

---

CM System

# Concepts Guide

Copyright © 1999-2013 Percussion Software.  
All rights reserved

The software contains proprietary information of Percussion Software; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

Due to continued product development this information may change without notice. The information and intellectual property contained herein is confidential between Percussion Software and the client and remains the exclusive property of Percussion Software. If you find any problems in the documentation, please report them to us in writing. Percussion Software does not warrant that this document is error-free.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Percussion Software.

AuthorIT™ is a trademark of Optical Systems Corporation Ltd.

Microsoft Word, Microsoft Office, Windows®, Window 95™, Window 98™, Windows NT® and MS-DOS™ are trademarks of the Microsoft Corporation.

This document was created using *AuthorIT™, Total Document Creation* (see AuthorIT Home - <http://www.author-it.com>).

**Percussion Software**

600 Unicorn Park Drive  
Woburn, MA USA 01801  
781.438.9900  
Internet E-Mail: [technical\\_support@percussion.com](mailto:technical_support@percussion.com)  
Website: <http://www.percussion.com>

# Contents

<b>Introduction.....</b>	<b>7</b>
The Basics of Content Management.....	8
Content Management in Percussion CM System .....	9
Percussion CM System and Item-based.....	10
Content Management.....	10
Using Percussion CM System .....	11
Documentation .....	11
<b>Percussion CM System Concepts.....</b>	<b>13</b>
Content Concepts .....	14
Content Items .....	14
Content Types .....	14
Content Editors .....	15
Locales.....	16
Assembly Concepts .....	17
Templates .....	17
Active Assembly.....	18
UI Concepts .....	20
Content Explorer .....	20
Folders.....	21
Views .....	22
Sessions.....	22
Publishing Concepts.....	23
Publishing.....	23

Sites.....	23
Editions .....	24
<b>System Concepts.....</b>	<b>25</b>
Workflow .....	25
Relationships.....	26
Security Concepts .....	27
Communities .....	27
Roles.....	27
<b>Percussion CM System Logical Architecture and Processing.....</b>	<b>29</b>
Content Service .....	30
Assembly Service.....	34
Relationship Service.....	37
Workflow Engine.....	41
Publishing Engine .....	44
<b>Clients and Interfaces .....</b>	<b>47</b>
Workbench.....	48
Server Administrator.....	52
Multi-Server Manager.....	62
Content Explorer .....	63
Web Services API .....	64
Full-text Search Engine .....	65
<b>Percussion CM System Modules .....</b>	<b>67</b>
Word Connector .....	68
<b>FastForward for Web Content .....</b>	<b>69</b>
<b>Management.....</b>	<b>69</b>

<b>Physical Architecture, Deployment, and Scaling</b> .....	<b>71</b>
Percussion CM System Physical Architecture .....	72
Deployment Scenarios .....	73
Percussion CM System Components.....	73
Multi-tiered Environment .....	74
Configuration Options for Development, Testing, and Production Tiers.....	76
Scaling the Percussion CM System Publishing Environment .....	78
Guidelines for Expanding a Percussion CM System .....	79
JBoss Application Server .....	80
Database .....	81
<b>Data Protection</b> .....	<b>83</b>
Security .....	84
Backup.....	85
Archiving .....	85
Purging .....	85
Backup Recommendations.....	85
<b>XSL in Percussion CM System</b> .....	<b>87</b>
<b>Index</b> .....	<b>89</b>



---

## CHAPTER 1

# Introduction

Percussion Content Management System (CM System) is a Content Management solution that provides a combination of out-of-the-box capabilities and customization options that enable companies to build high-impact, tailored functionality that evolves as their needs grow. A robust, scalable system, CM System manages Web and portal content, documents, digital assets, and scanned images. To provide greater efficiency and flexibility, CM System optimizes content delivery for multiple channels and enables customers to maximize the value of their content by providing content reuse capabilities.

### Concepts Guide

This guide provides an overview of Percussion CM System for managers, system administrators, and implementers that are new to the product. It introduces the concepts and architecture of CM System, and provides guidance for installation and deployment.

After reading this document, you should have a basic understanding of CM System. Users who require more detailed information after reading this document may find the following CM System documents useful:

- For complete documentation of Workflows in CM System, see the online Help for the Workflow tab in Context Explorer. Also see the *Percussion CM System Implementation Guide*.
- For complete documentation on publishing design, see the online Help for the Publishing Design tab in Content Explorer. Also see the *Percussion CM System Implementation Guide*.
- For complete documentation on publishing administration, see the online Help for the Publishing Runtime tab in Content Explorer. Also see the *Percussion CM System Administration Manual*.
- For complete documentation of System Administration in CM System, see the the *System Administrator Help*, accessible from the Percussion CM Server Administrator. Also see the *Percussion CM System Administration Manual*.
- For complete documentation of Content Explorer, see the *Content Explorer Online Help*, accessible from Content Explorer.
- For complete documentation of the Active Assembly Interface, see the *Active Assembly Interface Help*, accessible from the Active Assembly Interface.
- For help planning and designing your CM System, see *Modeling and Design of a Percussion CM System*.
- For a guide to CM System's Multi-Server Manager, see the document *Multi-Server Manager*.
- For information about using WebDAV with CM System, see the document *Implementing WebDAV in a Percussion CM System*.

---

# The Basics of Content Management

Content Management refers to a set of applications and capabilities that enable maintenance and publishing of items of data, such as documents or Web content.

Web Content Management (WCM) enables organizations to:

- manage information resources that are created and stored in countless places;
- deliver multiple types of targeted content to specific audiences;
- support easier content reuse throughout the content lifecycle.

A WCM system offers a strategic framework for storing enterprise content and communicating that content through multiple channels to multiple audiences. After being published, content may be delivered to customers through various types of applications including e-commerce, sales automation, customer relationship management, supply chain management, or employee portals.



---

# Content Management in Percussion CM System

The Percussion CM System manages content through the automation of content creation, maintenance, and delivery, enhanced with features such as content reuse, delivery to multiple sites, intelligent relationships between content, and simple content assembly for business users. One of CM System's key features is content reuse, which it achieves by separating content and formatting, enabling users to create a variety of outputs using the same data. Business users can assemble the same Content Items multiple times using different formats and include the Content Items in the output formats of other content. This enables CM System output pages to contain many Content Items assembled together.

Another key feature of CM System is its ability to publish content to multiple sites. CM System allows any number of applications to be used to present published content to consumers since the content delivery mechanism (a portal, Web server, or other application) is not connected or “coupled” to CM System. This “decoupled delivery” not only offers companies flexibility in choosing delivery applications, but also simplifies the process of delivering content to multiple media.

CM System provides all of its users with the interfaces and mechanisms they need to accomplish their tasks:

- Content contributors (business users) perform their work through the Active Assembly interface and the Content tab of Content Explorer. These graphical user interfaces provide these users with the ability to create and maintain content, and to assemble it into outputs. In addition, content contributors can author content through Microsoft Word or any other application that produces files in formats uploadable to CM System.
- Implementers and CMS Administrators use the functionality provided on Content Explorer's Workflow and Publishing tabs to configure CM System's components to meet their companies' needs, and use CM Server Administrator to maintain the Percussion Content Management Server (CM Server), security, and user settings.
- Designers customize the CM System elements and interfaces available to their content contributors through the Percussion CM System Workbench and use CM System's Multi-Server Manager to move components between development, testing, and production CM Servers.

---

## Percussion CM System and Item-based Content Management

Percussion CM System provides item-based content management, breaking all content into granular pieces that can be assembled, used and reused in many ways. With item-based content management, users can contribute, manage, preview and rearrange any type of content, regardless of how it will be assembled and used in the future.

Percussion CM System output pages can contain many Content Items assembled together. For example, a job posting may contain an image, a job description, a benefits description, and a legal disclaimer. CM System treats each of these as a distinct item. Page-based systems force users to treat this content as a single entity, causing them to reenter the same information into their system on each page that displays it. CM System allows content to be broken into its basic Content Items, which are then managed individually. Therefore, if the legal department needs to change the disclaimer on the bottom of all job postings, the change need only be applied once to the source item. The change is then automatically reflected across all job postings and anywhere else the disclaimer is used.

---

# Using Percussion CM System Documentation

In addition to the *Percussion CM System Concepts Guide*, the following documents are also available for CM System:

- *Getting Started with Percussion CM System*  
This document guides you through the process of installing a development server and publishing the FastForward sample content. It also includes a brief walk-through of basic CM System functionality.
- *Modeling and Designing a Percussion CM System*  
*Percussion CM System Implementation Guide*  
*Setting Up the Production Environment*  
Together, these three documents provide a tutorial-style guide to implementing CM System to deliver to a Web server file system.
- *Internationalizing and Localizing Percussion CM System*  
This document describes how to implement a localized CM System and how to internationalize Templates for localization.
- *Implementing the Word Connector*  
This document describes how to implement Microsoft Word as an alternate interface for contributing text content.
- *Implementing WebDAV*  
This document describes how to implement WebDAV as an alternate interface to Content Explorer Folders.

All of these documents are available on the Percussion CM System CD. The links to these documents assume that copies reside in the same directory as this document.

Different users of CM System will find particular documents useful:

- If you are going to implement CM System:
  - Read at least the following sections of the *Percussion CM System Concepts Guide*:  
*Percussion CM System Logical Architecture and Processing* (on page 29)  
*Clients and Interfaces* (on page 47)  
*Full-text Search Engine* (on page 65)  
*Percussion CM System FastForward for Web Content Management* (see "FastForward for Web Content Management" on page 69)
  - Read and use *Getting Started with Percussion CM System* to install your development server, test your implementation, and learn basic CM System functionality.

- Read and use *Modeling and Designing a Percussion CM System*, *Percussion CM System Implementation Guide*, and *Setting Up the Production Environment* to plan and develop your implementation.
- If you want to customize Content Explorer, implement internationalization and localization, or other speciality implementations, read and use the appropriate ancillary documents.
- If you are a Web Master who will administer CM System:
  - Read at least the following sections of the *Percussion CM System Concepts Guide*:  
*Percussion CM System Logical Architecture and Processing* (on page 29)  
*Clients and Interfaces* (on page 47)  
*Full-text Search Engine* (on page 65)
  - Read and perform the exercises in "A Brief Introduction to Percussion CM System" in *Getting Started with Percussion CM System*.
- If you will be managing a CM System implementation:
  - Read at least the following sections of the *Percussion CM System*:  
*Percussion CM System Concepts* (on page 13)  
*Percussion CM System Logical Architecture and Processing* (on page 29)  
*Full-text Search Engine* (on page 65)  
*Percussion CM System Modules* (on page 67)  
*Percussion CM System FastForward for Web Content Management* (see "FastForward for Web Content Management" on page 69)
  - Read and perform the exercises in "A Brief Introduction to Percussion CM System" in *Getting Started with Percussion CM System*.
- If you are a system administrator who is adding CM System to your network and software infrastructure, read at least the following sections of the *Percussion CM System Concepts Guide*:  
*Clients and Interfaces* (on page 47)  
*Physical Architecture, Deployment, and Scaling* (on page 71)  
*Data Protection* (on page 83)  
*System Requirements* (on page 87)

## CHAPTER 2

# Percussion CM System Concepts

To help you understand Percussion CM System, this section lists and defines some of the system's major concepts.

---

# Content Concepts

## Content Items

Content Items are the basic units of content in Percussion CM System. A Content Item may be a page or a portion of a page, or an image or other binary file, stored in a database. By defining output in terms of Content Items and related collections of Content Items, CM System allows users to modify only the parts of a page or other output that change and to reformat individual Content Items for multiple uses.

Content Items treat both traditional body content and metadata such as content titles and start dates as body content. Any type of data in the Content Item can be included in the Content Item output.

CM System stores the Content Item data in database tables and formats it prior to publishing it to a *Site* (see "Sites" on page 23). Each Content Item can be displayed in any number of formats and reused in multiple locations throughout a Site. Content Items can also include links to other Content Items which can be included in their outputs.

For more information, see *Content Service* (on page 30).

## Content Types

In Percussion CM System, a Content Type defines the fields and structure for storing information contained in a Content Item. The Content Type definition also controls the behavior of the *Content Editor (Web form)* (see "Content Editors" on page 15) used to create and modify the Content Items of that type. The implementer of each CM System defines a set of Content Types that store the different types of information displayed on the organization's Web Site(s). For example, one Content Type might store uploaded images while another could store promotional text. In addition to body content fields, each Content Type also includes fields for metadata associated with the main content.

CM System provides a developer interface for designing and customizing the functions of Content Types.

For more information, see *Content Engine* (see "Content Service" on page 30).

## Content Editors

A Percussion CM System Content Editor is a form for displaying and editing Content Items of a specific Content Type. Content Editors list and display the editable fields of a Content Item.

Figure 1: A Content Editor displaying a Content Item

Content Title (ID)	Creator	Created On	Last Modifier	Last Modified On
Before planning your estate, plan to see a lawyer (338)	admin1	Aug 18, 2004 - 12:00	admin1	Aug 18, 2004 - 12:00

State(ID)	Public	Checked Out	Assignees(Type)
Draft(1)		by me	Admin Author Editor Web_Admin
Community	Workflow	Locale	
Internet Community (1002)	Standard Workflow (5)	US English	

Figure 2: A Content Editor's Properties page

Each Content Editor is automatically created when an implementer creates a Content Type.

For more information, see *Content Service* (on page 30).

## Locales

A Percussion CM System Locale is a language including its regional variation, such as French Canadian or British English. CM System uses Locales in two distinct areas. First, each Content Item in the system is marked with a specific Locale. By default, the Locale of newly created items is set to that of the user interface. Second, the user interface Locale (the login user's default Locale or the Locale chosen by the user during login) affects the language of labels seen in Content Explorer during the session.

Locales enable customers to deliver similar content to audiences that speak different languages. If more than one Locale is available, the system is globalized, and it is possible for users to make Translation Copies of Content Items. For example, a customer may require two different sites for the Canadian market—one site in Canadian English and one site in Canadian French. Content contributors could begin by creating the site in Canadian English, and then translators could create duplicate Canadian French pages by creating Translation Content Items of each English page.

For information about Localizing your CM System, see the document *Internationalizing and Localizing Percussion CM System*.



# Assembly Concepts

## Templates

In Percussion CM System, a Template defines how to produce the formatted output (usually a Web page or a portion of a Web page) of a Content Item. The Template defines transformation and formatting rules that are applied to a Content Item's fields and includes placeholders called *Slots* for inserting other formatted Content Items. A Template can be associated with a specific Content Type or shared among Content Types, and each Content Type may have any number of associated Templates.

*CM System provides a developer interface for designing Templates and assigning them to Content Types.* (see "Assembly Service" on page 34)

The following graphics show the same Content Item formatted using two different Templates:



Figure 3: A Content Item formatted by a Page Template

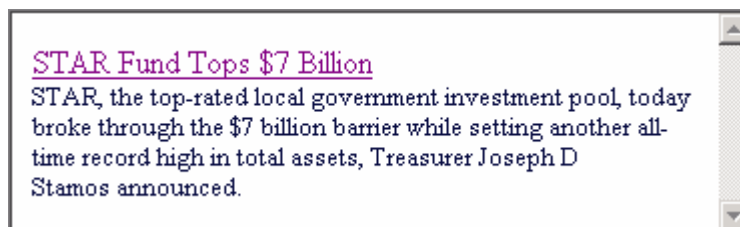


Figure 4: The same Content Item formatted by a Snippet Template

## Active Assembly

Active Assembly enables business users to assemble related content items together into the formatted pages, documents and other output they wish the CMS to generate. During any form of Active Assembly, a user is actually inserting a content item into a Slot on another content item's *Template* (see page 17). The Active Assembly interface provides a WYSIWYG browser UI that lets business users assemble content visually into Slots (as well as perform a range of other functions on them, including content item Workflow and creating new content items to add to Slots).



Figure 5: Active Assembly Interface

Active Assembly for Documents provides a split pane "tree" view that lets users assemble content hierarchically. In either case, users of Active Assembly perform searches against content in the system, then select the content items they wish to insert into a Template. They also select the Template of each content item to control how that item will appear in the final assembled output. Likewise, through Active Assembly, content items may have been added to Slots on the Templates inserted into the current item's Slots.

When a user opens a content item into Active Assembly, and adds another related content item into one of its Slots, the two items become linked by an Active Assembly Relationship. For more information, see *Relationships* (on page 26).

When a content item is previewed or published in one of its output formats, CM System automatically executes its Active Assembly Relationships in a process referred to as recursive rollup. The most deeply embedded related content items are formatted, then the content items that include them are formatted, then the content items that include these content items are formatted, and so on until the page content item that includes and displays all the others is formatted. The following diagram illustrates this process:

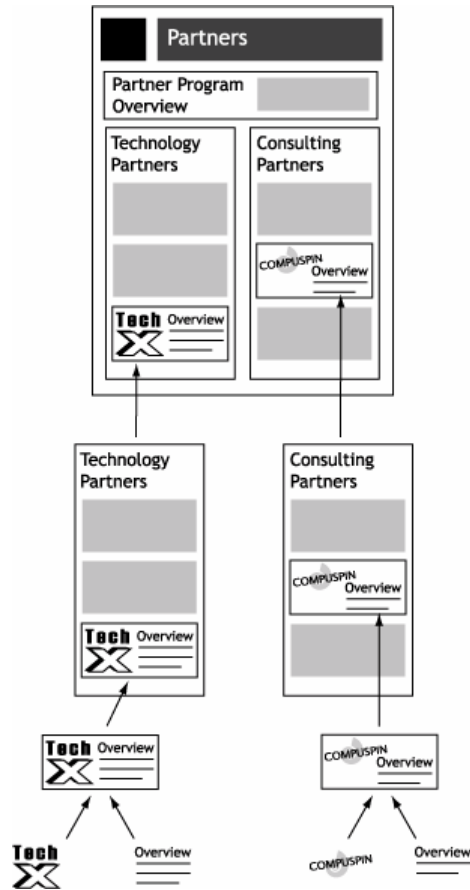


Figure 6: Recursive Rollup

NOTE: Active Assembly Relationships are predefined in CM System, but implementers can *modify their specific behavior* (see "Relationship Service" on page 37).

# UI Concepts

## Content Explorer

Content Explorer gives Web masters, CMS managers, and business users direct access to Content Items in the CMS including functions such as create, approve, and copy, as well as mechanisms to assist in organizing Content Items such as Folders, Views and Searches.

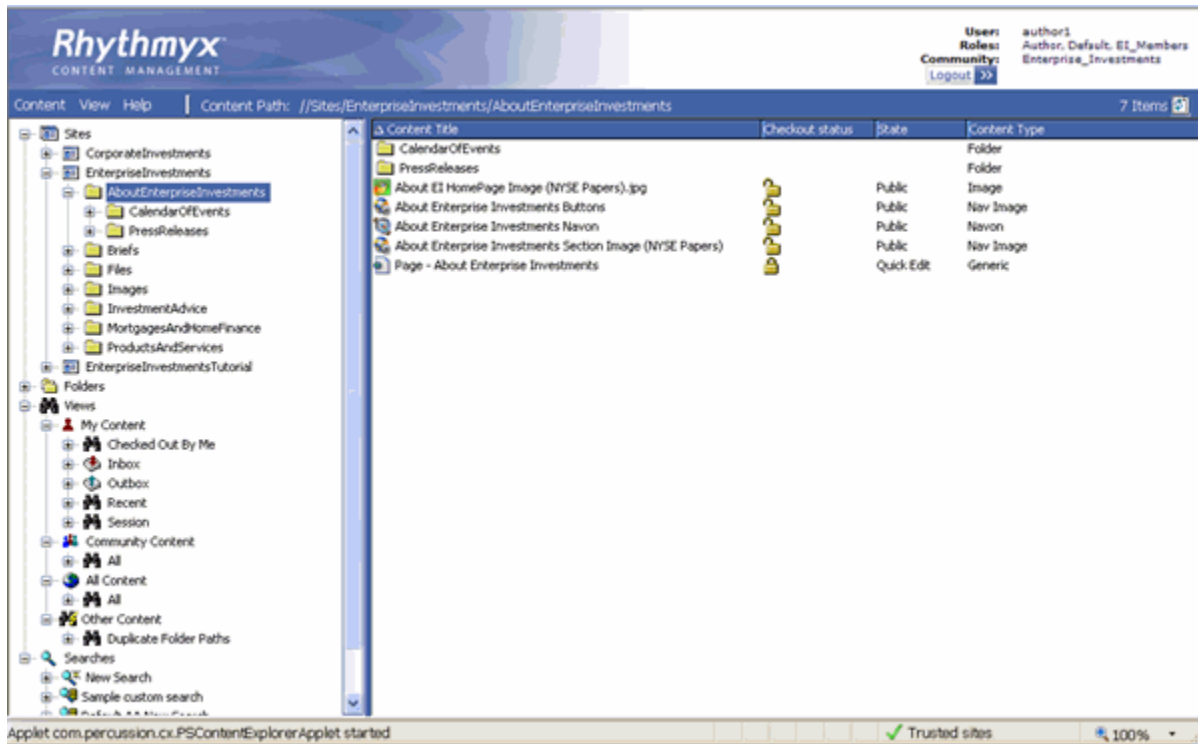


Figure 7: Content Explorer

Content Explorer includes a Navigation Tree with Folders, Views, and Searches that present Content Items by user-defined or system-defined categories. When users click on one of these nodes in the Navigation Tree, the Display pane to the right shows its contents.

Users can right-click on a Content Item to access an Action Menu of functions to perform on the Content Item, such as editing and performing *Workflow* (see page 25) transitions.

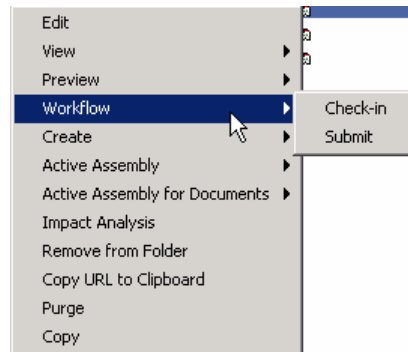


Figure 8: Action Menu

## Folders

Percussion CM System Folders in the Content Explorer navigation pane are similar visually to Folders in Microsoft Windows Explorer. While Windows Explorer folders organize files, Content Explorer Folders organize Content Items. However, underneath, CM System Folders are very different. In CM System, Folders are virtual locations for maintaining Content Items. A Content Item is not actually stored in a folder. All Content Items are stored in the CMS Repository (an RDBMS). A Content Item "in" a folder is actually linked to the Folder. Folders provide a method for users to arrange Content Items regardless of how they are stored. The same Content Item may appear in multiple Folders at the same time, since what the Folder holds is actually a link to the physical Content Item. Once an Item is altered, it is altered in the CMS as a whole, and will appear changed in all the Folders in which it appears.

The Navigation pane in Content Explorer contains two main Folder nodes, the Sites node and the Folders node. Folders in the Sites node are used to control the structure of Folders on a published Site. Administrators can use Site Folder Publishing to publish the contents of Site Folders to the same folder tree on the Site. Folders in the Folders node can be arranged in any way that is useful to the user; for example Folders could represent projects, or workgroups, or hold all content of a given Content Type.

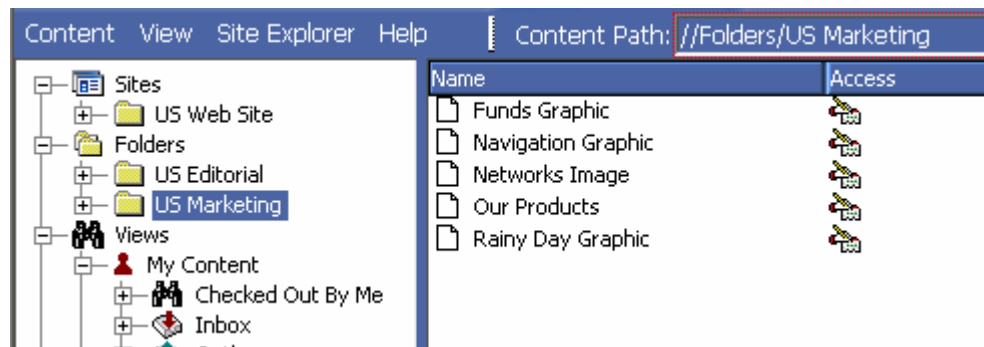


Figure 9: Folders in Content Explorer

## Views

Percussion CM System Views in the Content Explorer navigation pane display content sorted by parameters selected by CM System or a system implementer. Views are useful because they arrange content by categories that would be difficult to maintain in Folders, such as **Workflow States** (see page 25) and assigned users, and they allow users to see Content Items that are not stored in their folders. Implementers set up the views available to users; users can choose to display different views, but cannot modify any view.

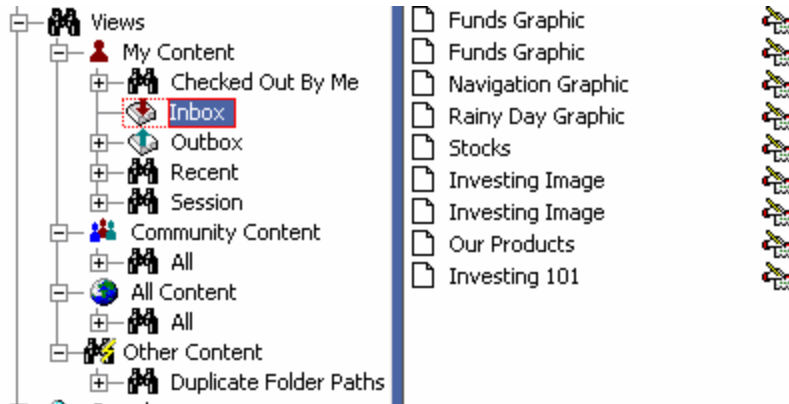


Figure 10: Content Explorer Views

## Sessions

A user begins a session by logging in to Percussion CM System. When the user logs out (or the system logs the user out) the session is finished. After logging out, a user cannot return to a previous session, but must log in again and begin a new session.

Each session takes the login username, and if enabled, the default or login Community and Locale as user session properties. CM System displays these user session properties and the login user's Role at the top of Content Explorer.

The default Locale and Community are configurable. The login Locale only displays at the top of Content Explorer with the other user session properties if the user has more than one Locale. The user can click on Community or Locale to change it for the active session.



Figure 11: User Session information in Content Explorer

Session properties determine the login user's access to Content Items and Action Menu options and other components of Content Explorer.

---

# Publishing Concepts

## Publishing

Percussion CM System can publish or deliver content to various locations including corporate internet sites, employee intranets, e-commerce applications, and print production systems.

In Content Management Systems, publishing is defined as either "coupled delivery" or "de-coupled delivery." With coupled delivery, each publishing application must directly request all content from the CMS, which stores the content to be published. With de-coupled delivery, each publishing application defines its own private content object store. The CMS transfers content from its repository into the different content object stores defined under each of the publishing applications. Whenever content in the CMS changes, the CMS publishes the appropriate changes to the publishing applications, allowing them to deliver only the appropriate content without an active connection to the CMS.

CM System uses de-coupled delivery. CM System uses a Publisher to transfer content to each delivery platform's repository or file system, or through platform specific APIs via Publisher plug-ins. CM System's decoupled delivery significantly simplifies the use of multiple delivery platforms, because each platform does not have to interface with the CMS, nor do all applications have to agree upon a common definition or schema for storing and delivering content objects.

## Sites

A Site defines a location where Percussion CM System Content Items are published. A Site may be a file system, an ftp location, or a database repository. Sites can also refer to multiple virtual locations on the same physical machine (such as a virtual Web folder or drive).

Site Folders in the Content Explorer navigation pane can dictate the structure of a Site. Administrators can use Site Folder Publishing to publish the contents of a Site Folder to the same folder tree on the Site.

## Editions

The Percussion CM System publishing process uses Editions to let administrators set up and manage the types of publishing activities that occur. Editions primarily define the content that is to be published and the Site where it is to be transferred. Publishing of an Edition begins via a schedule defined in CM System, a *workflow* (see page 25) or system action, or an end user manually initiating an Edition in the Publishing Runtime tab.

An Edition specifies the publishing Site and includes one or more Content Lists that define the content to be published. Each Content List specifies a given subset of Content Items to publish, including the sequence in which to publish them and the directory locations to which to publish them. For example, an Edition might specify that content is published to an Internet Site. Since *site folder publishing* (see "Sites" on page 23) is being used, the edition includes two Content Lists: one for publishing binary content (specifies that all binary Content Items in a Public State will be published to the Site) and one for publishing non-binary content (specifies that all non-binary Content Items in a Public State will be published).

An Edition can also specify additional tasks to be executed either before the publishing run begins or after it completes.



---

# System Concepts

## Workflow

A Percussion CM System Workflow is a set of States that a Content Item progresses through during its lifetime. Workflows often include the following States:

- development - In a development State, users create or modify a Content Item. It may be the initial State of a Content Item, a State in which a contributor regularly adds data, such as artwork, to an existing Content Item, or a State to which an approver returns a Content Item for modification.
- approval - In an approval State, administrators or Web Masters approve an existing Content Item for publication or reject it and return it to an earlier State for modification.
- public State - Content Items in a public State are ready to be published. During the next publishing run, these Content Items can be selected and published to a delivery Site. Depending on the Workflow assigned to them, Content Items in a public State may have moved through one or more approval States before reaching the public State or may have transitioned directly to the public State after being created.
- archive State - Content Items in an archive State are no longer publishable. Most Content Items in a public State reach a point at which they are no longer current; at this point, a user or automatic Transition moves them into an archive State. If a Content Item becomes unusable before it reaches a public State, it may also be transitioned to an archive State.

*Each Workflow includes Roles that define user access to content in each State, and Transitions or actions that specify how content can move from one State to another.* (see "Workflow Engine" on page 41)

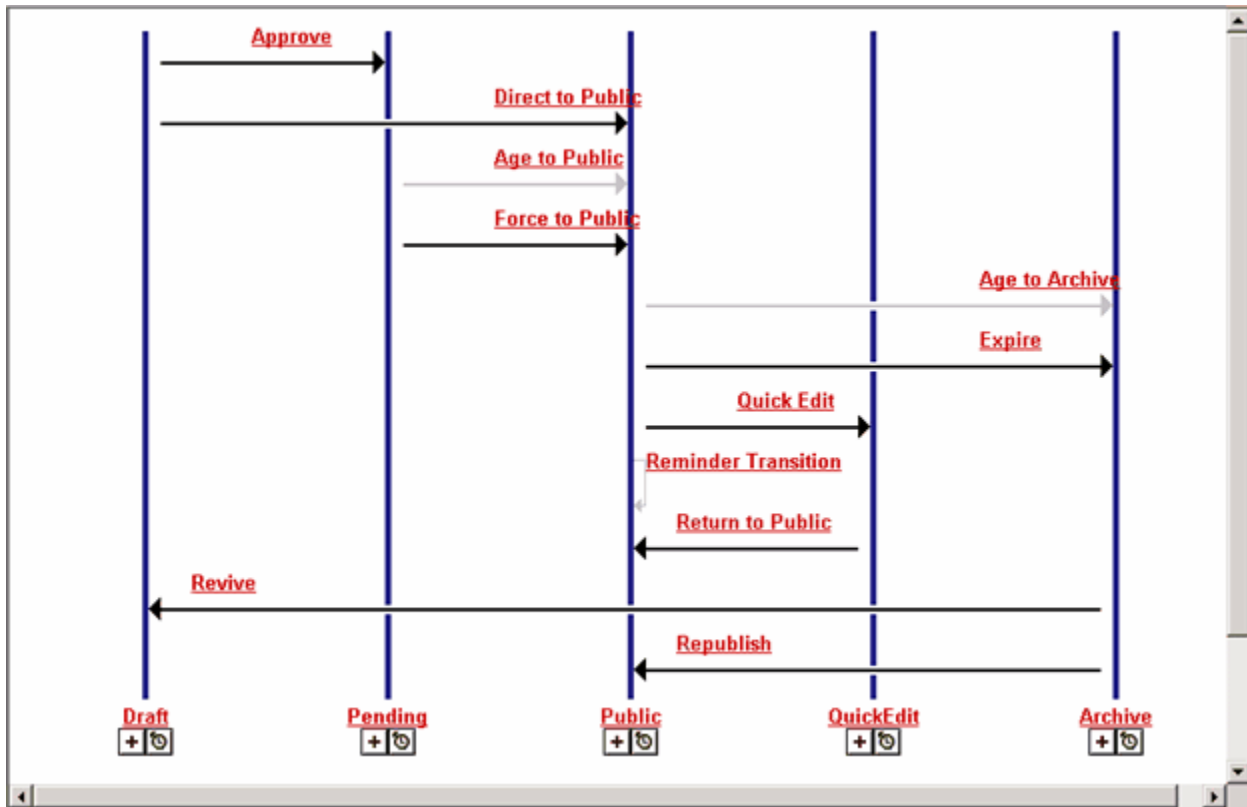


Figure 12: Workflow diagram. Each vertical line represents a State. Each horizontal arrow represents a Transition from one State to another. Transitions may be initiated by users, or by automated events, such as expiration based on date.

## Relationships

A Percussion CM System Relationship is an association between two objects in CM System (usually between Content Items). One object is the owner and the other is the dependent. For example, in an Active Assembly Relationship, an original Content Item is an owner, and a Content Item that is included in its output is a dependent; in a Translation Relationship, an original Content Item is an owner, and the same Content Item translated into another language is a dependent. A set of properties define the behavior for each type of Relationship in CM System. For example, in one type of Relationship, the owner and dependent must both be in a public State before either of them can be published; in another type of Relationship, when the dependent is published, the owner is transitioned to an archive State.

*CM System provides a developer interface for designing relationships and maintain rules governing relationship behaviors.* (see "Relationship Service" on page 37)

---

# Security Concepts

## Communities

Communities add efficiency to Percussion CM System by filtering the Content Editors, Templates, Workflows, Sites, and user interface components available to users. A Community represents a group of Roles that require access to similar information in CM System. A Role can be associated with more than one Community, and users belong to all of the Communities associated with their Roles. When users log in to the CMS interface, CM System only displays components associated with their Communities.

## Roles

A Role defines a collection of users associated with the same permissions and access in Percussion CM System. In *Workflows* (see page 25), Roles are assigned permissions by State to determine which users can access content in each State and which Transitions they can initiate. Roles also determine which portions of Content Explorer a user can view, Folder access, and other privileges in CM System. Roles are also used to define the members of a *Community* (see "Communities" on page 27).

Organizing users into Roles helps administrators manage users that have the same permissions. Instead of managing the permissions for each user, they define a Role and the permissions for it, and then assign users to it. Users assigned to that Role have the permissions specified for that Role.



## CHAPTER 3

# Percussion CM System Logical Architecture and Processing

Percussion CM System's components and their processes interact in a variety of ways. However, the majority of CM System's features and operations are associated with one of five logical services that make up the CMS:

- Content Service - Controls the creation and storage of Content Items.
- Relationship Service - Defines and processes Relationships between Content Items
- Workflow Service - Defines and controls the life cycles of Content Items.
- Assembly Service - Controls the formatting of Content Items
- Publishing Service - Implements the delivery of Content Items.

The topics in this section explain the processing of each service and the CM System components that it uses.

## Content Service

The Content Service includes the components and processes in Percussion CM System that allow users to create and store Content Items. Users can use these components in CM System to create and modify all of their content or can integrate them with outside applications such as Microsoft Word and Adobe PhotoShop to create content in other applications. CM System can upload content created in other applications through specially configured folders or one of the Connectors that CM System provides.

For information about configuring folders to upload content from other applications, see the document *Implementing WebDAV in Percussion CM System*.

### *Architecture*

The main components of the Content Service are Content Types, Content Editors, Content Items, Content Type definition files, and the backend repository tables that store local, shared, and system data for Content Items.

A Content Type defines a category of Content Item and the fields included in items of that type. It specifies the Content Editor or form used to create and modify the Content Items of that category. The Content Editor specified by the Content Type defines the body content and metadata fields for the Content Type.

A Content Type can include local, shared, and system fields which are defined in XML files. Local fields are unique to the Content Type; an implementer defines them. Shared fields are used by more than one Content Type in the CMS. System fields are used in all Content Types in the CMS. After the Content Type and therefore the Content Editor are created, CM System creates an XML file that defines all of the Content Type fields and builds a backend table for storing the values of local fields. The values for shared and system fields are also stored in backend tables.

Local, shared, and system fields may store what is traditionally distinguished as body content or metadata. However, CM System does not distinguish between body fields and metadata fields. They are defined in the same manner in Content Type definition files and backend tables, and any body or metadata field can be displayed in a Content Item's output.

Implementers create a Content Type in the Percussion CM System Workbench, specifying its Communities and Workflows during creation. After it is created, they can enter a Content Type editor to add fields and specify properties such as validations and special processing extensions (java plugins).

Users view Content Editors as Web browser forms that list and display the fields and the field contents of a Content Item. Manually entering data into a Content Editor is one way that users can create Content Items in CM System. After entering the data, the user clicks an Insert button to upload the data into the backend tables that store the Content Item. Users can then reopen the Content Items to view or edit them. Users can also create Content Items in third party applications and then upload them into CM System as specific Content Types using specially configured (WebDAV-enabled) folders or one of CM System's Connector applications. After CM System uploads these Content Items, users can open them in CM System Content Editors for viewing or editing.

A Content Item is an instance of a Content Type. It includes the body data and metadata for a particular piece of content and is stored as unformatted data in the CM System database tables representing the Content Type's local, shared, and system fields. When CM System requests the Content Item, raw data from the database fields are inserted into a virtual XML document for further processing by the CMS. When the processing is done, the contents of the updated virtual XML document are inserted into the database fields for storage.

The process of creating a Content Item in CM System can either involve a content contributor making a request to create the Content Item directly in CM System, or CM System ingesting a file or XML created in a third-party application and submitted through specially configured folders (WebDAV), a Connector, or through custom Web services API submission.

When a content contributor makes a request from Content Explorer to create a new Content Item, CM System determines the Content Type requested and generates a Web form for the Content Editor. The content contributor fills in the Web form with values for the Content Editor fields and clicks an insert button. When the insert button is clicked, CM System is directed to create the Content Item using the data entered into the form.

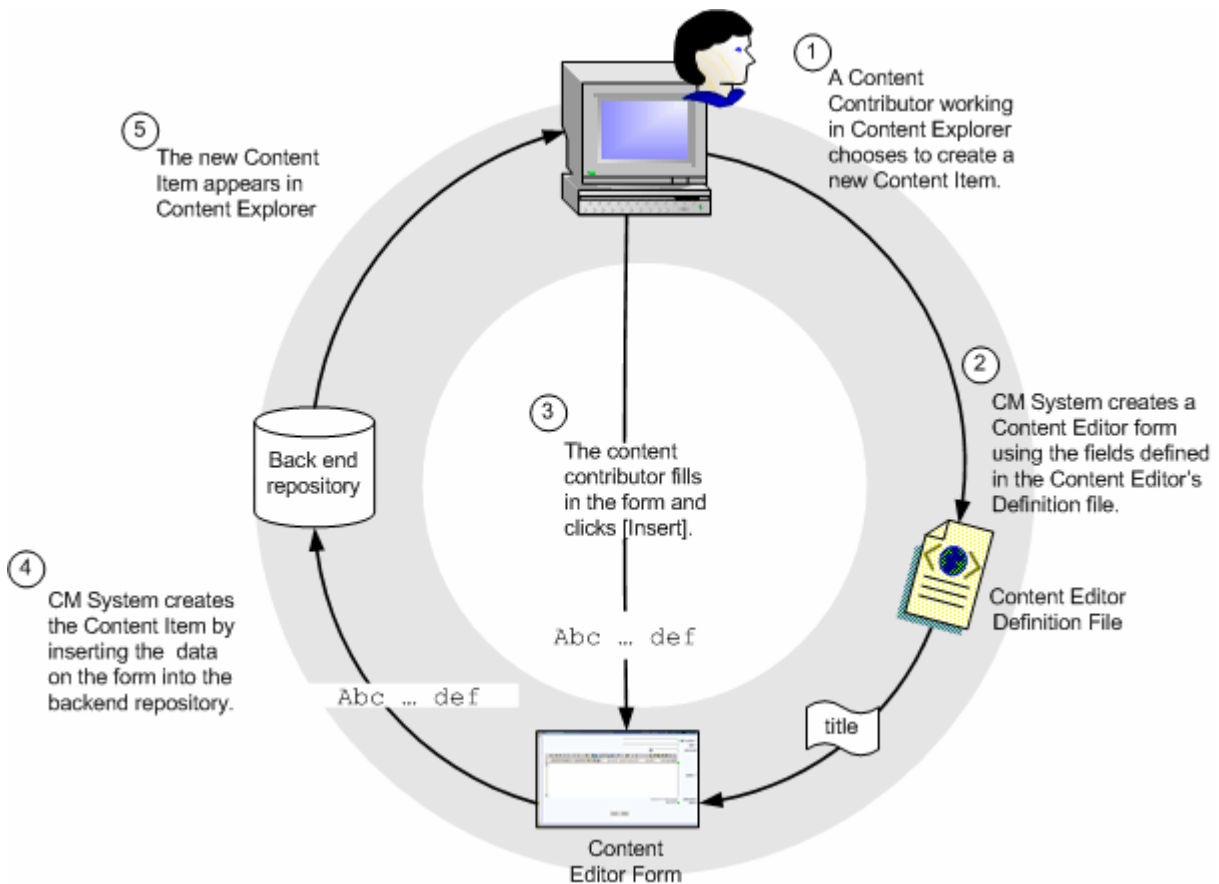


Figure 13: Creating a Content Item in Content Explorer

The process for uploading Content Items into CM System begins when a user creates a file in a third-party application. If the system uses WebDAV, the user saves the file using a WebDAV-enabled client (such as PhotoShop) or saves the file into a WebDAV-enabled folder in the underlying operating system. CM System uses mappings from a WebDAV configuration file to associate data from the file with a Content Type and to the fields defined for that Content Type. For information about implementing WebDAV in CM System, see the document *Implementing WebDAV in CM System*.

Other Connectors and a Web services API are also available for ingesting content into CM System. These all submit to the same Content Types using input field definitions and other rules from the same Content Editor definitions as are used by the Web forms and WebDAV submissions.

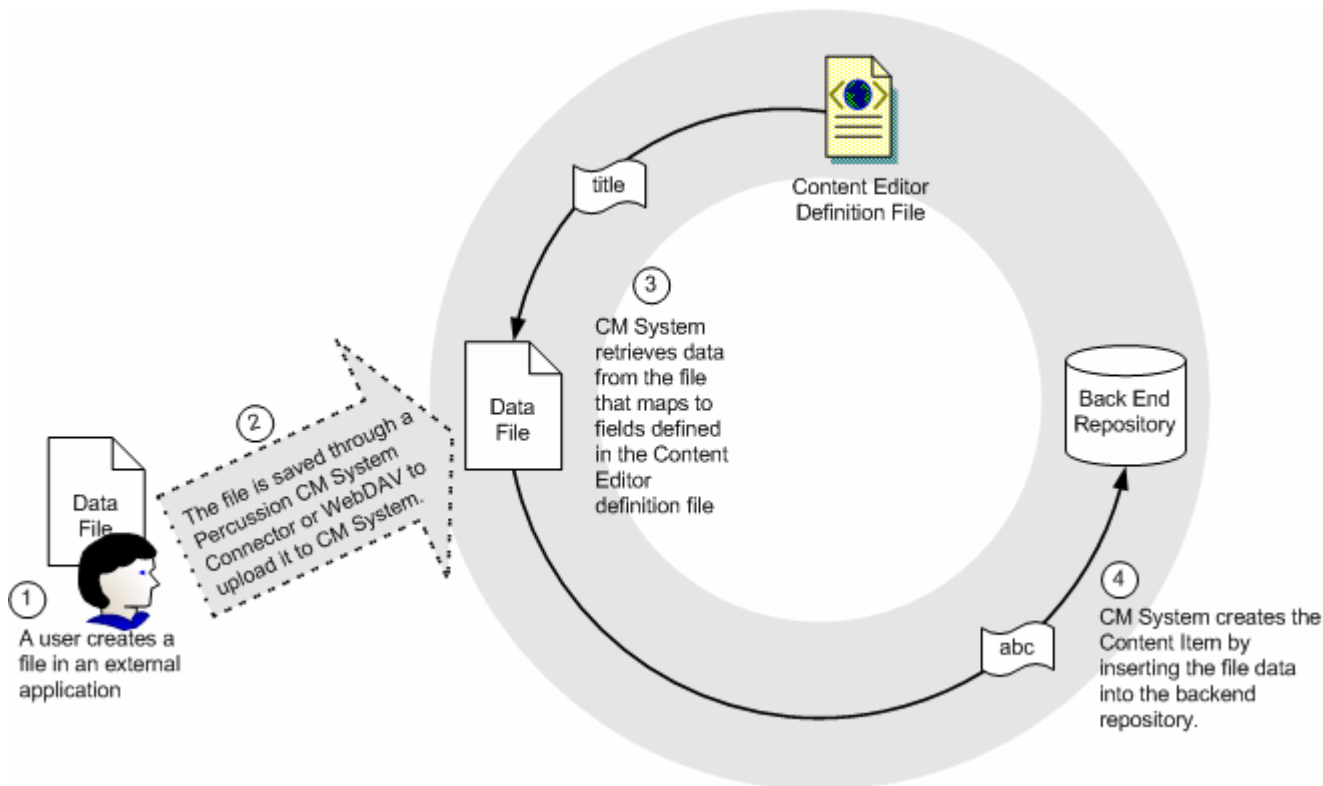


Figure 14: Creating a Content Item in an External Application

CM System uses the same components to present a Content Item for viewing or editing as it does to create a Content Item. In this process, a content contributor makes a request to view or edit the Content Item. In response to the request, CM System creates a virtual XML document using definitions from the Content Editor definition file and data from the backend repository tables for the specific Content Item. CM System applies Content Editor stylesheets that format this XML to present it in the desired form to the content contributor for viewing or editing.



If a Content Item is edited after entering a public State, a new revision of the Content Item is created and becomes the revision displayed as output. The former version of the Content Item is saved with the previous revision number, but in the future, it can be restored as the current revision that is displayed as output.

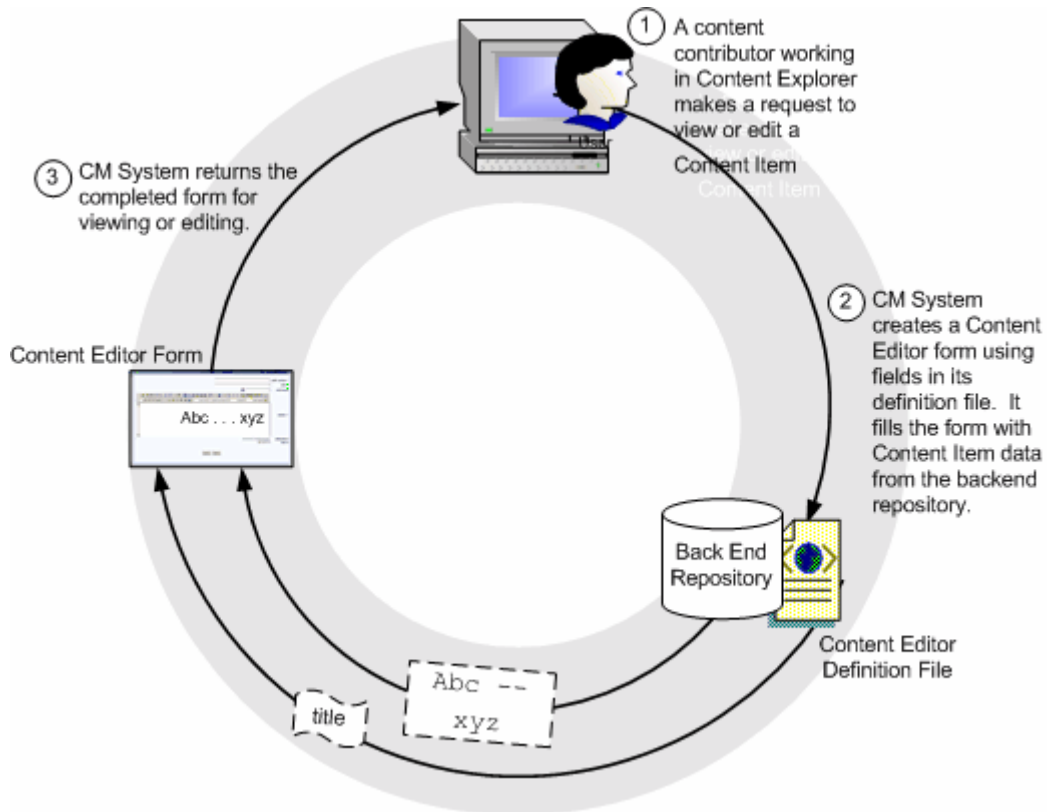


Figure 15: Viewing or Editing a Content Item

---

# Assembly Service

The Assembly Service is comprised of Templates and the Assembly plugin.

## *Architecture*

Most Templates are pages written in HTML markup and the Velocity templating language that specify where the local and related content will be added to the output page. Other Templates output binary data or choose one of a set of possible Templates that can be used to produce a final output. These Templates are defined as producing either Page output or Snippet output. Pages represent a complete Web Page and Snippets represent a portion of a Web Page.

When assembly is invoked, the Template is input to the Assembly plugin, as is the data of the requested Content Item. The Assembly plugin merges the Template formatting with the Content Item data to produce the formatted output.

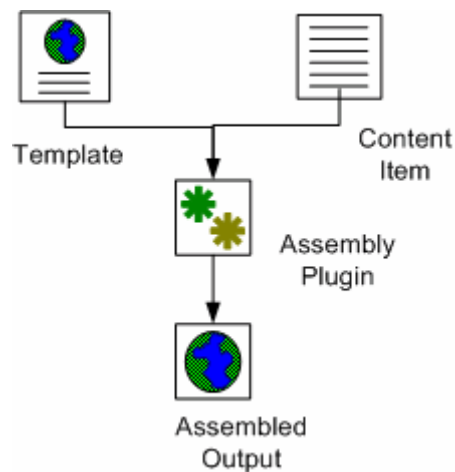


Figure 16: Components of a Template

*Processing*

To assemble a Content Item, CM System applies formatting to the item's local content and inserts Snippets into the Template's Slots. This process is referred to as recursive rollup because CM System must also format the embedded Snippet's local content and insert additional Snippets into its Slots, and format the local content of these Snippets and insert Content into their Slots, and so on. When the recursion is complete, all embedded Content is formatted and inserted and CM System returns the output Page or Snippet.

Assembly is triggered when a business user requests a preview of an assembled output of a Content Item or the Publishing service requests that a Content Item be formatted as a certain output for Publishing. When CM System receives the request, the server creates an Assembly Item, then loads the Template and Content Item data. If the Template includes any variable bindings, they are processed at this time. The Template, including binding results, and the Content Item are then passed to the Assembly plugin. The plugin merges the Template formatting with the Content Item data to produce the formatted output.

The processing is repeated each time a formatted Content Item is requested for a Slot, and the formatted output is merged into the Slot.

When the entire process is complete, the Assembly plugin returns the originally requested output for preview or processing.

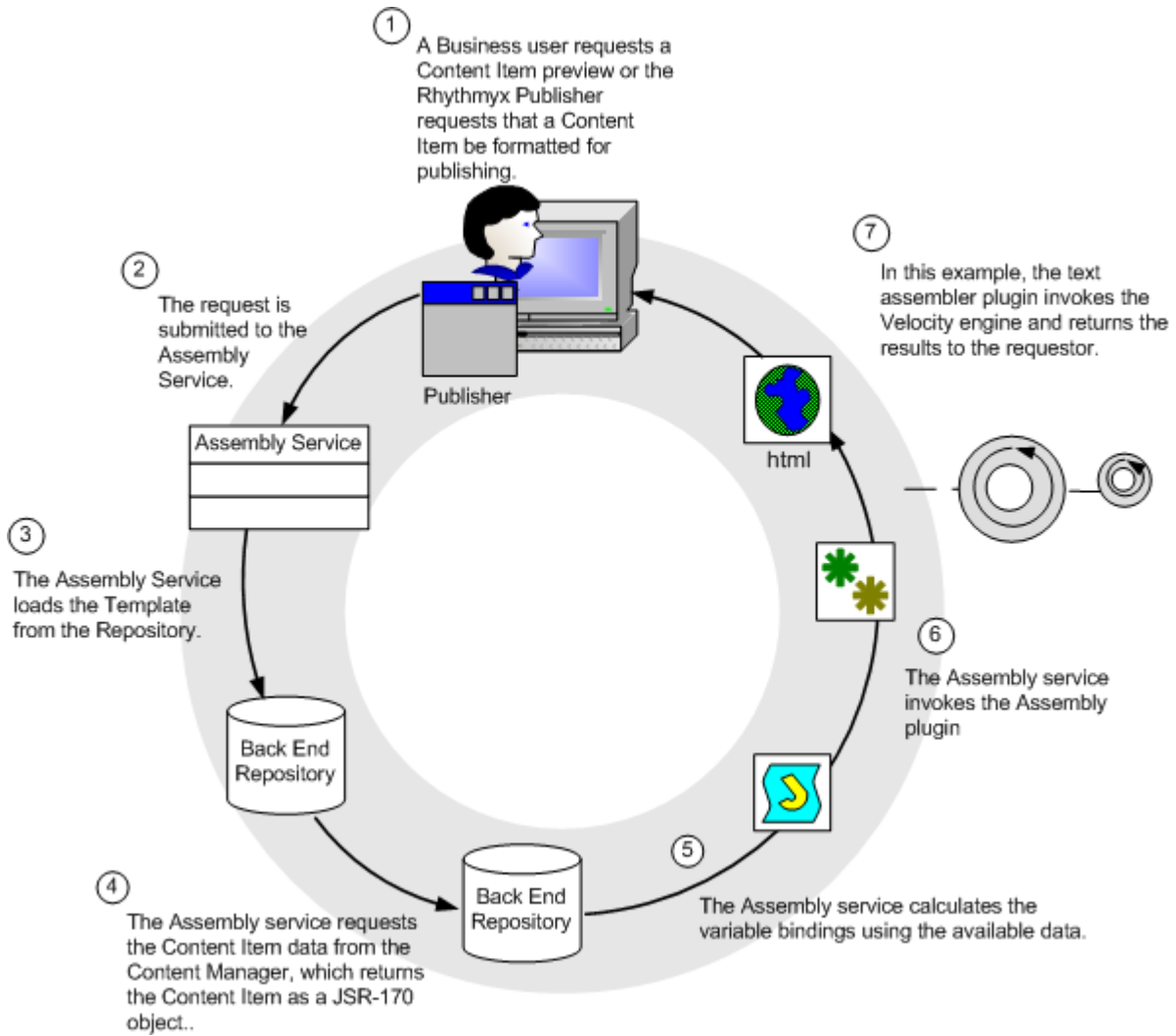


Figure 17: Assembly Process

## Relationship Service

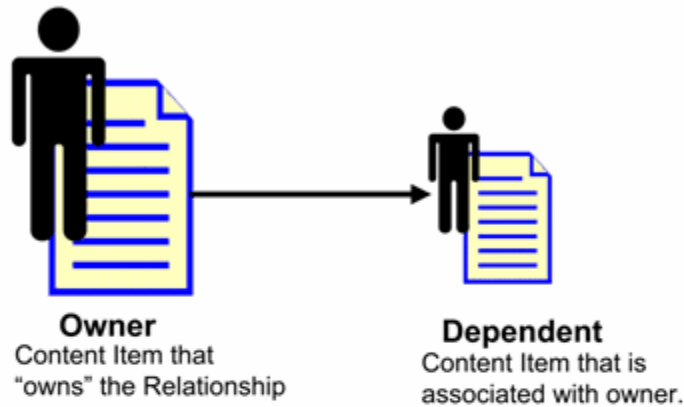
The Relationship Service ensures that CM System understands and manages all content associations and enforces all business rules defining those associations. The Relationship Service includes the interfaces for defining Relationships in CM System and the processes that implement the proper functioning of these Relationships.

### *Architecture*

A CM System Relationship is an association between two objects in CM System (usually two Content Items). One object is the owner and the other is the dependent. The dependent is associated with the owner according to rules or behaviors specified in the Relationship. Four components specify the rules and behavior of a Relationship: properties, cloning options, exits, and effects:

- **Properties** - Information defining the Relationship such as its Name, Expiration Date, and whether it is used in Active Assembly. CM System uses this information to determine actions such as when to end the Relationship.
- **Cloning Options** - Rules defining whether to duplicate the Relationship and how. For example, some Relationships may require that if the Relationship is duplicated, the dependent object must also be duplicated; others may require that just the Relationship be duplicated.
- **Exits** - Java code and scripts executed when the Relationship is created. For example, an exit may prevent CM System from creating the Relationship if an instance already exists.
- **Effects** - Java code and scripts executed when a Relationship is processed. For example, an effect may require that CM System notify certain users when the Dependent in a Relationship reaches a Public State.

## Relationship: Main Components



## Other Components

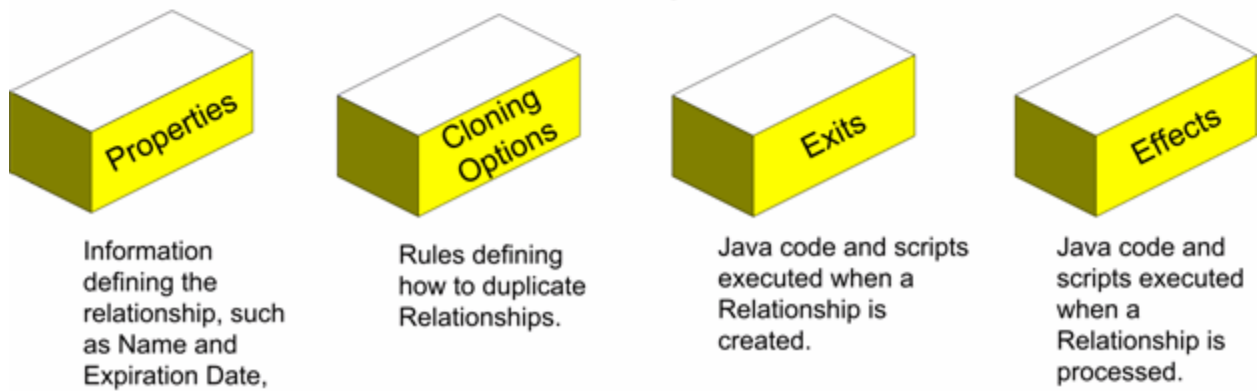


Figure 18: Relationship Architecture

### Processing

To begin understanding how Relationships work, let us look at a simple HTML page that consists of some text and a graphic:



Figure 19: A Page including text and a graphic

In CM System, this page consists of two Content Items, a Generic Page Content Item that contains the text and an Image Content Item used to manage the Image file. The two Content Items are associated through an Active Assembly Relationship.

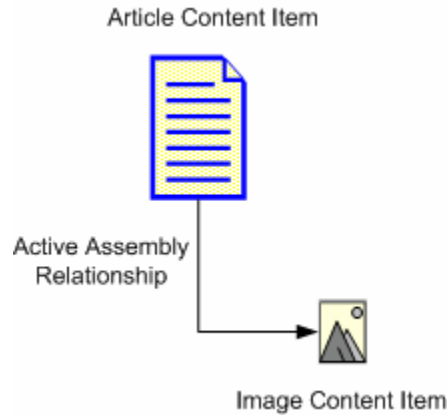


Figure 20: Active Assembly Relationship

The Active Assembly Relationship is a very basic Relationship. It simply points to a dependent Content Item to be inserted into a Slot in a specific Template of the owner Content Item. At assembly, when the individual Content Items are formatted, an HTML page is formatted with a reference to the image. The browser then renders the HTML as one page.

Now, let us suppose we want to ensure that the Article cannot go Public unless the associated graphic is also ready to go Public. The Active Assembly Relationship does not meet our needs because it does not put any constraints on the two Content Items. Each can go Public independently of the other. If the Article goes public and the image does not, the article will be published, but the Image will be removed. While this ensures non-public content will not be seen, it may make the article less compelling. In some cases, articles might depend on the image to make sense. The Relationship Service can then be used to manage these cases, where one piece of content is dependent on others. For example, to ensure that the Article cannot go Public unless the associated Image is ready to go Public, we could use the Active Assembly - Mandatory Relationship when linking the Image to the Article.

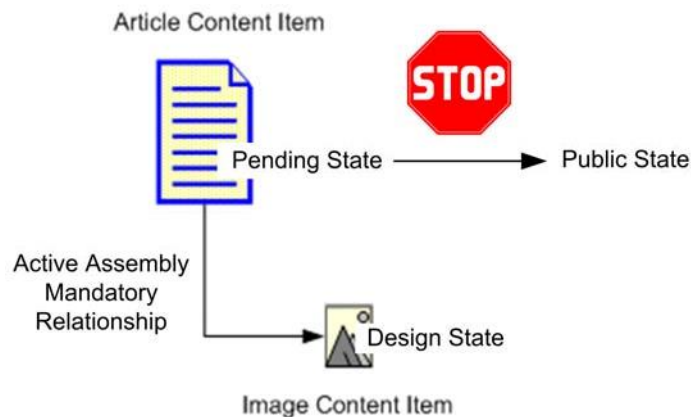


Figure 21: Active Assembly - Mandatory Relationship

The Active Assembly - Mandatory Relationship includes the Effect `sys_PublishMandatory`. This Effect prevents a Content Item from going Public if the associated dependent Content Item in the Relationship is not also Public. In this case, the Effect checks whether the Dependent is Public before allowing the Owner item to go Public. Thus, if the Image Content Item is not Public, we cannot Transition the owning Article to Public. The Article will wait in the Pending State until the Image Content Item has entered the pending State. When this Dependent enters the Pending State, we will be able to Transition the Article to Public. Other relationship Effects could be used to create different types of automation.

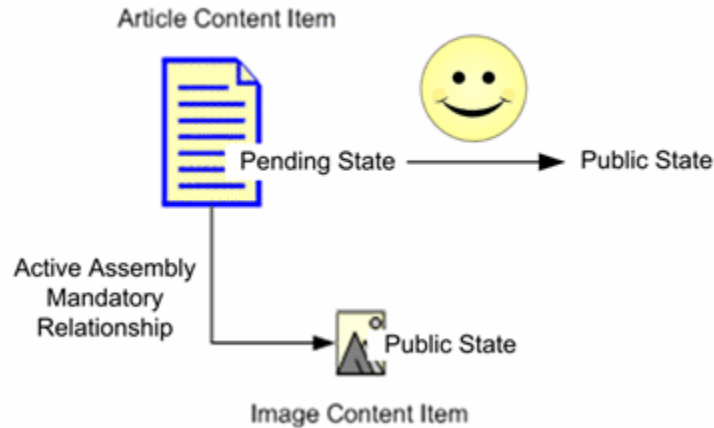


Figure 22: Transition to a public State



---

# Workflow Engine

The Workflow Service is composed of the components and procedures that define Workflows or business processes in CM System.

## *Architecture*

Workflows define the stages in the process of creating a Content Item and specify which users can access the items at each stage in the process. The Percussion CM System Publisher looks at the Workflow State to determine whether or not a Content Item is eligible to be published.

Workflows consist of States, Roles, Transitions, Transition Roles, and Notifications.

- States define the different stages of the life cycle of a Content Item.
- While a Content Item is in a certain State, only the people in Roles assigned to that State have access to the item. In Workflows, Roles should define a set of users by the function they perform in the process, such as “Author”, “Editor”, “Designer” and “Administrator”.
- Transitions define routing paths from one state to the next. There can be multiple transitions from one state to other states, and they may go forward, backward, and even "loop back" to the same state. Transitions are triggered manually when an assignee takes some defined "triggering" action on the Content Item, such as pushing the approve button or automatically, such as when a specified date is reached.
- Transition Roles specify the Roles that must perform a Transition on a Content Item before it is valid, and the number of users who must perform the Transition. For example, the Transition Role could specify that at least one person in the Editor Role and another person from the Marketing Role must perform the Transition before it is completed.
- Notifications or email messages can be assigned to certain Transitions. CM System can send the Notifications the e-mail addresses of users in Roles assigned to the originating or destination States of the Transition, as well as to arbitrary additional e-mail addresses defined for the Notification.

## Workflow Architecture

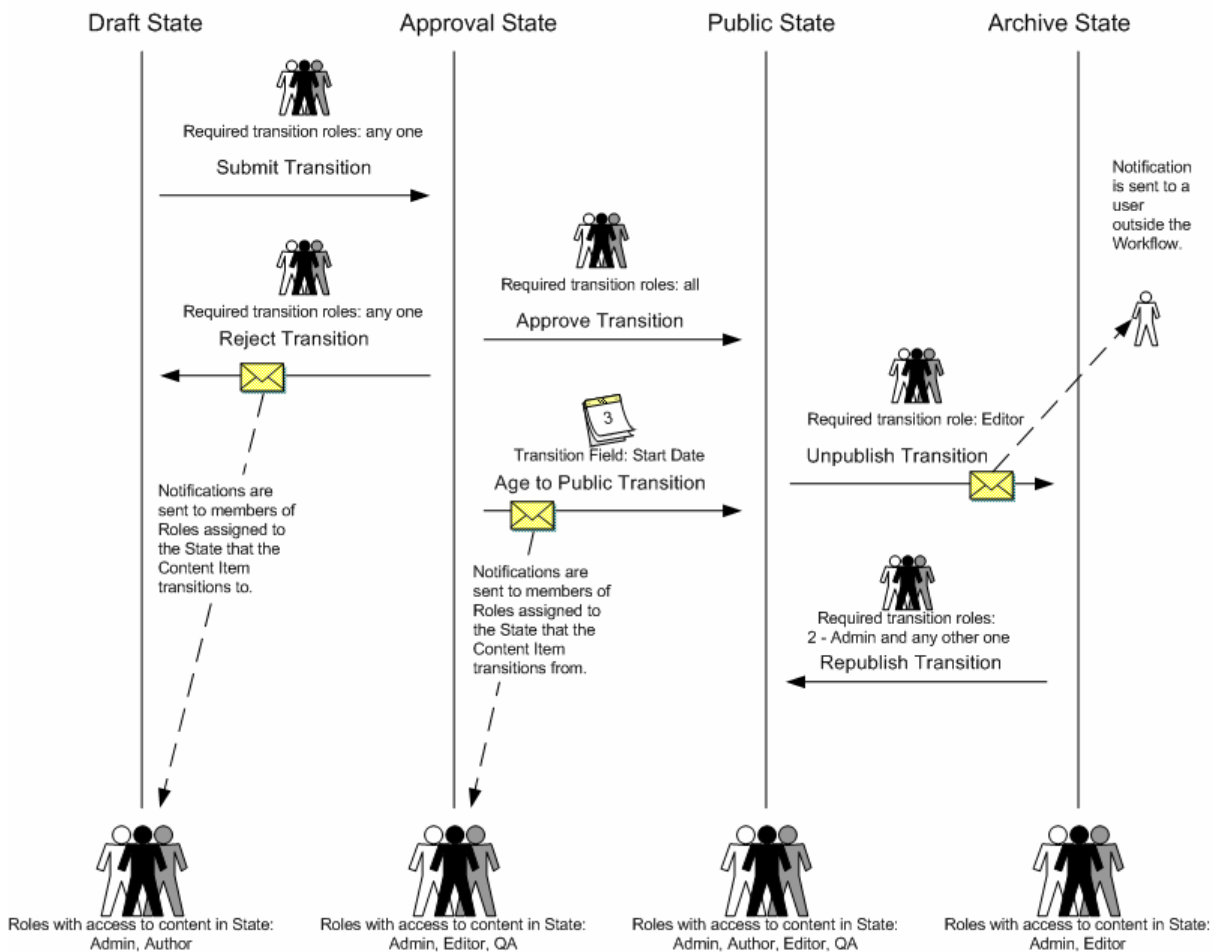


Figure 23: Workflow Engine Architecture

The Workflow architecture diagram, the vertical lines represent the States of this Workflow: Draft, Approval, Public, and Archive. The arrows coming from each State represent its Transitions. They point to the State that the Content Item transitions to. The figures at the bottom of each State represent the Roles that have access to Content Items in the State, and the figures above Transitions represent Transition Roles. Envelopes superimposed on Transition Arrows represent Notifications. Notifications point to the users who receive them when the Transition occurs.

### Processing

When a Content Item is created and saved, it automatically enters the first State in its Workflow. Either a Role with access to this State transitions the Content Item to the next State, or an Aging Transition automatically moves it to the next State when a specified date or interval is reached. The Content Item moves through each State in the Workflow in this manner until it reaches a public State. The Percussion CM System Publisher is generally configured to publish the Content Item when it reaches a public State. Once a Content Item is no longer valid or useful on a Web Site, a Role can transition it to an archive State, or it may enter an archive State through an aging Transition. Content Items in the archive State are unpublished from the Web Site and are saved for reference.

The following diagram shows the same Workflow as the diagram in the previous section, but here a Content Item's lifecycle is also recorded. The Content Item is created and saved by a user in the Author Role. Once it is saved, it enters the Draft State. The Author has access to the Draft State and performs a Submit Transition on the Content Item. Since any role may transition the Content Item, it moves to the Approval State. Since no users approve the Content Item before its Start Date is reached, on the Start Date, an Aging Transition moves the Content Item to the Public State. The Aging Transition automatically sends Notifications to all members of the Roles that had access to the Content Item in the Approval State. Eventually, the Content Item is no longer current on the Web Site. An Editor, the required Transition Role, performs an Unpublish Transition and the Content Item moves to the Archive State. The Unpublish Transition sends a Notification to the company auditor that the Content Item is in the Archive State.

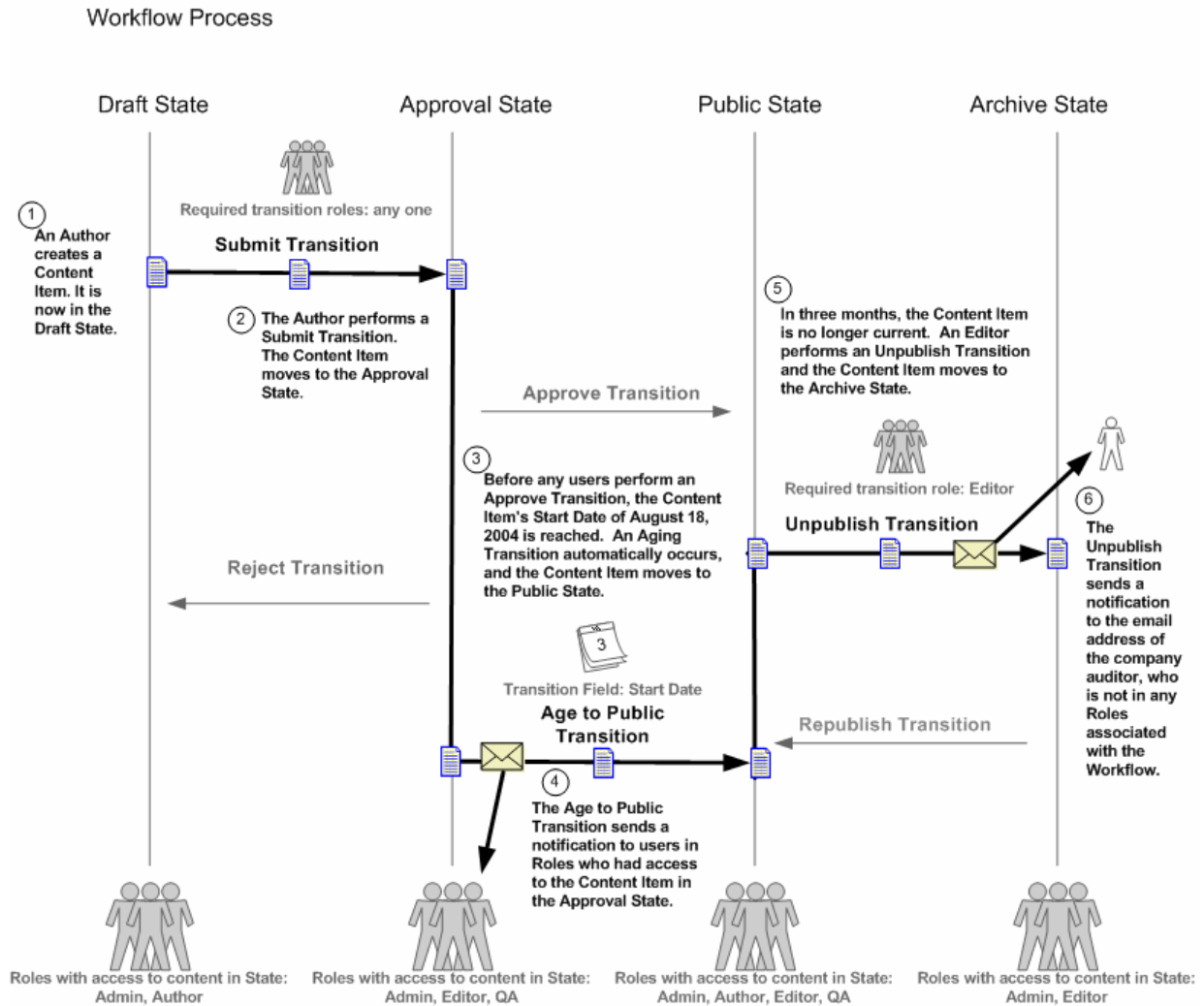


Figure 24: Workflow Processing

## Publishing Engine

The Publishing Service consists of the Sites, Editions, Content Lists, Publishers, applications, and Java code responsible for the delivery of CM System Content Items.

### Architecture

The main components of the Publishing service are the publishing subsystem of the CM Server, Sites, Content Lists, and Editions:

- A Content List specifies which Content Items to extract from the database for Publishing.
- An Edition specifies one or more Content Lists to publish and the order in which to publish them.
- The Publishing subsystem of the CM Server processes the Content Lists in the Edition to define a set of Content Items to publish. For each of these Content Items, it invokes the Assembly service to produce an assembled output, which it then sends to the destination.
- A Site defines a location where output will be published. A Site may be a file system or a database or some other destination (for example, a Portal). CM System can maintain multiple Sites on the same machine.

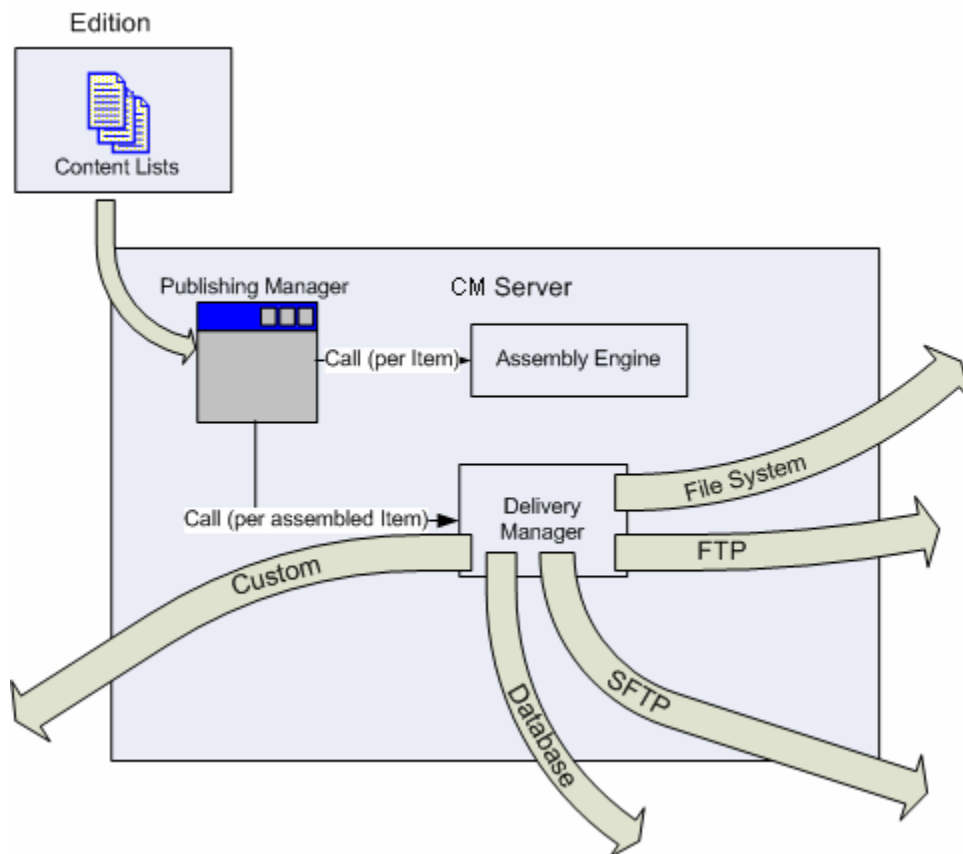


Figure 25: Publishing Engine

### *Processing*

Publishing can be initiated in one of two ways:

- A user can manually initiate a publishing request by clicking the **[Publish]** button next to an Edition on the Editions page.
- Publishing may be initiated automatically be a scheduled task in the CM Server. When Publishing is initiated, an Edition is submitted to the Publishing Manager in the publishing subsystem of the CM Server. The Publishing Manager processes the Content Lists in the Edition to determine the list of Content Items to publish. For each Content Item in the resulting list, the Publishing manager submits an assembly request to the Assembly service. When the assembled Content Item is returned, the Publishing Manager calls the Delivery Manager, which sends the completed Content Item to the output location specified in the Site.



## CHAPTER 4

# Clients and Interfaces

CM System provides several clients and interfaces that enable users to perform the functions particular to their Roles. In the Workbench, implementers create and arrange the majority of the components in their CM System, including objects used for content creation, content assembly, user interface design, and Community and Role definition. In the Server Administrator, administrators perform functions such as configuring back end connections and setting up security. Business Users work in the Content Explorer interface to create CM System content. Implementers use the Multi-Server Manager to move CM System components from one server to another, and they customize code with Web Services when sharing information with other applications on the Web.

---

# Workbench

The Workbench is the CM System environment in which implementers:

- Create and maintain the majority of components in their CM System, including Content Types, Templates, and Slots;
- Associate objects with Communities, and assign object security in general;
- Establish connections between objects (for example, which Templates are available to a Content Type).



Beginning with Version 6.0, the Percussion CM System Workbench runs as a plug-in to the Eclipse environment. Eclipse is an open source development environment that runs on various platforms and is extensible through plug-ins. For users who are currently using Eclipse, the CM System Workbench can run as an Eclipse plugin. Even if only the Workbench is visible to your users, various Eclipse features are present and available.

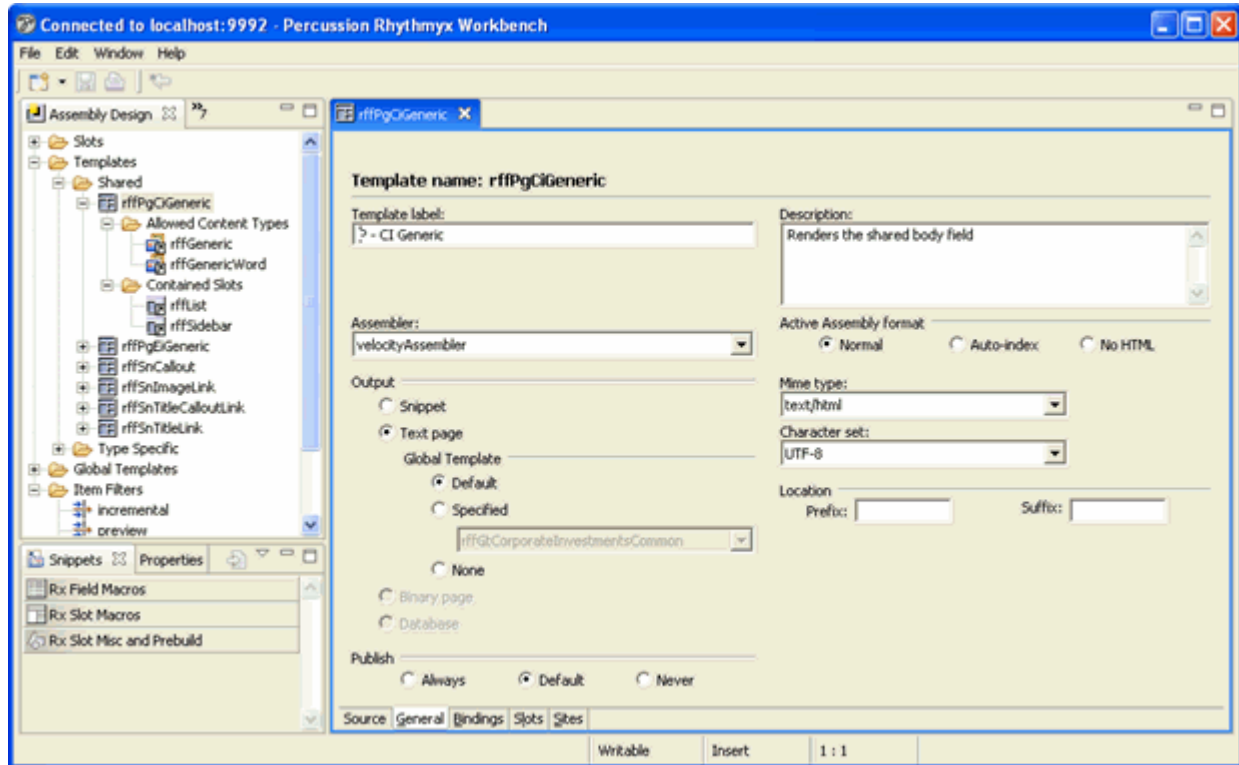



Figure 26: Rhythmyx Workbench

The CM System Workbench always displays a perspective, which is a particular selection of Workbench screens that you can view through Eclipse. The default perspective is shown in the graphic above. The default CM System perspective includes five main CMS views or windows which represent the structure of the CM System. To see all of these views and others, you would have to stretch the Assembly Design view, in the graphic to the right.

The five main views present a model of the CMS in which various CMS elements fit into the categories. Some of the main elements in each category are listed beside them in the bulleted list, below. A top-level folder appears for each element, and below it are sub-folders or objects for the element. From the views, implementers can add, modify and delete the objects they contain.

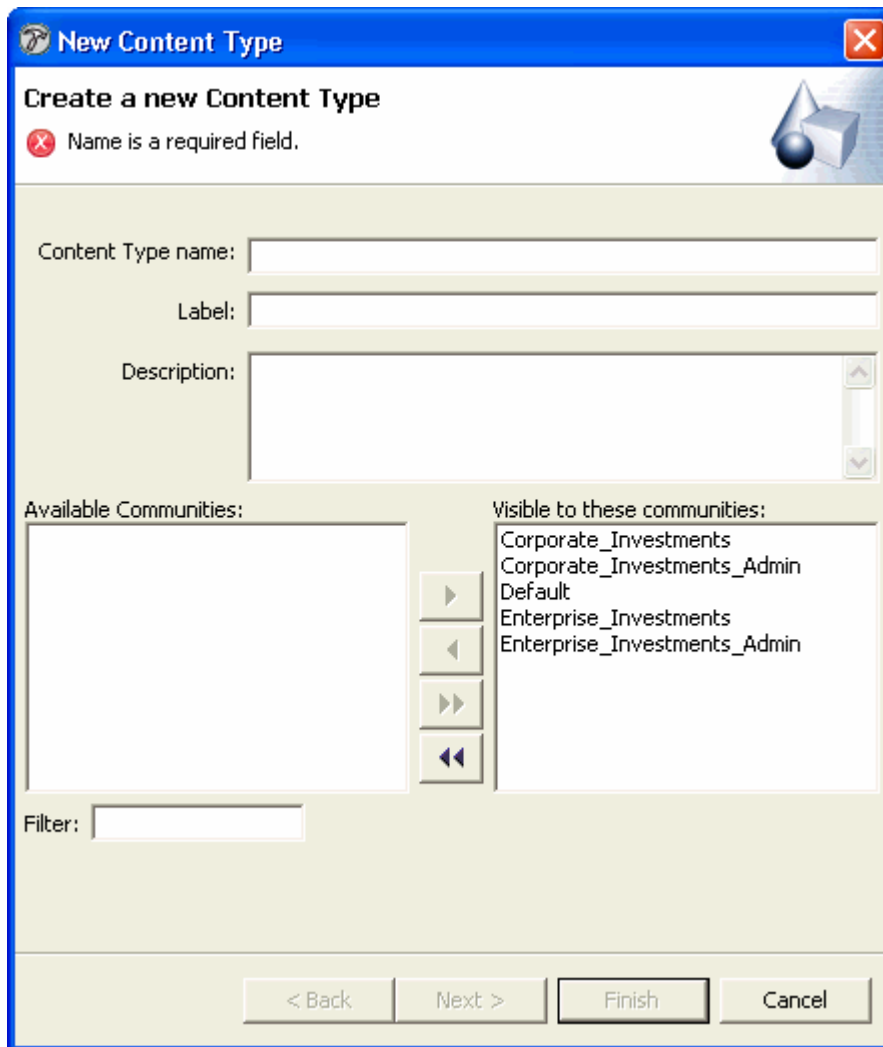
- Content Design - Content Types, Keywords, Shared Fields
- Assembly Design - Slots, Templates
- UI Elements Design - Content Editor, Content Explorer
- Security Design - Roles, Communities
- System Design - Relationship Types, CMS Files, Extensions

Views display CMS objects in a manner which displays their relationships to one another. Note that some of the views contain references to objects in other views rather than the objects themselves. References are indicated by the  icon.

The Workbench includes views other than the five main views, such as the Community Visibility view, which enables you to see all the objects available to each Community, and the Object Sorter view, which makes it easier to find an object by displaying all of the embedded objects under a selected node at the same hierarchical level.


Creating a new object in the Workbench usually involves a Wizard and an Editor, and modifying an object requires you to open it in the editor. To delete an object, simply access a right-click menu for the object and choose the delete option.

A Workbench wizard opens in a separate window from the Workbench, and depending on the amount of information required, consists of one dialog or a sequence of dialogs. When the correct information is entered, the **[Finish]** button is enabled. Clicking **[Finish]** creates the object in the appropriate Workbench view, closes the wizard, and causes the editor for the object to open in a window on the Workbench. The editor may include fields that are not available in the wizard.



**New Content Type**

**Create a new Content Type**

 Name is a required field.

Content Type name:

Label:

Description:

Available Communities:

Visible to these communities:

- Corporate\_Investments
- Corporate\_Investments\_Admin
- Default
- Enterprise\_Investments
- Enterprise\_Investments\_Admin

Filter:

< Back   Next >   Finish   Cancel

Figure 27: New Content Type Wizard, Screen Title page

CM System Workbench editors may consist of a single window or any number of tabs. The editor for an object may duplicate the information entered into the object's wizard, or it may include additional optional or required fields. A box at the top or side of each window may include instructions and display error information if invalid or insufficient data is entered.

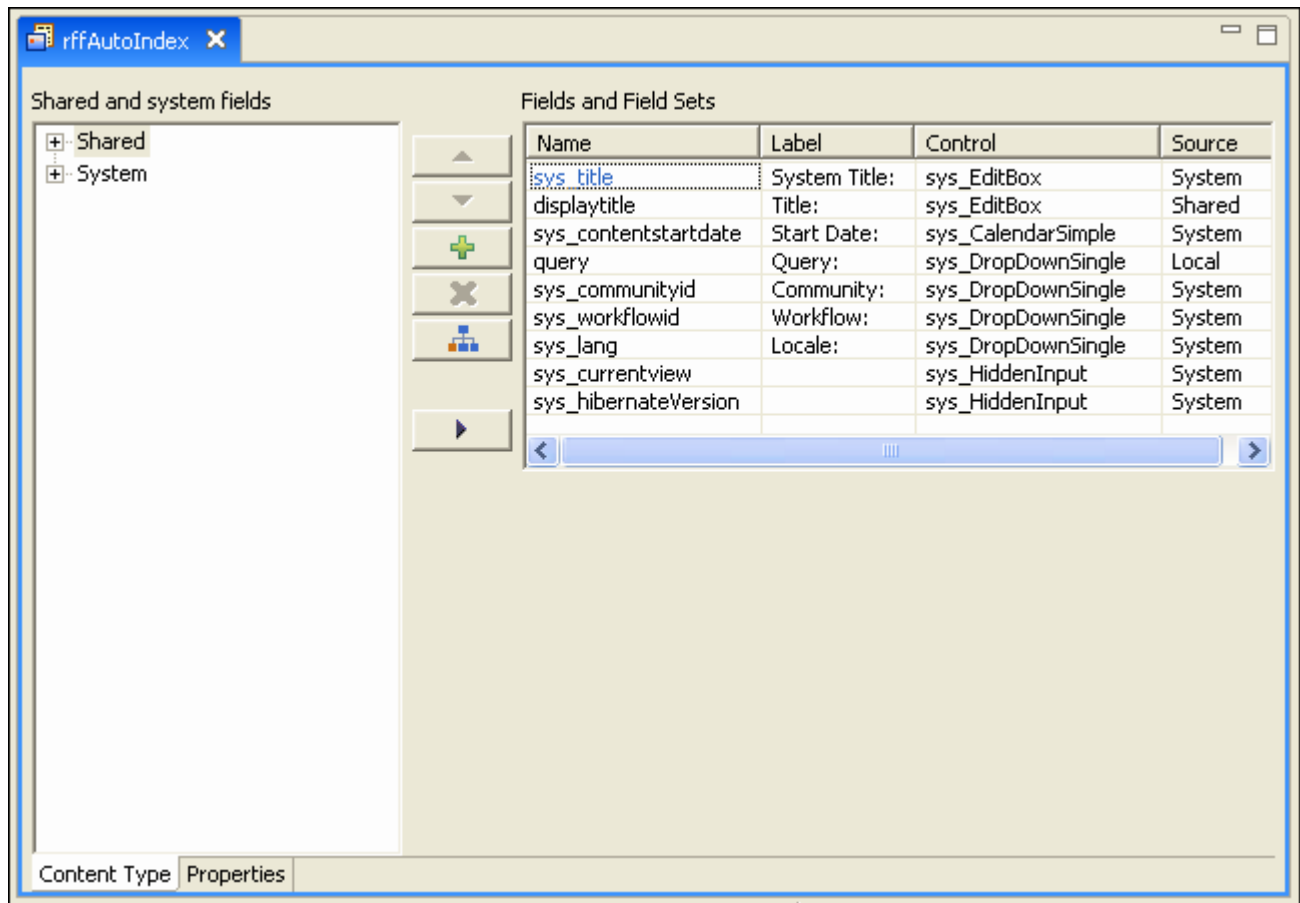


Figure 28: Field Editor

# Server Administrator

The Server Administrator is the CM System interface that the System Administrator uses to maintain the CM Server.

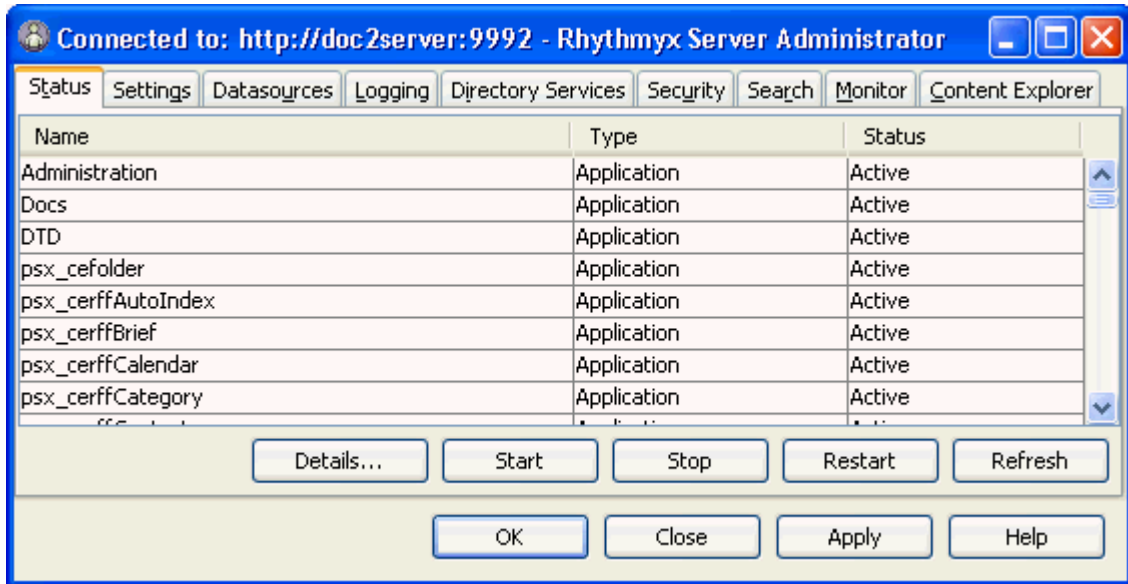


Figure 29: Status Tab

The Server Administrator separates administration tasks into nine categories displayed by its upper tabs.

## Status

In the Status tab, an administrator can stop and start CM System applications and monitor their statistics.

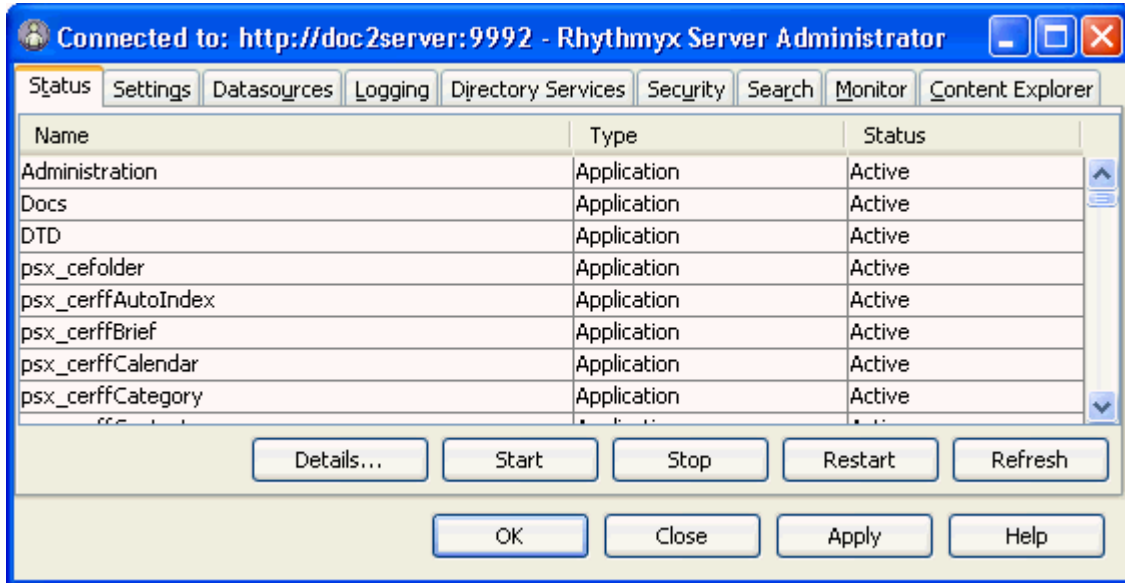


Figure 30: Status Tab

## Settings

The Settings tab lets administrators optimize CM System's performance by specifying idle time limits and maximum connections. In addition, this tab lets the administrator enable or disable the Server Cache and set its size.

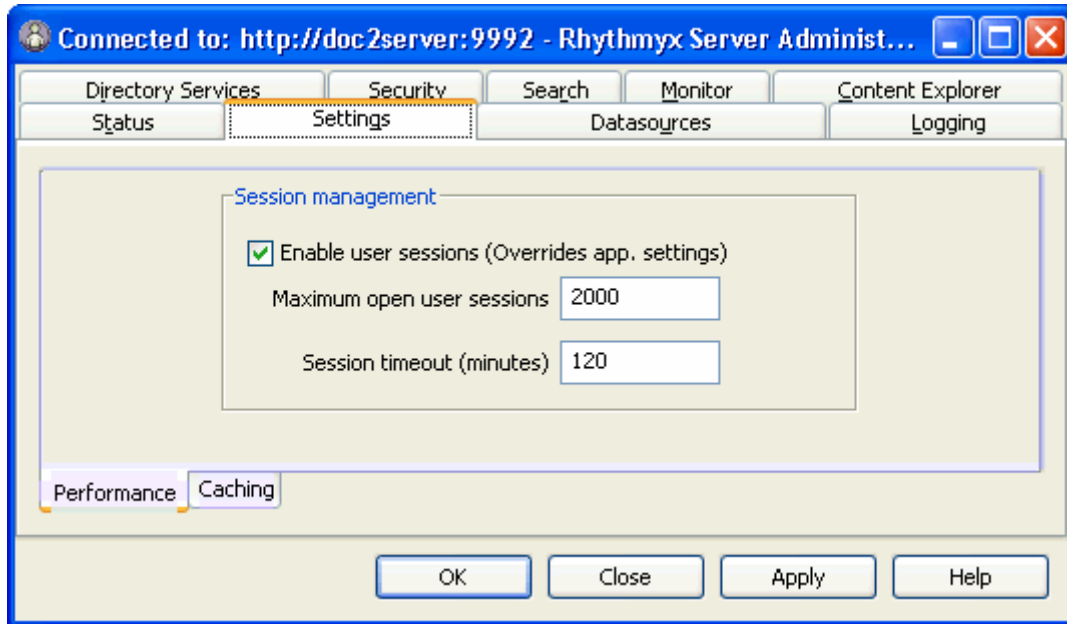


Figure 31: Settings Tab

## Datasources

In the Datasources tab, an administrator can maintain the data CM System uses to connect to an RDBMS and to a specific database or schema in the RDBMS. The administrator can maintain database driver definitions, JNDI datasource configurations, and specify the database or schema to which CM System can connect.

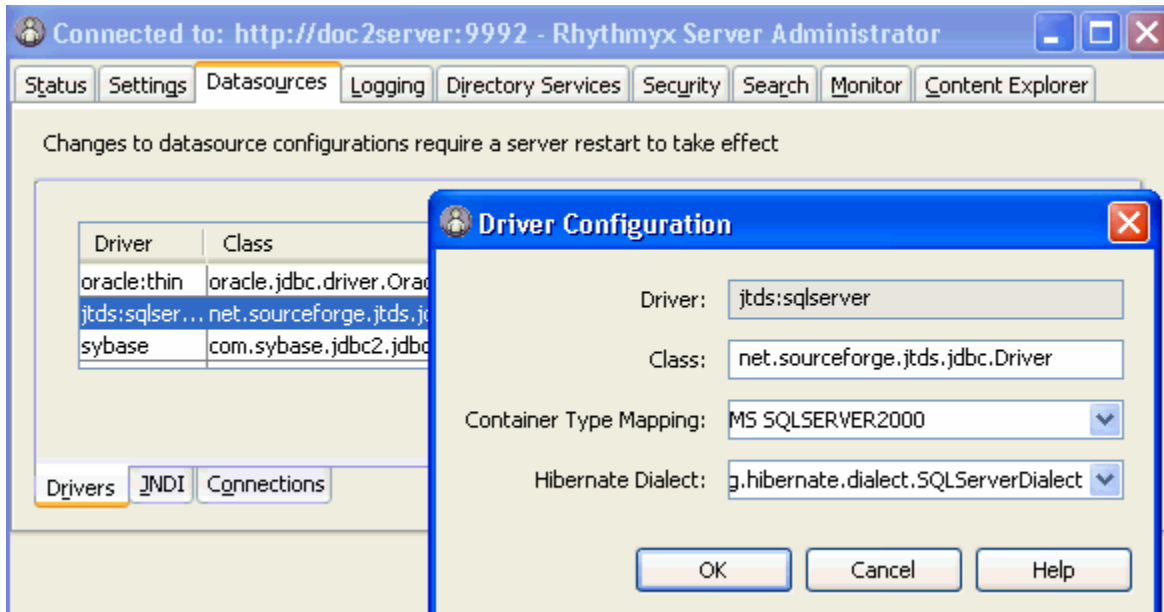


Figure 32: Datasources Tab

## Logging

In the Logging tab, an administrator can set the types of events to log, specify how long to save log files, and query log records for information.

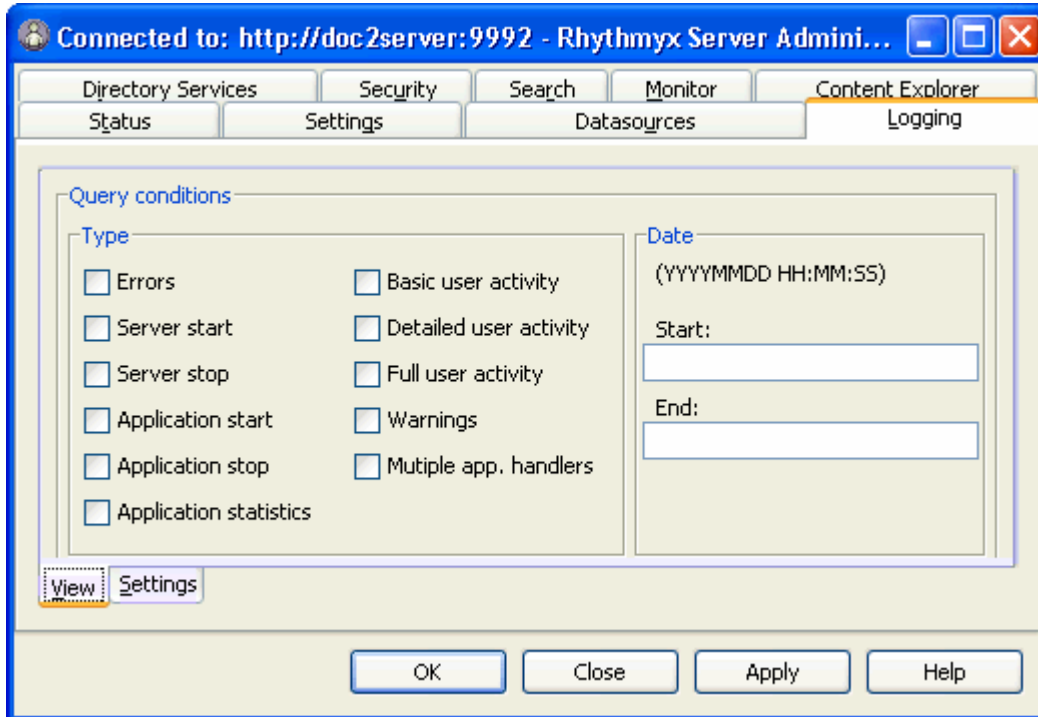


Figure 33: Logging Tab



## Directory Services

Directory Servers are third-party systems that store various types of user data. In CM System, Directory Services can provide authentication. In the Directory Services tab of the Server Administrator, administrators can register Directory Services, add authentication information for users connecting to a Directory Service, and provide other data necessary for connecting to and using Directory Services.

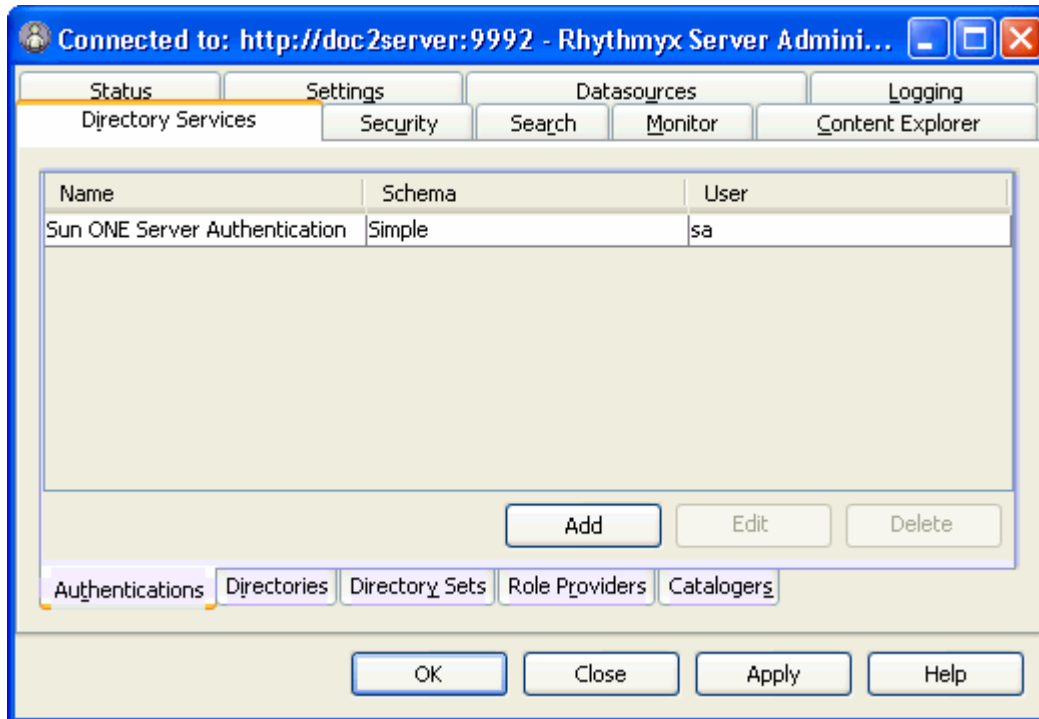


Figure 34: Directory Services tab

## Security

The Security tab allows administrators to set up most of CM System's security features. It includes sub-tabs for configuring Security Providers, Roles, and Server ACLs.

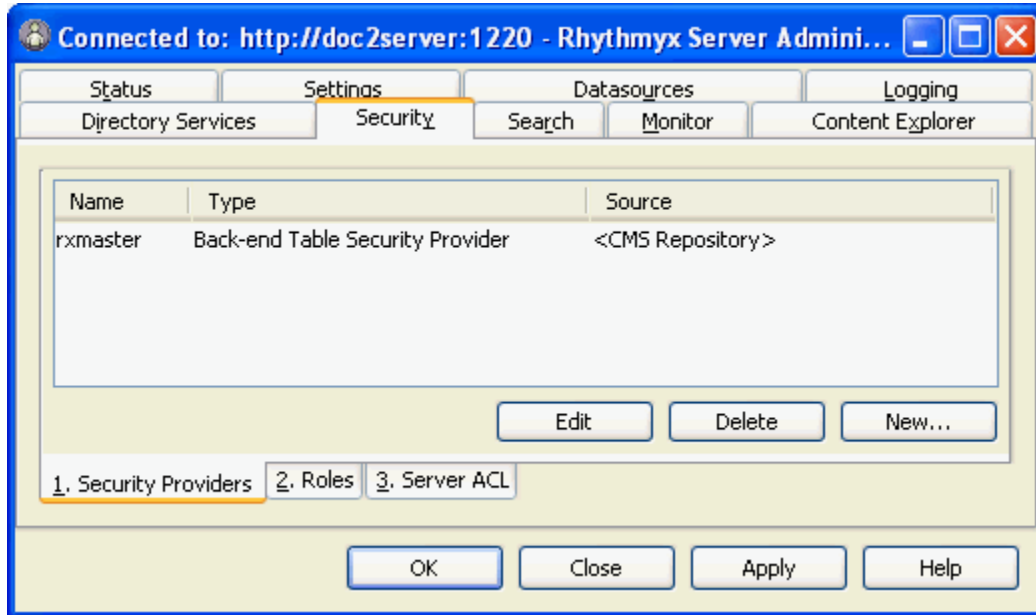


Figure 35: Security tab

In the Security Providers sub-tab, administrators register security providers and associate them with servers or users. Security providers perform authentication for servers. For more information about them, see **Data Protection** (on page 83).

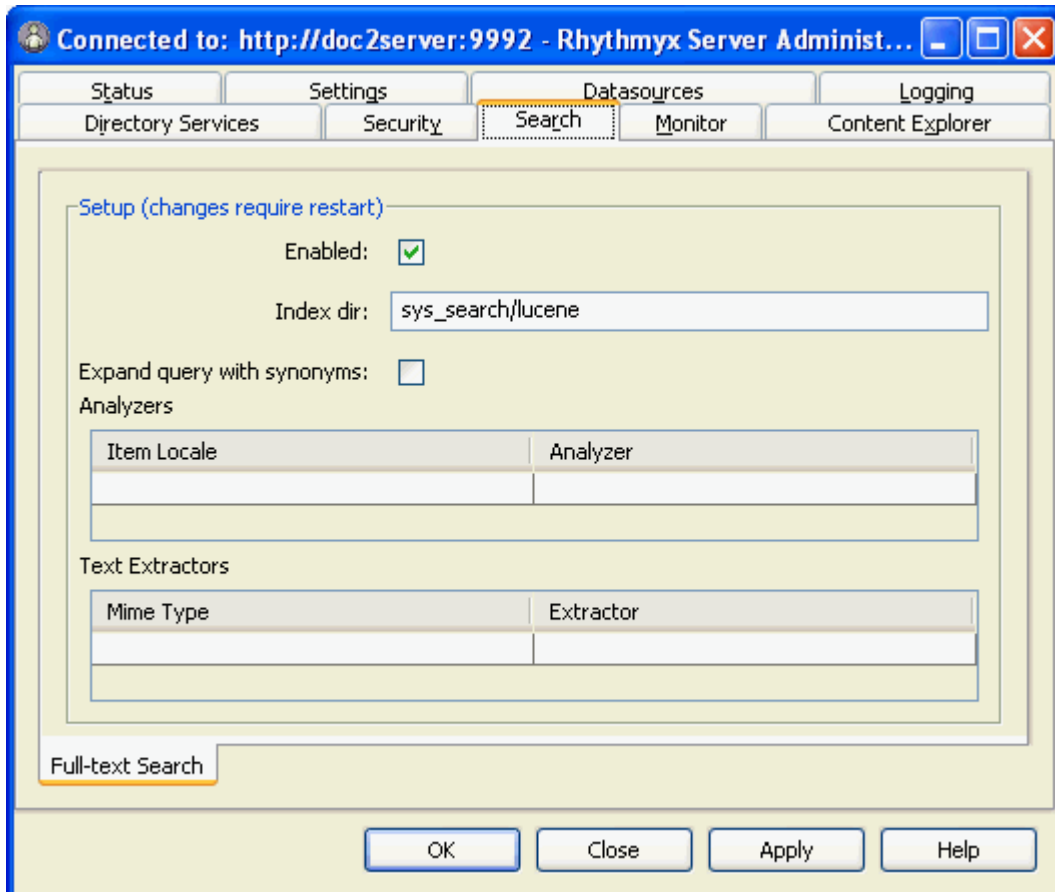
In the Roles sub-tab, administrators add or edit CM System Roles and their members on the CM Server. Roles must be defined here before Workflows and Communities can use them.

In the Server ACL sub-tab, administrators define the level of access users and Roles have to the CM Server.

## Search

CM System uses the Lucene Search Engine, which offers users advanced search features such as the ability to search content in content item fields and the ability to search for synonyms of search terms.

The Search tab allows administrators to change or override the default settings for CM System's full-text search engine on the Full-text Search sub-tab.



The Full-text search sub-tab allows you to enable and disable the full-text search, change the default index directory and override the default indexing interfaces.

For more information, see the section “Search Configuration” in the Server Administrator online help.

## Monitor

The Monitor tab lets administrators who are accessing a CM Server remotely enter Server commands and view the CM Server response on a console.

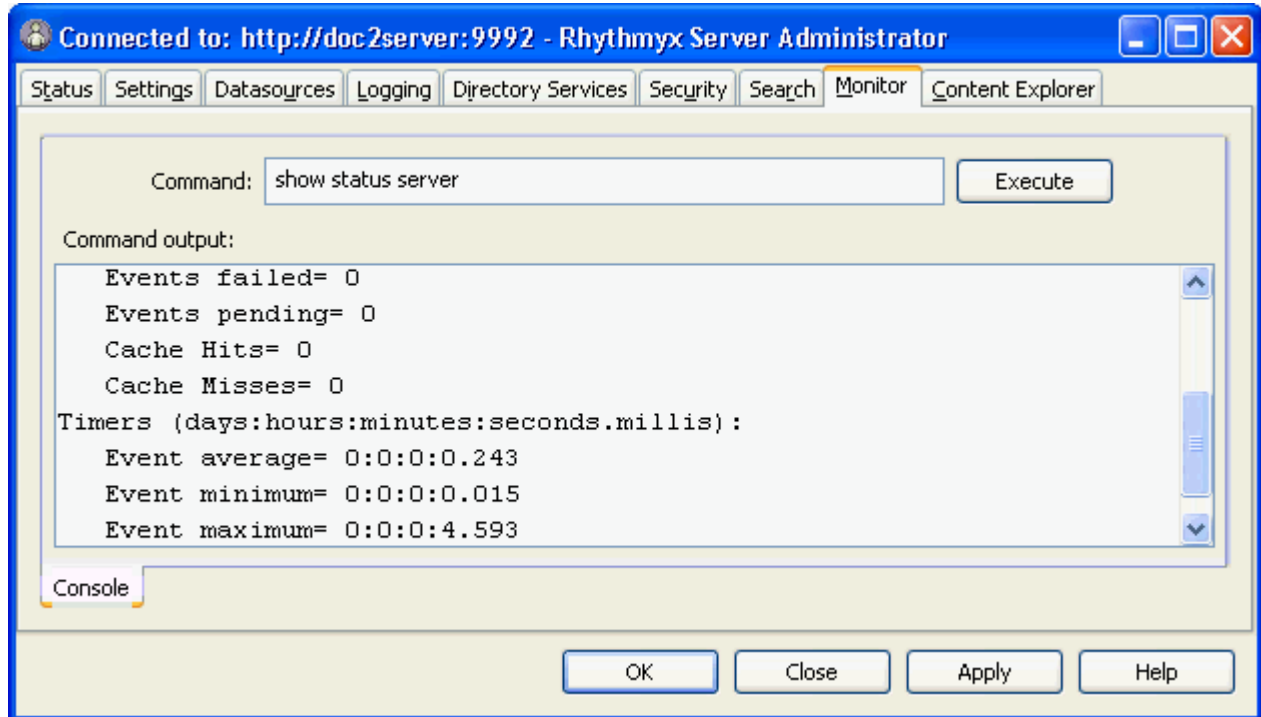


Figure 36: Monitor Tab

## Content Explorer

In the Content Explorer tab, administrators can configure which Java Plugin Content Explorer uses and its download location, and choose when to refresh the Content Explorer screen.

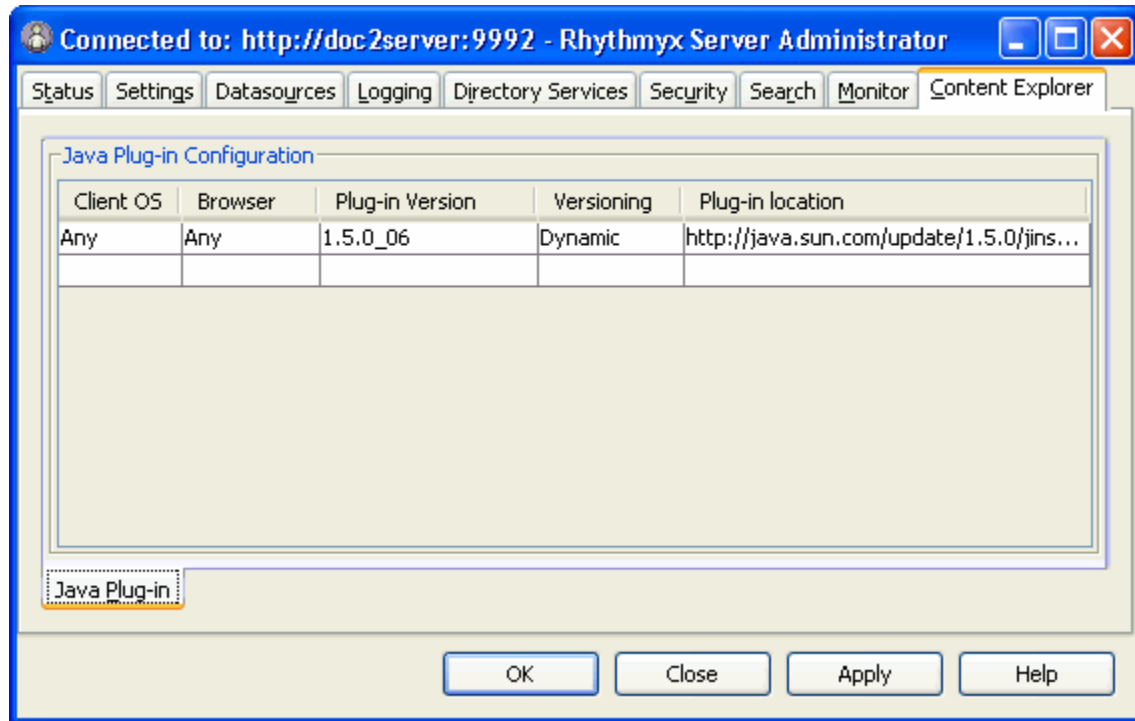
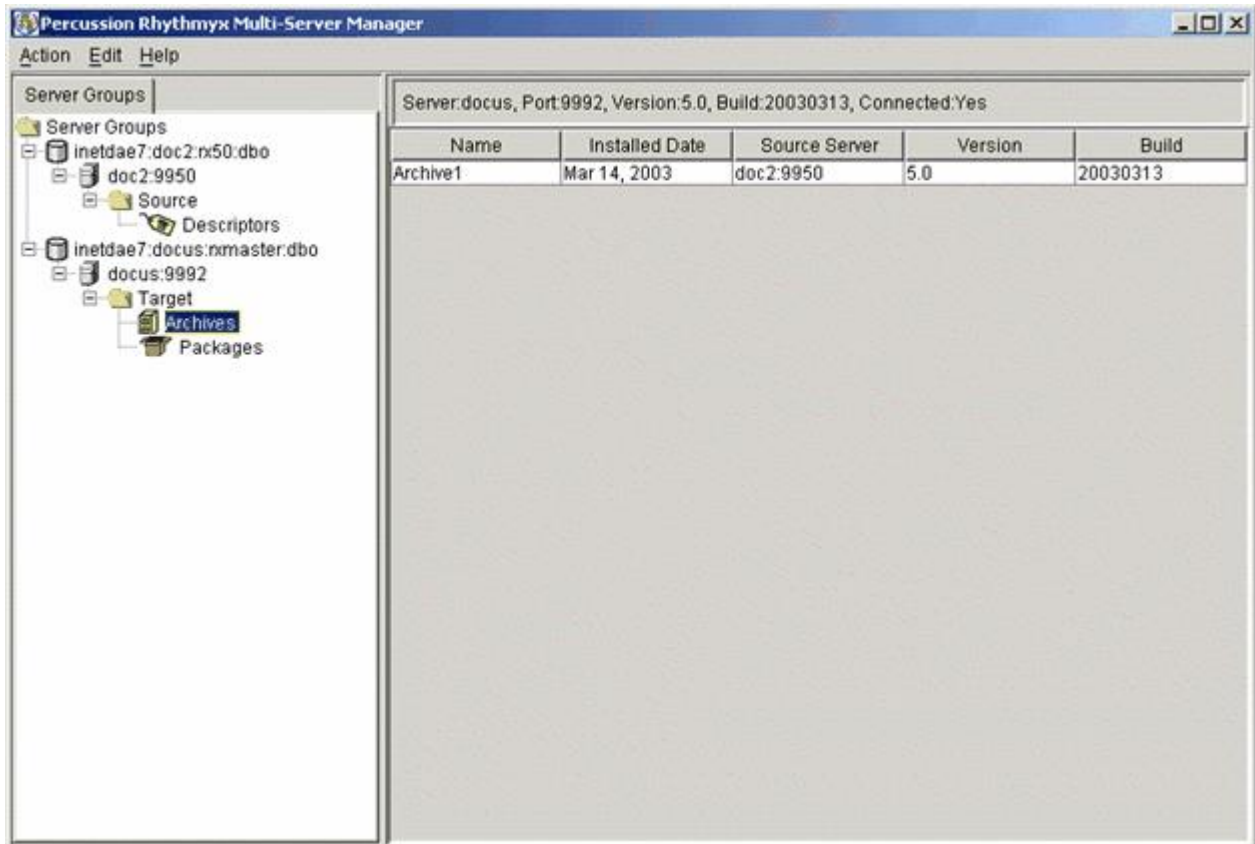


Figure 37: Content Explorer Tab

## Multi-Server Manager

Percussion CM System implementers use the Multi-Server Manager to move a CM System or portions of a CM System from a source server to a target server.



*Figure 38: Rhythmyx Multi-Server Manager*

The Multi-Server Manager guides users through each step of the installation process including selecting elements to install, packaging the elements in an archive file, and installing the archive file's contents on the target server.

The parts of a CMS that developers can install using the Multi-Server Manager include Action Menus, Templates, Content Items, Content Lists, Content Types, Folders, Locales, Relationships, Sites, Slots, and Workflows.

The Multi-Server Manager is a stand-alone tool that functions as a client and interacts with the target and source servers. The CM System Installer loads it with other client tools into the CM System root directory.

# Content Explorer

Content Explorer is the default home page when users log into CM System. It is the interface through which end users create, modify, and process Content Items. The Content Explorer interface appears and functions similarly to the Windows Explorer interface: the left pane displays a Navigation Tree and the right pane is a Display pane that lists the contents of the node selected in the Navigation Tree. The Content Explorer also includes a Menu Bar with drop menus, and a banner with user session properties.

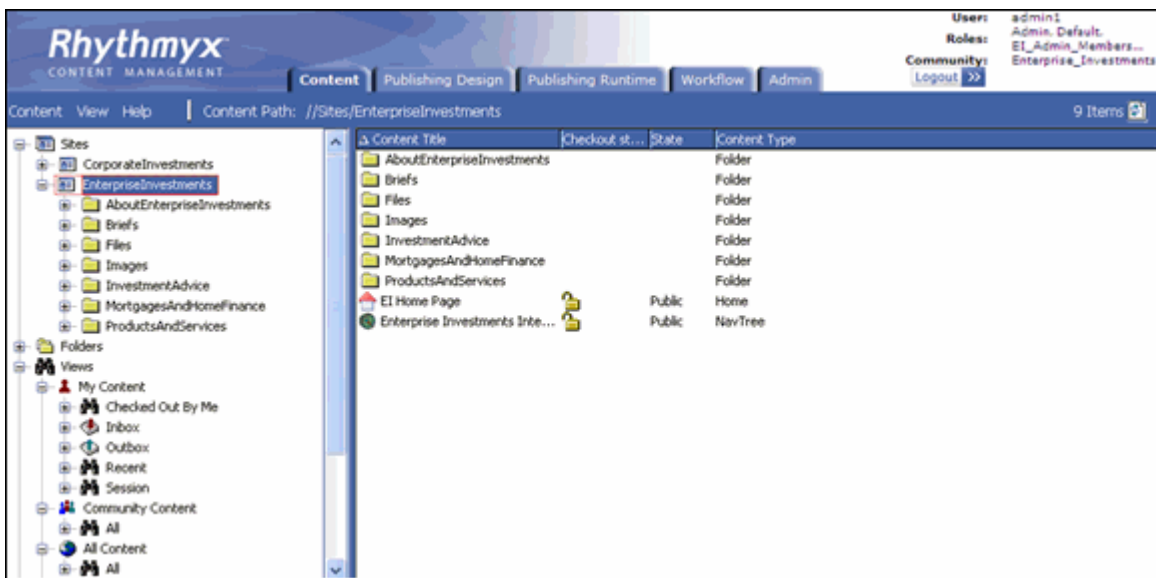


Figure 39: Content Explorer

In the Display Pane, users can select a Content Item node and right-click to open a menu of options for acting on the Content Item.

In the Menu Bar, the Content menu displays options for working with folders, views, and saved searches in the Navigation Tree. Users can access a menu of similar options by right-clicking on a node in the Navigation Tree. The other Menus display options for refreshing the page, changing its appearance, and accessing Help.

At the top of Content Explorer, the Community entry and Locale entry (if it appears) link to pages that let users change these properties.

Users' Roles determine the Content Explorer options and functions and the Content Items that they may access.

The tabs at the top of the page are only visible to users who have access to windows other than Content Explorer. They access the Publishing Design tab, the Publishing Runtime tab, the Workflow tab and the Admin tab, which allow CMS Administrators to register the Publishing, Workflow, and System components of the CMS and to publish content and review it. Most business users do not have access to these tabs.

---

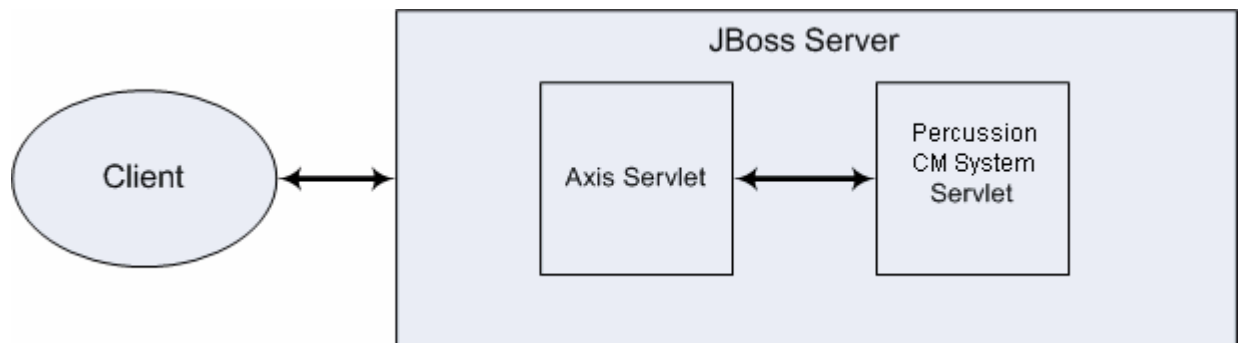
## Web Services API

Web Services enable applications or processes to interact based on standard Internet protocols. CM System includes a Web Services API that serves as an interface between the CM Server and remote client applications or other servers. Percussion CM System Web Services let third-party applications access CM System functionality within their own environment. For example, through Web Services a third party application could create a Content Item or move a Content Item from one Folder to another.

Percussion CM System Web Services are based on Axis, which is an implementation of SOAP. SOAP is a lightweight, XML-based protocol for exchanging structured information in a decentralized, distributed environment. SOAP consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.

The API is defined in schemas and WSDL's found under the CM System installation directory <root>/WebServices/6.0.0.

The graphic below shows the general architecture of CM System Web Services:



*Figure 40: Web Services Configuration*

Each message (except login and logout) to CM System includes a header with valid CM System session information and a body that includes the name of the function requested and parameters that the function requires.



# Full-text Search Engine

CM System provides the Apache Lucene full-text search engine to offer users advanced search features such as the ability to search content in content item fields and the ability to search for synonyms of search terms. Out of the box, the full-text search engine can convert seven common file formats into searchable text, but administrators can easily extend its ability to convert additional file formats.

The full-text search engine resides within the CM Server, and therefore must remain on the same machine as the server. However, if the full-text search indexes consume too much disk space, the administrator can move them to another machine and update the search configuration in the CM Server Administrator to specify the new location of the indexes.

Administrators also have the ability to disable the full-text search engine, which causes CM System's database search engine to become enabled. The database search engine uses the query functionality embedded in the database system.

For more information about configuring the full-text search engine, see the CM Server Administrator online help.

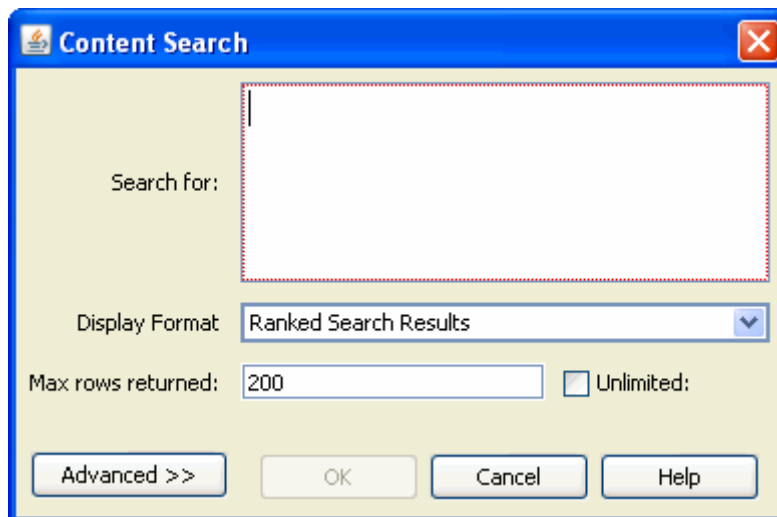


Figure 41: Full-text Search dialog



## CHAPTER 5

# Percussion CM System Modules

Percussion CM System is a Web Content Management system that offers the add-on modules listed in the table below. These modules are transparent to both the logical architecture of the CM Server and the actual physical software components that make up the CM System and do not impact deployment, scaling, or performance considerations.

The following Modules are available in CM System:

<b>Module</b>	<b>Description</b>
Digital Asset Management	Percussion CM System's Digital Asset Management function is a lightweight module that includes a specialized image editor to allow organizations to include digital assets such as images, graphics and other media files in documents and Web content.
Word Connector	Percussion CM System includes a <i>Word Connector</i> (see page 68) that allows content contributors who work primarily in Microsoft Word to continue to create and edit content in their familiar Word environment and easily save it to CM System as CM System content items.

---

## Word Connector

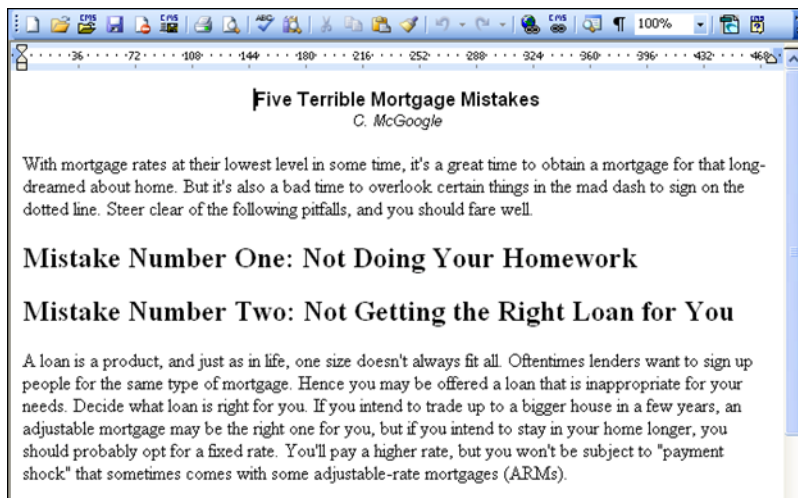
CM System includes a Word Connector that allows content contributors who work primarily in Microsoft Word to continue to create and edit content in their familiar Word environment and easily save it to CM System as CM System Content Items.

The Word Connector uses Word-based Content Editors to let users create Content Items in Word and save them to CM System. A Word-based Content Editor opens a Word document with a Word template that associates Word styles with CM System Content Editor body and metadata fields.

The Connector also enables various CM System features to appear in Word. For example, the Save to CMS control lets a user save a document with an alternate save command to upload it to CM System as a CM System Content Item. CM System saves the Word.doc file in a column in the backend table for the Word-based Content Editor. The Inline Link button lets users add links to CM System Content Items within the body of the Word document.

After a Word-based Content Item is created and saved, a user can either open it in CM System and access Microsoft Word from a control within the Content Editor or open it directly in Microsoft Word. The user edits the content in Word and saves it back to CM System.

CM System uses a Word Macro to manage Word menu extensions, and an ActiveX control to manage the browser-to-Word launching behaviors.



*Figure 42: A Microsoft Word document using a Word Connector template. Icons for saving to the CMS and including CMS links and graphics appear in the Toolbar.*

## CHAPTER 6

# FastForward for Web Content Management

Part of the Web Content Management (WCM) Module, FastForward for WCM is a complete reference implementation which includes a set of pre-built components and applications that allow a company to quickly bring their Web site content under content management. Managers therefore can focus their customization budgets on projects that offer a measurable return, rather than on those that simply enable basic Percussion CM System capabilities. FastForward's pre-built components include:

- A set of commonly used Content Types and Templates to help administrators quickly plan the content and output required by the system, and to save implementers development time.
- Content Types and Templates used to create navigational Content Items, such as site maps, breadcrumb lists, and navigation menus.

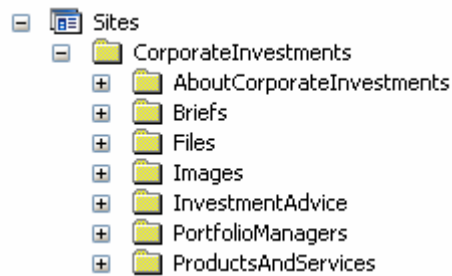


<b>About Enterprise Investments</b>
Press Release
<b>Investment Advice</b>
Insurance Advice
Estate Planning
Retirement
Tax
<b>Mortgages and Home Finance</b>
Home Purchase
Home Equity
<b>Products and Services</b>
Mortgages
Funds
Insurance Products

Figure 43: Output of a Navigation Content Item

- Global Templates. A Global Template is a Template that defines common features for pages on a Web Site or in a document.

- Site Folder Publishing components. Site Folder publishing enables users to publish all of the content in a Content Explorer Site Folder tree to the same folder structure on a Site.



*Figure 44: FastForward's Internet Site Folder Trees*

- Two Workflows that reflect the business processes used by most customers.
- Communities that allow customers to assign users according to Sites and access levels.

## CHAPTER 7

# Physical Architecture, Deployment, and Scaling

The physical architecture of each Percussion CM System depends on a variety of factors including the degree of security used and the location of the Web Server. The number of servers and the optional components included in each system depend on the system load, backup requirements, and business processes. This section offers general information and guidelines for planning the deployment and scaling of a CM System.

## Percussion CM System Physical Architecture

All Percussion CM Systems have the same general physical architecture. Content contributors, who may be either local or remote, access the CM Server via HTTP (default port is 9992) or HTTPS (default port is 9443). CM System must have access to a database, where the Repository resides. Output is generally published either to a file system or to a database. When the Web server is local to the CM Server, the content is published directly into the Web server's file system. If the Web server is on a remote machine, FTP (or SFTP) is used to transfer the files. When publishing to a database, the content is transferred using the proprietary protocol of the RDBMS.

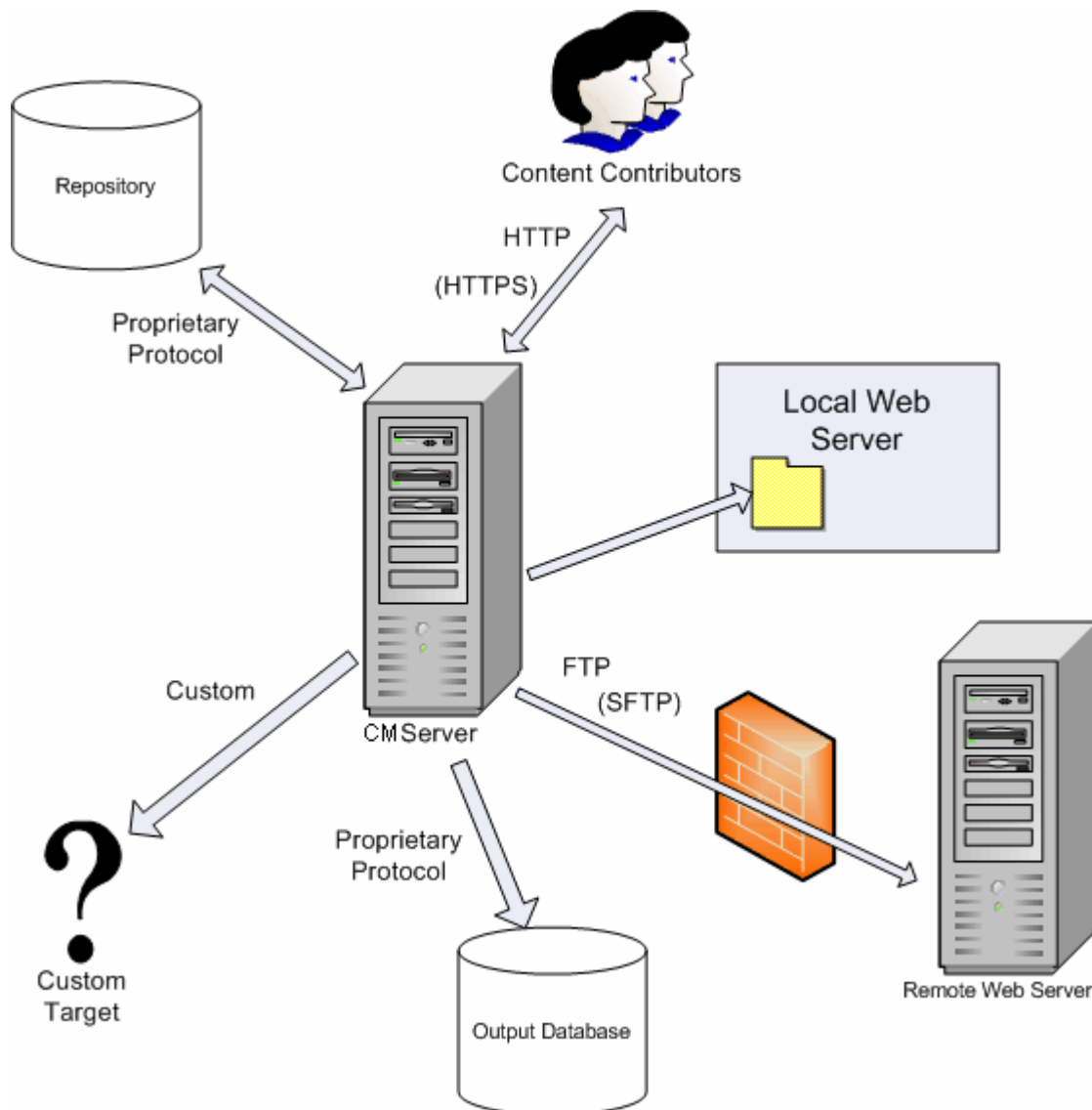


Figure 45: Generic Percussion CM System Architecture



## Deployment Scenarios

The optimal deployment scenario for a Percussion CM System depends on the CM System components included in the system, the number of users working on the system, the level of integration of users' work, and other factors. This section reviews the elements affecting the way a system should be deployed, and suggests deployment models depending on the characteristics of the system.

### Percussion CM System Components

As stated earlier, the Percussion CM System is made up of the same fundamental components regardless of the Modules or licensing options purchased. CM System components include Servers, servlets, end user clients, implementer clients and administrative clients. Client components deployment is straightforward and not covered in this section. Server components can provide for greater scaling, performance, and availability options, as well as more robust development and testing. This section will focus on the server components and the deployment options they provide for the system as a whole.

The available components include:

Server Components	Description	Notes
CM Server (also referred to as: Production Server or System Master)	The fundamental processing and publishing service of the CMS. Runs as a stand alone Server.	Requires a Repository.
Repository	Defines a specific database schema for storing and managing system metadata for all content managed by CM System. May be installed in a variety of standard RDBMS systems, either local or remote to the CM Server. The same repository may be shared by multiple Servers.	External data and content from outside this Repository may also be used by the CMS on a virtual basis.
Servlets and/or Application Server	A set of input/output oriented capabilities. These may be deployed in the native CM System Application Server, or as a set of servlets in a number of J2EE containers.	
Test Server	A CM Server, restricted by license to be used for QA and for User Acceptance Testing of the CMS implementation.	Because of de-coupled delivery, a Test Server is not required to test content (such as Web Sites). The test server is only needed for testing CMS implementation design (such as forms, content types, workflow definitions, etc.)

Server Components	Description	Notes
Development Server	A CM Server, restricted by license to be used for development of the CMS implementation.	

## Multi-tiered Environment

In the Percussion CM System environment that Percussion recommends, implementers create new CM System elements and features on a development server, perform user testing on a testing server, and install tested, new functionality to a production server. In technical terms, this is called a multi-tiered environment and each server level represents a tier.

NOTE: During delivery, CM System content and its associated metadata are rolled out to one or more content delivery applications. The multi-tiered CMS environment should not be confused with the multiple tiers that may exist in each of the delivery channels. In the CMS tiers, CMS functionality, such as forms, workflow, and content type definitions are implemented and rolled out to production users of the CMS. This section will cover only deployment of CMS functionality using the multi-tiered CM System.

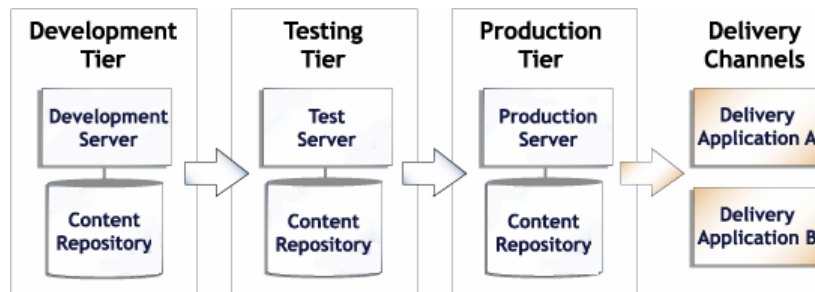


Figure 46: A multi-tiered environment includes development, testing and production tiers.

### Development Tier

Developers create new functionality in the development tier. Unique tasks that developers perform on the development tier include:

- Developing CM System elements such as Content Types and Templates;
- Testing and fixing bugs in CM System elements;
- Publishing to development servers and local file systems;
- Installing Multi-Server Manager archives from production and integration servers to modify existing functionality and from other development servers to adapt other developers' functionality;
- Installing sample Content Items from production or integration tiers with the Multi-Server Manager for testing or bug fixing.

## Testing Tier

Developers test new features and fix bugs on the integration tier. Other tasks include:

- User Acceptance Testing (UAT);
- Publishing to an integration testing Web server;
- Installing archives from the development tier for testing and integration and from the production tier for bug fixing.

## Production Tier

Tasks on the production tier include:

- Content creation and approval;
- Publishing to a staging Web server (to test a production configuration);
- Publishing to a production Web server;
- Installing archives of tested new functionality from the testing tier.

## Configuration Options for Development, Testing, and Production Tiers

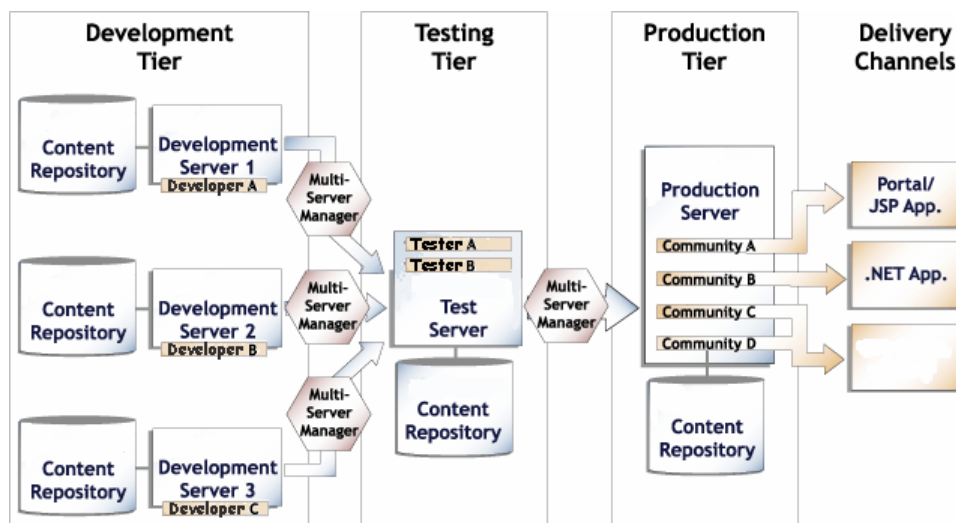
In the development tier, implementers should deploy separate CM Servers with separate repositories if developers work on independent projects or are separated geographically. If developers work together on projects, implementers should deploy a single development server or a single development repository to allow them to view and integrate with one another's work. Put developers who are working on independent projects or who are geographically separated on different servers and repositories so they can avoid conflicting with other developers' work.

Depending on an organization's content-sharing requirements, implementers may deploy one shared CM Server and repository or multiple, separate, CM Server and repositories on the testing tier. Testing on a single, shared CM Server on this tier is recommended if users share a CM Server that integrates all functionality in the production tier. Deploying multiple, separate instances of CM System on this tier can be beneficial if CM System Communities are separated onto different CM System instances with different features and different publishing destinations in the production tier. In many situations both CM System configurations are deployed in the integration tier.

Likewise, implementers can deploy a shared CM Server or separate CM Servers on the production tier. Configuration depends on whether or not Roles and Communities share content and applications and whether or not they always publish to the same or different Sites.

In many cases, tiers include both shared CM Servers and separate CM Servers that are used by individual developers or certain Communities or Roles.

The following graphics show two possible configuration scenarios for a multi-tiered environment; in each graphic, at least one tier includes both a separate and a shared CM Server repository. In the first graphic, three developers work separately on three different CM Server repositories and Servers, but deploy all of their new functionality to a single CM Server instance for testing. Two testers work on the testing CM Server and deploy the tested components to a single production server and repository. Four different Communities share the production Server, although two of the Communities publish content to unique Sites.



In the second graphic, two developers work on individual CM Servers and repositories and two developers share a third Server and repository. The functionality developed on each Server is deployed to a separate testing Server. Different users test the functionality on each testing Server and deploy the tested components to separate production Servers. Two Communities work on their own production Servers and repositories and two additional Communities share a third production Server and repository. Content created in each production Server is published to a different destination.

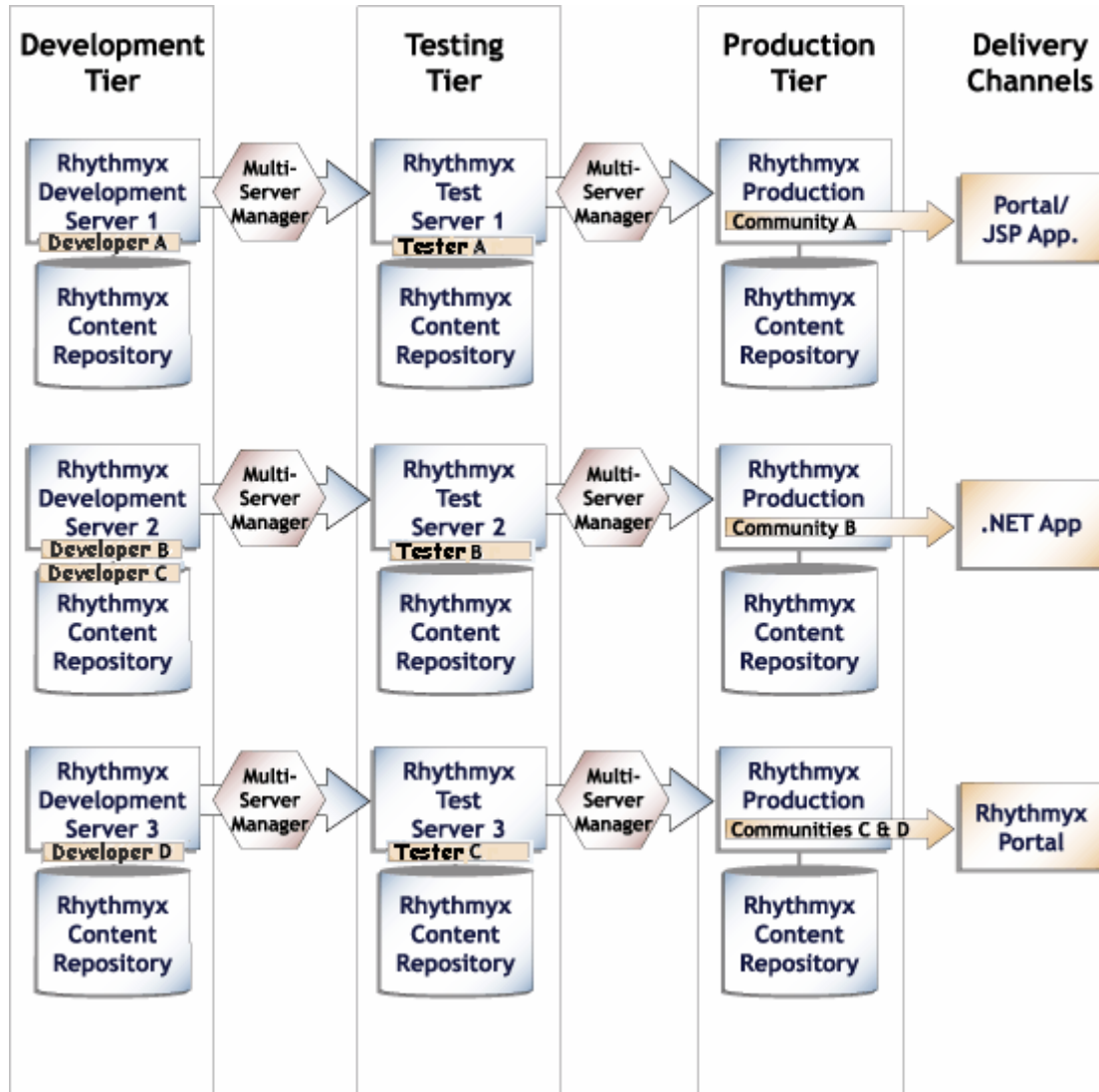


Figure 47: A decentralized CM System environment

## Scaling the Percussion CM System Publishing Environment

After selecting a deployment approach, an Administrator must consider what number of Publishing Hubs will meet an organization's publishing requirements.

The number of Publishing Hubs required depends on the amount of content published to each target environment.

In the following graphic, the amount of content published requires two publishing hubs, each handling the entire volume published to one destination and sharing the task of publishing to another destination.

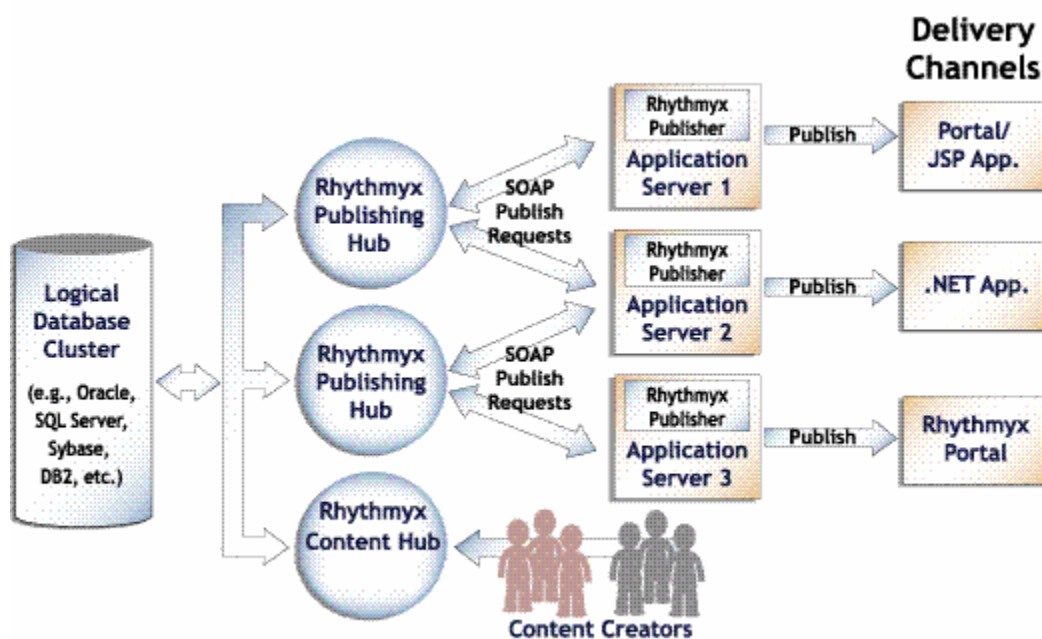


Figure 48: Scaling CM System Publishing Hubs

Percussion recommends scaling by increasing Publishing hubs because increased productivity in CM System causes greater CPU use rather than more network traffic. This architecture allows for the segmentation of the CPU cycles across multiple servers based on typical user activity and on the expected publishing requirements of multiple communities and multiple delivery locations.

## Guidelines for Expanding a Percussion CM System

A Percussion CM System Web Content Module or Document Management Module includes one Content Hub and one Publishing Hub. Depending on the factors discussed in this section, a company may initially choose to purchase additional publishing hubs, a test server, or a hot standby server, or may purchase them later, as its system develops and grows.

The following are guidelines for determining when to expand a CM System:

Optional Component	Scaling Guidelines
Additional Publishing Hub	If the publishing runs in your incremental publishing cycle overlap each other (one publishing run begins before the previous one is finished) your system may begin to republish Content Items and waste resources. You may acquire an additional publishing hub to handle some of the publishing volume.
Database Publisher	If you store content in a database and deliver it dynamically, you should consider purchasing a Database Publisher when you purchase your CM System. Otherwise, purchase a Database Publisher when you implement a database storage and delivery system for your Web site.
Test Server	<p>Access to development servers is restricted to users with Workstation licenses; therefore a test server is necessary if you want business users to test your changes before you move the changes to a production server.</p> <p>A test server is also useful if:</p> <ul style="list-style-type: none"> <li>▪ Your implementers need the ability to merge their development work on a common server.</li> <li>▪ You want to make new functionality available to some users for testing, but not to others.</li> </ul>
Hot Standby Server	If customer need or industry regulations require that you have a highly available and updated system, you may acquire hot standby servers to back up your CM Server and other servers that you have added to your CM System.

---

## **JBoss Application Server**

Percussion CM System runs in a Java 2 Enterprise Edition (J2EE) environment and is installed as a Web application within a JBoss application server. JBoss offers several advantages over other application servers, including its open source status, its widespread use, and its technical capabilities. JBoss functions as a Java servlet container that lets CM System run the Percussion CM System Publisher and Web Services and several of the CM System standalone applications as Java Servlets. In addition, JBoss includes a Java Server Pages (JSP) engine which enables CM System users to deliver dynamic Web Pages with efficiency and flexibility, the Java Message Service (JMS) for enabling communication between J2EE servers, and the Java Management Extension (JMX) to control Java applications.



---

## Database

Percussion CM System stores unformatted content and metadata in third-party database applications. Most of the database features are managed through CM System while backup is handled separately by the database. A large number of functions may be accomplished through CM System interfaces, such as the Server Administrator, the Workbench, and the Multi-Server Manager. For example:

- entering backend server timeout and connection limits, setting up database users and passwords, and configuring backend table security providers.
- deploying table schemas and data from one backend database to another.
- creating backend tables that store local Content Editor data.

Each CM Server may connect to its own backend database (or databases) or CM Servers may share databases depending on the Servers' functions. See *Configuration Options for Development, Testing, and Production Tiers* (on page 76).



## CHAPTER 8

# Data Protection

Data protection in Percussion CM System includes a range of security functions that limit access to CM System Content Items, interfaces, and applications to users with the proper credentials. Since CM System content is stored in external database repositories, CM System does not include its own backup or archiving features; however Percussion recommends that customers carefully follow the procedures provided by their database systems.

---

## Security

Percussion CM System provides various mechanisms for preventing users without the proper permissions from gaining access to Content Items and other CM System components.

At the most basic level, users must have a username and password to log in to CM System. The System Administrator first assigns these in a security provider, and then adds the user to CM System through the Percussion CM Server Administrator.

Security providers are systems that maintain directories of usernames and credentials for the purpose of authenticating users who attempt to access a server or an individual application. By default, CM System uses a backend table security provider that validates usernames and passwords. CM System systems that access CM System through a Web application server may use a Web Server security provider which relies on an existing Web server or Web application server authentication.

CM System also uses Access Control Lists (ACLs) to define users' access levels to servers, applications, Content Explorer Folders and Content Items. The CM System Administrator defines a set of Roles with different privileges in the Server ACL and then assigns users to one or more Roles.

Although Communities and Locales can limit the content and user interface components that are available to users, they are not security mechanisms. Communities and Locales simplify what users see by filtering out information that is not relevant to them; however, they do not provide the security available through the server and ACL settings.

To prevent the loss of data that might occur if multiple users attempted to modify and save the same Content Item simultaneously, CM System includes a check in and check out feature that prevents more than one user from having edit access to a Content Item at the same time. A Content Item must be checked in for a user to have edit access to it. The user must check it out to edit it; checking it out locks it to all other users.

For more information, see the section “About Security” in the online help accessible through the CM Server Administrator.

---

# Backup

Backing up Percussion CM System includes implementing procedures for archiving data and backing up the server and database.

## Archiving

All Percussion CM System content is stored in the database; therefore administrators can use standard database archiving procedures to meet content archiving requirements.

---

NOTE: When Content Items are removed from a Publish State in a Workflow, they are often transitioned to an Archive State. No intrinsic connection exists between Content Items that are in an Archive State and Content Items archived through database archiving.

---

## Purging

Purging permanently destroys all records of a Content Item in the system. A company's retention rules may require that purge actions only be available for Content Items that have been stored through database archiving.

## Backup Recommendations

### CM Server Backup

To back up the CM Server, the Network Administrator can use any type of back up software or hardware to make a copy of the CM System Tree (the entire contents of the CM System installation root). Depending on the amount of server use and the significance of the data entered, the Network Administrator can choose appropriate backup software and hardware and determine an optimal back up schedule.

In most back up scenarios, the development CM Server is backed up more frequently than the production server because applications are modified more frequently in development. For example, an Administrator might schedule full weekly backups and partial nightly backups for an active development environment, but perform a backup of the production environment only when a change is introduced.

### Database Backup

The Network Administrator determines the procedure and schedule for database backup based on the methods available for the specific database that the CM System system uses. CM System works with any database backup and recovery scheme. Most CM System implementations use a single database/schema, and the backup procedure depends on its physical storage configuration.

As a general rule development databases should have the same backup schedule as the corresponding development Server tree. In most cases, the production repository backup schedule depends on the frequency of content contribution to the system.



## APPENDIX I

# XSL in Percussion CM System

Percussion CM System Version 6.0 and later versions use Velocity Templates to provide assembly processing and formatting. CM System Version 5.7.1 and earlier used XSL to provide assembly processing and formatting, in an object called a *Variant*. Variants are still available in Percussion CM System

Version 6.0 and later, usually to support backwards-compatibility.

A Variant consists of the CM System XML application resource that selects and processes the Content Item data to be processed, and the XSL stylesheet used to format the final output. In addition, a Variant defines the rules dictating how the Variant interacts with other Variants. While a Velocity Template can be associated with multiple Content Types, each Variant can be associated with only one Content Type. (But each Content Type can still be associated with multiple Variants.)





# Index

## A

ACLs • 84  
 Active Assembly • 18  
 Apache Lucene • 65  
 Application Server • 73, 80  
 Archiving • 85  
 Archiving content • 85  
 Assembly Concepts • 17  
 Assembly Service • 17, 34  
 Axis • 64

## B

Backup • 85  
 Backup Recommendations • 85

## C

Clients and Interfaces • 11, 12, 47  
 Communities • 27  
 Configuration Options for Development,  
 Testing, and Production Tiers • 76, 81  
 Content Concepts • 14  
 Content Editors • 14, 15  
 Content Explorer • 20, 61, 63  
 Content Items • 14  
 Content lists • 24  
 Content Management • 8, 9, 10  
 Content Management in Percussion CM System  
 • 9  
 Content Service • 14, 15, 30  
 Content Types • 14

## D

Data Protection • 12, 58, 83  
 Database • 81  
 Database Publisher • 79  
 Datasources • 55  
 Deployment Scenarios • 73  
 Development Server • 73, 74, 76  
 Development Tier • 74  
 Digital Asset Management • 67

Directory Services • 57

## E

Editions • 24  
 Enterprise Content Connector • 73

## F

FastForward • 69  
 FastForward for Web Content Management • 11,  
 12, 69  
 Folders • 21  
 Full-text search engine • 65  
 Full-text Search Engine • 11, 12, 65

## G

Global Templates • 69  
 Guidelines for Expanding a Percussion CM  
 System • 79

## H

Hot Standby Server • 79

## I

Introduction • 7

## J

J2EE • 80  
 JBoss • 80  
 JBOSS Application Server • 80  
 JSP • 80

## L

Locale • 16  
 Locales • 16  
 Logging • 56  
 Lucene • 65

## M

Monitor • 60  
 Multi-Server Manager • 62  
 Multi-tiered Environment • 74

## N

Navigation • 69

## P

Percussion CM System and Item-based Content  
 Management • 10  
 Percussion CM System Components • 73  
 Percussion CM System Concepts • 12, 13

Percussion CM System Logical Architecture and Processing • 11, 12, 29  
Percussion CM System Modules • 12, 67  
Percussion CM System Physical Architecture • 72  
Permissions • 27  
Physical Architecture, Deployment, and Scaling • 12, 71  
Production Server • 74, 75, 76  
Production Tier • 75  
Publishing • 23  
Publishing Concepts • 23  
Publishing Engine • 44  
Publishing Hub • 73, 78, 79  
Purging • 85  
Purging content • 85

## R

Relationship Service • 19, 26, 37  
Relationships • 18, 26  
    Active Assembly • 26  
    Translation • 26  
CM System  
    Application Server • 73, 80  
    Deployment • 73, 76, 78  
    Development Server • 73, 74, 76  
    Logical architecture • 29, 30, 34, 37, 41, 44  
    Multi-Server Manager • 62  
    Physical Architecture • 72  
    Publishing Hub • 73, 78, 79  
    CM System Database • 81  
    CM System Repository • 73, 81  
    CM System Server • 73  
    Server Administrator • 52, 53, 54, 55, 56, 57, 59, 60, 61  
  
    Test Server • 73, 74, 75, 76, 79  
    Web Server • 72  
    Workbench • 48  
CM System Database • 81  
CM System Repository • 73, 81  
CM System Server • 73  
Roles • 25, 27

## S

Samples • 69  
Scaling the Percussion CM System Publishing Environment • 78  
Search • 59, 65

Server Administrator Search tab • 59  
Search and Replace • 59  
Security • 58, 84  
Security Concepts • 27  
Security providers • 84  
Server Administrator • 52  
    Content Explorer • 61  
    Datasources • 55  
    Directory Services • 57  
    Logging • 56  
    Monitor • 60  
    Search • 59  
    Settings • 54  
    Status • 53

Sessions • 22  
Settings • 54  
Site Folder Publishing • 69  
Sites • 14, 23, 24  
SOAP • 64  
States • 25  
    approval State • 25  
    archive State • 25  
    development State • 25  
    public State • 25  
Status • 53  
System Concepts • 25  
System Requirements • 12, 87

## T

Templates • 17, 18  
Test Server • 73, 74, 75, 76, 79  
Testing Tier • 75  
The Basics of Content Management • 8  
Tiered Environment • 74, 76

## U

UI Concepts • 20  
Using Percussion CM System Documentation • 11

## V

Variants • 87  
Views • 22

## W

Web Content Management • 8, 67, 69  
Web Server • 72  
Web Services • 64  
Web Services API • 64

Word Connector • 67, 68  
Workbench • 48  
Workflow • 21, 22, 24, 25, 27  
    Roles • 27  
Workflow Engine • 26, 41

**X**

XSL in Percussion CM System • 87