form* or *formal*
- *??form* matches *reform* but not *perform*

❖ **%** (**percentage symbol**) (*database searches only*)

This wildcard stands for zero or more alphanumeric characters. For example

- *form%* matches *form* and *formula*
- *%orm* matches *form* and *reform*
- *%form%* matches *form*, *formula*, *reform*, and *performance*

❖ **_** (**underscore**) (*database search only*)

This wildcard stands for one (and only one) alphanumeric character. For example

- *form_* matches *forms* and *forml*, but not *form* or *formal*
- *__form* matches *reform* but not *perform*

Performing a Full-Text Search

A content server instance can be set up for full-text searching (in addition to [metadata searches](#)). *Full-text* searching enables you to find a content item based on the text contained in the file itself (as opposed to metadata searches, which only search in the information *about* the content item). When a content item is checked into the content server, the indexer stores all of the words in the web-viewable version of the content item (PDF, HTML, text, or other supported file formats) in an index. When you perform a full-text search, the search expression is compared with the index, and any content items and discussions that contain your search text are returned in the search results.

A full-text search expression can include the following elements:

- ❖ **Strings**—partial words (such as *addr*)
- ❖ **Words**—individual whole words (such as *addresses*)
- ❖ **Phrases**—multiple-word phrases (such as *new addresses*)
- ❖ **Operators**—logic applied to words and phrases (such as *news AND addresses*)



Important: Search queries always take your user privileges into account. In other words, the search results will only show those files that you have been granted access to.



Note: Case sensitivity for full-text searches varies depending on how your system administrator has configured the content server. For more information see the *Content Server User Guide*.

Full-Text Search Operators

You can use special search operators to broaden or narrow your full-text search.

The available full-text search operators depend on the [search solution](#) that is used with the content server. Some common search operators include:

- ❖ **AND**

This operator finds all content items that contain all of the specified terms.

Example: If you search for *address AND name*, the query returns all content items that contain both the word *address* and the word *name*.

- ❖ **OR**

This operator finds all content items that contain at least one of the specified terms.

Example: If you search for *safety OR security OR protection*, the query returns all content items that contain at least one of these three words.

❖ **NOT**

This operator finds all content items that contain the term that precedes the operator (if any), and ignores content items that contain the term that follows it.

Example: If you search for *NOT server*, the query returns all content items that do not contain the word *server*. If you search for *internet NOT server*, the query returns all content items that do contain the word *internet*, but not the word *server*.



Note: If you are using Verity as the full-text search engine, you can use localized (i.e., non-English) search operators. However, you need to enclose these in angle brackets, for example: “système <ET><SAUF> gestion”.

Using Repository Manager to Look Up Content

Repository Manager is one of the tools accessed from the Administration page. It is primarily a diagnostic and maintenance tool, and should not be used for routinely checking in and out content. You should use the normal check-in and check-out features of Content Server for that.

Repository Manager provides a number of powerful features for administrators to look up content. You can start Repository Manager from within the Content Server software or as a stand-alone application. Administrators and sub-administrators with Rights.Apps.RepMan rights can use Repository Manager to see what content is currently in the system and check the revision and conversion status. They can use filters to narrow down the view. For example, they can filter the content to only see the content of a particular type or author, or only content that was checked during the last week.

By default, the content items are listed in ascending sort order by content ID (0-9, a-z). If you click on any of the other column headings, the content item list is resorted by that column in ascending order. If you click the same column heading again, the sort order is reversed to descending.



Important: The Repository Manager view will always depend on your user privileges. In other words, you will always only be able to look up and view files that you have been granted access to.



Note: A maximum of 200 content items are displayed. If more than 200 items meet the selection criteria, not all of them will be listed.

Within Repository Manager, you can perform a number of administrative tasks on specific content items (using the buttons of the Functions drop-down menu):

- ❖ Check in new content
- ❖ Add a new revision to an existing content item
- ❖ Delete a specific revision of a content item
- ❖ Delete all revisions of a content item (effectively deleting it entirely from the system).
- ❖ View the content information of a content item
- ❖ Update the content information of a content item
- ❖ View subscription information of a content item
- ❖ Check out a content item
- ❖ Undo the check-out of a content item
- ❖ Approve a content item in a workflow
- ❖ Reject a content item in a workflow
- ❖ Resubmit a content item to Inbound Refinery



Note: You can only check in new content and revisions if Repository Manager is started as a stand-alone application. This functionality is not available if Repository Manager is started from the Administration page in Stellent Content Server.

ADMINISTERING CONTENT SERVER

OVERVIEW

Content Server provides a variety of features that help administrators and sub-administrators set up and manage a Content Server environment.

This section focuses on the following topics:

- ❖ [Administration Tools](#) (page 9-2)
- ❖ [Reviewing the System Settings](#) (page 9-3)
- ❖ [Admin Server](#) (page 9-4)
- ❖ [System Properties](#) (page 9-5)
- ❖ [User Admin](#) (page 9-6)
- ❖ [Batchloader](#) (page 9-6)
- ❖ [Repository Manager](#) (page 9-8)
- ❖ [Archiver](#) (page 9-9)
- ❖ [Content Server Analyzer](#) (page 9-13)
- ❖ [Environment Packager](#) (page 9-14)
- ❖ [Log Files](#) (page 9-14)
- ❖ [Backup Strategy](#) (page 9-15)

ADMINISTRATION TOOLS

The system administrator uses a number of administration tools to configure and manage the Content Server environment. These are the main administration and diagnostics tools:

❖ **Web Layout Editor**

This administration tool is used to customize the portal page and create the library hierarchy. For details refer to [Customizing the Portal Page](#) (page 10-5) and [Creating the Library Hierarchy](#) (page 10-6).

❖ **User Admin**

This administration tool is used to set up and manage the user base, security, and aliases. For details refer to [User Admin](#) (page 9-6).

❖ **Repository Manager**

This administration tool is used to perform a number of file management and indexing functions, and to define and manage subscriptions. For details refer to [Repository Manager](#) (page 9-8).

❖ **Archiver**

This administration tool is used to transfer, back up, and reorganize information within a Content Server server instance or between server instances. For details refer to [Archiver](#) (page 9-9).

❖ **Workflow Admin**

This administration tool is used to define and manage [workflows](#). For details on workflows refer to chapter 7.

❖ **Configuration Manager**

This administration tool is used to define and manage content types, file formats, and information (metadata) fields. For details refer to [Grouping of Content](#) (page 8-2) and [Content Information \(Metadata\)](#) (page 8-3).

❖ **Admin Server**

This administration tool is used to start, stop, and restart a content server. In addition, it enables you to review or edit a number of system settings. For details refer to [Admin Server](#) (page 9-4).

❖ **Batchloader**

This administration tool is used to check in large volumes of content into Content Server. For details refer to [Batchloader](#) (page 9-6).

❖ **System Properties**

This administration tool is used to set a number of system properties. For details refer to [System Properties](#) (page 9-5).

❖ **Content Server Analyzer**

This administration tool is used to confirm the integrity of the content server repository components, including the file system, database, and search index. For details refer to [Content Server Analyzer](#) (page 9-13).

❖ **Environment Packager**

This administration tool is used to “package” state directories, log files, and other component and resource directories for diagnostic and support purposes. For details refer to [Environment Packager](#) (page 9-14).

All of these tools (except Environment Packager) can be launched as stand-alone applications at the server, and all but Batchloader, System Properties, and Content Server Analyzer can also be started from within Stellent Content Server (from the Administration page).



Note: You need administrator privileges to access the administration tools above (with the exception of System Properties).

REVIEWING THE SYSTEM SETTINGS

The main system settings can easily be reviewed by clicking the ‘Configuration for *[Instance Name]*’ link on the Administration page of the content server instance. This opens a page which provides an overview of the main system settings, including server parameters and options, installation directories, Internet properties, database properties, version information, Java properties, and content security options. In addition, it lists all installed server components or custom components that are currently enabled and disabled



Note: The overview page is for information purposes only—you cannot change anything on it. If you want to modify any of the settings, you need to do that elsewhere (see [Admin Server](#) on page 9-4 and [System Properties](#) on page 9-5).

ADMIN SERVER

The Admin Server administration tool is used for the following main tasks:

- ❖ [Starting, stopping, and restarting the content server](#) (see Chapter 6)
- ❖ [Enabling and disabling custom components](#) (see [Using Custom Components](#) (page 10-7))
- ❖ [Setting a number of general configuration options](#) (see below)
- ❖ [Viewing server output and server logs](#) (see below)



Caution: In addition to the options mentioned above, Admin Server also provides an Edit Server link in the left navigation menu. This link enables you to edit some configuration settings which are critical to proper system operation. Use the utmost care when editing these system-critical settings. If you set them to incorrect values, the Content Server system may shut down entirely.



Note: You must be logged in as the system administrator or a user with the sysmanager role to access the Admin Server.



Note: If you make any changes, you may need to restart the content server. If that is the case, a message is displayed notifying you of this.



Note: For further details on Admin Server, refer to the *Managing System Settings and Processes* guide or Content Server's online help.

Setting General Configuration Options

The Admin Server administration tool also enables you to view the output logs created by the content server:

1. Make sure you are logged into Content Server as an administrator.
2. On the Administration page in Content Server, click the Admin Server link.
3. Click the button of the appropriate content server to the right of the start-stop-restart symbols.
4. The left navigation menu now provides a number of links which enables you to change system configuration settings.

You can change settings in the following main categories:

❖ **General configuration**

Here you can enable and disable a number of optional Content Server features (for example, multiple-file download and upload, automatic generation of content IDs, etc.). In addition, you can define the major and minor revision label sequences, and specify any additional global configuration variables.

❖ **Content security**

Here you can set a number of content security settings that overrule the default settings. For example, you may specify that authors are allowed to delete their own revisions without delete privilege. (By default, users need to have delete privilege to be able to delete anything.)

❖ **Internet configuration**

Here you can specify the mail server address and administrator e-mail address. It also shows some other Internet-related configuration settings (if you want to change these, you need to use System Properties, see below).

Viewing Server Output and Server Logs

The Admin Server administration tool enables you to view the server output and server logs:

1. Make sure you are logged into Content Server as an administrator.
2. On the Administration page in Content Server, click the Admin Server link.
3. Click the button of the appropriate content server to the right of the start-stop-restart symbols.
4. Click View Server Output or View Server Logs on the left navigation menu.



Note: For further details on log files, refer to [Log Files](#) (page 9-14).

SYSTEM PROPERTIES

System Properties is a stand-alone application (not a Java applet) which must be run locally from the server:

❖ **Microsoft Windows:** Start—Programs—Stellent Content Server—<Instance Name>—Utilities—System Properties.

❖ **UNIX:** Run the System_Properties script, which is located in the *bin* subdirectory of the Content Server installation directory.

You can use System Properties to review and edit the following application settings:

- ❖ Check-in and search options (for example, automatic content ID allocation)
- ❖ Content security options (for example, delete and check-out permissions)
- ❖ Internet configuration settings (see important caution below!)
- ❖ Java Database Connectivity (JDBC) settings
- ❖ Server configuration settings (for example, system locale and time zone)
- ❖ Localization settings (enabling, disabling, or editing locales)
- ❖ Paths to some important files and directories



Caution: Some of the settings that you can change using System Properties are critical to proper system operation (most notably the options on the Internet tab). Use the utmost care when editing these system-critical settings. If you set them to incorrect values, the Content Server system may shut down entirely.



Note: Most of the options in System Properties can also be set using Admin Server.



Note: If you make any changes, you need to restart the content server before they take effect.

USER ADMIN

The User Admin administration tool is used to manage the user base, set up security (by assigning roles and permissions to users), define aliases, and manage security groups.

For details on user management and security refer to [Chapter 5 \(Content Server Security\)](#).



Note: For further details on User Admin, refer to the *Managing Security and User Access* guide or Content Server's online help.

BATCHLOADER

Batchloader is a stand-alone application (not a Java applet) which must be run locally from the server:

- ❖ **Microsoft Windows:** Start—Programs—Stellent Content Server—*<Instance Name>*—Utilities—Batchloader.

- ❖ **UNIX:** Run the Batchloader script, which is located in the *bin* subdirectory of the Content Server installation directory.

Batchloader is a utility that lets you automatically check in (insert), delete, or update a large number of content items into Content Server at one time. This can save a lot of time and effort.

Batchloader is useful in a variety of situations including:

- ❖ You just purchased Content Server and you want to automatically check in all existing content with metadata fields that already exist in a database.
- ❖ You want to migrate to Content Server from another document management system and you need an interface that will enable you to “import” all existing data into Content Server.
- ❖ You want to build a system that automatically checks content into Content Server. External applications can interface with Batchloader to manipulate content (for example, check new content in).



Caution: Improper use of the Batchloader utility may cause loss of content and metadata. Make sure that you read the documentation and have a backup available before using Batchloader.



Note: You need to provide a valid administrator user name and password before you can use Batchloader.

Batchloader executes and performs actions specified in a batch load file. This file contains metadata fields and content records which describe the actions to be performed. The entries in a batch load file consist of name/value pairs that specify the metadata fields for each content item as well as the action to be performed. Valid actions are ‘insert,’ ‘update,’ and ‘delete’. Each content record must end with the <<EOD>> (end-of-data) marker. A pound sign (#) indicates a comment, which is ignored when Batchloader processes the file.

A simple batch load file could look something like this:

```
#This is comment...
Action=insert
dDocName=Sample1
dDocType=Report
dDocTitle=Title of first document to be checked in
dDocAuthor=sysadmin
dSecurityGroup=Public
primaryFile=sample1.doc
dInDate=5/14/04
<<EOD>>
```

The Content Server software is provided with a number of sample batch load files. They are located in the `samples/batchloader` subdirectory of the Content Server installation directory.



Note: Any errors in the batchloading process are logged in the server log.



Tech Tip: In addition to using Batchloader's application interface, you can also run it from the command line. If you run Batchloader from the command line, no message boxes are displayed. The syntax to run Batchloader from the command line is as follows:

```
BatchLoader /q /n[filepath]
```

where the parameter `/q` prevents the Batchload interface from displaying (required), and `/n[filepath]` is the path to the batch load file (not required). If you do not specify the `/n` parameter, Batchloader will look in the Batchloader path as specified in the `intradoc.cfg` configuration file (located in Content Server's `bin` subdirectory).



Note: For further details on Batchloader, refer to the *Managing System Settings and Processes* guide or Content Server's online help.

REPOSITORY MANAGER

Repository Manager has three main functions:

- ❖ View the status of any content items in Content Server, and perform a variety of administrative tasks on specific files
- ❖ Define and manage subscriptions
- ❖ Update or rebuild the search index.

Viewing the Status of Content

You can use Repository Manager to look up content items, view their status, and perform a number of administrative tasks on the content items. For details refer to [Using Repository Manager to Look Up Content](#) (page 8-22).

Defining and Managing Subscriptions

You can use Repository Manager to set up and manage subscriptions to content items. For details on subscriptions refer to [Content Subscriptions](#) (page 8-11).

Updating and Rebuilding the Search Index

The indexing features of Repository Manager allow you to update and rebuild the search index. (For details on the search index refer to [Search Collection](#) on page 3-6).



Note: For further details on Repository Manager, refer to the *Managing Repository Content* guide or Content Server's online help.

ARCHIVER

Archiver is a Java applet that is used to transfer and reorganize content server files and information. Archiver has four main functions:

- ❖ **Export**—Use the export function to copy native and web-viewable files out of the content server instance for backup, storage, or import to another content server instance. You can also export content types and user attributes. You export to an archive, which contains the exported files and their metadata in the form of batch files. (See [Exporting Content](#) on page 9-10.)
- ❖ **Import**—Use the import function to retrieve files and content server information from an exported archive. Importing is typically used to get a copy of content from another content server or to restore data that has been in storage. You can also change metadata values during an import. (See [Importing Content](#) on page 9-10.)
- ❖ **Transfer**—Use the transfer function to transfer content from one content server instance to another over sockets. This function is typically used to move or copy content across a firewall or between two content server systems that do not have access to the same file system. (You can also use the Transfer function to transfer archive files between content server systems that have access to the same file system.) (See [Transferring Content](#) on page 9-11.)
- ❖ **Replicate**—Use the replicate function to automate the export, import, and transfer functions. For example, you can use replication to automatically export from one content server instance, transfer the archive to another computer, and import to another content server instance. (See [Replicating Content](#) on page 9-12.)



Caution: Do not use Archiver as your primary method of disaster recovery; use standard backup systems for the database and file system.



Note: For further details on Archiver, refer to the *System Migration Guide* or Content Server's online help.

Exporting Content

Archiver's export function is used to copy native and web-viewable files out of the content server instance for backup, storage, or import to another content server instance. You can also export content types and user attributes. When performing an export, Archiver copies or moves content and its associated metadata from a Oracle content server to a storage area which is a directory structure typically located under the *archives* subdirectory of the Content Server installation directory. This storage areas contains definition files, lists of batch files, configuration files, other optional files, and one or more subdirectories. Files are stored in the same structure as in Content Server (with 'vault' and 'weblayout' subdirectories).

Typical uses of Archiver's export function include:

- ❖ Copying files from an intranet to make them available to an extranet for vendor or customer viewing.
- ❖ Creating an archive of content items that will then be imported back to the same instance with different metadata.
- ❖ Removing content from the content server for permanent or temporary storage. For example, if space becomes limited or performance drops, you could remove all but the latest revision of each file.
- ❖ Copying files, content types, and user attributes from a development content server instance for use in a production instance.

Importing Content

Archiver's import function is used to retrieve files and content server information from an exported archive. Importing is typically used to get a copy of content from another content server or to restore data that has been in storage.

Import rules define how revisions are added, replaced, or deleted during import. During import, Archiver compares each revision being imported with the existing revisions in the importing content server. The import rule specifies which action to take (add, replace, delete, or ignore), depending on comparison of a number of metadata fields.

Typical uses for Archiver's import function include:

- ❖ Placing data archived from an intranet on an extranet for vendor or customer viewing.

- ❖ Changing metadata for a large number of content items. For example, if an employee leaves the organization, you could export all of their content items and then import them with another user specified as the author.
- ❖ Restoring content that was inadvertently deleted or configuration information that was inadvertently changed.
- ❖ Copying files, content types, and user attributes from a development content server archive to a production instance.

Transferring Content

Archiver's transfer function is used to move or copy content from one content server to another over sockets. The transfer function can be used to transfer files between content servers on a shared file system, but transfers do not require a shared file system. Transferring files between non-shared file systems requires an outgoing provider on the source content server instance.

Typical uses for Archiver's transfer function include:

- ❖ Exporting and importing over a firewall.



Note: To transfer through a firewall, you may need to configure the firewall to permit the outgoing provider's socket to pass through it.

- ❖ Transferring content between content server instances in different physical locations (buildings, cities, or countries).
- ❖ Transferring content between content server instances using a shared drive. (A transfer over a file system share can handle large archives better than a socket transfer.)
- ❖ Avoiding the need to build an FTP or HTTP interface to move files from one file system to another.
- ❖ Combining the batch files from two archives into a single archive.

There are three types of transfers, listed in order from simplest to most complex:

- ❖ [Local Transfer](#) (page 9-12)
- ❖ [Pull Transfer](#) (page 9-12)
- ❖ [Push Transfer](#) (page 9-12)

Local Transfer

A *local transfer* is a transfer between local archives, which belong to collections that both the source and target content servers can reach through a mapped or a mounted drive. An outgoing provider is not required. This type of transfer is typically used to combine the batch files of two archives.

Pull Transfer

A *pull transfer* is a transfer that is owned by the proxied (remote) content server, which is the instance that is the target of the outgoing provider. If you are running a pull transfer across a firewall, you might need to configure the firewall to permit the outgoing provider's socket to pass through it.

Push Transfer

A *push transfer* is a transfer that is owned by the local content server, which is the instance on which the outgoing provider is set up. If you are running a push transfer across a firewall, you might need to configure the firewall to permit the both providers' sockets to pass through it.

For performance monitoring of a push transfer, you also should set up an outgoing provider from the target (proxied) content server back to the source (local) content server. This "talkback" provider can then notify the source content server when each transfer is complete. A push transfer will work without the talkback provider, but the source content server would not be aware of transfer completion or problems.

Replicating Content

Archiver's replication function is used to automate the export, import, and transfer functions. To set up replication, the exporting instance ("exporter") must create an archive and allow the importing instance ("importer") to access the archive. Replication requires a shared directory between the importing and exporting instances.

With replication, an export is performed each time a file that meets the export criteria is indexed. Indexing takes place when you check in new files or every five minutes. Import replication occurs automatically, about once a minute. Import rules govern automated imports (replication).



Note: For performance reasons, replication is not recommended for large archives (in the order of 20,000 files or more). Export and import of large archives should be run manually, during periods of non-peak usage if possible.

Typical uses for Archiver's replication function include:

- ❖ Automatically exporting from one content server instance and importing to another content server instance to synchronize two web sites.
- ❖ Copying content automatically between two contribution/consumption servers.
- ❖ Automatically moving certain documents from a contribution server to a higher-security content server.
- ❖ Automatically moving old content to a storage location.

CONTENT SERVER ANALYZER

Content Server Analyzer is a stand-alone application which must be run locally from the server:

- ❖ **Microsoft Windows:** Start—Programs—Stellent Content Server—*<Instance Name>*—Utilities—Content Server Analyzer.
- ❖ **UNIX:** Run the `Content_Server_Analyzer` script, which is located in the `bin` subdirectory of the Content Server installation directory.

Content Server Analyzer is a utility that enables you to confirm the integrity of the content server repository components, including the file system, database, and search index. It can also assist system administrators in repairing some problems that occur and are detected in the repository components. The Content Server Analyzer utility enables system administrators to do the following:

- ❖ Confirm the accuracy of synchronization between three important content server database tables: Revisions, Documents, and DocMeta.
- ❖ Confirm that the `dRevClassID` and `dDocName` fields are consistent across all revisions of content items.
- ❖ Determine if the file system (*vault* and *weblayout* file repositories) contains duplicate or missing files.
- ❖ Ensure the accuracy of synchronization between the search index and the file system.
- ❖ Ensure the accuracy of synchronization between the search index and the Revisions database table.
- ❖ Ensure that the file system contains all necessary files.
- ❖ Remove duplicate files from the content server repository either permanently or provisionally by moving them into the log directory (*<Instance_Name>/logs*).

- ❖ Produce a general report on the state of content items in the content server.



Note: For further details on using Content Server Analyzer, refer to the *Troubleshooting Guide* or Content Server’s online help.

ENVIRONMENT PACKAGER

Environment Packager is a diagnostic tool that is primarily used for support purposes. It enables you to “package up” the content server Java and operating system environment—that is, it creates a zip file with the desired state directories, log files, and other component and resource directories. Environment Packager is started from the Administration page in Content Server.

LOG FILES

The system stores status information and errors in log files, which can be accessed from the Administration page in Content Server and using the Admin Server tool. The information contained in the log files can be used to monitor system performance or troubleshoot errors. All log files are sorted by date in descending order (which means the most recent item is at the top). They are created only once each day at the time the first status event, error, or fatal error occurs. (Please note that no empty log files are created.)

There will never be more than 60 log files. If new ones are needed, the oldest log files are deleted. This means that log files do not endlessly consume disk space.



Tech Tip: Bookmark your log file pages. This will help you troubleshoot problems, even if Content Server is unavailable. Also, know where your configuration files are, so you can find them if Content Server is unavailable.

Log File Types

There are three types of log files:

- ❖ **Server logs**

These log files contain status events and errors generated by the Stellent Content Server software.

- ❖ **Archiver logs**

These log files contain all status events and errors generated by the archiving process (see [Archiver](#) on page 9-9).

❖ Inbound Refinery logs

These log files are only generated if the Content Server Inbound Refinery add-on is installed and running. They contain status events and errors generated by Inbound Refinery.

Log File Columns

Each log file contains three columns:

❖ Type

This column specifies the kind of incident that prompted the log entry: info, error, or fatal (see below).

❖ Time

This column lists the date and time the log entry occurred.

❖ Description

This column describes the incident that occurred.

Log File Entries

The following types of log entries are generated (specified in the first column):

❖ Info

Displays basic status information. For example, status information is logged when the content server is started.

❖ Error

Displays user/administration errors that occur but do not stop the software from functioning. For example, an error is logged if users request secure information they are not allowed to access.

❖ Fatal

Displays errors that stop the software from functioning. For example, a fatal error is logged if the software cannot access the database on startup.

BACKUP STRATEGY

This section covers the following topics:

- ❖ [Using Backups](#) (page 9-16)
- ❖ [Backup/Recovery Methods](#) (page 9-16)

- ❖ [Disaster Recovery](#) (page 9-18)

Using Backups

It is important that you back up your files on a regular basis to make disaster recovery as efficient as possible. Use standard backup principles when setting up a back-up strategy, taking the following considerations into account:

- ❖ There are basically four storage areas in Content Server (see also [Basic Architecture](#) on page 3-3):
 - the native file repository (“Vault”)
 - the web-viewable file repository (“Web Layout”)
 - the content information database
 - the search index

Of these four storage areas, the two file repositories and the content information database are the most important—a Oracle application can never be successfully recovered without them. Without the actual files, there is no content. Without the content information database, Content Server knows nothing about the content, which effectively means it does not exist.

- ❖ The backed-up versions of the file repositories and the content information database should be synchronized as closely as possible. Ideally, the content server and database should be shut down before making a backup. This achieves maximum synchronization, since no content can be checked in or out while the backup is being made. If this is not an option, make sure that the backup is made during periods of low system activity (for example, at nights or on weekends).
- ❖ It is recommended that you include the entire Content Server directory structure in your backup strategy. This ensures that you include all configuration files and other important application-specific files (for example, custom components).

Backup/Recovery Methods

Content Server is created to utilize both an RDBMS (Oracle, SQL Server, etc.) and an operating system’s file structure. With this in mind, the options vary with each organization’s implementation model. When encountering a disaster recovery, the first question that should be asked is what is an acceptable risk that an organization can handle in the event a data loss occurs and what will be your recovery protocol. Here are two options to consider:

- ❖ Nightly backups
- ❖ Incremental backups

With this understanding, there are backup/recovery methods that can be used as guidelines for customer strategies.

Incremental Archive

Content Server's Archiver utility can perform incremental updates. These updates are triggered anytime an object is checked in or modified in the content server. You can keep an archive on a separate server in another location. Recovery is performed by importing the archive into either the same server or another server. By importing the archive, Content Server returns to the state of the last object modification. This is a method suggested for organizations that have an active contributor base.

Manual Archive

Content Server's Archiver utility can perform "on-demand updates." Recovery is performed by importing the archive into either the same server or another instance. This method is suggested for organization that have a small contribution community. Optionally, a component can be built to allow a third-party scheduling application to initiated the Archiver server on an automated bases.

Replication

You can set Content Server's Archiver utility to perform to replicate the content in the content server. Replication can be implemented using either the Incremental or Manual Archive functionality. These replications can be used to create a mirrored instance of a content server. If the incremental option is used, an archive can be automatically transferred and imported to the mirrored server on any object modification on the main content server. This method is suggested for organizations that want to implement the content server in an automatic fail-over environment.

Full Database/File System Restore

Oracle's open architecture leverages industry-standard database and file systems. Therefore, Content Server can take advantage of the backup/restore capabilities of each. Most databases have the capability to do full back-up and incremental logging. Combined with a built-in or third-party file backup application, this solution can provide an organization the option to completely restore the state of a database and file system. This

option should be used in conjunction with one of the archive or replication solutions as a secondary disaster recovery plan.



Tech Tip: It is important to back up your files on a regular basis to make disaster recovery as efficient as possible. We recommend that you back up the following:

- All changes to the content server’s directory structure on a daily basis
- All files in the content server’s directory structure on a monthly basis
- The software after you install it and before you update it with a new version.

Disaster Recovery

Disasters are unusual; however, in the event of a total machine failure, it is best to have strong disaster recovery procedures in place. Following a total machine failure, perform the following tasks:

1. Reinstall all required third-party software.
2. Configure the platform to be the same as it was when the last backup was made.
3. Reinstall the database software and the Content Server software.
4. Copy the backed up Content Server software directory structure over the fresh installation.
5. If the native file repository (“Vault”) and web-viewable file repository (“Web Layout”) are not on the same computer as the content server, copy them to their respective original locations.
6. Copy the backed-up database over the empty database that was created when the software was reinstalled.
7. Restart the software.
8. Run the [Content Server Analyzer](#) to confirm the integrity of the content server repository components and clear up any discrepancies between the file system and the database (see page 9-13).

Chapter 10

CUSTOMIZATION AND PERSONALIZATION

OVERVIEW

Content Server is highly customizable. The out-of-the-box look-and-feel and functionality can be modified to suit virtually any application. The customizations can be strictly cosmetic (for example, different graphics on the Content Server portal page) or dramatic, with a new look-and-feel and new functionality added to the out-of-the-box product. This enables companies to adapt the Content Server software to their specific needs.

Content Server also supports personalization. This means that the presentation of program features and content depends on the identity and privileges of individual users.

Customization and personalization can be done in a number of areas including:

- ❖ [Using Interface Layouts and Skins](#) (page 10-2)
- ❖ [Providing Localized User Interfaces](#) (page 10-4)
- ❖ [Customizing the Portal Page](#) (page 10-5)
- ❖ [Personalizing the Interface](#) (page 10-6)
- ❖ [Creating the Library Hierarchy](#) (page 10-6)
- ❖ [Using Custom Components](#) (page 10-7)

USING INTERFACE LAYOUTS AND SKINS

Content Server is provided with a number of standard interface layouts and skins. Users can set their own interface layout and skin in their User Profile page.

Layouts

Layouts define the general interface and navigation structure of the user interface.

Three standard layouts are provided with the Content Server software:

- ❖ Trays (default)
- ❖ Top Menus
- ❖ Classic

Skins

Skins encompass the “look” or color scheme of the layout. A number of standard skins are provided with the Content Server software, including:

- ❖ Oracle (default)
- ❖ Oracle1
- ❖ Classic

Figure 10-1 and Figure 10-2 below show two examples of layout and skin combinations.



Note: Custom skins and/or layouts can be designed or existing skins can be modified. For further details about this, refer to the *Modifying the Content Server Interface* guide.

Figure 10-1 Trays layout with a sample skin

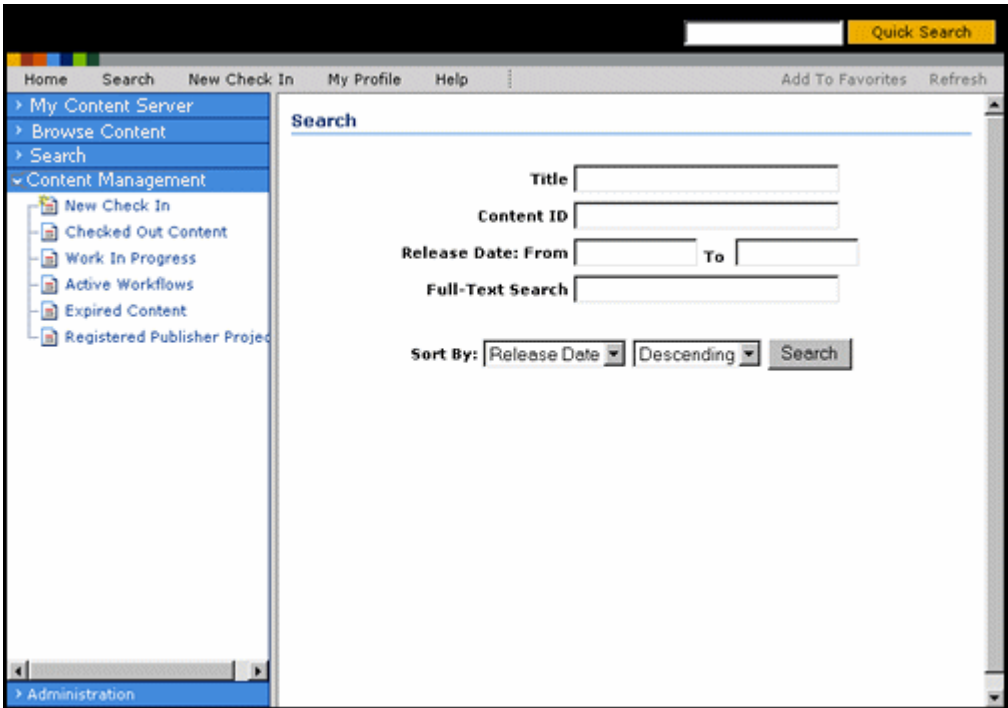
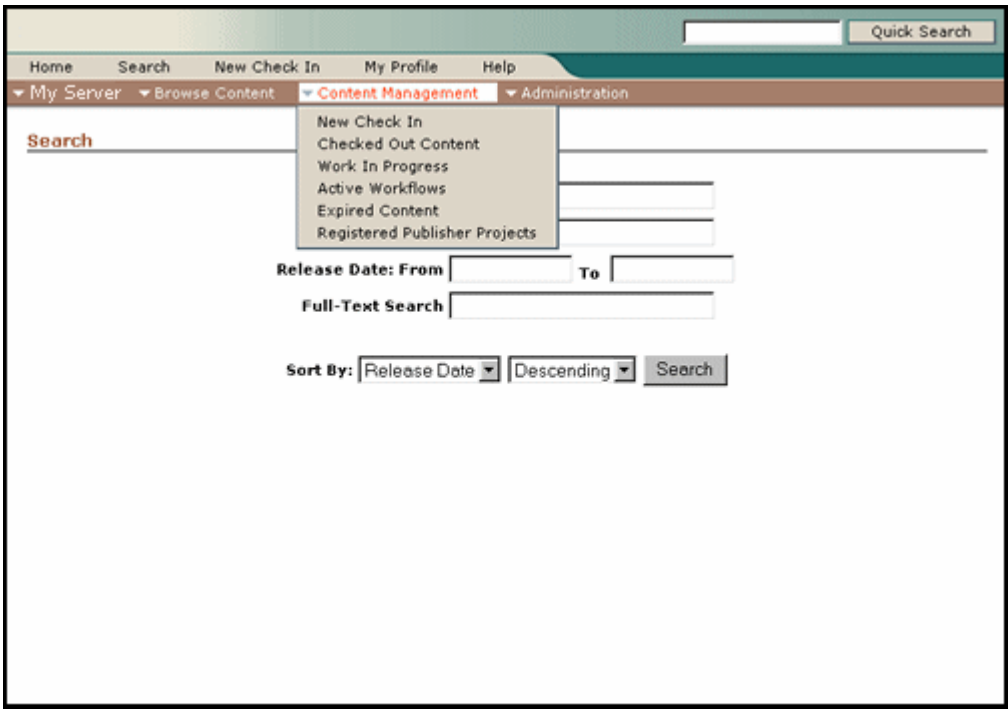


Figure 10-2 Top Menus layout with a sample skin



PROVIDING LOCALIZED USER INTERFACES

System administrators can make localized interfaces available to non-English users of the Content Server system. This is accomplished through optional language packs, which are installed along with the Content Server software and can also be downloaded separately from the support site once they are available. After the files in a language pack have been installed (either as part of the Content Server software setup or separately), you need to explicitly enable the associated locale before it is available to end-users. This is done using the [System Properties](#) utility (see page 9-5), which is executed locally from the server and cannot be started from within Content Server.

Users can choose their preferred locale from all enabled locales in their user profile, which is accessible from the content server's portal page. The Content Server user interface will then be presented in the selected language, with date/time format and alphabetization following the conventions of that language.

As an example, Figure 10-1 shows the Japanese Content Server user interface. Please note that the date format is in accordance with the Japanese convention (yyyy/mm/dd).

Figure 10-3 Japanese user interface (with Trays layout and a sample skin)

The screenshot displays the Japanese user interface for the Content Server. The main window is titled 'コンテンツ チェックイン フォーム' (Content Check-in Form). The interface features a navigation menu on the left with options like 'ホーム', '検索', '新規チェックイン', 'プロフィール', and 'ヘルプ'. The main form area contains several input fields and dropdown menus:

- タイプ** (Type): A dropdown menu with the selected value 'ADACCT - Acme Accounting Department'.
- タイトル** (Title): An empty text input field.
- 作成者** (Author): A text input field containing 'sysadmin' and a dropdown menu also showing 'sysadmin'.
- セキュリティグループ** (Security Group): A dropdown menu with the selected value 'Public'.
- プライマリファイル** (Primary File): An empty text input field with a '参照...' (Reference...) button.
- 代用ファイル** (Replacement File): An empty text input field with a '参照...' (Reference...) button.
- コンテンツ ID** (Content ID): An empty text input field.
- 改訂** (Revision): A text input field containing the number '1'.
- Comments**: A large text area for entering comments.
- リリースの日付** (Release Date): A date and time field showing '2004/04/07 23:37'.
- 有効期限** (Expiration): An empty text input field.

At the bottom of the form, there are three buttons: 'チェックイン' (Check-in), 'リセット' (Reset), and 'クイック ヘルプ' (Quick Help). The top right corner of the window has a search bar labeled 'クイック検索' and a 'リフレッシュ' (Refresh) button.



Note: For further details on locales, refer to the *Using Content Server in International Environments* guide.

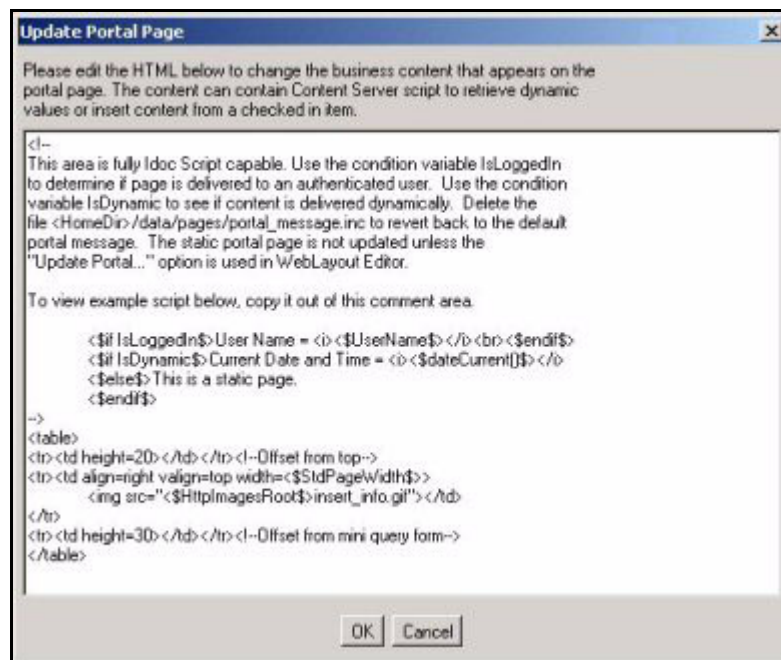
CUSTOMIZING THE PORTAL PAGE

The main page of the Content Server environment is called the “portal page.” This is the page that users see when they log on to the system. Custom features and content can be added to the portal page. This is done using the Web Layout Editor, which is one of the tools accessed from the Administration page. If you choose Options—Update Portal in the Web Layout Editor, a window is presented in which you can use HTML and Idoc Script code to modify the look-and-feel and functionality of the portal page (see Figure 10-1 below). For example, you can change the graphics or display system variables (for example, the date and time).



Note: You need Content Server administrator privileges to access the Web Layout Editor.

Figure 10-1 Customizing the Content Server portal page



Tech Tip: The portal page—and all other user interface pages—can also be customized using [custom components](#) (see page 10-7).

PERSONALIZING THE INTERFACE

Individual users can also personalize their specific user interface. For example, they can add shortcuts to frequently used Content Server functions and search queries. Users can personalize their navigation menu by clicking the Portal Design link in their User Profile page. This opens the Portal Design page, where users can set a number of navigation menu parameters.



Note: Not all personalization features are relevant to all interface layouts, so some options may not be available to users (depending on their current layout settings).



Note: The available personalization options are also determined by the user's authorization level. For example, all administrative options are only available to users with the proper administrator privileges.

CREATING THE LIBRARY HIERARCHY

In addition to customizing the portal page, the Web Layout Editor is also used to define and build the library hierarchy, which consists of the tree-like structure that users can step through (“drill down”) to find the information they are looking for. The library hierarchy is unique to each application and is defined locally by the system administrator.



Note: You need administrator privileges to access the Web Layout Editor.

By clicking the Browse Content link in the navigation menu, users start at the highest level of the hierarchy. From there, they go down level by level until they have found the information they were looking for. Generally, the deeper in the hierarchy, the more specific the information gets. For example, the highest level might refer to company divisions, and the levels below that to main locations, departments, document types, and years, until you reach the document that you need.



Important: Your user privileges are always taken into account when you move through the library hierarchy. This means that you will only see the links to information that you have been granted access to.

There are four different link types in the library hierarchy:

❖ **Local pages (directory)**

A directory is a local page link to another web page of links to local pages, URLs, or

queries. The main page of the library hierarchy (called `index.htm`) is a directory local page.

❖ **Local pages (reports)**

A report is a local page link that links to either an active or historical report. Reports can be used to list system and status information (for example, a list of all defined users or all items currently in a workflow).

Active reports perform a renewed query on the database every time they are opened, which means they always reflect the most current information.

Historical reports, on the other hand, contain information that was queried at the time they were created. They do not perform a renewed database query every time they are opened. An historical report provides a “snapshot” of the system at a certain moment in time. By regularly creating historical reports of the same system view, you can easily monitor what has changed between the moments the historical reports were created.

❖ **External URLs**

An external URL is a page link to a specified web address. You can link to any URL or web page on the Internet, an intranet, or extranet (for example, <http://www.oracle.com>).

❖ **Queries**

A query link jumps to a search result page which contains a hyperlinked list of content that meets the criteria of the query. The criteria are defined by one or more metadata fields. When defining a query link, you can specify the query expression that should be used to filter the content (for example, all content of a particular type and published after a particular date). Every time a user clicks on a query link, the query is performed and the results are displayed. This means that they always provide up-to-date information.

USING CUSTOM COMPONENTS

Content Server components are small modules of program code, resources, and templates that modify or enhance the functionality of Content Server. They are a tremendously powerful tool to customize Content Server to suit specific application needs.



Note: You need administrator privileges to create and work with components.

Creating Components

Component Wizard is a tool that helps developers create custom components and prepare them for use with Content Server. It is a stand-alone utility, which is executed locally from the server and cannot be started from within Content Server. The Component Wizard assists in creating or updating component definition files, component HDA files, and resource files.

Before a component can be used, it must first be made available to the Content Server system. The first step in this process is “building” the component, which means that all files associated with the component are packaged into a compressed zip file. This zip file can then be distributed for use with one or more content servers.

Enabling Components

After a component has been “built,” it needs to be loaded into Content Server and enabled. This is done in either of two ways:

- ❖ Using Component Manager, which is part of Admin Server (one the administration tools).
- ❖ Using Component Wizard, which is a stand-alone utility and cannot be started from within Content Server.

If a new component has been enabled, the content server needs to be restarted before the new functionality is available.

Software Developer’s Kit (SDK)

The Software Developer’s Kit (SDK) is a set of documents which provide information on how to modify the functionality of the out-of-the-box Content Server software.

Idoc Script

Idoc Script is a powerful, proprietary server-side scripting language which is used to modify the functionality and look-and-feel of Content Server. It is used extensively in custom components. It provides the ability to reference variables, conditionally include content in HTML pages, and loop over results returned from queries. In addition, Idoc Script provides a way to process page elements after the browser has made a request, but before the request page is returned.

GLOSSARY

absolute path

The path to a file location that includes the drive (if required), starting or root directory, all attached subdirectories, and ends with the file or object name. For example, *c:/stellent/weblayout/images/logo.gif*. See also [relative path](#) (page 1-17).

absolute URL

The complete URL (web address) of a web page, including the transfer protocol, network location, and optional path and file name. For example, *http://localhost/intradoc-cgi/idc_cgi_isapi.dll*. See also [relative URL](#) (page 1-17).

account

A Oracle security element that enables greater flexibility and granularity in a security structure than security groups provide. Content can be assigned to a particular account upon checkin, and users can access the content only if they have the appropriate permission to that account. See also [security group](#) (page 1-20), [documents without accounts](#) (page 1-7), [#none](#) (page 1-13), and [#all](#) (page 1-3).

active report

A list that contains the results of a database query that is dynamically generated each time the report page is accessed. This type of report always reflects current content server information. Active reports are built with the Web Layout Editor and accessed through the Library. See also [historical report](#) (page 1-9).

administration application

One of the following Java applications that are used for administration purposes: [User Admin](#) (page 1-23), [Workflow Admin](#) (page 1-26), [Web Layout Editor](#) (page 1-25),

[Repository Manager](#) (page 1-18), [Configuration Manager](#) (page 1-4), [Archiver](#) (page 1-3)
These applications can be run as a Java applet from a Java-enabled browser, or in stand-alone mode from the content server computer. See also [administration utility](#) (page 1-2).

administration rights

Permission to use to [administration application](#) (page 1-1). A user must also have [Admin permission](#) (page 1-2) to at least one security group to be able to use the administration applications for which they have rights.

administration utility

One of the following Java applications that are used for administration purposes:
[Component Wizard](#) (page 1-4), [Batch Loader](#) (page 1-4), [System Properties](#) (page 1-22)
These utilities can be run only in stand-alone mode from the content server computer. See also [administration application](#) (page 1-1).

administrator

Person in an organization that manages all or part of a content server system. See also [consumer](#) (page 1-4), [contributor](#) (page 1-5), [subadministrator](#) (page 1-21), and [system administrator](#) (page 1-21).

Admin permission

The permission level that allows users to perform the following tasks within a particular security group: view; check in; check out; get copy; delete; check in with another user specified as the author; Use the User Admin, Workflow Admin, Web Layout Editor, and Repository Manager administration applications (the user must have also [administration rights](#) (page 1-2) for the application)
See also [Read permission](#) (page 1-17), [Write permission](#) (page 1-26), and [Delete permission](#) (page 1-6).

admin role

A standard [role](#) (page 1-18) that gives the user Read, Write, Delete, and Admin permission to all security groups and rights to all administration tools. This role does not allow a user to access the Admin Server. See also [guest role](#) (page 1-9), [contributor role](#) (page 1-6), and [sysmanager role](#) (page 1-21).

Admin Server

An administration tool that enables the system administrator to set system properties and start, stop, and restart content servers (remotely, if necessary).

alias

A name that represents one or more users in workflows and subscriptions. For example, the “Mktg” alias could include all users in a marketing department.

#all

A special [account](#) (page 1-1) classification that can be used to assign a user a single permission level for all accounts.

alternate file

A web-viewable version of the primary file, or a version that can be converted to a web-viewable format upon checkin. The alternate file must be specified and checked in at the same time as the primary file. See also [primary file](#) (page 1-15).

archive

A set of content files and their associated batch files that are exported by Archiver.

Archiver

An [administration application](#) (page 1-1) that is used to transfer and reorganize content server files and information. Archiver has four main functions: [export](#) (page 1-7), [import](#) (page 1-9), [transfer](#) (page 1-22), and [replication](#) (page 1-17).

authentication

The process by which the content server system validates a user’s logon information. The user name and password are compared against an authorized list. If the system detects a match, access is granted to the extent specified in the permissions list for that user.

authorization type

The way that Oracle groups users, depending on how their user attributes are defined: [local user](#) (page 1-11), [global user](#) (page 1-9), or [external user](#) (page 1-7). Also referred to as *user type*.

basic subscription

A subscription to an individual piece of content by Content ID. See also [criteria subscription](#) (page 1-6), [forced subscription](#) (page 1-8), and [open subscription](#) (page 1-13).

basic workflow

A type of workflow where specific content items are entered into a workflow by an administrator. See also [criteria workflow](#) (page 1-6).

Batch Loader

An [administration utility](#) (page 1-2) that is used to check in (insert), delete, or update multiple content items at a time.

check in

To submit a file to the Content Server repository.

check out

To lock a content item in the Content Server repository so that no other users can revise it. Other users can still view and get a copy of a checked out content item. See also [undo checkout](#) (page 1-23).

component

An optional module that adds to or changes the functionality of the out-of-the-box installation of Content Server.

Component Manager

An Admin Server feature that enables administrators to remotely enable, disable, upload, and download content server components.

Component Wizard

An [administration utility](#) (page 1-2) that is used to create and manage custom components.

Configuration Manager

An [administration application](#) (page 1-1) that enables the system administrator to create and manage content types, file formats, and metadata fields.

consumer

A content server user who only needs to find, view, and/or print content. See also [contributor](#) (page 1-5), [subadministrator](#) (page 1-21), and [administrator](#) (page 1-2).

consumption site

A web site that content consumers use to view content. Content contributors use a different web site (a [contribution site](#) (page 1-5)) to check in content.

content

A collective term for the content items in the Content Server repository.

content folio

A logical grouping of related content items that can be managed as a single unit.

content ID

A standard, required metadata field that provides a unique identifier for each content item.

content information

See [metadata](#) (page 1-12).

content item

A file that has been checked in to the Content Server repository. A content item includes a [primary file](#) (page 1-15) and [metadata](#) (page 1-12), and can include an [alternate file](#) (page 1-3).

content profile

A set of one or more rules that can be used to control the display of metadata fields on the check-in and search pages. Content profiles are used to filter what information is displayed on these pages, based on user attributes, content attributes, or a combination of the two. This enables system administrators to make the check-in and search pages less complex and more specifically geared to particular user or content types.

content repository

The place where content files are stored. See also [Web Layout](#) (page 1-25) and [Vault](#) (page 1-24).

content type

See [type](#) (page 1-23).

contribution site

A web site that content contributors use to check in content. Content consumers access a different web site (a [consumption site](#) (page 1-4)) to view the content.

contributor

A content server user who creates and revises documents. See also [consumer](#) (page 1-4), [subadministrator](#) (page 1-21), [administrator](#) (page 1-2).

contributor role

A standard [role](#) (page 1-18) that gives the user Read and Write permission to the Public security group. See also [guest role](#) (page 1-9), [admin role](#) (page 1-2), and [sysmanager role](#) (page 1-21).

conversion

The process of changing an electronic file to a different file format (for example, changing a Microsoft Word document into PDF or HTML format).

core

The basic components of Content Server.

criteria subscription

A subscription to a group of content items based on metadata criteria. See also [basic subscription](#) (page 1-3), [forced subscription](#) (page 1-8), and [open subscription](#) (page 1-13).

criteria workflow

A type of workflow where a content item automatically enters the workflow if the security group and one metadata field match predefined criteria. See also [basic workflow](#) (page 1-3) and [sub-workflow](#) (page 1-21).

custom component

A user-defined Oracle [component](#) (page 1-4).

custom metadata field

A user-defined [metadata field](#) (page 1-12).

database provider

An Application Programming Interface (API) that establishes a connection from a content server to a database.

Delete permission

The permission level that allows users to perform the following tasks within a particular security group: view; check in; check out; get copy; delete.

See also [Read permission](#) (page 1-17), [Write permission](#) (page 1-26), and [Admin permission](#) (page 1-2).

dependent choice list

An option list in which the options depend on what is selected in a different option list. For example, if there is an option list for the Continent field and another option list for the Country field, the available choices in the Country option list depend on which Continent is selected.

document profile

See [content profile](#) (page 1-5).

documents without accounts

A special [account](#) (page 1-1) classification that assigns a permission level for content items that do not have an account specified. Same as [#none](#) (page 1-13).

document type

See [type](#) (page 1-23).

download

To copy a file from Content Server to your local hard disk.

download applet

An optional Java applet that enables users to download multiple content items at the same time. The content items can also be checked out during the download process. See also [multiple-file check-out](#) (page 1-13).

export

To copy files out of a Content Server instance to an [archive](#) (page 1-3) using Archiver.

external user

An [authorization type](#) (page 1-3) where user security attributes (password, roles, and accounts) are stored in an external storage system. External users might use a Microsoft network login or another type of provider (LDAP) login. See also [local user](#) (page 1-11) and [global user](#) (page 1-9).

Extras component

A supported Oracle [component](#) (page 1-4) that adds functionality to the base content server. Extras components are provided in the Extras directory on the content server DVD or CD-ROM.

field mapping

Copying exported values from one metadata field to another metadata field during import. See also [value mapping](#) (page 1-24).

file format

The structure of a file. The file format is determined by the application used to create the file, and can typically be determined by the file extension (such as *.pdf* or *.doc*).

file type

See [type](#) (page 1-23).

Folders component

An [Extras component](#) (page 1-7) that provides a hierarchical folder interface to content in Content Server. This component is required as a foundation for WebDAV.

folio

See [content folio](#) (page 1-5).

forced subscription

A subscription where users and/or aliases are assigned to the subscription by an administrator. If individual users are assigned, each user can unsubscribe if they wish. If an alias is assigned, the users in that alias cannot unsubscribe. See also [open subscription](#) (page 1-13), [basic subscription](#) (page 1-3), and [criteria subscription](#) (page 1-6).

full-text indexing

The process of creating a searchable index that includes every word in a file.

full-text search

A search that compares the query expression against every word in a file. See also [metadata search](#) (page 1-12).

full-text search operator

A word or symbol that refines the query expression for a full-text search (for example, AND, OR, and double quotation marks ").

GenWWW status

The revision status that indicates that the revision is being converted to web-viewable format or is being indexed, or has failed conversion or indexing.

global user

An [authorization type](#) (page 1-3) where user security attributes (password, roles, and accounts) are stored on master content server, but the user has access to proxied content servers as well. Global users are considered “lightly managed” users, as this authorization type limits some of the user functions in order to enhance scalability and performance. See also [local user](#) (page 1-11) and [external user](#) (page 1-7).

guest portal page

The web page that users see after they start Content Server but before they log in. See also [login portal page](#) (page 1-11).

guest role

A standard [role](#) (page 1-18) that gives the user Read permission to the Public security group. A login is not required to access content items in the security groups for which the *guest* role has permission. This role is assigned to anonymous users by default. See also [contributor role](#) (page 1-6), [admin role](#) (page 1-2), and [sysmanager role](#) (page 1-21).

historical report

A list that contains the results of a database query that was performed at a specific date and time. This type of report provides a “snapshot” of content server information as it existed at a particular moment. Historical reports are built with the Web Layout Editor and accessed through the Library. See also [active report](#) (page 1-1).

Home page

See [guest portal page](#) (page 1-9) and [login portal page](#) (page 1-11).

HTTP provider

A connection that allows communication between content servers using the HTTP protocol. This type of provider requires the Proxy Credentials Extension component.

Idoc Script

A proprietary server-side script language that is used to modify the functionality and look-and-feel of content management products. Idoc Script tags are in the format `<$script$>`.

import

To copy content from an archive into Content Server.

Inbound Refinery

Software included with Content Server that converts native files to web-viewable files, along with specific conversion add-on modules (such as PDF Converter or XML Converter).

incoming provider

A connection initiated from an outside entity, such as a browser or client application. The content server listens on a specified port to be aware of incoming connections.

information field

See [metadata field](#) (page 1-12).

installation directory

The directory where the content management software is installed.

instance

A single copy of Content Server. See also [master content server](#) (page 1-12) and [proxied content server](#) (page 1-16).

internal members

Users and groups who are employees of your company or members of your organization.

item

A content item, folder, or discussion.

latest revision

The most recent version of a piece of content.

LDAP provider

A connection initiated to an LDAP (Lightweight Directory Access Protocol) server for managing user access to the content server.

Library

A hierarchical structure of links that organizes content items based on metadata criteria. Users can “drill down” through the levels of the Library to find content items they are looking for. The Library hierarchy is built using the Web Layout Editor application, and is accessed by users through the Library link in the portal.

Lightly Managed Content

An [Extras component](#) (page 1-7) that enables external collections of files to be managed in the content server. Lightly managed content items are full-text indexed and their metadata is stored in the database, but the native files are not stored in the content server vault.

Users can search for, delete, and update metadata for lightly managed content, but only one revision is maintained, and file conversion is not performed.

local archive

An archive that belongs to a local collection.

local collection

A collection that the content server can reach by file access using a mapped or mounted network share.

locale

A setting that specifies the language of the content server interface and defines how the content server handles language-specific issues, such as date formatting and full-text indexing. See also [system locale](#) (page 1-21), [user locale](#) (page 1-24), and [Verity locale](#) (page 1-25).

local transfer

A transfer between local archives. Both the source archive and the target archive are in a local collection.

local user

An [authorization type](#) (page 1-3) where user security attributes (password, roles, and accounts) are stored on master content server, and the user is included in all local content server functions. See also [global user](#) (page 1-9) and [external user](#) (page 1-7).

log in

To gain access, or sign in, to the Oracle system. Logging in requires users to identify themselves by entering their user name and password, which Oracle uses to grant security permissions to content and rights to administrative functions.

login

See [user name](#) (page 1-24).

login portal page

The web page that users see after they log in. See also [guest portal page](#) (page 1-9).

major revision

The primary revision label (for example, the number portion of the revision sequence 1a, 1b, 2a, 2b). See also [minor revision](#) (page 1-12).

mapping

See [field mapping](#) (page 1-8).

master content server

The main Content Server instance of a group of instances that share the core code, but have separate content repositories and databases. See also [proxied content server](#) (page 1-16).

metadata

Information about a content item, such as Title, Author, Security Group, and so on. Metadata is used to describe, find, and group content items. Also referred to as *content information*.

metadata field

A field on a web page that is used to define metadata during checkin, or to define search criteria. Also referred to as *content information field*.

metadata search

A search that compares the query expression against metadata field values. See also [full-text search](#) (page 1-8).

Microsoft Login

An optional security implementation that enables users to be authenticated in Content Server through their Windows NT user name and password.

minor revision

The secondary revision label (for example, the letter portion of the revision sequence 1a, 1b, 2a, 2b). See also [major revision](#) (page 1-12).

multiple-file check-in

An optional feature of Content Server that enables users to check in multiple content items as a single compressed zip file. See also [upload applet](#) (page 1-23).

multiple-file check-out

An optional feature of Content Server that enables users to check out and download multiple content items at the same time. See also [download applet](#) (page 1-7).

native application

A software application that was used to create an original file that was checked in to the content server (for example, Microsoft Word or Adobe Photoshop).

native file

The original file that is checked into the Content Server repository. See also [primary file](#) (page 1-15).

native file format

The [file format](#) (page 1-8) that an original file was created in.

#none

A special [account](#) (page 1-1) classification that assigns a permission level for content items that do not have an account specified. Same as [documents without accounts](#) (page 1-7).

notification

The act of informing users through e-mail. In Content Server, e-mail notification is used in subscriptions and workflows.

NTLM authentication

User authentication using the NT LAN Manager method.

open subscription

A type of subscription where users manually subscribe to content items through a Basic or Criteria subscription. See also [forced subscription](#) (page 1-8), [basic subscription](#) (page 1-3), and [criteria subscription](#) (page 1-6).

option list

A drop-down list on a Oracle web page from which users can choose an item.

original file name

The name of the native file that was checked in as the primary file.

outgoing provider

A connection initiated to an outside entity, such as another content server.

pass through

The process of storing a native file in the *weblayout* repository without converting it to a web-viewable format. The Inbound Refinery may pass through a file if it cannot be converted (for example, because the native application is not supported), if a pass-through file format is specified during checkin, or if there is no need to convert the file (for example, if the original file is already in PDF format).

permanent license

A Content Server license for customers who have determined their long-term host name and instance name. A permanent license does not expire for the current version of the software, and it does not permit the removal of licensed add-on products. See also [temporary license](#) (page 1-22).

permission

The access that a user has to a particular security group or account. See also [Read permission](#) (page 1-17), [Write permission](#) (page 1-26), [Delete permission](#) (page 1-6), and [Admin permission](#) (page 1-2).

persistent URL

The URL (web address) that always points to the most recent version of a content item. When a new revision is checked in, the file name of the previous revision is changed to reflect the revision number (for example, 002050~1.pdf, 002050~2.pdf, and so on), and the new revision takes on the Content ID as the file name (for example, 002050.pdf). The directory location of the file remains the same as long as the security group, Type, and account are not changed.

personal URL

A link from the [portal navigation bar](#) (page 1-15) to a URL (web address) that a user accesses frequently.

personalization

Customization to a person's specific needs and preferences.

portal

Secure, personalized point of access to a company's resources and information. It typically includes a search engine for internal documents as well as the ability to customize the portal page for different user groups and individuals.

Portal Design

A Content Server feature that enables users to personalize their [portal navigation bar](#) (page 1-15).

portal navigation bar

The customizable navigation area on the left side of most web pages. See also [portal page](#) (page 1-15)

portal page

See [guest portal page](#) (page 1-9) and [login portal page](#) (page 1-11).

portlet

Dynamic components within a portal that serve as content channels (providing, for example, the latest headlines or personalized search capability).

preview provider

An outgoing provider connection to Oracle Content Publisher, for use with the optional HTML Preview component.

primary file

The original file that is checked in to the Content Server repository. See also [native file](#) (page 1-13) and [alternate file](#) (page 1-3).

privilege

See [permission](#) (page 1-14).

profile

See [content profile](#) (page 1-5).

provider

An Application Programming Interface (API) that establishes a connection with outside entities, such as other Content Server instances, LDAP servers, databases, or server sockets. There are six types of providers: [incoming provider](#) (page 1-10), [outgoing](#)

[provider](#) (page 1-14), [database provider](#) (page 1-6), [preview provider](#) (page 1-15), [LDAP provider](#) (page 1-10), and [HTTP provider](#) (page 1-9).

proxied

See [proxied content server](#) (page 1-16).

proxied archive

An archive that belongs to a proxied collection.

proxied collection

A collection on another content server that the local content server can reach through an outgoing provider.

proxied content server

A secondary Content Server instance in a group of instances that share the core code, but have separate content repositories and databases. A proxied content server is usually installed on the same machine as the master content server. See also [master content server](#) (page 1-12).

proxied source transfer

A transfer where the source archive is a proxied archive and the target is local.

proxied target transfer

A transfer where the source archive is a local archive and the target is proxied.

Public

A predefined [security group](#) (page 1-20). By default, no login is required to view Public content.

pull transfer

A transfer over an outgoing provider that is owned by the proxied (remote) content server.

push transfer

A transfer over an outgoing provider that is owned by the local content server.

query

See [search](#) (page 1-19).

query expression

A statement that specifies the criteria to be matched during a search. See also [search criteria](#) (page 1-19).

query result page

See [search results page](#) (page 1-20).

Read permission

The permission level that allows users to perform the following tasks within a particular security group: view; get copy.

See also [Write permission](#) (page 1-26), [Delete permission](#) (page 1-6), and [Admin permission](#) (page 1-2).

refinery

See [Inbound Refinery](#) (page 1-10).

relative path

The path to a file location that takes the current working directory as the starting point. For example, */weblayout/images/logo.gif*. See also [absolute path](#) (page 1-1).

relative URL

The URL (web address) of a web page that takes the location of the current page as the starting point. For example, *news/toc.htm*. See also [absolute URL](#) (page 1-1).

relative web root

The root directory name used by the web server to resolve URLs (web addresses). For example, */oracle_2/*.

release date

The date and time when a revision is available for searching and viewing in the content server.

rendition

A particular file associated with a content item, such as the [primary file](#) (page 1-15), [alternate file](#) (page 1-3), or [web-viewable file](#) (page 1-25).

replication

The process of automatically exporting, importing, or transferring archives using Archiver.

report

A list that contains the results of a content server database query. There are two types of reports, [active report](#) (page 1-1) and [historical report](#) (page 1-9), which are built with the Web Layout Editor and accessed through the Library.

repository

See [content repository](#) (page 1-5).

Repository Manager

An [administration application](#) (page 1-1) that is used to: manage content items (view status, delete revisions, and so on); create criteria subscriptions and assign users to subscriptions; update and rebuild the search index

reviewer

A user who can approve or reject a workflow revision but cannot check it out for editing. See also [export](#) (page 1-7).

revision

A new or revised version of a content item. By default, revisions are numbered sequentially starting with Revision 1, and every time the content item is checked out and checked in again, the revision number is incremented by one.

revision history

A record of all revisions for a particular content item. The files and metadata can be accessed for all previous revisions that have not been deleted.

revision status

The status of a revision in the content server. Revision status can be Done, Edit, GenWWW, Released, Pending, Expired, Deleted, or Review.

rights

The access that a user has to any of the following administration applications: [User Admin](#) (page 1-23); [Repository Manager](#) (page 1-18); [Workflow Admin](#) (page 1-26); [Web Layout Editor](#) (page 1-25).

See also [subadministrator](#) (page 1-21).

role

A set of permissions for each security group. Each user is assigned one or more roles that define their access to content. There are four predefined roles: [guest role](#) (page 1-9),

[contributor role](#) (page 1-6), [sysmanager role](#) (page 1-21), and [admin role](#) (page 1-2).
See also [permission](#) (page 1-14) and [security group](#) (page 1-20).

R permission

See [Read permission](#) (page 1-17).

RW permission

See [Write permission](#) (page 1-26).

RWD permission

See [Delete permission](#) (page 1-6).

RWDA permission

See [Admin permission](#) (page 1-2).

saved query

A link from the [portal navigation bar](#) (page 1-15) to a particular search that a user performs frequently.

score

A search results sorting option that rates each file with a number to determine how closely it matches the full-text search criteria. The higher the score, the closer the match.

search

To retrieve a list of content items that match specified criteria.

search collection

See [Verity collection](#) (page 1-25).

search criteria

The metadata values and/or full-text words and phrases to be matched during a search. See also [query expression](#) (page 1-17).

search engine

Software included with Content Server that performs metadata and full-text searches. See also [search index](#) (page 1-20).

search index

A set of files that contain metadata information and the full-text indexes. The search index is created by the Indexer and is read by the [search engine](#) (page 1-19).

search operator

A word or symbol that can be used in a query expression to refine the search criteria (for example, AND, OR, NOT, Substring, Matches, and so on).

search results

A list of content items that match the search criteria.

search results page

Standard page that displays the results of a query. Also referred to as *query result page*.

Secure

A predefined [security group](#) (page 1-20). By default, only the system administrator has access to this security group.

security group

A set of content items to which users are granted permission based on their roles. Each content item is assigned to a security group during checkin. There are two predefined security groups: [Public](#) (page 1-16) and [Secure](#) (page 1-20).

security model

The specific configuration of security groups, roles, and accounts that is defined for an organization.

self-registration

A function that enables users to create their own logins (user name and password).

source archive

An archive that contains batch files to be transferred.

step

A sequential stage in a workflow that defines which users can review, approve, and reject a revision.

subadministrator

A user who has rights to at least one [administration application](#) (page 1-1). See also [consumer](#) (page 1-4), [contributor](#) (page 1-5), [administrator](#) (page 1-2), and [system administrator](#) (page 1-21).

subscribe

To indicate that you wish to be notified by e-mail when a new revision of a particular content item is checked in to the Content Server repository.

subscriber

A user who is subscribed to a content item.

subscription

A function that notifies subscribed users by e-mail when a particular content item has been revised. See also [criteria subscription](#) (page 1-6), [basic subscription](#) (page 1-3), [forced subscription](#) (page 1-8), and [open subscription](#) (page 1-13).

sub-workflow

A type of workflow that does not have an initial contribution step. A file can enter a sub-workflow only through a jump from a [criteria workflow](#) (page 1-6).

sysadmin

A predefined user assigned to the [admin role](#) (page 1-2) and [sysmanager role](#) (page 1-21). This user cannot be deleted. See also [user1](#) (page 1-23).

sysmanager role

A standard [role](#) (page 1-18) that gives the user Read permission to the Public and Secure security groups, and access to the Admin Server. See also [guest role](#) (page 1-9), [contributor role](#) (page 1-6), and [admin role](#) (page 1-2).

system administrator

A user who has full administrative [permission](#) (page 1-14) and [administration rights](#) (page 1-2) to manage the Oracle system. See also [sysadmin](#) (page 1-21).

system locale

A setting that specifies the language of the content server interface and defines how the content server handles language-specific issues on a system-wide basis. See also [user locale](#) (page 1-24) and [Verity locale](#) (page 1-25).

System Properties

An [administration utility](#) (page 1-2) that is used to set global options and customize the content server environment.

talkback provider

In a [push transfer](#) (page 1-16), an outgoing provider configured at the target server which is used to notify the source server that the transfer is complete. Without the talkback provider, the transfer can still be completed, but the source must poll the target to confirm that the transfer is finished.

targetable archive

An archive that is enabled to be a target archive.

target archive

An archive that receives transferred batch files.

temporary license

A Content Server license for customers who have not determined their long-term host name or instance name. Temporary licenses expire at a specified date and permit the removal of add-on products. See also [permanent license](#) (page 1-14).

thumbnail

A miniature representation of a page or image. In Content Server, thumbnails are created by the Inbound Refinery and displayed on search result pages.

title

A descriptive name for a content item.

token

A piece of Idoc Script that defines variable users in a workflow.

topic

The subject of a discussion post.

transfer

To copy or move Archiver batch files and their associated content files from one archive to another. There are three types of transfers: [local transfer](#) (page 1-11), [pull transfer](#) (page 1-16), and [push transfer](#) (page 1-16).

transfer owner

The content server instance that performs and monitors a transfer.

transfer source

See [source archive](#) (page 1-20).

transfer target

See [target archive](#) (page 1-22).

type

A required, standard metadata field that is used to group similar content by category. Also referred to as *content type*, *document type*, or *file type*.

undo checkout

To cancel a content item checkout without creating a new revision.

unsubscribe

To cancel a [subscription](#) (page 1-21).

update

To modify the metadata for a revision without checking out the content item or adding a revision.

upload applet

An optional Java applet that enables users to check in multiple content items as a single compressed Zip file. See also [multiple-file check-in](#) (page 1-12).

user

A person who has been assigned a user name and password for Content Server.

user1

A predefined user assigned to the [contributor role](#) (page 1-6). This user cannot be deleted. See also [sysadmin](#) (page 1-21).

User Admin

An [administration application](#) (page 1-1) that is used to manage content server users and security access.

user ID

See [user name](#) (page 1-24).

user information

Information about a user, such as User Name, Full Name, E-mail Address, and so on.

user information field

A field that is used to define user information on the User Profile page.

user locale

A setting that specifies the language of the content server interface and defines how the content server handles language-specific issues for an individual user. See also [system locale](#) (page 1-21) and [Verity locale](#) (page 1-25).

user login

A Content Server user name and password.

user name

The name of a user, as recognized by Content Server.

User Profile

Personal information about a user, such as User Name, Full Name, E-Mail Address, and so on. User information can be changed by an administrator through the User Admin administration application, or by the user on their User Profile page.

user type

See [authorization type](#) (page 1-3).

value mapping

Changing exported metadata values to new values during import. See also [field mapping](#) (page 1-8).

Vault

The repository where native files are stored in Content Server. The default directory location for this repository is `<install_dir>/vault/`. See also [Web Layout](#) (page 1-25).

Verity

The standard indexer and [search engine](#) (page 1-19) included in Content Server.

Verity collection

A set of files that are created by Verity for indexing and searching. Also referred to as *index collection* or *search collection*.

Verity locale

A Verity setting that extends the Verity search function to work with languages other than English. See also [system locale](#) (page 1-21) and [user locale](#) (page 1-24).

web browser

Program that serves as a user's front-end to the World Wide Web (WWW) on the Internet or other networks or interfaces based on web technology.

WebDAV (Web-Based Distributed Authoring and Versioning)

A protocol that provides a way to remotely author and manage content using clients that support WebDAV. For example, you can use Microsoft Windows Explorer to check in, check out, and modify content in the Oracle repository rather than using Oracle's web browser interface.

Web Layout

The repository where web-viewable files are stored in Content Server. The default directory location for this repository is `<install_dir>/weblayout/`. See also [Vault](#) (page 1-24).

Web Layout Editor

An [administration application](#) (page 1-1) that is used to create the Library hierarchy, define reports, modify search result pages, and update the portal page.

web root

See [relative web root](#) (page 1-17).

web-viewable file

A file in a format that can be viewed using a web browser, such as PDF or HTML.

workflow

The process that routes a file for review and approval before it is released to the content server. Users are notified by e-mail when they have a file to review. There are three types of workflows: [basic workflow](#) (page 1-3), [criteria workflow](#) (page 1-6), and [sub-workflow](#) (page 1-21).

Workflow Admin

An [administration application](#) (page 1-1) that is used to set up and manage workflows.

Write permission

The permission level that allows users to perform the following tasks within a particular security group: view; check in; check out; get copy.

See also [Read permission](#) (page 1-17), [Delete permission](#) (page 1-6), and [Admin permission](#) (page 1-2).



#

- % (wildcard), 8-20
- * (wildcard), 8-20
- ? (wildcard), 8-20
- _ (wildcard), 8-20
- /in account names, 5-15

A

- accounts, 5-2, 5-12
 - all accounts, 5-16
 - assigning --, 5-16
 - documents without accounts, 5-16
 - hierarchy, 5-15
 - overview, 5-12
 - and permissions, 5-14
 - vs. security groups, 5-12
 - slashes in account names, 5-15
 - special classifications, 5-16
- Acrobat Reader, 4-3
- Active Directory, 5-18
- active reports, 10-7
- active workflows, 7-2
- add-ons
 - Content Categorizer, 8-3
- Admin permission (A), 5-8
- Admin role, 5-11
- Admin Server, 9-2, 9-4
 - Component Manager, 10-8
 - setting configuration options, 9-4
 - vs. System Properties, 9-6
 - viewing output logs, 9-5
 - viewing server logs, 9-5
- administration, 2-8
- administration tools, 9-2
 - Admin Server, 9-2, 9-4
 - Archiver, 9-2, 9-9
 - Batchloader, 9-2, 9-6
 - Configuration Manager, 9-2
 - Content Server Analyzer, 9-3, 9-13
 - Environment Packager, 9-3, 9-14
 - Repository Manager, 9-2, 9-8
 - System Properties, 9-3, 9-5
 - User Admin, 9-2, 9-6
 - Web Layout Editor, 9-2
 - Workflow Admin, 9-2
- administrator rights, 5-17
 - RepMan, 5-17
 - UserAdmin, 5-17
 - Web Layout, 5-17
 - Workflow, 5-17
- administrators, 5-4, 5-16
- aliases, 7-4
- all accounts, 5-16
- Analyzer, 9-3, 9-13
- AND (full-text search operator), 8-21
- application modules, 2-8
- application, running Content Server as an --, 6-2
- applications servers, 3-17
- architecture (Content Server), 3-3
 - content information database, 3-6
 - content server, 3-5
 - native file repository ("Vault"), 3-7
 - search collection, 3-6
 - web browser, 3-8
 - web server, 3-5
 - web-viewable file repository ("Web Layout"), 3-7
- architecture (Universal Content Management), 2-4
 - application modules, 2-8
 - core services, 2-5
 - integration, 2-14
 - repository, 2-5
- Archiver, 9-2, 9-9
 - exporting content, 9-10
 - importing content, 9-10
 - incremental archives, 9-17
 - log files, 9-14
 - manual archives, 9-17
 - replicating content, 9-12, 9-17
 - transferring content, 9-11
- Archiver logs, 9-14
- authorization types, 5-4

- external users, 5-5
- global users, 5-5
- local users, 5-4
- auto-contribution step in workflows, 7-3

B

- backups, 9-16
- basic workflows, 7-2
- batch load files, 9-7
- Batchloader, 3-8, 9-2, 9-6
- BEA WebLogic portal server, 2-15

C

- checked-out content, displaying a list of --, 8-16
- checking in files, 8-12
- checking out files, 8-14
- CIS (Content Integration Suite), 2-15
- clustering, 3-14
- collaboration, 7-1
- collaboration management, 2-12
- command line, starting content server from --, 6-3
- Common Object Request Broker Architecture (CORBA), 3-16
- communication protocols, 3-15
 - see also 'integration options'
- component architecture, 3-13
- Component Manager, 10-8
- Component Object Model (COM), 3-16
- Component Wizard, 3-13, 10-8
- components, 3-13, 10-7
- Configuration Manager, 9-2
- console, starting content server from --, 6-3
- consumers, 5-3
- Contains (metadata search operator), 8-19
- content, 8-1
 - checking in files, 8-12
 - checking out files, 8-14
 - content information page, 8-14
 - creating a new revision of --, 8-16
 - displaying a list of checked-out --, 8-16
 - exporting --, 9-10
 - file names, 8-6
 - folios, 8-11
 - full-text searches, 8-21
 - grouping, 8-2
 - identification, 8-4
 - identifiers, 8-4
 - importing --, 9-10
 - maximum file sizes, 8-7
 - metadata, 8-3

- navigating to --, 8-17
 - profiles, 8-10
 - replicating --, 9-12, 9-17
 - searching for --, 8-17
 - searching metadata, 8-17
 - storage, 8-8
 - subscriptions, 8-11
 - transferring --, 9-11
 - updating --, 8-16
 - using the library hierarchy, 8-17
 - viewing the status of --, 9-8
- Content Categorizer, 8-3
- content check-in form, 8-12
- content folios, 8-11
- content ID, 8-4
- content identifiers
 - content ID, 8-4
 - content title, 8-5
 - dID, 8-4
- content information, 8-3
 - storage of --, 3-6
- content information database, 3-6
- content information fields, see 'metadata fields'
- content information page, 8-14
- content information, see also 'metadata'
- Content Integration Suite (CIS), 2-15
- Content Portlet Suite (CPS), 2-15
 - BEA WebLogic, 2-15
 - IBM WebSphere, 2-16
 - Plumtree, 2-16
 - Sun ONE, 2-16
- content profiles, 8-10
- content publisher, 2-10
- Content Server
 - communication protocols, 3-15
 - integration, 3-15
- content server, 3-5
 - architecture, 3-3
 - clustering, 3-14
 - debug information, 6-3
 - hardware and software requirements, 3-2
 - integration options, 3-15
 - interfaces, 3-8
 - log files, 9-14
 - master and proxied servers, 3-11
 - maximum file sizes, 8-7
 - restarting (UNIX), 6-7
 - restarting (Windows), 6-5
 - running as a service, 6-2
 - running as an application, 6-2
 - starting (UNIX), 6-6
 - starting (Windows), 6-2
 - stopping (UNIX)
 - stopping a content server

- UNIX, 6-7
 - stopping (Windows), 6-4
- Content Server Analyzer, 9-3, 9-13
- content title, 8-5
- content types, 8-2
- contribution step in workflows, 7-3
- Contributor role, 5-11
- contributors, 5-4
- conversion, 2-7, 3-8, 3-13
- CORBA, 3-16
- core services, 2-5
 - administration, 2-8
 - conversion, 2-7
 - library services, 2-5
 - personalization, 2-6
 - search, 2-8
 - security, 2-6
 - user input, 2-7
 - workflow, 2-7
- CPS (Content Portlet Suite)
 - BEA WebLogic, 2-15
 - IBM WebSphere, 2-16
 - Plumtree portal server, 2-16
 - Sun ONE, 2-16
- criteria subscriptions, 8-12
- criteria workflows, 7-2
- custom components, 3-13, 10-7
- customization, 10-1
 - library hierarchy, 10-6
 - navigation menu, 10-6
 - portal page, 10-5

D

- database, 3-6
 - maximum capacity, 8-7
 - supported databases, 3-3
 - see also 'content information database'
- dDocName, 8-4
- dDocTitle, 8-5
- debug information, 6-3
- Definitive Guide to Stellent Content Server
 - Development, 4-2
- Delete permission (D), 5-8
- Desktop, 2-11, 2-13
- dFileSize, 8-7
- dID, 8-4
- digital asset management, 2-9, 2-13
- disaster recovery, 9-16, 9-18
- document management, 2-9, 2-10
- documentation, 4-1

- online help, 4-2
 - PDF files, 4-2
- documents without accounts, 5-16
- Dynamic Converter, 2-11
- dynamic converter, 2-10

E

- e-mail
 - and subscriptions, 8-11
 - and workflows, 7-3
- EJB, 3-18
- Ends (metadata search operator), 8-19
- Enterprise JavaBean (EJB), 3-18
- Environment Packager, 9-3, 9-14
- Error log file entry, 9-15
- exporting content, 9-10
- external security, 5-18
- external URLs, 10-7
- external users, 5-5, 5-18

F

- FAST search engine, 3-6, 3-6
- Fatal log file entry, 9-15
- file conversion, see 'conversion'
- file names of content items, 8-6
- file repositories
 - native files, 3-7
 - web-viewable files, 3-7
- file size, maximum --, 8-7
- file sizes, 8-7
- file storage, 8-8
- file subscriptions, 8-11
- file system capacity, 8-7
- files names
 - native files, 8-6
 - web-viewable files, 8-6
- folios, 8-11
- full text, storage of --, 8-10
- full-text searches, 8-21
 - search operators, 8-21

G

- global users, 5-5
- glossary, 1-1
- graphical workflows, 7-5
- grouping of content, 8-2
- Guest role, 5-11

H

hardware requirements, 3-2
 Has Prefix (metadata search operator), 8-19
 Has Word (Has Word search operator), 8-19
 Has Word Prefix (metadata search operator), 8-19
 help system, 4-2
 hierarchy of accounts, 5-15
 historical reports, 10-7

I

IBM WebSphere portal server, 2-16
 IDC Content Admin Service, 6-2
 IDC Content Service, 6-2
 ldcAdmin.exe, 6-4
 ldcCommand, 3-15
 ldcServer.exe, 6-3
 identification of content, 8-4
 ldoc Script, 10-5, 10-8
 and workflows, 7-3
 Image Manager, 2-13
 importing content, 9-10
 Inbound Refinery, 3-8
 log files, 9-15
 Inbound Refinery logs, 9-15
 incremental archive, 9-17
 Info log file entry, 9-15
 integration, 3-15
 content server
 integration, 2-14
 integration kits
 BEA WebLogic, 2-15
 Content Integration Suite (CIS), 2-15
 IBM WebSphere, 2-16
 Plumtree, 2-16
 Sun ONE, 2-16
 Verity search engine, 3-6
 integration methods, 2-14
 integration options, 3-15
 Common Object Request Broker Architecture
 (CORBA), 3-16
 Component Object Model (COM), 3-16
 ldcCommand, 3-15
 Java API, 3-15
 Java Server Page (JSP), 3-16
 Open Document Management API (ODMA), 3-16
 Simple Object Access Protocol (SOAP), 3-16
 Web Distributed Authoring and Versioning
 (WebDAV), 3-16
 interfaces, 3-15
 content server, 3-8
 internal security, 5-18

intradoc_users newsgroup, 4-1

J

J2EE, 3-17
 Java API, 3-15
 Java Server Page (JSP), 3-16
 Java Virtual Machine (JVM), 3-3
 JSP, 3-16
 jumps in workflows, 7-3
 JVM (Java Virtual Machine), 3-3

L

language packs, 10-4
 Launcher, 6-3
 LDAP, 5-18
 library hierarchy, 8-17, 10-6
 library services, 2-5
 link types in library hierarchy
 external URLs, 10-7
 local pages (directory), 10-6
 local pages (reports), 10-7
 queries, 10-7
 local pages (directory), 10-6
 local pages (reports), 10-7
 local transfers, 9-12
 local users, 5-4
 locales
 localized user interface, 10-4
 localized user interfaces, 10-4
 log file entries
 error, 9-15
 fatal, 9-15
 info, 9-15
 log files, 9-5, 9-14
 Archiver logs, 9-14
 Inbound Refinery logs, 9-15
 server logs, 9-14

M

manual archive, 9-17
 master servers, 3-11
 Matches (metadata search operator), 8-18
 maximum file sizes, 8-7, 8-7
 metadata, 8-3
 editing --, 8-16
 searching --, 8-17
 storage, 8-9
 metadata fields, 8-13
 metadata searches, 8-17

- search operators, 8-18
- wildcards, 8-20
- metadata storage, 3-6

N

- names of content files, 8-6
- naming
 - native files, 8-6
 - web-viewable files, 8-6
- native file repository ("Vault"), 3-7
- native files
 - naming of --, 8-6
 - storage, 3-7, 8-8
- navigating to content, 8-17
- navigation menu
 - personalization, 10-6
- newsgroup, 4-1
- NOT (full-text search operator), 8-22
- Not Has Word Prefix (metadata search operator), 8-20
- Not Matches (metadata search operator), 8-18
- Not Substring (metadata search operator), 8-18

O

- ODMA, 2-7, 3-16
- online help, 4-2
- Open Document Management API (ODMA), 3-16
- OR (full-text search operator), 8-21
- output logs, 9-5

P

- passing through content, 3-13
- PDF Converter, 2-11
- PDF documentation, 4-2
- permissions, 5-2, 5-8
 - and accounts, 5-14
 - admin (A), 5-8
 - assigning --, 5-8
 - delete (D), 5-8
 - read (R), 5-8
 - and roles, 5-8
 - write (W), 5-8
- persistent URL, 8-6
- personalization, 2-6, 10-1
 - navigation menu, 10-6
- Physical Content Manager, 2-12
- Plumtree portal server, 2-16
- Portal Design, 10-6
- portal page
 - customization, 10-5

- portal servers, 3-17
- portals, 3-17
- portlets, 3-17
- predefined roles, 5-11
- predefined security groups
 - security groups
 - predefined --, 5-7
- predefined user logins, 5-6
- profiles, 8-10
- protocols, 3-15
- proxied servers, 3-11
- Public security group, 5-7
- pull transfers, 9-12
- push transfers, 9-12

Q

- query links, 10-7

R

- Read permission (R), 5-8
- records management, 2-9, 2-12
- Records Manager, 2-12
- recovery, 9-18
- replicating content, 9-12, 9-17
- replication, 9-17
- RepMan right, 5-17
- reports, 10-7
 - active --, 10-7
 - historical --, 10-7
- repository, 2-5
- Repository Manager, 9-2, 9-8
 - updating and rebuilding the search index, 9-9
 - using -- to look up content, 8-22
 - viewing the status of content, 9-8
- restarting a content server
 - UNIX, 6-7
 - Windows, 6-5
- restoring database/file system, 9-17
- reviewer step in workflows, 7-3
- reviewer/contributor step in workflows, 7-3
- reviewing system settings, 9-3
- revisions, 8-16
- rights
 - RepMan, 5-17
 - UserAdmin, 5-17
 - WebLayout, 5-17
 - Workflow, 5-17
- roles, 5-2, 5-8
 - admin, 5-11
 - contributor, 5-11

- guest, 5-11
- predefined --, 5-11
- sysmanager, 5-11

S

- script templates, 7-5
- scripting language, 7-3, 10-8
- search collection, 3-6
- search index, 3-7
 - updating and rebuilding the --, 9-9
- search operators
 - in full-text searches, 8-21
 - in metadata searches, 8-18
- search operators (full text)
 - AND, 8-21
 - NOT, 8-22
 - OR, 8-21
- search operators (metadata)
 - Contains, 8-19
 - Ends, 8-19
 - Has Prefix, 8-19
 - Has Word, 8-19
 - Has Word Prefix, 8-19
 - Matches, 8-18
 - Not Has Word Prefix, 8-20
 - Not Matches, 8-18
 - Not Substring, 8-18
 - Starts, 8-19
 - Substring, 8-18
- search solutions, 2-8
- searching, 2-8, 3-6
- searching for content, 8-17
 - full-text searches, 8-21
 - navigating to content, 8-17
 - searching metadata, 8-17
 - using Repository Manager, 8-22
- Secure security group, 5-7
- security, 2-6
 - accounts, 5-2
 - administrator rights, 5-17
 - external, 5-18
 - internal, 5-18
 - overview, 5-1
 - permissions, 5-2, 5-8
 - roles, 5-2, 5-8
 - security filter, 3-5
 - security groups, 5-1, 5-7
 - setting up --, 5-20
 - users, 5-1, 5-3
 - web server and --, 3-5
- security filter, 3-5
- security groups, 5-1, 8-2
 - vs. accounts, 5-12
 - defining and managing --, 5-7
 - overview, 5-7
 - Public, 5-7
 - Secure, 5-7
- server logs, 9-5, 9-14
- service
 - IDC Content Admin Service, 6-2
 - IDC Content Service, 6-2
 - running Content Server as a --, 6-2
- setting up security, 5-20
- Simple Object Access Protocol (SOAP), 3-16
- site studio, 2-10
- slashes (/) in account names, 5-15
- SOAP, 3-16
- Software Developer's Kit (SDK)
 - custom components, 10-8
- software requirements, 3-2
- starting a content server
 - UNIX, 6-6
 - Windows, 6-2
- Starts (metadata search operator), 8-19
- Stellent
 - customization, 10-1
 - hardware and software requirements, 3-2
 - personalization, 10-1
- steps in workflows, 7-3
- stopping a content server
 - Windows, 6-4
- storage
 - content, 8-8
 - content files, 8-8
 - full text, 8-10
 - metadata, 3-6, 8-9
 - native files, 3-7, 8-8
 - web-viewable files, 3-7, 8-9
- sub-administrators, 5-4, 5-16
- sub-workflows, 7-3
- subscriptions, 8-11
 - criteria --, 8-12
 - e-mail notifications, 8-11
 - file --, 8-11
 - types, 8-11
- Substring (metadata search operator), 8-18
- Sun ONE portal server, 2-16
- support, 4-1
 - before contacting --, 4-1
 - newsgroup, 4-1
- Sysadmin user login, 5-6
- Sysmanager role, 5-11
- System Properties, 9-3, 9-5
- system settings, 9-3

T

templates for workflows, 7-4
 tokens in workflows, 7-4
 training, 4-1
 transferring content, 9-11
 transfers
 local --, 9-12
 pull --, 9-12
 push --, 9-12

U

Universal Content Management
 application modules, 2-8
 architecture, 2-4
 collaboration management, 2-12
 core services, 2-5
 digital asset management, 2-9, 2-13
 document management, 2-9, 2-10
 overview, 2-2
 records management, 2-9, 2-12
 repository, 2-5
 web content management, 2-9, 2-9
 updating content, 8-16
 URLs
 links to external -- in library hierarchy, 10-7
 persistent --, 8-6
 User Admin, 9-2, 9-6
 assigning permissions, 5-8
 assigning rights to sub-administrators, 5-17
 defining and managing security groups, 5-7
 defining and managing users, 9-6
 user input, 2-7
 user interface
 localized --, 10-4
 user locales
 enabling --, 10-4
 user logins
 sysadmin, 5-6
 user1, 5-6
 User1 user login, 5-6
 UserAdmin right, 5-17
 users, 5-1
 administrators, 5-4, 5-16
 authorization types, 5-4
 consumers, 5-3
 contributors, 5-4
 defining and managing --, 9-6
 external --, 5-5, 5-18
 global --, 5-5
 local --, 5-4
 sub-administrators, 5-4, 5-16

user types, 5-3

V

Vault, see 'native file repository'
 Verity search engine, 3-6
 full-text search operators, 8-21
 Video Manager, 2-13
 Visio 2002 and workflows, 7-5

W

web browser, 3-8
 web content management, 2-9, 2-9
 Web Distributed Authoring and Versioning
 (WebDAV), 3-16
 Web Layout Editor, 9-2
 building library hierarchy, 10-6
 customization of portal page, 10-5
 Web Layout, see 'web-viewable file repository'
 web server, 3-5
 and security, 3-5
 web servers
 supported --, 3-2
 web-viewable file repository ("Web Layout"), 3-7
 web-viewable files
 naming of --, 8-6
 storage, 3-7, 8-9
 WebDAV, 2-7, 3-16
 WebLayout right, 5-17
 WebLogic portal server, 2-15
 WebSphere portal server, 2-16
 wildcards
 % (percentage symbol), 8-20
 * (asterisk), 8-20
 ? (question mark), 8-20
 _ (underscore), 8-20
 in metadata searches, 8-20
 Workflow Admin, 7-1, 9-2
 Workflow Designer, 7-5
 Workflow right, 5-17
 workflows, 2-7, 7-1
 active --, 7-2
 aliases, 7-4
 auto-contribution steps, 7-3
 basic --, 7-2
 contribution steps, 7-3
 criteria --, 7-2
 e-mail notifications, 7-3
 graphical workflows, 7-5
 jumps, 7-3
 reviewer steps, 7-3

Index

reviewer/contributor steps, 7-3
script templates, 7-5
steps, 7-3
sub--, 7-3
templates, 7-4
tokens, 7-4
types, 7-2
and Visio 2002, 7-5
Workflow Designer, 7-5

workflow templates, 7-5
working with content, 8-1
Write permission (W), 5-8

Y

Yahoo newsgroup, 4-1

