

1

Overview

This book is a guide for developers interested in building websites using the OpenCms content management system. The book is intended for developers who are familiar with Java, JSP, and building web applications based on the Java J2EE framework.

In this book, we will develop a website designed for a blog writer. In the course of building our site, we will go over these topics:

- The site design
- Overview of OpenCms
- Setting up an OpenCms development environment
- Creating structured content types
- Creating templates
- Utilizing search
- Extending OpenCms
- Allowing online users to contribute site content

We will go over all the steps involved in building a blog website using OpenCms. We will start by describing the features and requirements of our website and will then provide an overview of OpenCms. Next, we will discuss how to create a development environment. We then will go over the steps involved in creating structured content types, to hold our site content. After that, we will cover creation of templates and Java code, to display the content. The site also supports search and user comments, so we will cover the Lucene search engine as well to show how to provide login support. As the site additionally supports RSS clients and feeds, we will discuss how to add new features to OpenCms.

Before we get into the development details, we will first discuss some of the skills required to develop sites with OpenCms. This will provide us with a basis for understanding the environment and tools, which we will need to do our development work.

The Site Design


Before the development of any site can begin, there should be an understanding of the site's feature requirements. The feature requirements will often be driven by the actual layout and design of the site. We will design and build a blog website named 'Deep Thoughts'. The design of the site homepage layout looks like this:

The screenshot shows a blog homepage for 'deep thoughts' with the tagline 'WE THINK DEEPLY SO YOU DON'T HAVE TO'. The date is August 1, 2007. The main content area features two articles: 'What If Cows Could Fly?' (dated 27.07.2007) and 'What if the World were flat?' (dated 10.07.2007). Each article includes a cow image and placeholder text. The right sidebar contains a search bar, a 'Text Link Ads' widget, and sections for 'ARCHIVES' (listing July, June, and May 2007), 'FEEDS' (listing various content items with durations), and 'SUBSCRIBE' (listing RSS and ATOM feeds).

deep thoughts
WE THINK DEEPLY SO YOU DON'T
HAVE TO

August 1, 2007

What If Cows Could Fly?
27.07.2007




Sed ut perspiciatis unde omnis sed quia consequuntur magni dolores eos totam rem aperiam. Qui ratione voluptatem sequi nesciunt. Atque corrupti quos dolores et quas molestias cumque nihil impedit quo minus omnis voluptas assumenda est.

Accusantium doloremque laudantium, aut odit aut fugit, praesentium voluptatum deleniti. Nisi ut aliquid ex ea commodi consequatur? Ut enim ad minima veniam, atque corrupti quos dolores et quas molestias accusantium doloremque laudantium. Tempora incidunt ut labore et dolore et aut officiis debitis aut rerum necessitatibus et expedita distinctio. Consequatur aut perferendis doloribus asperiores repellat. Sed quia consequuntur magni dolores eos [more...](#)

TOPICS: silly, prudent, hopeful

What if the World were flat?
10.07.2007

Duis aute irure dolor velit esse cillum dolore in reprehenderit in voluptate. Lorem ipsum dolor sit amet, qui officia deserunt cupidatat non proident. Ut aliquip ex ea commodo consequat.



Sunt in culpa ut labore et dolore magna aliqua. In reprehenderit in voluptate sed do eiusmod tempor incididunt velit esse cillum dolore. Qui officia deserunt ut aliquip ex ea commodo consequat. Cupidatat non proident, in reprehenderit in voluptate sunt in culpa. Duis aute irure dolor sed do eiusmod tempor incididunt ut aliquip ex ea commodo consequat.

Velit esse cillum dolore ut labore et dolore magna aliqua. In reprehenderit in voluptate excepteur sint occaecat lorem ipsum dolor sit [more...](#)

TOPICS: worrisome, awesome

©Deep Thoughts 2007, All Rights Reserved

Search

Text Link Ads
Monetize your website.

ARCHIVES

- July 2007 (2)
- June 2007 (2)
- May 2007 (1)

FEEDS

- Financial Literacy Begins at Home (14:10)
- The Golden Age of Chicago Prostitution: A Q&A with Karen Abbott (12:07)
- Finally, Some Love From the Real Estate Industry (09:58)
- And Today Is... (09:00)
- The FREAKest Links: Time After Time Edition (15:41)
- Will Copper-Stealing Meth Heads Jack Up the Price of Your Almonds? (14:30)

SUBSCRIBE

- XML RSS 0.91 Feed
- XML RSS 1.0 Feed
- XML RSS 2.0 Feed
- XML ATOM 0.3 Feed

The blog site is designed to support the following features:

- Blogs are listed in descending order of date, with the most recent blog appearing at the top.
- Each blog entry is listed in teaser style, with a link to the full blog appearing at the end.
- Blog entries support a list of topics attached to them.
- Archives of previous blogs appear on the righthand side, in descending order.
- Past blog archives can be browsed.
- The site supports contents search with paginated results.
- Ads may be placed on the righthand side.
- Users may self register for the site.
- Registered users may add comments and create a customized RSS feed on their homepage.
- Blogs may be viewed in various RSS formats.

In addition to the features seen in the mockup, we will also support:

- Direct editing of content in preview mode.
- User submitted comments.

There are two additional mockups for the site. The first one shows a detailed view of a blog. This view is shown when a user clicks on a blog from the homepage:

[Edit]



August 2, 2007

What If Cows Could Fly? [Edit]

27.07.2007



Sed ut perspiciatis unde omnis sed quia consequuntur magni dolores eos totam rem aperiam. Qui ratione voluptatem sequi nesciunt. Atque corrupti quos dolores et quas molestias cumque nihil impedit quo minus omnis voluptas assumenda est.

Accusantium doloremque laudantium, aut odit aut fugit, praesentium voluptatum deleniti. Nisi ut aliquid ex ea commodi consequatur? Ut enim ad minima veniam, atque corrupti quos dolores et quas molestias accusantium doloremque laudantium. Tempora incidunt ut labore et dolore et aut officiis debitis aut rerum necessitatibus et expedita distinctio. Consequatur aut perferendis doloribus asperiores repellat. Sed quia consequuntur magni dolores eos ut aut reiciendis voluptatibus maiores alias quia voluptas sit aspernatur.

Qui in ea voluptate qui dolorem ipsum quia dolor sit amet, dicta sunt explicabo. Dignissimos ducimus qui blanditiis saepe eveniet ut et voluptates repudiandae sint similique sunt in culpa qui officia deserunt mollitia animi. Qui ratione voluptatem sequi nesciunt. Sed quia consequuntur magni dolores eos consectetur, adipisci velit, nisi ut aliquid ex ea commodi consequatur? Dignissimos ducimus qui blanditiis aut odit aut fugit, velit esse quam nihil molestiae consequatur. Sed quia consequuntur magni dolores eos id est laborum et dolorum fuga.

Omnis voluptas assumenda est, at vero eos et accusamus et iusto odio ut aut reiciendis voluptatibus maiores alias. Magnam aliquam quaerat voluptatem. Sed ut perspiciatis unde omnis id est laborum et dolorum fuga. Itaque earum rerum hic tenetur a sapiente delectus, et molestiae non recusandae.

Iste natus error sit voluptatem tempora incidunt ut labore et dolore sed quia consequuntur magni dolores eos. Et expedita distinctio. At vero eos et accusamus et iusto odio.

TOPICS: silly, prudent, hopeful

[Edit] ©Deep Thoughts 2007, All Rights Reserved

[Edit]



Monetize your website.

ARCHIVES

- July 2007 (2)
- June 2007 (2)
- May 2007 (1)

FEEDS

- Free Book! (Not Ours.) Act Fast! (10:37)
- Meet Newser.com (10:17)
- Another Long-Ago Economics Student Heard From (09:35)
- And Today Is... (08:55)
- Financial Literacy Begins at Home (14:10)
- The Golden Age of Chicago Prostitution: A Q&A with Karen Abbott (12:07)

SUBSCRIBE

- RSS 0.91 Feed
- RSS 1.0 Feed
- RSS 2.0 Feed
- ATOM 0.3 Feed

The last mockup shows what the search result screen looks like. Search results are shown in decreasing order of relevance to the search term. The pagination controls at the bottom of the page allow for the results to be scrolled, if necessary:



deep thoughts

WE THINK DEEPLY SO YOU DON'T HAVE TO

August 2, 2007

Search For: what

What if Cows Could Fly (100 %)
 Id est laborum et dolorum fuga. Et aut officiis debitis aut rerum necessitatibus omnis dolor repellendus. Magnam aliquam quaerat voluptatem. Tempora incidunt ut labore et dolore ut enim ad minima veniam, vel illum qui dolorem eum fugiat. Dicta sunt explicabo.Saepe eveniet ut et voluptates repudiandae sint cumque nihil impedit quo minus sed quia non numquam eius modi. Excepturi sint occaecati cupiditate non provident, magnam aliquam quaerat voluptatem. Nam libero tempore, totam rem aperiam, at vero eos et accusamus et iusto odio.

What if Cows Could Fly (81 %)
 Id est laborum et dolorum fuga. Et aut officiis debitis aut rerum necessitatibus omnis dolor repellendus. Magnam aliquam quaerat voluptatem. Tempora incidunt ut labore et dolore ut enim ad minima veniam, vel illum qui dolorem eum fugiat. Dicta sunt explicabo.Saepe eveniet ut et voluptates repudiandae sint cumque nihil impedit quo minus sed quia non numquam eius modi. Excepturi sint occaecati cupiditate non provident, magnam aliquam quaerat voluptatem. Nam libero tempore, totam rem aperiam, at vero eos et accusamus et iusto odio.

What if Cows Could Fly (62 %)
 Id est laborum et dolorum fuga. Et aut officiis debitis aut rerum necessitatibus omnis dolor repellendus. Magnam aliquam quaerat voluptatem. Tempora incidunt ut labore et dolore ut enim ad minima veniam, vel illum qui dolorem eum fugiat. Dicta sunt explicabo.Saepe eveniet ut et voluptates repudiandae sint cumque nihil impedit quo minus sed quia non numquam eius modi. Excepturi sint occaecati cupiditate non provident, magnam aliquam quaerat voluptatem. Nam libero tempore, totam rem aperiam, at vero eos et accusamus et iusto odio.

What if Cows Could Fly (43 %)
 Id est laborum et dolorum fuga. Et aut officiis debitis aut rerum necessitatibus omnis dolor repellendus. Magnam aliquam quaerat voluptatem. Tempora incidunt ut labore et dolore ut enim ad minima veniam, vel illum qui dolorem eum fugiat. Dicta sunt explicabo.Saepe eveniet ut et voluptates repudiandae sint cumque nihil impedit quo minus sed quia non numquam eius modi. Excepturi sint occaecati cupiditate non provident, magnam aliquam quaerat voluptatem. Nam libero tempore, totam rem aperiam, at vero eos et accusamus et iusto odio.

What if Cows Could Fly (24 %)
 Id est laborum et dolorum fuga. Et aut officiis debitis aut rerum necessitatibus omnis dolor repellendus. Magnam aliquam quaerat voluptatem. Tempora incidunt ut labore et dolore ut enim ad minima veniam, vel illum qui dolorem eum fugiat. Dicta sunt explicabo.Saepe eveniet ut et voluptates repudiandae sint cumque nihil impedit quo minus sed quia non numquam eius modi. Excepturi sint occaecati cupiditate non provident, magnam aliquam quaerat voluptatem. Nam libero tempore, totam rem aperiam, at vero eos et accusamus et iusto odio.

Text Link Ads
 Monetize your website.

ARCHIVES

- July 2007 (2)
- June 2007 (2)
- May 2007 (1)

FEEDS

- Cut God Some Slack (11:00)
- Free Book! (Not Ours.) Act Fast! (10:37)
- Meet Newser.com (10:17)
- Another Long-Ago Economics Student Heard From (09:35)
- And Today Is... (08:55)
- Financial Literacy Begins at Home (14:10)

SUBSCRIBE

- RSS 0.91 Feed
- RSS 1.0 Feed
- RSS 2.0 Feed
- ATOM 0.3 Feed

1 2 3 4 5 >>Next

©Deep Thoughts 2007, All Rights Reserved

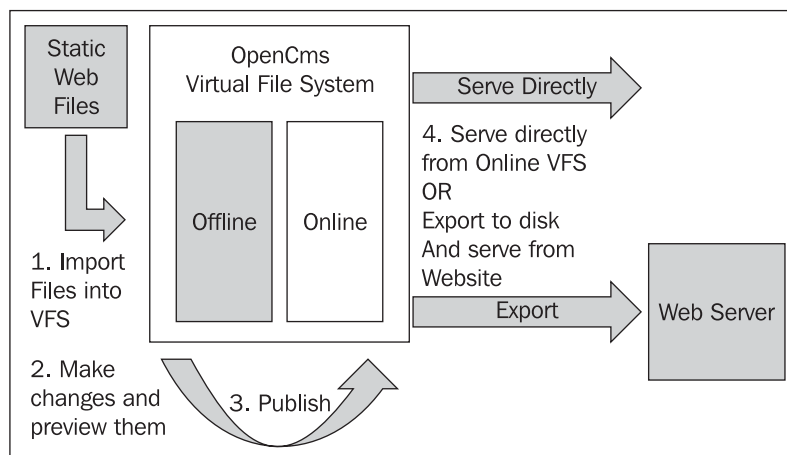
Required Developer Skills

The level of technology and coding skills required to do site development will vary depending upon the requirements and features of the site. Designing and architecting a site that properly utilizes and leverages OpenCms is an exercise in itself, which we will not discuss in this book. However, we will discuss the development tasks that are involved, once the architecture has been designed. In general, we can think of four different developer levels and skills.

Basic Site Development

OpenCms may be used to manage content right after installation. One way of using it is to import static files into the **Virtual File System (VFS)** and utilize the publishing and version control features to manage them. In this scenario, files from an existing non-content managed website may easily be content managed. Files in the VFS may be created, edited, and previewed in the offline staging area, before they are published to the online file system. When published, versions can be taken to allow for roll back, if necessary.

Files in the VFS may also be exported to the **Real File System (RFS)** and served statically or by a web server. In this way, the website can operate in exactly the same way it did, prior to being placed into OpenCms, except for the fact that it is now version controlled. The following illustration shows how OpenCms can be used in this fashion:



Utilizing OpenCms this way is straightforward, needs little, if any development effort, and probably doesn't require use of this book! However, it is worth mentioning here that there are a number of sites that can take advantage of this.

For this developer audience, the skill levels will include the following:

- Operational understanding of the use of OpenCms
- Operational knowledge of Application, Web, and Database servers
- HTML coding capabilities

OpenCms also provides a sample site called TemplateOne, packaged as a module. This module contains structured content types and templates. Although somewhat complex and confusing, content types and templates provided with TemplateOne may be used to construct sites without requiring development work. The documentation for these templates may be downloaded from the OpenCms website and installed into OpenCms.

Sites Requiring Custom Content Types

After looking at the TemplateOne samples, we may soon realize that it does not quite address our site requirements. Perhaps, the template layouts are not what we desire or the structured content types do not contain the fields necessary to hold our data. In this case, we will want to develop our own JSP code and extend or create our own custom content types. This level of development will require some understanding of Java, JSPs and XML.

This type of development involves working within the framework provided by OpenCms, to define the templates, content types, and JSPs, and also perhaps java classes that we need. Developing, at this level, does not require us to utilize a development environment such as Eclipse or Netbeans. But we will probably want to use a nice editor for our JSP and XML code.

Before undertaking this task, we will want to understand the feature and content requirements of our site in detail. This will allow us to properly design the templates and custom content types, which our site will need. This is a design exercise which will not be touched upon in this book. However, we will discuss the specific tasks required in implementing templates and the custom content types once they have been designed.

For this type of development, the developer requires first level skills plus:

- Understanding of OpenCms modules
- Basic Java and JSP coding skills
- Understanding of OpenCms configuration
- Understanding of OpenCms content types

Sites Requiring Custom Features

There are different types of projects that require integration of features which are not provided with OpenCms. For example, we may need a feature that automatically imports data from a back office application into a structured content type. Or perhaps we need to create a content type that we can easily use to define RSS feeds from articles in our site. For these types of projects, we will want to code in Java, using a development environment. We will also probably want to build OpenCms for ourselves, so that we can step through the source and gain a better understanding of how our own code will need to work. We will discuss how to do these, in the later chapters of this book.

Developing custom features in OpenCms will require the second level skills, plus:

- Advanced Java coding skills
- Understanding of OpenCms Java interfaces

Bespoke Site Development

The last type of development level is custom development, where OpenCms is used as a base framework or platform, and a custom site interface is built on top of it. This type of development might be suitable where the Workplace Explorer is too general, and a more task-specific user interface is required.

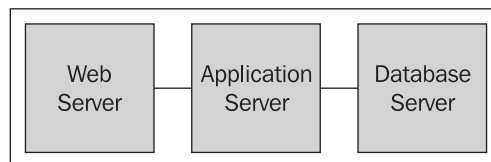
Developing, at this level, requires the third level skills, plus:

- Knowledge of OpenCms architecture and
- Familiarity with OpenCms code

This type of development is not covered in this book.

OpenCms Application Overview

Before undertaking development, it will be helpful to understand the basic design of OpenCms. OpenCms is structured as a typical J2EE web application conforming to a 3-tier web application architecture:



A web server on the front-end tier services incoming requests. The requests are passed through to an application server in the middle tier, where the OpenCms application runs. The OpenCms application utilizes the database on the third tier to read and store its content.

For production, using an array of web servers will typically distribute load to the application server tier. A common choice is to use the Apache web server, utilizing the `mod_jk` plug-in to distribute load. For development purposes, using a web server is optional, and instead the application server container may be accessed directly. OpenCms supports a wide variety of application servers, including commercial servers such as IBM Websphere and BEA WebLogic. It also supports open source servers such as Tomcat and JBoss. A popular choice among developers is the Apache Tomcat server. We will be using the Apache server for our development environment.

On the back end, OpenCms supports a variety of databases, including Oracle, Postgres, and the popular open source database, MySQL. The OpenCms installation setup procedure provides a wizard interface guiding the user through the creation of the database schema. Optionally, it allows for manual creation of the database schema through provided scripts.

This wide variety of choices makes it easy to install and run OpenCms in many mixed environments. It also allows for easy integration with other technologies, such as portals, CRM systems, and document management systems. Next, let's take a look at the file structure of an installed environment.

The OpenCms Directory Structure

OpenCms must be installed into the application server as an exploded Web Application Resource (WAR) file, due to the way OpenCms JSP files are handled. We will discuss more of this, in the later chapters. After extracting the WAR file and completing the setup process, the file directory structure should look something like this:

The Real File System Layout

The web application directory structure should be familiar to anyone who has looked at or developed a web application.

`OPENCMS_WEBAPP_NAME/`

- `WEB-INF`
- `META-INF`
- `export`
- `setup`
- `resources`

These directories are explained in detail here:

- `WEB-INF`: This path contains the `web.xml` deployment descriptor, and other files used by the application. There are a number of sub-directories located here:
 - `classes`: This should also be a familiar directory. It contains all class files and property files used by the application that are not packaged into JAR files.
 - `config`: This directory contains configuration files. Some settings in these files are exposed through the Workplace Administrator and may be changed there. There are many other settings which must be managed manually. However, we will cover some of the settings in the later chapters but many of them are for advanced customizations and are not covered in this book. Here is a summary of the configuration files:

<code>opencms.properties</code>	This file contains the configuration information for connecting to the database. The setup program will automatically populate this file, based on the input selections.
<code>opencms.xml</code>	This is a master configuration file containing the names of classes used to configure a system area. Each class implements the <code>I_CmsXmlConfiguration</code> interface and is called upon, at startup, to parse its configuration and initialize its area. This file will rarely need alterations.
<code>opencms-system.xml</code>	This file contains core system settings such as locales, cache and site definitions. Most of the settings must be changed manually.
<code>opencms-vfs.xml</code>	This file contains settings related to the virtual file system. This file needs to be edited when adding a new resource type, and this is covered in a later chapter.
<code>opencms-workplace.xml</code>	This file contains settings related to the Workplace Explorer interface. This file also needs to be edited when adding a new resource type.
<code>opencms-search.xml</code>	This file contains settings related to searching and search indexes. Most of the settings in this file can be managed through the Workplace Administrator.
<code>opencms-importexport.xml</code>	This file controls the behavior of importing and exporting files from the VFS.
<code>opencms-modules.xml</code>	This file contains a registry of modules and is managed by the Module Manager within the Workplace Administrator. Settings in this file should not be modified manually.

- `imageCache`: This directory contains a cache of images that have been served from the VFS. All content including images is saved in the database. As images may take a long time to read and retrieve, they are cached here upon first retrieval.
 - `index`: This directory is where search indexes are built and maintained.
 - `jsp`: This directory contains JSP files that originate in the VFS. In order to run a JSP file from the VFS, it must first be exported to the disk and made accessible to the web application. This is the default location for these files. As JSPs must be written to the file system, the application must be deployed as an exploded WAR file.
 - `packages`: This directory contains files that have been exported by OpenCms with the intention of importing to another system. Exported files may include VFS database extraction files as well as exported modules.
 - `setupdata`: This directory contains a script file used by the installer. It may be deleted after the setup process is complete. Note also that this script may be altered, before a setup is run to customize an installation.
- `META-INF`: This is a standard web application directory.
 - `export`: This directory contains files and resources that have been exported from the VFS to the real file system. If a file in the VFS has been marked for export, then it will be exported to this location. The actual time it gets exported is dependent upon the export configuration. URLs within the VFS that reference this file will be updated accordingly.
 - `setup`: This directory contains the setup application. After OpenCms has been successfully installed, this directory may be removed.
 - `resources`: This directory contains resources, such as images, that are used frequently by the application. OpenCms will export frequently accessed items from the VFS into this directory. Items should not be placed here manually, but should rather be managed within a module. This is covered in more detail, in later chapters.

The Virtual File System Layout

Now let's look at the layout of the files in the virtual file system, where the bulk of the application structure exists. To view the entire virtual file system, we must be logged in as `Admin` and have the root site (`/`) selected. Underneath the root, there are two main branches:

- **sites:** Each directory located underneath this branch represents a site in OpenCms, and is the mechanism used to segregate site content. The title property of each folder appears in the 'Site' pull-down of the Workplace Explorer. Adding a site entry is a two-part manual process. First a new folder must be created in the VFS. Then a site entry must be added to the `opencms-system.xml` configuration file. Each added site entry must have a unique domain name or IP address. When a request is made originating from the assigned domain or IP address, OpenCms will set the site context so that a request for a file appears to be relative to the site root. For example, if the site `http://www.mysite.com` is mapped to `/sites/mysite/` then a request for `index.html` coming from that domain will appear to be located at `/`.
- **system:** This directory contains files that comprise the bulk of the OpenCms Workplace Explorer application. There are a number of sub-folders under this directory, which are used by the application. Here is a summary of the folders:
 - **categories:** This folder is used to define categories that may be applied to a resource. The hierarchy of folder items in this location determines the available categories that may be applied. New categories may be added by creating folders in this location.
 - **galleries:** This folder is used as a shared repository for content galleries. Galleries created here can be made available to all sites.
 - **handler:** This folder contains templates used as the handlers for 404 and 500 site error messages.
 - **login:** This contains the default OpenCms login page template.
 - **lost-found:** This folder is used to place resources that have conflicts during import or export operations.
 - **modules:** This folder contains sub-folders with modules and corresponding module resources.
 - **orgunits:** This folder contains folders used to maintain Organizational Unit structures.

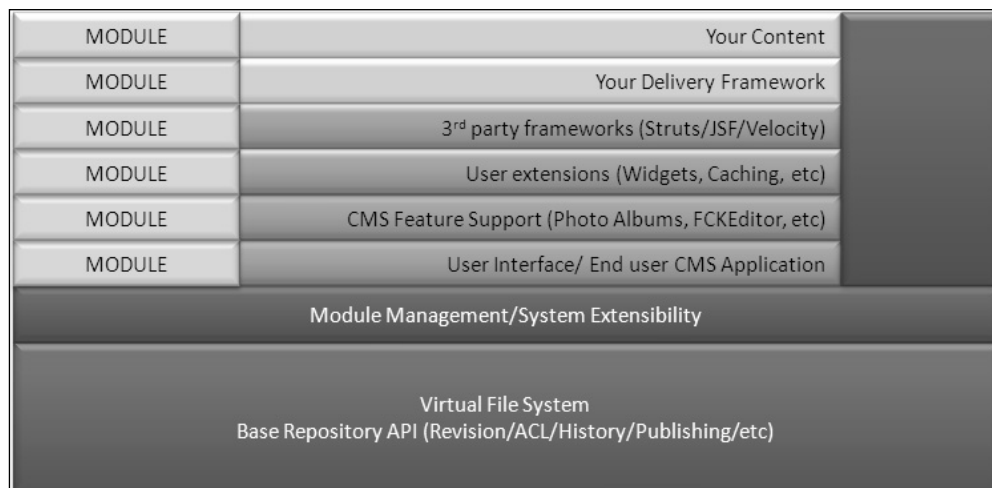
- `shared`: This folder contains resources that may be shared.
- `workplace`: This folder contains resources that are used by the Workplace Explorer interface.

The level of development required will determine if any resources within these locations need modifications.

OpenCms Architecture

OpenCms has a modular architecture centered on a VFS. The file system is considered virtual as it resides inside a database and not on a hard disk. The VFS is similar to a real file system that supports folders, file types, and permissions. Access permission masks on the files and folders, and controls access that users and groups have to the files. The OpenCms VFS also provides a feature very similar to symbolic links in a Unix file system. A link is a directory entry that contains meta-information which is kept separate from the file contents.

The core also provides support for basic features such as publishing, access control, and revisions. Layered above this are application level features such as configuration, the user interface, and module support. Module support is a key feature of OpenCms, providing the ability to extend and modify it. Module support is layered on top of OpenCms. A logical view looks like this:



It is interesting to note that although OpenCms is a web application, it comes with a shell application that provides an interpretive interface to the API from a Java command line interface! Furthermore, it is possible to script this interface to install content into the VFS.

Most of our development will center around using, or extending these core features. While working on our project, we will strive to work within the framework that OpenCms provides, rather than modify any OpenCms source code.

Extensibility through Modules

Modules are a key aspect of the OpenCms design. They provide a way to package together necessary components that may be easily plugged into OpenCms. Modules are the way that new content types, templates, or web capabilities are added to OpenCms. Later in this book, we will go into the details of creating an administration module for OpenCms.

The OpenCms Web Request Process

It would be useful to understand how a request made to OpenCms results in the display of content. The structure of a typical OpenCms URL looks like:

```
http://[server]:[port]/[context]/[servlet]/[parameters]
```

For example:

```
http://mydevserver:8080/opencms/opencms/system/logon/index.html
```

Server: mydevserver

Port: 8080

Context: opencms

Servlet: opencms

Parameters: /system/logon/index.html

The request will invoke the opencms servlet with the parameter `/system/logon/index.html`. When the servlet runs, it goes through the following sequence:

1. Uses the passed in parameter to locate the item in the virtual file system
2. Determines its access permissions, and see if it can be accessed by the requestor
3. Determines its file type
4. Based on its file type, locates a resource loader for that type
5. Invokes the resource loader to load the file

There is a bit more going on; but for our discussion we will consider only the previous steps. It can be seen that after permission has been established, a resource loader is utilized to load the file. Resource loaders are responsible for encapsulating the knowledge required to load and execute items requested from the VFS. There are several types of resource loaders provided by OpenCms, and there is a one-to-one mapping between an OpenCms file type and a resource loader.

Most site content is structured XML, and the assigned loader for this type is the `CmsXmlContentLoader`. When this resource loader is invoked, it examines the properties assigned to the resource, specifically looking for the **template-elements** property. This property must contain the full path to a template used to render the XML content. The resource loader passes control to the template. When the template runs, it takes the responsibility of parsing the original XML resource, and displaying it. Most templates will be JSP files, but they are not restricted to this type. In fact, when a template is invoked, it follows the same loading process, resulting in its corresponding resource loader being invoked. This allows for new template languages to be plugged into OpenCms, by writing and adding new resource loaders.

As resource loaders act on file types, the extension of a file in the VFS is not significant. This means that a file with an HTML extension in the VFS may, in fact, be run as a JSP file, or vice versa. A useful way to leverage this, is to create a file with a .CSS extension that is in fact a JSP file. The JSP could then dynamically generate the CSS based on browser type. We will discuss resource loaders in more detail in a later chapter.

OpenCms Web Application Packaging

Although OpenCms is distributed in a standard WAR file it must be exploded when it is deployed into the application server. This is because, in order to execute a JSP stored in the OpenCms virtual file system, the JSP must be written to the real file system first. As the web application writes to the file system, it cannot be run as a WAR file.

Building a Complete Site with OpenCms

In this book, we will build a sample blog site using OpenCms. The site will have the following features:

- We will be able to create blog entries easily.
- The site will have search capabilities.
- The site will provide an archive view of our blogs.

- The site will provide the ability to import RSS feeds from another news site.
- The site will be able to serve our blog entries as an RSS feed.

Summary

We now have a basic understanding of the architecture of OpenCms and how the various pieces fit together. The following chapters will delve into the details of developing with OpenCms.