

.....

JCMS v.2.0-a1-dev

Project Documentation

Table of Contents

1 General Information	
1.1 Welcome to JCMS	1
1.2 Overview	2
1.3 JCMS Features	4
1.4 News and Status	6
1.5 Roadmap	7
1.6 FAQs	8
2 General Documentation	
2.1 Architecture	9
2.2 installation	12
2.3 Configuration files	13
2.4 Maven goals	14
3 Design	
3.1 Overview	15
3.2 Design	16
3.3 Current implementations	17
3.4 Object Model	18
3.5 Services	19
3.5.1 Model Service	20
3.5.2 Security Service	21
3.5.3 Search Service	22
3.5.4 Versioning Service	23
3.5.5 Lock Service	24

1.1 Welcome to JCMS

Welcome to JCMS

JCMS Goals

JCMS is an open source project for building content management & document management systems. JCMS is component based solution, so you can use it in any Java applications.

Its main goal is to avoid strong coupling between your application and a specific CMS technology. Each CMS service is pluggable and can be replaced easily by another CMS product.

Send us your remarks and suggestions on the mailing lists and obviously you are welcome to contribute.

Portal Integration

JCMS (Jetspeed Content Management System) is the default CMS solution for [Jetspeed 2 Portal Server](#).

Overview and Features

See the sections [Overview](#) and [Features](#) to get more information on JCMS.

Download

1.2 Overview

What is JCMS ?

General Overview

JCMS offers a complete platform for creating, managing, and publishing content in your portal or in any other kind of java application. JCMS integrates content repositories, workflow, collaboration and personalization via existing open source projects and powerfull standards like WEBDAV.

JCMS includes features like taxonomy, content versionning, fine grained access control, collaborative editing, publication workflow, scheduling, indexing, searching and more. It will also support many document types like XML, HTML, PDF, MS Office, Open office, ect.

Look here to see the full list of JCMS's [features](#) .

JCMS Use Cases

There is a huge amount of applications that can uses JCMS. Some good examples are :

1. Content catalog like Yahoo style portal : you can classify your content in a virtual content tree which group together different content repositories.
2. On-line content management : web site content is updated directly by the end-users. They can update news, HTML pages and all information which has to be stored into CMS repositories.
3. Document management : different teams can share documents required for their job in a secure fashion. Following a customized authoriting workflow, they can receive notifications when documents have been updated; takes some actions on the existing documents; and at the end, publish the final publications in different formats (PDF, HTML, Word, ...)

Technical Overview

JCMS is, in fact, a flexible set of components, services and portlets which can transparently support different CMS engines. JCMS is packaged with a default cms repository : JCMS Store but it also supporting a plugin design. Existing plugin are OJB and Webdav. You can also write your own plugin in order to access to a proprietary CMS repository.

The CMS implementation can be conformant on three levels:

1. Level One Conformant - implements the required Content and Lock component interfaces.
2. Level Two Conformant - implements Search , Version and Security component interfaces in addition to Level One required interfaces.

3. Level Three Conformant - implements Workflow component interface in addition to Level One and Level Two interfaces.

The Content Management resolution is performed in a pluggable fashion; consequently, Jetspeed applications are independent of underlying CMS technologies. New or updated technologies can be substituted without requiring modifications to the application itself. An implementation for a particular CMS technology is determined at run-time (see the architecture detail to get more information).

The JCMS components are based on PicoContainer IOC framework. PicoContainer is the default service framework for Jetspeed 2 but JCMS can be used in other portal applications or in other kind of Java platform as well.

1.3 JCMS Features

JCMS Features

Here is a summary of the main JCMS features.

CMS Reposirotly Services

Creating flexible CMS services which can transparently support different CMS repositories. By this way, Jetspeed can access to heterogenous content store. It should be possible to support CMS repositories which are providing a Java API or a WEBDAV layer or the upcoming JSRs (JSR 147 & JSR 170).

Features	Description
Model Service	Manage all CMS objects (documents, folders, ...). Create, retrieve, update and delete content. Store links between contents. Provides an infrastructure for recording information about content.
Lock Service	Manages concurrency access to the CMS objects Enable remote, collaborative authoring of any media type. Resources can be locked as "exclusive" or "shared".
Version Service	Manages content versioning. Infrastructure for versionned resources : check in/ check out with comments and metadata; versionning graphs and histories; browse and retrieve old versions.
Security Service	Manages "fine grained" permissions across the complete content tree Each CMS service implementations has to be integrated with the Jetspeed security services or another security service; Grant permissions are based on the association between a content reference, an action type (eg. : read, write) and an actor type (user, group or role).
Search & indexing Service	Search documents and folders via full text search and/or metadata criteria.; Possibility to implement a new query language. Minimal query language can be : boolean search for property existence or value; or search for substring in a resource content. Indexing documents can vary in function of the language and the media type.
Workflow Service	Manages the different publication activities. Content can be attached to a workflow, which is a series of tasks what must be completed. Workflow can consist of any number of manual tasks, performed by users and groups, as well as automated tasks which are performed. Workflow tasks can be assigned to groups, in which case any member of a designated groups may approve, edit, or re-route content as appropriate. The default workflow can be customized and linked for reuse in other projects. Users are notified of new tasks via email notifications or notification portlet.
Event Service	Triggers events when content is created, update, deleted or retrieved; Events can be customized in order to index the document or makes auditing, ...
Integration Service	Manages external repositories consolidated into one virtual content tree. The user doesn't worry about the physical content location.

JCMS Store

[TO BE COMPLETED]

CMS portlets

Accessing thoses CMS Services across predefined JSR-168 portlets. Providing or integrate CMS editor portlets in order to update the CMS repository content. Managing the content publication across a customized content workflow.

So, we plan to build portlets for the following usages :

- 1.Administration : managing the content tree (add, delete folders & documents, ...), setting permissions, managing external server references, ...
- 2.Content browser & viewer.
- 3.Content workflow/notification.
- 4.Content editor.
- 5.Specialized portlets like news, forum and many others which have to store their content into a content repository.

Other Features

Support for any media type (XML, HTML, binaries like Ms Word, ...).

WEBDAV support.

Transactionnal engine & transactionnal event handlers.

Tools like scheduler, report generator, asynchronous publication, ...

JCMS JSP tags and templates

1.4 News and Status

Status

The JCMS birth

JCMS development is just starting. The code is not 100% stable and not all jcms features are implemented.

Obviously, you are welcome to send your comments and to contribute !

News

29 March 2004

Model and security services are available.

First implementation is based on Slide. it is possible to store your content and their metadata in a file system or MySQL DB.

We have just starting the JCMS documentation.

1.5 Roadmap

Roadmap

2.0-a1-dev

JCMS Store implementation for the main CMS repository services (based on OJB) :

1. Mode Service.
2. Simple Version Service.
3. Search service based on Lucene.
4. Security service.
5. Lock Service.
6. Event Service.

Simple demo portlets.

2.0-a2-dev

Integration service.

Plugin architecture for at least JCMS Stores and WEBDAV servers.

2.0-a3-dev

Notification service.

Workflow service.

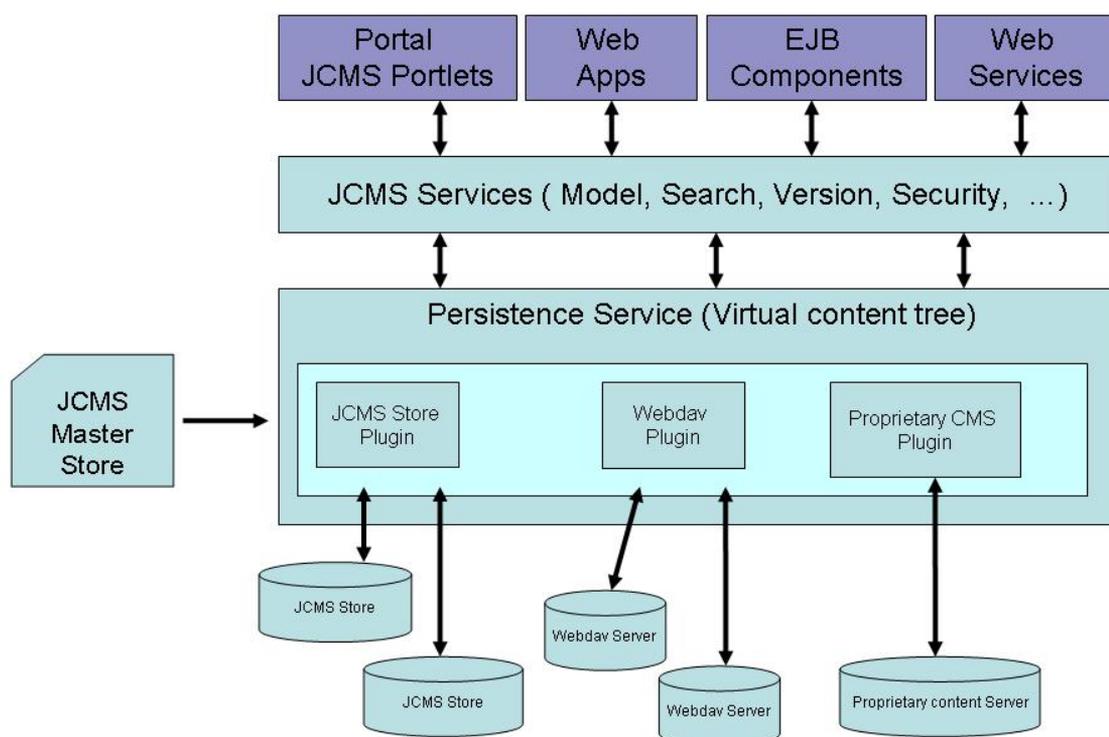
Portlets & tools :

1. Admin portlets
2. Editors & viewers

1.6 **FAQs**

2.1 Architecture

JCMS Engine Architecture



Architecture Neutral

Thanks to the PicoContainer framework, JCMS can be used in different kind of architectures :

1. Portal server : JCMS is the default Jetspeed CMS engine.
2. Any other web applications can used JCMS.
3. EJB components.
4. Web services.

Virtual Content Tree and JCMS Master Store

JCMS groups together different stores into the same virtual content tree. By this way, your application can access to heterogenous content servers. The principal store requires by the JCMS engine is the JCMS Master Store. It contains all store references and knows how to mount those stores into the virtual content tree. JCMS Master Store can also contain some CMS objects but it is not mandatory.

The JCMS Engine

The JCMS engine is a container based on 2 distinct layers : The "JCMS services" layer and the "Persistence service" Layer.

The first one, the "JCMS Services" deals with an high level of abstraction. It is composed of different PicoContainer components which are providing an generic API. This API is independant of any CMS content store. Each PicoContainer component solves one of the CMS aspect. For example, the model service manages the CMS objects lifecycle (instantiate, create, update, remove) and the search service gives a way to search content across the complete content tree, In order to have more abstraction on the content location, the "JCMS Services" layer dispatchs all requests to the "Persistence Service" layer.

The core JCMS services are :

1. The Model Service.
2. The Version Service.
3. The Security Service.
4. The Search Service.
5. The Index Service.
6. The Workflow Service.
7. The Event handler Service.

Thanks to the PicoContainer assembling script, it is possible to add new JCMS services.

The "Persistence Service" layer is the key component for building the virtual content tree. After reading all store references from the JCMS Master Store, it can mount each of them into the content tree. Mounting a store is based on an URI mapping. All servers are associated to an URI prefix. So, a CMS object is stored into the store which has the same URI prefix. In the current version it is possible to mount a server only on the root level. Later, we plan to mount a server anywhere in the content tree. The "Persistence Service" is based on a plugin design. By this way, it is possible to have different kind of stores (Webdav, JCMS Store or proprietary store). The "Persistence Service" knows which plugin to use with which server. The plugin goals is to convert a JCMS request (eg. create a document) into a specific CMS API.

The JCMS Store plugin

JCMS Store plugin can be used to access to a predefined DB schema provided with the JCMS distribution. This plugin implementation is based on OJB.

The Webdav plugin

[TO BE COMPLETED]

2.2 **installation**

2.3 Configuration files

2.4 Maven goals

Maven Goals

In addition the standard maven goals, the following goals have been defined.

Goal	Description
testdb-slide-create	Create the DB used for unit testing. The parameters used for this goal are defined in the project.properties file.

3.1 Overview

3.2 **Design**

3.3 **Current implementations**

3.4 **Object Model**

3.5 **Services**

3.5.1 **Model Service**

3.5.2 **Security Service**

3.5.3 **Search Service**

3.5.4 **Versionning Service**

3.5.5 **Lock Service**
