Administration Guide

ScientificCMS
Content Administration Series

Draft

Administration Guide

Version 2.0

This guide contains instructions and reference information to administrate a ScientificCMS installation. It is focused on Red Hat Enterprise Linux or Fedora Linux distributions but may be used for other Linux distributions or Unix operating systems as well. It also contains some hints for working on the Windows operating system.

ScientificCMS:
Content Administration Guide Release 2.0

Release 1.9.1 (2012-05)

edited by: Peter Boy, University of Bremen (pb@zes.uni-bremen.de)

Acknowledgments

This guide uses material from administration manuals for previous versions of Red Hats documentation for the Collaboration and Content Management system (CCM) and APLAWS+, a system specifically tailor-made for Great Britain Local Authorities based on CCM. Large parts had been written by Oliver Sharpe (Red Hat) and Arturo Dell (Camden). It also contains information, gathered from various contribution of users, namely on discussion fora and wikis.

Document History:

Version	Date	Origin	Written by	

Table of Contents

Preface

What is ScientificCMS

ScientificCMS is an open source application that makes the creation and maintenance of websites easier for academic research institutes, at the same time that it facilitates scholarly communication and collaboration.

ScientificCMS shares the codebase and cooperates closely with the *APLAWS+* project, a CMS that was designed for local governmental authorities in the United Kingdom. Both projects are user-group-specific adaptations of (Open)CCM, a general Web content management framework, which was originally developed at Massachusetts Institute of Technology by Philip Greenspun and others.

Scope of This Guide

This guide contains step by step instructions as well as reference information to maintain content in each of the various modules of a ScientificCMS installation. All but a few tasks are performed in the WEB based graphical user interface. Some – rarely required – tasks, especially setting software configuration parameters, have to be done on the server in a command line environment. Their descriptions are focused on Red Hat Enterprise Linux or Fedora Linux distributions but may be used for other Linux distributions or Unix operating systems, as well. On Windows operating systems the necessary step are very similar and can easily performed accordingly. Some hints are given, nevertheless.

Specifically, this guide provides

- an overview about the WEB Content Management System application (CMS). It is often considered the main application.

In the examples we assume an installation on a standalone server for sake of simplicity. If your ScientificCMS instance is running in a cluster and multi tier environment, you have to modify the instructions accordingly by choosing the correct server.

Intended Audience

This guide is written with the publisher, site planner, or information architect, i.e. the content administrator, in mind who have to plan and configure a site. He or she needs a sound knowledge of the intended informational purpose of the site, the organisation of people contributing, and the decision making processes. This guide should support her or him to acquire the required knowledge of the internals of ScientificCMS. It provides a high level, overall perspective.

- Introduce the APLAWS+ Content Management System
- Describe the features of the CMS
- Define the terminology of the CMS
- Explain the concepts of content management
- Describe sample workflows that you can use when first setting up the CMS, maintaining it, and customizing it
- Provide step-by-step instructions for the different tasks

A content administrator must either be the (software system) administrative user in ScientificCMS or at least member of the system administrator group. For those tasks to be performed on the server systems command line, limited administration privileges on the server machine are required as well (or assistance of a server system administrator).

Release Information

This guide is specifically based on **version 2.0** of ScientificCMS but should be applicable for all 2.x versions. As soon as changes or new functions will be included, a new version will be released.

Content Administration Series

The Content Administration Series provides information for the publisher and information architect how to set up and configure the software internals as well as the processes to publish adequate content, choose the appropriate tools to deliver information online and organize people working on it.

Previous Versions of This Guide

This guide is based on several manuals created by Red Hat for the CCM framework as well as by the sister project, APLAWS+. It updates the information and adds a description of procedures and functionality that were missing there. Especially it supersedes

- the "LAWS Administration Guide For: CMS Administrator Guide (Last Updated: 24/03/2004) (LAWS_CMS_CMS_Admin_guide.pdf)
- "Red Hat Content Management System 6.1 – CMS Administrator Guide", 2004 (rhea-ag-cms-en-6.1-1.pdf)
- "Red Hat Content Management System rickshaw – CMS Administrator Guide", 2003 (rhea-ag-cms-en-rickshaw-1.pdf)

Part I
Introduction and Planning

This part introduces into ScientificCMS, describes it's features, and defines concepts and terminology.

It will enable you to decide, weather ScientificCMS might fulfil your needs, to develop your own working strategy, and give you a picture what has to be done to get the system up and running.

1. Introduction

ScientificCMS is an open source application that makes the creation and maintenance of websites easier for academic research institutes, at the same time that it facilitates scholarly communication and collaboration.

Research institutions have particular needs and characteristics that complicate the creation and maintenance of a website, and can pose particular problems for large academic institutions:

- They deal with a large volume of highly complex scientific and scholarly material.
- They must be able to process information from publications and research projects according to the precise rules of various disciplines, a process that can be extremely labour intensive.
- Information may be equally relevant in a number of different context. A publication, for example, may be the result of work in several research projects.

Such institutions require a detailed **information processing** system that allows users to easily access and search this high volume of complex material. The presentation of information must be adapted to Web browsers and website reading habits, and information that is required on several different pages should nevertheless be accessed from a single source. Furthermore, because researchers tend to work collaboratively with individuals at other institutes, often in distant lands, Web technology should be fully exploited to provide a forum where a group of individuals who are collaborating on a project can share texts and data, or edit and comment on a common text.

Most popular content management systems (CMS) are based on individual Web pages, making them unsuitable for these purposes. The idea of ScientificCMS, on the other hand, is to **manage content not pages.** Information is stored as specially formatted **documents** that are organized by **category** and automatically prepared for Web presentation.

ScientificCMS shares a codebase and cooperates closely with the *APLAWS+* project, a CMS that was designed for local governmental authorities in the United Kingdom. Both projects are user-group-specific adaptations of *(Open)***CCM**, a general Web content management framework, which was originally developed at Massachusetts Institute of Technology by Philip Greenspun and others.

2.

The CCM Framework

The **Content and Collaboration Management System** (*CCM*) is an open source framework and set of applications for managing the creation and delivery processes of various kinds of content and integrates it into one or more websites as well as WEB based collaboration. It is specifically designed to manage *large* and *complex* content. It includes version control, archive function, defined work processes, safety features, style sheets, templates and much more.

2.1 Features

Features that set the Content and Collaboration Management platform apart from other open source content management systems and make it specifically suitable to handle extensive and complex information include:

Information Centric

> The system is designed as an pool of information upon which different views can be set up to build a specific web site. Other content management system are build as a managed set of web pages.

Strong Categorization System

> The system allows for poly-hierarchic categorization and builds an *information retrieval* system which makes complex content available to website visitors with varying interests and perspectives.

Semantic Structure

> Information is organized into "*content types*" based on its semantic structure, this allows information to be accessed and systematically reused in different contexts.

Modular and stable API

> All functions are provided in modules interconnected by a strong, long living, and well documented API. General purpose modules and application or customer specific modules can thus be combined to produce a stable system that is specifically tailored to the users or to various applications domains needs.

Enterprise Grade

> The system is ready to run in a multi tier enterprise environment providing Load Balancing and High Availability.

At the level of **work organization**, the CCM framework establishes clearly defined roles and documented and transparent roles, permissions and work-flow processes. The processing of all content is clearly and completely separated from technical or web-specific features and functions.

By enabling and allowing every individual to create and maintain content, a large website can have up-to-date content in every area. The individuals closest to the content are responsible for updating and modifying it and can work on the task without having to rely on web experts.

2.2 Structure

Characteristic for the CCM framework is its *modular structure*. The framework is basically a set of modules integrated by a well defined application programming interface (API).

Base is a core module, formerly known as Red Hat Web Application Framework (WAF), and JEE© technology.

On top of this base various modules to handle specific tasks. The most notable: Content Management System (CMS).

Additional: a suite of collaboration applications: forum, questionnaire, document management among others

A Set of integration application: portal, theme support

Kind of content it can handle: Web pages, document repositories, discussion forum, conduction of questionnaires.

high integration, same content structure throughout the system.

2.2.1 Core

The core module (Web Application Framework) handles functions such as

- User and group management
- Permissions services
- Work-flow services
- Template and deployment services

These functions may be used by all modules based on it. So it delivers a sound integration layer for all modules, allowing to exchange content and functionality between modules or to use a single authentication and authorisation for all modules.

On top of the core other modules (or applications) implement additional, task specific functionality, especially a set of *content management applications,* specifically the CMS application which combines powerful Web content management functionality with the flexibility to tailor deployments to the specific product environments and processes of your organization. The CMS application is accompanied by other modules, which manage or otherwise support the presentation of the content in various ways, e.g. the navigation module which presents content in a very systematic, straightforward way. Together they make the creation, maintenance, and publication of websites easy and efficient to perform.

Another type of applications, the *collaboration applications*, support various ways of WEB based communication, among others a discussion forum, a weblog, or a document repository including meta-data. In combination these modules facilitate scholarly communication and collaboration.

2.2.2 CCM Content Management Applications

The CCM content management applications provide end-to-end functionality for managing all types of Web content, establishing clearly defined roles and documented *work-flow* processes. The applications enable *content creators*:

- Work with an easy-to-use, configurable Web interface that works in a variety of browsers on different platforms

- Author content in multiple languages
- Manage the work-flows for content creation, editing, and publication processes
- Categorize, version, and search for content throughout the system
- Manage content life-cycles
- Syndicate content via RSS
- Easily extend and configure the application to meet specific business requirements

The most important characteristic from an administrators point of view is **flexibility and control**.

As the needs of your organization grow, additional functionality can be integrated. Use the CMS straight out of the box, adding the necessary pieces for your company on demand by choosing the optimal system configuration.

As the needs of your organization change, you can take advantage of the flexibility and control, customizing the CMS as desired. Configurable aspects of an instance would include:

- The work-flow, or set of tasks, required for production. Common tasks include entering meta-data, authoring content, selecting keywords and categories, and associating images and other media assets.
- Rules for deploying content, such as publishing immediately or according to a schedule
- Staffing, such as who is assigned or authorized to perform specific tasks

The software allows different *instances* of the application to be mounted under different managed content sections of the overall site. Each instance can be configured differently to handle a particular production process, yet all interface with a single *content repository* (database).

You configure projects and departments through an administrative interface, called the *Content Center*, that is also listed as an application. Administrative authority is limited to the same branch of the site that includes the Content Center. This flexibility allows administrators to adapt the CCM Web Application Framework and the CCM Content Management System to existing business processes, rather than forcing your organization to adapt to processes imposed by inflexible software.

You can also create a unique look and feel for your websites and easily extend the data model, business logic, and presentation layer as your business grows and requirements change, using a CMS that does not require modifications when upgrades are released.

With the CMS, you can have your company running a fully-functional content management system in as little as two months.

2.2.3 CCM Collaboration Management

(coming soon)

2.3 General Capabilities

Accessible Output

The CMS produces standard HTML files that have been used to power Section 508 sites.

Archive

The CMS includes a system to save older versions of content items, enabling you to rollback if necessary. You can view the differences between versions. See Version Control.

Category Management

Create organizational categories for content items, which are browsable and searchable.

Content Reuse

You can reuse images throughout CMS, using the image authoring step for articles.

Cross-browser Support

The HTML pages produced by the CMS can be viewed by various browsers on different platforms.

Database Repository

The CMS uses a database to store content items (text, images, hyperlinks, and so on), which enables easy retrieval, linking, and searching capabilities.

Extensible

The CMS includes documented APIs.

External Systems Integration

You can integrate LDAP, Kerberos, Web services, RSS, and other portlets designed to be used by developers. See the *CMS Deployment Guide*.

Form Builder

The CMS includes a form builder wizard for creating online forms.

Integrated Authoring Environment

An easy-to-use form-based interface produces structured content, which includes text, images, hyperlinks, and so on. Content creators add information within fields, which prevents incorrect structure and formatting, and can upload content from other applications, such as Microsoft Office and OpenOffice.

Lifecycles

The CMS has a highly flexible publishing model for managing content lifecycles, enabling users to schedule publishing and removing of content items, and their durations and date ranges.

Linking

The CMS includes a RelatedLink type, which allows any content item to have a sortable list of "related links."

Managing Users, Groups, and Roles

The CMS can help you manage an unlimited number of groups, users, and roles in a simple, coherent way that is fully integrated with the access control system. It provides role-based permissions at multiple levels including content item granularity, supports integration with LDAP authentication, and provides security and permissions for groups and users.

Metadata creation

The CMS supports the arbitrary import of metadata specifications, including Dublin Core.

Multilingual

The CMS has native support for managing and syncing content in an unlimited number of languages. All static strings in the CMS are globalized, enabling easy localization. Content items are grouped into content bundles, which enables you to publish a single piece of content to different languages.

Multiuser authoring

Users can create, edit, and modify the content they are responsible for while others are working on their items. The CMS tracks which users are working on specific items and prevents other users from making simultaneous modifications to the same piece of content.

Notifications

Include automatic email notifications for workflows and tasks. You can configure the CMS to send emails when a task is overdue, enabled, disabled, finished, or rolled back.

Publishing and Removing Content Items

The CMS includes a default publishing interface that can produce customized output. You can schedule content for publication and removal by using content lifecycles. Preview content items before publishing, using the associated templates.

Scalable

The CMS, built on proven J2EE architecture, has built-in support for HTTP cache headers and publishing to a file system to maximize the scalability. Extensive testing has shown the CMS to be capable of handling large numbers of users and administrators in a variety of roles.

Search

The CMS supports full-text searching of all content, regardless of the database. This includes Intermedia on Oracle® for full-text search of text and binary files, the open source Lucene search engine for database-independent searches, and support for navigation to content based on categories.

Security

A single access control system allows administrators to specify which groups, users, and/or roles can perform specific actions by assigning permissions. Administrators can delegate administrative privileges to others. The CMS provides a standard content contribution wizard for authors that uses workflows to determine the content creation process.

Separation of content and presentation

The CMS includes a standards-based Cascading Style Sheet template system which uses either XSLT or JSP™ templates to consistently format content. The CMS supports an unlimited number of custom content templates for publishing to any channel (for example, Web, WAP, and PDF). You can upload your own HTML templates and the CMS supports pre-populated and user-defined content types.

Stylesheets

Use the default stylesheets or customize them to match your corporate branding.

Support for Multiple Output Formats

The CMS produces XSL output, which can be transformed to HTML pages, PDF files, and more. It also supports RDF/RSS standards for syndicating content.

Version Control

The CMS includes the WAF versioning system that recognizes the differences between iterations of content items, which are displayed on the History tab in the user interface. Developers can use this data to build conflict resolution and branching capabilities.

Workflow

Content moves through a series of steps during the creating process. Each step can be assigned to a different role or user. This lets you design a customized, role-based workflow that includes routing, email notification, and task assignments. The CMS documents a full audit trail.

2.4 History

It was developed from the ArsDigita Community System (ACS) that was initiated at MIT by Philip Greenspun, who, together with several MIT colleagues, founded the company ArsDigita. ArsDigita was later acquired by Red Hat, and ACS was developed into the Red Hat Content and Collaboration Management System. As part of the changes in Open Source software distribution Red Hat made CCM a **community driven project** in 2004. Several projects emerged but all development soon concentrated at the *APLAWS* project, a special configuration of the CCM that was designed to enable local governments in the U.K. to deliver services online and incorporated into the Local Authority Websites National Project. *APLAWS* absorbed most of the development efforts over time and evolved into the central place of development. The source code is hosted as part of fedorahosted.org now.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 2
- Red Hat Content Management Systems 3.1: CMS Administrator Guide, chapter 1
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 1

It supersedes these guides and provides updated information.

3. Understanding the CCM Framework – Underlying Concepts

You will use and have to understand the following concepts in CCM:

- Content Type
- Content Item
- Assets
- Category
- Lifecycles
- Permissions and Privileges
- Templates
- Users, Groups, and Roles
- Workflows

3.1 Content Type

Content types are the backbone of the CCM framework. During installation the necessary content types are included. You can add new content types by installing the appropriate packages.

Content types map a specific *type of information*, e.g. an event, and represent their semantic structure in a set of properties. The CCM framework distinguishes unique **basic properties**, which specifically assemble the typical information elements, and more general **additional properties** (e.g. a free text description) as well as global **associated properties** (e.g. related links or downloadable files).

When you create a piece of information to publish on the site, you specifically create a concrete instance of a content type, the *content item* (see below) and you have to specify it's content type first. This loads the appropriate information for the item, including the unique, **basic properties** that the author will fill in. These basic properties provide consistency across CCM applications so that all content items of a specific type include fields for the necessary information and are displayed the same.

In addition to the unique basic properties of each content type, you can also use global **associated properties**, e.g. categories, related links or any metadata scheme (The system *optionally* offers the Dublin Core Metadata Initiative – DCMI).

Each default content type is associated with a *parent* content type. You cannot modify these content types and you will only use them if you add new content types programmatically. The CMS includes the following parent content types:

- **ContentPage**
  Lets you add different types of content to the content item, including images and text.
- **GenericArticle**
  Lets you add information appropriate for an article, including a captioned image.
- **GenericPerson**
  Lets you add information appropriate for a person, including a captioned image.
- **GenericOrganizationalUnit**
  Lets you add information appropriate for an organization.
- **GenericAddress**
  Lets you add information appropriate for a generic address, which may be associated with a person or an organisation.
- **GenericContact**
  Lets you add information appropriate for basic contact information, which may be associated with a person or an organisation.

Each content type includes specific **basic properties**. If one of the default content types does not meet your needs, you can add two kinds of new content types:

- Add new content types through the UI, known as a **user-defined content type** (UDCT). You can base the UDCT on an existing content type or create it from scratch. Make sure you specify a lifecycle before it is time to publish any content items based on the content type or users won't be able to publish it.

  Note
  If you don't base the UDCT on an existing content type, you

have to add the appropriate authoring kit items or users won't be able to add any content to the content item.

After creating the child content type, modify it to meet your needs.

- Add new parent or child content types **programmatically**. You typically have this done when you have specific requirements, so you have to create *custom content types* to meet your needs. The CCM framework supports the creation of *custom content types* by a strong API and capable built-in reusable components. For more information, see the CCM *Developer Guide*.

Various content types are available, among others:

- Address
- Agenda
- Article
- Event
- FAQ Item
- File Storage Item
- Glossary Item
- Job
- Legal Notice
- Minutes
- Multi-part Article
- News Item
- Organization
- Press Release
- Service

For a detailed description of all content types refer to chapter 3.1

Be aware that many content types have their own configuration options or even an administration interface. So a content item may have it's own administration tasks!

3.2 Content Item

The content is comprised of your content items, which is a set of information that can include other content items, links, images, file attachments, and more, known as *assets* (cf. below). Each content item includes basic properties as defined by its content type, which include workflow, lifecycle, and template. Refer to 3.1 Content Type.

3.3 Assets

Assets are any object that you can use to add content to the web page. In contrast to content items they don't map a self-contained meaningful type of information, but a deliberate / coincident piece of information. This includes files and images.

3.4 Category

Categories are an extremely powerful way to organize the content items for the visitors who will use your public site, and one of the most important *building components* of the CCM framework.

For example, one way to design a category structure is to include broad categories at the top of the structure, followed by narrower categories. Think carefully about your categories and be consistent.

Typically, categories are managed by someone in an information architect role. Discussion on information architecture is beyond the scope of this guide.

The CCM framework lets you create **category hierarchies**, which lets you provide an overall purpose for the category, resulting in a **thesaurus** or even a **taxonomy**.

You can also link one category to another to create a relationship between categories in different parts of the tree. For example, suppose that two of the subcategories on your category tree are Literacy and Education. By linking the Literacy category to the Education category, users can search Education and get results from both categories. Literacy would be displayed as a child to both its own parent category and to Education. By this technique a **polyhierarchical thesaurus** may be created.

The CCM framework is capable to support and provide **several independent** category hierarchies, resulting in multiple thesauri or taxonomies.

Each category hierarchy is managed in the CCM framework either as "*use context*" or as a more advanced implementation known as "*category domain*" (terms).

Of course, you can assign a content item to more than one category and to more than one category hierarchy. For example, you might have a "Contact Us" content item that you want listed under multiple categories. You would create the content item and then assign it to the individual categories so that users would be able to access it from several contexts and perspectives. This also lets you take advantage of **reusability**, as edits made to the content item are reflected in all categories. This in turn enables **redundancy free** data storage.

View and manage subcategories

Subcategories let you further subdivide a category for more organization. For example, you might have a parent category of Entertainment that includes subcategories for Music, Movies, and Television. If Entertainment is a child category of your root category, then Music is actually a grandchild of the root level and Entertainment is both a parent (of Music) and a child (of root).

For subcategories, view and manage linked categories.

By linking one subcategory to another, you create a relationship between categories in different parts of the category tree by assigning it a virtual parent. Your users can search the "true" parent or the virtual parent and find results within that subcategory, which is displayed in both places on the tree. Users will not realize that the categories are linked.

3.5 Lifecycles

Lifecycles define the start and end publishing dates of a content item. If a content item is not assigned a lifecycle, it never gets published.

Lifecycles consist of at least one phase, which controls its start delay and duration. The start delay determines the amount of time between when a user with publish privileges publishes a content item and when the item is displayed on the website. The duration determines how long the item remains on the website.

While the default lifecycle functionality lets you control when an item gets published to and removed from the website, you can programmatically add more functionality. For example, you could create phases that publish the item for one week, move the item to a different area of the website after that week, and then move it again after a month. For more information, refer to *CCM System Developer Guide*.

3.6 Permissions and Privileges

A **privilege** is the ability to perform specific actions, such as create, edit, delete, or preview, on the different objects in the CCM System. Privileges are pre-defined and cannot be modified. You can only add new privileges programmatically.

Permissions are when you assign privileges to users, either individually, through their pre-defined groups, or through their roles. If you do not assign users any privileges, then they cannot do anything within the CCM System (except receive email alerts).

You can set permissions for the roles you define or the predefined roles at the content section level. You can also set permissions on a folder that are different from the global permissions on the content section. By inheritance, the items and folders within a folder you set permissions for will have the same permissions as that folder. So, if you specify that only a particular group can create items within a folder, that restriction will hold true for all folders within that folder as well.

Each application may manage their own set of privileges and provide permissions for a user to access its resources (e.g. access to administration pages or manage public and non-public access).

3.7 Templates

Templates are Java Server Pages (.jsp) or XSL files associated with content types (or content items) that specify which user interface components will be included in a given page of content. Creating a new template is similar to creating a new content item.

The CCM framework supports uploading modified XSL and JSP templates for each *content type*. If you do not upload modified versions, the installation defaults are used. Content items use the templates assigned to the appropriate content type, or the defaults if none have been assigned. The GUI enables an administrator to check what templates have been assigned to the content type.

3.8 Users, Groups, and Roles

When you manage permissions in the CCM, you assign users to a role. Each role has a particular set of privileges needed to create, maintain, publish, and view content. Developer usually create required default groups programmatically and include them in the installation process. Administrators may create additional roles and groups using the various user interfaces (e.g. core administration GUI or Portal administration portlet), which lets you further organize users. You can assign users, groups, and roles to specific workflows, thus streamlining the process for creating content.

3.9 Workflows

Workflows are composed of *tasks* that control the process of producing a content item from creation through publication. The CCM includes default workflows that you can use or modify, and you can design an unlimited number of workflows.

Tasks can be dependent on earlier tasks. These dependencies control how tasks are assigned to different users as the content item is processed. For example, if the authoring task must be finished before editing, and editing must be finished before deployment, then select the authoring dependency when creating the Edit task, and select the editing dependency when creating the Deploy task. You do not need to select both authoring and editing as dependencies for deployment. You can only enter dependencies if you have added more than one task.

As a content item progresses through the CMS, assigned users and users who are members of the assigned role must finish their task or the content item stalls. When a user assigned to the Deploy task locks the task, selects the **Publishing** tab, and applies a Lifecycle, the content item is published.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 5
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 4
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 4

It supersedes these guides and provides updated information.

Part II
Getting Started

Learn about the typical (suggested) workflows for the initial setup, customizing, and maintaining the CMS.

4. Application Administration Tools

There are two types of tools to perform administration tasks:

- GUI tools
- Command-line tools

Because of the modular structure each module has its own set of administration tools.

4.1 Application Administration GUI Tools

Many modules have their own GUI administration interfaces. This section lists all of the (often unlinked) administration pages.

By convention, site wide administration pages are under `/ccm/admin/`, while per-application administration pages are at `/ccm/[app url]/admin/`.

Table 4.1.: Administration Pages

URL	Description
/ccm/content-center	The content centre where you will manage the content of the site
/ccm/admin/	User and group administration
/ccm/admin/auth-http/	Administration of HTTP (NTLM/Win2k Kerberos) authentication. This is only present if the `ccm- auth-http` application is loaded
/ccm/admin/themes/	Administration of XSL Themes
/ccm/admin/shortcuts/	Administration of URL shortcuts.
/ccm/admin/subsite/	Administration of subsites .
/ccm/admin/terms/	Administration of terms (controlled lists)
/ccm/search/admin/	Administration of search sponsered links and SOAP remote search hosts.
/ccm/navigation/admin/	Administration of category navigation templates
/ccm/services/admin/	Administration of category services templates
/ccm/atoz/admin/	A-Z admin

URL	Description
/ccm/ds/	Web developer support
/ccm/portal/admin/	Administration of portal administrators
/ccm/portal/edit.jsp	Administration and styling of portal page.
/ccm/channels/rss/admin/	Administration of RSS feeds
/ccm/content/admin/	Administration of content
/ccm/forms/admin/	Administration of forms
/ccm/register/change-password	Change your password
/ccm/register/edit-profile/	Edit profile
/ccm/register/logout	Log out

Each administration task is described in the corresponding chapter(s) of this guide.

4.2 Command Line Administration Tools

This chapter details the functionality available in various command line tools.

`ccm`

`ccm-run`

5. Getting started – pre-installation tasks

Before you start to initially install the software, you should perform some preparational work.

5.1 Determine the modules to install

Read chapters 2 and 3 to determine the set of modules your installation requires. Then consult the installation manual to decide about the appropriate installation bundle.

5.2 Check the modules configuration

Check for each module, especially for each content type, whether the default configuration (or the installation default) is as needed. Otherwise determine the configuration parameters and the appropriate value to set them as a post-installation procedure.

6. Getting Started – Initial Setup

After the CMS has been installed, administrators should set up and define the following pieces of information before proceeding. This workflow has been designed to streamline the process. However, you can complete the steps in any order, with the exception of finishing the Content Type Definition, which requires defined workflows, lifecycles, templates, and so on. For more information, see XX *Terminology* and the appropriate step-by-step procedures in this Administrator Guide.

The goal is to get the system up and running and content creators can start adding content.

The following tasks have to be performed:

6.1 Organize the content storage

6.2 Create users and groups

6.3 Assign staff roles

6.4 Assign permissions

6.5 Create a Navigation Category Tree

6.6 Build your own custom theme (site design)

6.7 Check and Verify the content types

6.8 Create the basic content

6.9 Start Using the CCM

6.1 Organize the content storage

Create content sections and folders, basic structure:

content sections:

- default content section: content

additional sections depend on organizational requirements. Reconsider: Each content sections allows for a separate group of users, set of permissions, roles, etc.

Folders

- best practices
  - Folder for “formalia” (imprint, contact, webmaster, etc.)
  - Folder for start page and elated material.
- Remember: Folder structure (as well as sections) are in no way related to the public structure of the site (e.g. the navigation menu).

Refer to chapters XX and XX for further information.

6.2 Create users and groups

Use ~/ccm/admin to create users and groups.

6.3 Assign staff roles

Default roles for CMS include Alert Recipient, Author, Editor, Manager, and Publisher. Other applications, e.g. discussion forum, have their own set of users and groups. Check the corresponding administration guide.

When creating a role, add the name, description, and privileges for that role. Users assigned to the role will be able to only do those things they have permission for, such as creating new items or publishing. The Alert Recipient role does not have any privileges.

Add members (users) and administrators, if desired, to each role. When the role is assigned to a task, associated users are automatically assigned at the same time.

Refer to XX *Users, Groups, and Roles* for more conceptual information.

6.4 Assign permissions

You can assign each of these different permissions to roles, users, and groups. Users inherit permissions from their groups and groups inherit their permissions from their roles. You can assign individual users different permissions from their groups or roles to cover special circumstances (for example, you can assign permissions to a user who is an author, editor, and publisher without assigning the user to a specific group or role).

Groups can be a subset of roles or can be defined through the /admin in Web Application Framework.

Refer to XX Users, Groups, and Roles for more conceptual information.

6.5 Create a Navigation Category Tree

By default a new installation provides a minimal default category hierarchy for navigation.

Use

- ~/ccm/admin/terms
- ~/ccm/content-center → Categories

to adjust the categories to custom requirements

Use the category tree to determine how your content will be organized. The tree always starts with a Root folder, followed by some number of subcategories. You can add more second level categories and each level can have more sublevels, letting you make your category tree as granular as necessary. Programming is required to add new top level categories and direct programmatic input is recommended for companies with large amounts of information.

You can link categories, assign a purpose to top-level categories (which lets you use multiple logical category hierarchies), and assign category templates by content type.

Refer to XX Category for more conceptual information.

6.6 Build your own custom theme (site design)

You may either modify the build-in default theme

- ~/webapps/ROOT/themes/mandalay

or use themedirector to create and manage your own theme.

6.7 Check and Verify the content types

Each installation bundle comes with a decent set of applications and content types. Check it all applications and content types are available, specifically if each content section can access the needed content types. It not refer to the CMS administration guide how to add a content type to a content section.

Check whether the configuration of each content type is appropriate.

Refer to XX *Content Type* for more conceptual information.

6.8 Create the basic content

- ~/ccm/content-center
- Create site start page:
  - Assign navigation root category
  - Assign index page in root navigation category

  alternativly
  - Use Portal page (portalworkspace)
- Create index pages for each menu item
  - Create content item in appropriate folder
  - Assign appropriate navigation category
  - Change index page appropriately
- Create pages for formal stuff: imprint, contact, etc., accordingly

6.9 Start Using the CCM

Typical Workflow for Using the CMS

After CMS is set up, administrators and users:

1. Create new content items by specifying the content type
2. Add attributes and content to content item
3. Categorize content item
4. Approve content item
5. Publish content item

See the [cjt: CMS User's Guide] for more information

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 4
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 3
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 3

It supersedes these guides and provides updated information.

7. Getting Started – Further Customization

Once you have set up the core information in CMS, you can customize various pieces to match your corporate branding and processes. You do not have to follow these steps in any particular order and you can implement them at any time. Some of these items require programming (you cannot implement them through the CMS interface).

7.1 Create workflows

The default workflow is called "Production Workflow," which includes tasks for Authoring, Approval, and Deploy.

When creating a workflow, add the name and description, then add the tasks. At that point, you can assign users and add roles. (If you assigned users to roles in the previous step, those users will be assigned automatically.)

For the workflow to be available, you must assign users or roles to at least one of the tasks. Otherwise, you will not be able to apply it as a default workflow.

Refer to 4.17 *Users, Groups, and Roles* and 4.18 *Workflows* for more conceptual information.

7.2 Add lifecycles

The default lifecycle is called "Simple Publication," which includes one phase with no delay that lasts forever.

When creating a lifecycle, add the name and description, then add the phases.

Refer to 4.17 *Users, Groups, and Roles* and 4.11 *Lifecycles* for more conceptual information.

7.3 Create additional Category Trees

For several purposes additional category trees may be required, e.g. the rss server requires a dedicated category tree to deliver content.

Use

- ~/ccm/admin/terms
- ~/ccm/content-center → Categories

to create new category domains.

Remember:

Use the category tree to determine how your content will be organized. The tree always starts with a Root folder, followed by some number of subcategories. You can add more second level categories and each level can have more sublevels, letting you make your category tree as granular as necessary. Programming is required to add new top level categories and direct programmatic input is recommended for companies with large amounts of information.

You can link categories, assign a purpose to top-level categories (which lets you use multiple logical category hierarchies), and assign category templates by content type.

Refer to XX Category for more conceptual information.

7.4 Verify and modify the content type definitions

Each installation bundle comes with a decent set of applications and content types.

You can add child content types through the UI. When creating a content type, add the name, description, parent content type, default lifecycle, and default workflow, then assign the appropriate template.

Refer to 4.6 *Content Type* for more conceptual information.

7.5 Create a new content type

Within the interface, you can create new content types that are children of the existing content types. You can also programmatically add new parent and children content types to the CMS.

Refer to 4.6 *Content Type* for more conceptual information.

7.6 Add custom stylesheets

Programming is required to implement new stylesheets, which change the look and feel of the output. This lets you match corporate branding or create public and private sites that are displayed differently.

Refer to the *CMS Deployment Guide* for more information.

7.7 Design forms

Use the Forms section to create and modify forms, which you use to create templates for surveys, "contact us" forms, and more. Once the forms have been created, content creators can create new forms in the CMS that you know will include the correct information.

Refer to XX *Forms* for more conceptual information.

7.8 Verify and customize templates

The default templates are alternate, public, and summary.

When the CMS is installed, all three are identical. You will probably customize one of the existing templates to create a new template. Customizing templates requires a programmer or graphic designer. You can upload a new template via the user interface and publish it, then assign it as the default to the content type or content item.

Refer to 4.15 *Templates* for more conceptual information.

7.9 Modify templates

Programming is required to modify templates, which are used to define the necessary pieces for the different content types.

Refer to XX *Templates* for more conceptual information

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 4
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 3
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 3

It supersedes these guides and provides updated information.

Part III Global System Administration

8. Base Administration

The Core module provides basic services for every application modul. This includes specifically:

- User and group management
- Notification service
- Mail service
- Permission service
- Workflow service

So it provides a sound integration layer for all modules, allowing to exchange content and functionality between modules or to use a single authentification and authorisation for all modules.

8.1 Administration Tasks

- Administration of users and groups using a graphical user interface
-

Part IV
CMS Administration

Organising Work on Publishing WEB Content

9. Content Management System Administration

Welcome to the CMS Administrator Guide. The purpose of this guide is to:

- Introduce the Red Hat Content Management System (CMS) Application.
- Provide an administration task overview.
- Explain the various special concepts of the CMS application.
- Provide step-by-step instructions for the different tasks.

9.1 Purpose

Content management systems enable an organization to scale their creation and maintenance processes.

The CCM CMS application – being build upon the CCM framework – specifically allows every individual to create and maintain content, thus a large website can have up-to-date content in every area. The individuals closest to the content are responsible for updating and modifying it.

In addition, centralized content management can improve an organization's capability for knowledge sharing and communication. By making content readily available to the right people in the right format, content management systems improve users' efficiency when finding information while simultaneously reducing barriers to communicating relevant information.

9.2 Administration Goal

- Prepare and maintain the CMS application to enable users – authors, editors, publishers, managers (refer to CCM *CMS User Guide*) – to work on and publish content

- To some extent the same tasks as the manager role, described in the *CMS User Guide*.

9.3 Administration Tools

- Graphical User Interface
  - Address: http://your.server.dom/ccm/content-center
  - Same GUI for content creators, but additional capabilities
- Command Line Interfaces
  - Set various configuration parameters
  - Perform some special tasks as bulk publishing

9.4 Working with the Administration GUI

The GUI for administration is the Content Center , the same as for users to work on content (`~/ccm/content-center`).

Abbildung 1: The Content Center

You see a list of content sections (in most cases only one section has been installed). For a detailed description of the actions to perform here refer to the *User Guide*.

As an administrator you have to decide whether to use only one section for all content or to use a granular distribution of content over several sections. Refer to 9.5.1 Content Storage about this administration tasks.

If you select a section its administration screen is displayed.

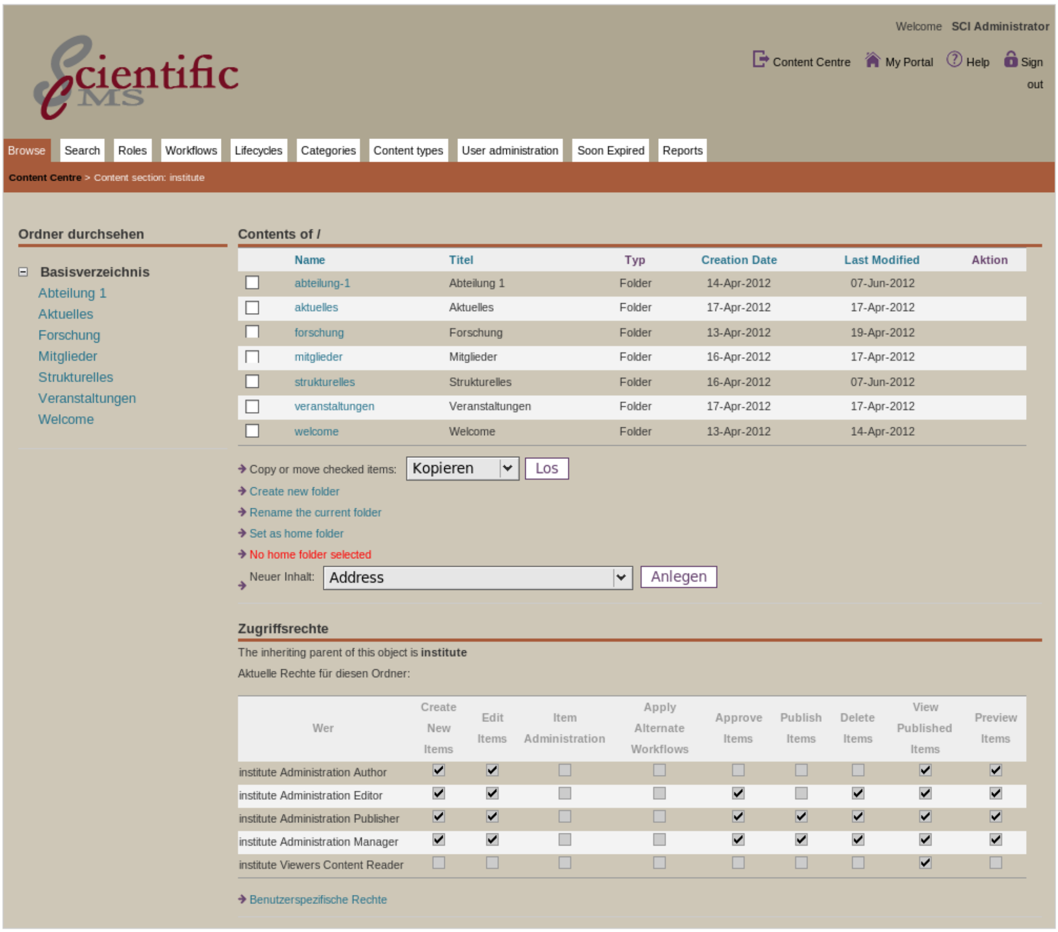


Abbildung 2: A Content Section

While the screen is the same for administrators and for users, depending on the configuration users without administration privilege may see only a subset of the tab options.

In the Browse tab all the content stored in this section is displayed, usually stored in a hierarchy of folders and sub-folders. Refer to 9.5.1 Content Storage about administration tasks here.

The other tabs are associated with various administration tasks.

9.5 Administration Tasks

Administration tasks to perform include:

- 9.5.1 Content Storage
- 9.5.2 Permissions and Privileges
- 9.5.3 Roles

9.5.4 Workflows

9.5.5 Lifecycles

9.5.6 Categories

9.5.7 Content Types

9.5.8 User Administration

9.5.1 Content Storage

All content is stored in content *sections* and within a section in *folders*. CMS users may create folders, but can not create content sections or delete or rename folders. It is an administrators obligation to create the required content sections, but he or she should also create the basic folder layout and folder policy.

Browse button displays folders and content items inside a section.

Refer to chapter 9.5.1 Content Storage below for detailed information.

9.5.2 Permissions and Privileges

Permissions and privileges are one of the CCM framework's methods to organize the work of many contributors in a safe and reliable way.

Permissions work on folder level and enable or prevent users to apply specific actions to all content items in a folder, e.g. a user may be able to modify a content item, but not to create a new one.

Sub-folders inherit from it's parent folders by default unless a different set of privileges is set by the administrator. Every folder and sub-folder may have its own set of privileges.

For details refer to chapter "Permissions".

While there is no required order for entering information into the CMS, you must set permissions for users and roles before they can modify content items.

9.5.3 Roles

Roles are another element of the CCM framework's powerful organisation infrastructure to empower a large number of users to contribute content at the same time and keep a large site current in any section.

Every action in the CMS is bound to a role, most important are e.g. author, editor and publisher. Each role may have different members and every member may have one or multiple roles. This way a decently grained system of responsibilities may be established.

During installation a predefined set of user groups and roles are set up, so everything works right out of the box. It is an administrators responsibility adjust this predefined set to the organization's requirements. Refer to chapter “Roles” for further information.

9.5.4 Workflows

Workflows are yet another element of the CCM framework's powerful organisation infrastructure to empower a large number of users to contribute content at the same time and keep a large site current in any section. Workflows are composed of tasks that control the process of producing a content item from creation through publication. Refer to chapter 3.9 for further conceptual information about the workflow concept.

Whereas a default workflow is established during installation so that the system works right out of the box, it is the administrator's responsibility to add further workflows and probably to adjust the predefined workflow to the organisations concrete requirements.

As an example, if a user creates a new content item based on a content type that does not have an assigned workflow, the author has the burden to select one before any further work on it. So the administrator should ensure that every content type has an appropriate workflow or set of workflows associated.

While there is no required order for initially entering information into the CMS, you must add a workflow before you can assign it to a content type, either during creating or editing.

9.5.5 Lifecycles

Lifecycles are another important element of the CCM framework's powerful organisation infrastructure. They empower administrators and publishers to automate the publication process and help to keep the site current without consuming a great amount of resources.

Lifecycles define the start and end publishing dates of a content item. If a content item is not assigned a lifecycle, it never gets published.

The default lifecycle contains one phase and publishes items immediately, with no expiration date. It is an administrators responsibility to adjust this predefined lifecycle to the organization's requirements. Refer to chapter “Lifecycle” for further information.

While there is no required order for initially entering information into the CMS, you must define a lifecycle so that it can be applied before you can publish a content item.

9.5.6 Categories

Categories are a keystone of the CCM framework. Refer to chapter 3.4 for details.

Categories are primarily for the sites visitors to access information. While categories make it easier for users to find information on the website, administrators and content contributors use *folders* to organize and find content items, instead (refer to section 9.5.1). **Remember**: both are *completely independent* from each other! For example, if there are several administrators who create content items, each of you might have a folder labelled with your name. The CMS is flexible enough for any organizational structure that you want to impose.

In addition to the standard implementation of categories in the CCM frameworks core module there is an extension in the *Terms* module (refer to chapter XX for detailed information). *Terms* uses the concept of *Domains* for category hierarchies and specifically supports the building and maintenance of system wide classification models. Terms is completely compatible with the standard implementation and both administration tools can be used interchangeable besides a few exceptions.

All installation bundles create at least one category tree for use as a navigation menu. It is an administrators responsibility to adjust this category tree to the organization's requirements and to create additional category trees as needed. Refer to chapter “Categories” for further information.

While there is no required order for initially entering information into the CMS, you must assign a category before you publish the content item if your website uses categorization. Otherwise, the content item is not organized in the public site and users have to know the direct URL to access it. Therefore an administrator has to care about categories in an early stage of the site development!

9.5.7 Content Types

(coming soon)

9.5.8 User Administration

The CMS includes a **User Administration** tab that lets you reset any user's home folder. If you are a site-wide administrator, you will also see a link to the **Standard Administration Page** in the Core User Administration Interface, where you work with users and groups.

9.5.9 Search

The CMS includes two types of search:

- A Search tab lets you search for content items in the CMS.
- A Search prompt is displayed when you need to find users in the user database to assign them to various functions of the CMS.

Search Tab

The Search tab is displayed in the Content Center and in each instance of the CMS that is installed. You can search the CMS by query string, category, launch date, and content type.

Search User Database

When you need to assign a user to a function, you will use the Search prompt. Search is used throughout the CMS to find users and assign them to:

- Permissions
- Roles (either as a member of the role or as an administrator)
- Groups
- Tasks and task roles within workflows
- Resetting a user's home folder

To search the user database, type the user's first name, last name, email address, or some combination of the three. Only users who have registered are displayed in the results. If you are not sure of the user's information, or to see all users who are registered, type *%* in the **Search** field. When the list is displayed, select the appropriate checkboxes for the users you want to add.

References:
This chapter uses material from

6. APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 4
7. Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 3
8. Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 3

It supersedes these guides and provides updated information.

10. Content Storage

Content is stored in content *sections* and within a content section in *folders*.

10.1 Content Section

Content sections are separate instances of a CMS. Your installation can have multiple content sections and each will act like a mini-CMS, with its own users, groups, roles, workflows, lifecycles, and content types.

Multiple content sections are not linked to one another. This means that a user with permission to edit in one content section does not automatically get permission to edit content in another. Refer to XX Permissions and Privileges.

An initial set of content sections is created during installation and depends on the installation bundle. You can see the already available content sections by selecting the Task/Sections tab in the CMS workspace. Refer to XX for an example.

10.1.1 Administration Tasks

It is an **administrators task** to *create* the content sections as required and name them appropriately for the organisation beyond the initial installation. There is currently no GUI to create a new content section. They are created and configured using CMS properties.

Refer to XX to get a list of configuration parameters and use the `ccm set` command to set the parameter

```
com.arsdigita.cms.contentsection.new_section_name=[name]
```

to the appropriate name.

The name has to fulfil URL requirements (all lower case, one word, letters, digits, hyphen, etc) because it is part of the URL of any conent item in the section.

To have the section actually created restart the server. There is no need to unset the parameter after creation. If a content section of the specified name already exist, the parameter is ignored.

10.1.2 User Tasks

The **Tasks/Sections** tab in the Content Center allows **users** to:

- See the sections that have been created in this Content Center
- See a list of tasks (locked, unlocked, or all) that corresponds to the user login
- For all tasks in the list, see the item title, current action (Authoring, Approval, or Deploy), due date, status, and locking. You can sort the columns in alphabetical or reverse alphabetical order by clicking the headings.
- In the *special case* the site is configured *not* to use category based navigation (but folder based navigation) see the names of the Public Sites that correspond to those sections.

  Currently no installation bundle uses folder based navigation and therefore there are no names of public sites which can be displayed.

No special administration task is required to let users see that information.

10.2 Folders

Inside each content section content is organized in folders and subfolders, building a folder hierarchy. With two exceptions, users and administrators can use the same folder actions.

Folders can be created by an administrators as well as user (i.e. author, editor, etc.) providing the users' credentials allow folder creation (which is default).

The exceptions are:

- Renaming a folder
- Deleting a folder

Refer to CMS User Guide for information about the other folder actions.

10.2.1 Administrator Tasks

Renaming a Folder

To rename a folder:

- In the **Folder Browser**, click the folder name of the folder you want to rename.
- Click **Rename the current folder**. The screen is redisplayed

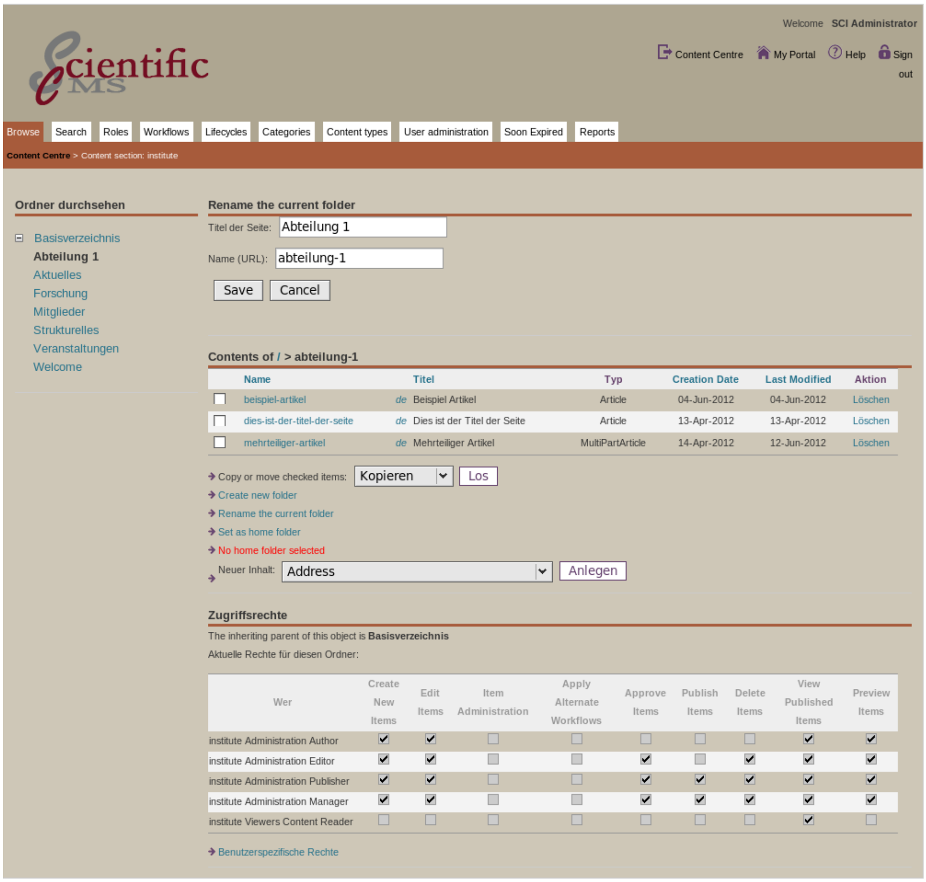


Abbildung 3

with the Rename the current folder section at the top.

- Type the new name in the **Page Title** field.
- Type the name of the page in the **Name (URL)** field (URL compliant).
- Click **Save**.

Deleting a Folder

To delete a folder:

- In the **Folder Browser**, click the folder name of the folder you want to delete.
- Click **Delete**.

Note:
You cannot delete a parent folder (it has to be empty). If the folder cannot be deleted, **Delete** is not displayed.

Creating a Folder

Although user may be able to create folders it is the administrator's duty to provide an initial folder structure which fulfils the organisation's requirements. Remember that the folder structure is completely independent from any public presentation and navigation menu. It is good practice to map the folder structure to work and user organisation. Refer to XX for further details.

To create a folder:

- In the **content section**, click the `Create Folder` button.
- Enter an appropriate folder name. Follow carefully the organisation's naming conventions if any!
- Click **Create**.

It is an administrator’s task to set the permissions within each folder appropriately. Refer to section XX for detailed information.

10.2.2 User Tasks

Depending on permissions users may create folders at some locations in the folder tree of a content section. It is the user's duty to comply to an organisation's naming conventions.

11. Permissions

This chapter

- provides *conceptual information* about permissions and privileges in the content management applications
- and discusses the *settings available* on the **Browse tab** in a content section.

11.1 Permissions and Privileges

The general concept of permissions and privileges is discussed in chapter 3.6 “Permissions and Privileges”. Permissions administration in the CMS application uses the Core services but adds CMS specific privileges.

The CMS application includes the following **privileges** (which are added to the core set of privileges during installation):

- **Item Administration**
  Lets users create, modify, and delete roles, workflows, lifecycles, categories, and content types. You should assign item administration privileges to other administrators only. This helps preserve the integrity and security of the CMS.
- **Apply Alternate Workflows**
  Lets users assign a different workflow to the content item.
- **Categorize Items**
  Lets users assign items to the appropriate category within the CMS.
- **Create New Items**
  Typically assigned to the Author role, lets users create new content items.
- **Delete Items**
  Lets users remove content items from the content section, as long as they are not live (actively published).
- **Edit Items**
  Lets users modify existing content items.

- **Preview Items**
  Lets users see what the published item will look like when it is deployed to the server, while there is still time to make any necessary changes.
- **Publish Items**
  Lets users publish and unpublish content items by applying life-cycles. The Deploy task must be finished before the life-cycle information is displayed on the **Publishing** tab.
- **View Published Items**
  Typically assigned to users who have a Viewer Role (the audience), lets users see the published content items.

11.2 Administration Tools

All administration tasks are done in the **Browse** tab, which must be active in a content section.

To activate the Browse tab:

- Open a content section. The **Browse** tab is displayed by default. Refer to the Red Hat Content Management System *User Guide* for more information.
- If you are already working in the content section, select the Browse tab.

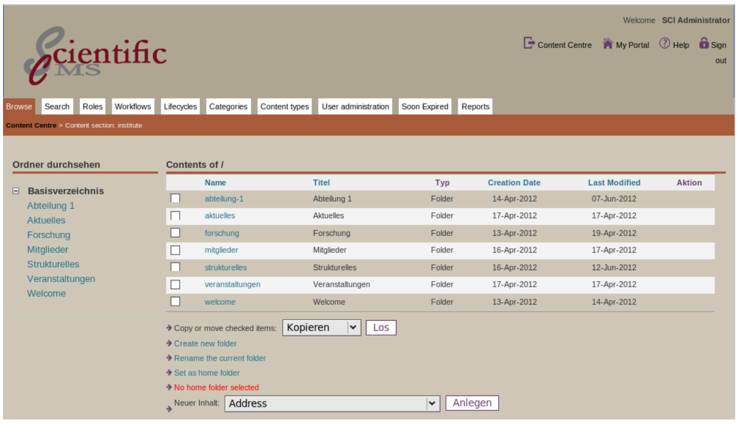


Abbildung 4

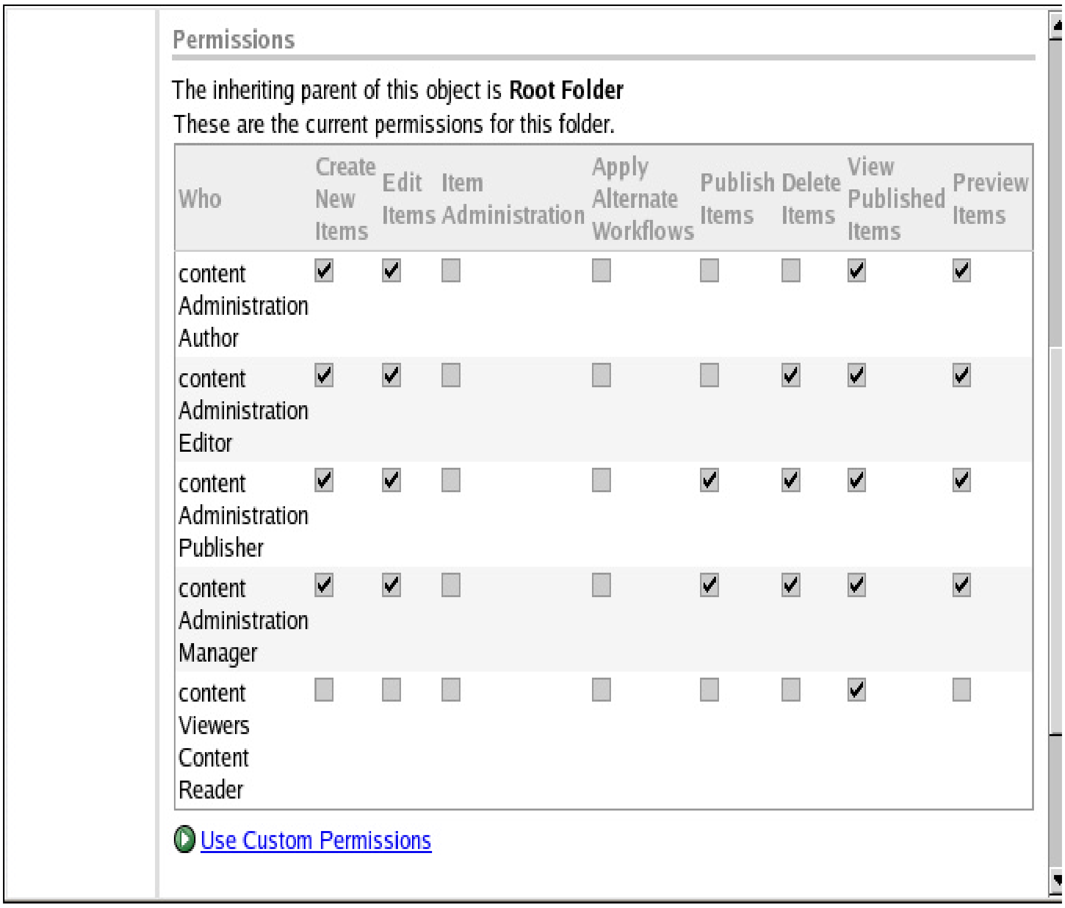
11.3 Administration Tasks

You can search the user database, assign custom permissions or restore default permissions to a folder, and add or remove an administrator.

- Setting Permissions
- Searching for a Person or Group
- Adding an Administrator
- Removing an Administrator

11.3.1 Setting Permissions

When you select a folder from the **Folder Browser** list, a Permissions section is displayed at the bottom of the window.


Permissions

The inheriting parent of this object is **Root Folder**
These are the current permissions for this folder.

Who	Create New Items	Edit Items	Item Administration	Apply Alternate Workflows	Publish Items	Delete Items	View Published Items	Preview Items
content Administration Author	☑	☑	☐	☐	☐	☐	☑	☑
content Administration Editor	☑	☑	☐	☐	☐	☑	☑	☑
content Administration Publisher	☑	☑	☐	☐	☑	☑	☑	☑
content Administration Manager	☑	☑	☐	☐	☑	☑	☑	☑
content Viewers Content Reader	☐	☐	☐	☐	☐	☐	☑	☐

Use Custom Permissions

Figure 5-3. Default Permissions

When you create a new folder, it uses the default permissions for each group. You can recognize the default permissions folder because the checkboxes are displayed with a gray background. Checkmarks indicate which permissions have been enabled.

Setting Custom Permissions

When you select a folder from the **Folder Browser** list, a **Permissions** section is displayed at the bottom of the window.

Override the default permissions at any time by clicking **Use Custom Permissions**. The window is redisplayed and the check-boxes are now displayed with a white background, indicating that you can toggle the selection for the group. You can also search for specific users to apply permissions to for this folder.

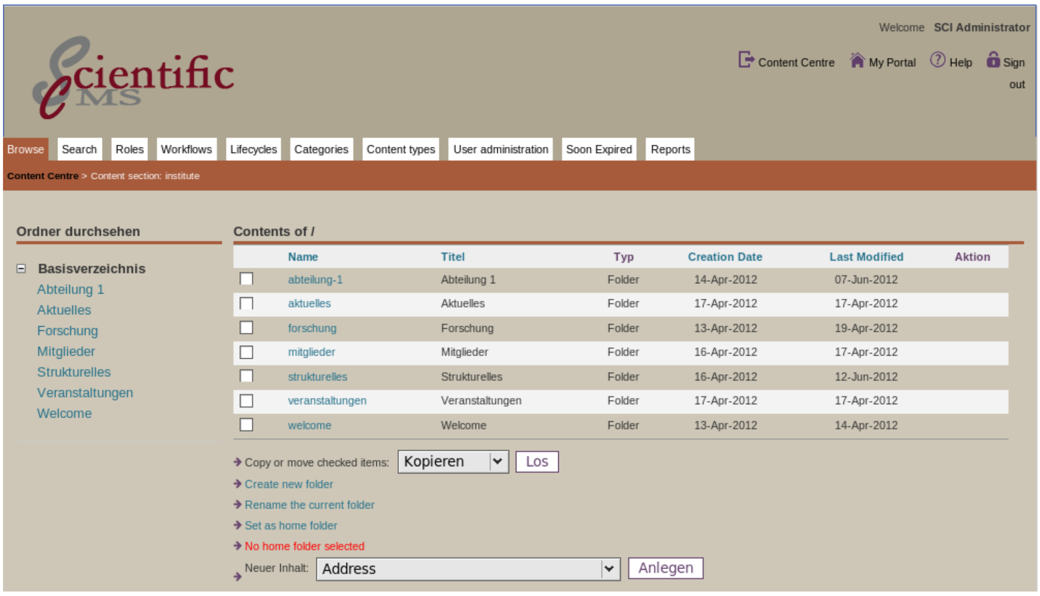


Abbildung 5

To set custom permissions:

- When default permissions are in effect, click Set Custom Permissions. The screen is redisplayed.
- For each group, select the appropriate checkbox for to apply the custom permission. The screen is redisplayed each time you select a checkbox.

Setting Direct Permissions

To apply direct permissions:

9. When custom permissions are in effect, type a user's name in the **Search** field.
10. Click **Search**. If there are no results, the message *“Your search matches no users or groups, search again”* is displayed.
11. If the search is successful, the screen is redisplayed with the search results and the list of permissions.
12. Select the checkbox for the users you want to apply direct permissions to.
13. Select the checkboxes for the permissions you want to apply to the users.

14. Click **Save**. The **Browse** tab screen is redisplayed and the users with direct permissions are displayed at the bottom of the Permissions list.

Restoring Default Permissions

To restore default permissions:

- When custom permissions have been set, click **Restore Default Permissions**.

11.3.2 Searching for a Person or Group

There are two ways to search the user database:

- Type the user's name in the **Search** field at the bottom of the Browse or Categories screens.
- Type the user's name in the **Search** field on the Roles, Workflows, and User administration screens.

Note
You can type a user name in the **Creation User** or **Last Modified By** fields on the Search tab when searching for specific content items. Refer to the CCM User Guide for more information about the Search tab.

To search the user database:

1. Type % in the **Search** field or leave the field blank to see a list of all possible members or type several characters from the member's name or email address.
2. Click **Search** or **Cancel** to return to the previous screen.
   If you clicked **Search**, the results are displayed. This example is from searching the database to add members to a role.
3. Select the users you want to add and click the appropriate button to continue or Cancel to return to the previous screen.

***Insert here**: Figure 5-5. Search Results screen*

4. Select the users you want to add and click the appropriate button to continue or Cancel to return to the previous screen.

11.3.3 Adding an Administrator

To add an administrator:

1. If necessary, click **Use Custom Permissions**.
2. At the bottom of the screen, click **Add administrator**. The screen is redisplayed and the **Search** field is displayed at the bottom of the screen.
3. Use the **Search** field. Refer to section *Searching for a Person or Group* for information on using the Search field. The Search results are displayed.
4. Select the checkboxes for the users you want to add as administrators.
5. Click **Add Administrators**.

11.3.4 Removing an Administrator

To remove an administrator:

1. At the bottom of the screen, click **remove** next to the administrators' name or group. The screen is redisplayed.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 6
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 5
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 5

It supersedes these guides and provides updated information.

12. Roles

Roles are another element of the CCM framework's powerful organization infrastructure to empower a large number of users to contribute content at the same time.

- Large number of contributors, expert in their domain but with no technical skills
- achieve rich content in every section of a large site

This chapter

- provides *conceptual information* about roles in the content management applications (section 12.1)
- and discusses the *settings available* on the **Roles tab** in a content section (remaining sections of this chapter).

12.1 Roles, Users, and Groups

Administrators may create additional roles and groups using the various user interfaces (e.g. core administration GUI or Portal administration portlet), which lets you further organize users. You can assign users, groups, and roles to specific workflows, thus streamlining the process for creating content.

While there is no required order for entering information into the CMS, you must create users and roles before you assign them to workflows. You should add your roles and users (and groups, if you will be using them) before making any other modifications to the CMS.

Because the CMS can be divided into multiple sections, and each section is treated independently, you need to create any new roles, modify existing roles, and apply permissions to roles in all sections. You do not have to assign the same users to roles in all sections (unless their job description includes roles in all sections). For more information on permissions and privileges, refer to section *Permissions and Privileges*.

While roles are configurable, common roles within the CMS are:

Alert Recipient

Receives Workflow Alerts, Overdue Workflow Task Alerts, and Content Expiration Alerts by email. This role has no privileges. The CMS automatically creates the Alert Recipient role when a new content section is created. You can customize the emails that are sent. Refer to the *CMS Deployment Guide*.

The Content Expiration Alert is dependent on the selections chosen when publishing the content item. The CMS sends content expiration alert emails to members of the Alert Recipient role in each content section. These emails are triggered by the expiry date of the lifecycle, which is set when the content item is published.

Workflow and Overdue Workflow Task alerts are dependent on the configuration of the workflow notification and mail services. These services must be properly installed and configured prior to activating alerts.

Each content section defines which alerts it sends on a state change for each of the three task types (Author, Edit, and Deploy). The CMS sends Workflow Alerts when a task starts, ends, is disabled, or is rolled back. It sends Overdue Workflow Task Alerts when a task exceeds the amount of time it was assigned (refer to section Workflows). The emails that are sent describe the state that has either been completed or started, the name of the content item the task was performed on, the name of the person who triggered the task completion or start, the latest comment on the workflow task, and a link back to a page on the CMS where recipients can begin their own task.

Each content section contains parameters that let you specify how long a task can remain incomplete before it is considered overdue. When a task is overdue, the system can send alerts to participants, reminding them that they have to finish their task.

Author

Creates new content items. The privileges for this role include creating, editing, and previewing items.

Editor

Verifies new content items. The privileges for this role include editing and previewing items.

Manager

Oversees the overall content or forms section. The privileges for this role include administering roles, workflows, lifecycles, categories, and content types.

Publisher

Approves and deploys content to the website. The privileges for this role includes previewing, publishing, and categorizing items.

Trusted User

Can choose their own workflow. Content section only.

Content Reader

Users able to read private sections of the public website. (Public sites are available, by definition, to anyone.) Content readers are also called "Viewers." If you have any questions about roles at your company installation, ask your System Administrator.

For each role, you can assign members (users or groups) and administrators by searching the CMS user database. Role administrators can make modifications to any role member, as compared to the Manager role, who can make modifications to any role (including the role administrator).

While there is no required order for initially entering information into the CMS, you must assign users to roles before you can assign a role to a task in a workflow, or you will not be able to create content items.

12.2 Administrations Tools

All administration tasks are done in the **Roles** tab, which must be active in a content section.

To activate the Roles tab:

- If necessary, open a content section. Refer to the Red Hat Content Management System User Guide for more information.
- Select the **Roles** tab.

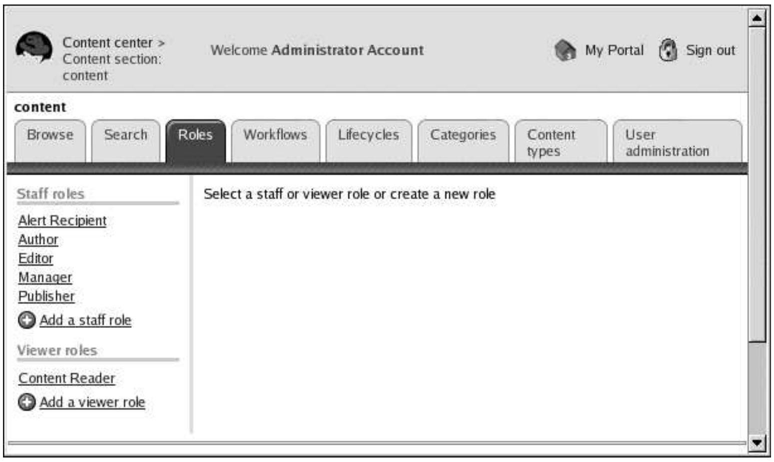


Figure 6-1. Roles Tab

12.3 Administration Tasks

You can add roles, assign members and administrators to roles, edit and delete roles, and remove members and administrators.

For more information on roles, refer to Section 4.17 Users, Groups, and Roles.

- Section 6.1 Adding a Role
- Section 6.2 Editing a Role
- Section 6.3 Deleting a Role
- Section 6.4 Adding a Member to a Role
- Section 6.6 Adding an Administrator to a Role
- Section 6.5 Removing a Member from a Role
- Section 6.7 Removing an Administrator from a Role

12.4 Adding a Role

To add a role:

- Click **Add a staff role** or **Add a viewer role**. The **Add role** section is displayed on the right.

Note

The list of privileges that are displayed depend on your installation.

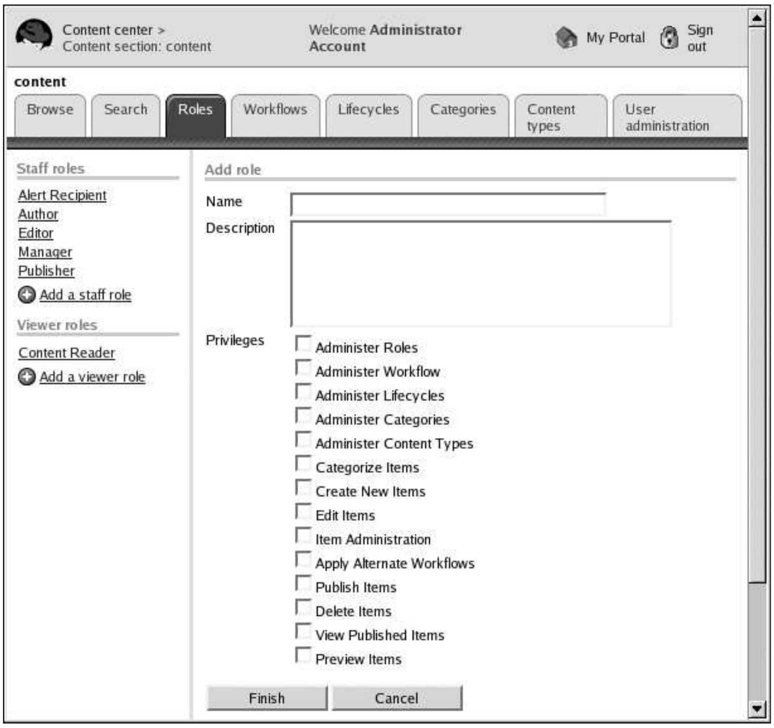


Figure 6-2. Add Role screen

- Type a name for the role (required).
- Type a description for the role (optional).
- Select the privileges for this role by selecting the appropriate checkboxes.
  Note: You do not have to select any privileges for a role. However, users assigned to roles without any assigned privileges only receive email alerts.
- Click **Finish** to save your entries or **Cancel** to return to the previous screen. When you click **Finish**, the **Role details** screen is displayed and a new group is added to the database for the role. The group is named *foo Administration role*, where *foo*

is the name of the current Content Section and *role* is the name of the role.

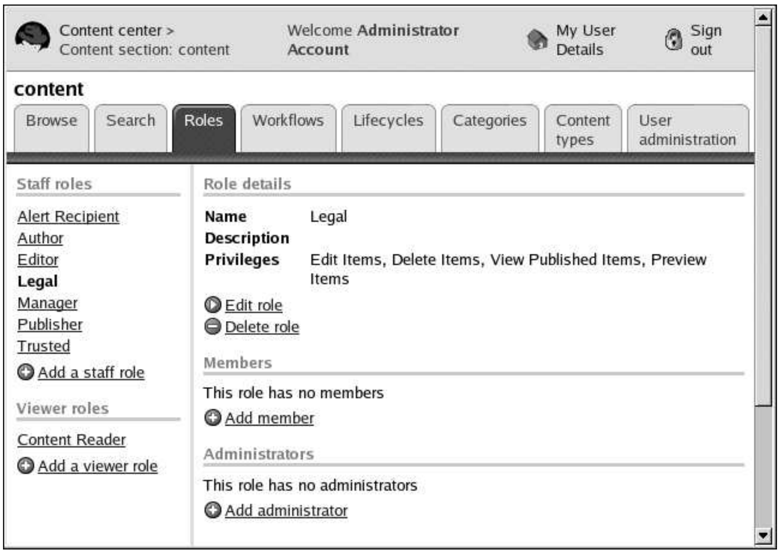


Figure 6-3. Roles Detail screen

From the **Roles detail** screen, you can:

15. View the information about the role
16. Add another role
17. Edit the current role information
18. Delete the current role
19. Add members and administrators to the role
20. Remove members and administrators from the role

12.5 Editing a Role

To edit a role:

- Click a role name in the left pane.
- Click **Edit Role** and the Edit Role section is displayed. This screen is the same as the Add role screen. Refer to Figure 6-2.
- Make your changes to the role's name, description, or privileges.
- Click **Finish** to save your entries or **Cancel** to return to the previous screen.

12.6 Deleting a Role

To delete a role:

- Click a role name in the left pane.
- Click **Delete role** and the prompt **Are you sure you want to delete this role?** is displayed.
- Click **Delete** to delete the role or **Cancel** to return to the *Role Details* screen. If you select **Delete**, the group that was created at the same time as the role is also deleted.

12.7 Adding a Member to a Role

To add a member to a role:

5. Click a role name in the left pane.
6. Click **Add member** and the Search prompt is displayed. Refer to Section *Searching for a Person or Group* for more information on how to use search.
7. After selecting the members you want to add, click **Add Members**. The **Role details** screen is displayed.

12.8 Removing a Member from a Role

To remove a member from a role:

6. Click a role name in the left pane. The members for the role are listed in the **Members** section.

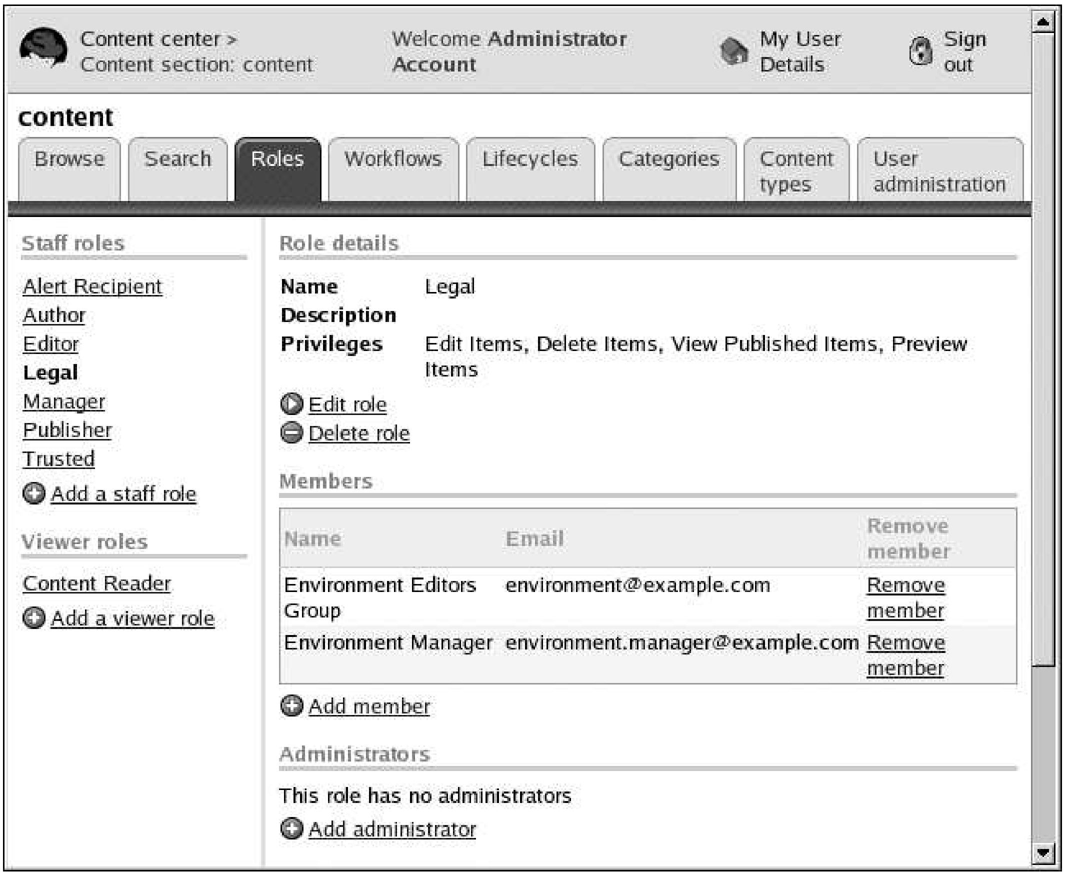


Figure 6-4. Role details screen with assigned members

7. Click **Remove member** next to the member you want to remove from the role. The member is immediately removed.
   Note: No confirmation prompt is displayed when removing members.

12.9 Adding an Administrator to a Role

To add an administrator:

2. Click a role name in the left pane.
3. Click **Add administrator** and the Search prompt is displayed. Refer to 5.3 *Searching for a Person or Group* for more information on how to use search.
4. After selecting the members you want to add as administrators, click **Add Administrators**. The **Role details** screen is displayed.

12.10 Removing an Administrator from a Role

To remove an administrator:

- Click a role name in the left pane. The administrators for the role are listed in the **Administrators** section.
- Click **Remove administrator** next to the administrator you want to remove from the role. The administrator is immediately removed.
  Note: No confirmation prompt is displayed when removing members.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 6
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 5
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 5

It superseeds these guides and provides updated information.

13. Workflows

Workflows are yet another element of the CCM framework's powerful organisation infrastructure to empower a large number of users to contribute content at the same time and keep a large site current in any section.

This chapter

- provides *conceptual information* about workflows in the content management applications (section 13.1 below).
- discusses the settings available on the **Workflows tab** in a content section (remaining sections of this chapter).

13.1 Workflow Characteristics

When you create a workflow, you must enter the name and description, and then you must add the tasks. When you create a task, you must enter the name and description. The task name is displayed on the Workflow Details and Task Details screens. This gives you the flexibility to name your task according to the needs of your organization, while still using the three task types in the CMS. You can create any number of tasks that correspond to these task types and you can create new task types programmatically.

Each task is associated with one of three task types: Author, Edit, or Deploy. If you do not select a different task type, Author is selected by default.

After you enter the basic task information, assign users and roles. You can assign multiple users and roles to a task, in which case only one user has to sign off on the task for it to proceed. However, when you use multiple tasks to define the workflow process, each task must be signed off.

13.2 Workflow Administration Tools

Before modifying any workflow or task information, the **Workflows** tab must be active in a content section.

To activate the Workflows tab:

- If necessary, open a content section. Refer to the Red Hat Content Management System User Guide for more information.
- Select the **Workflows** tab.

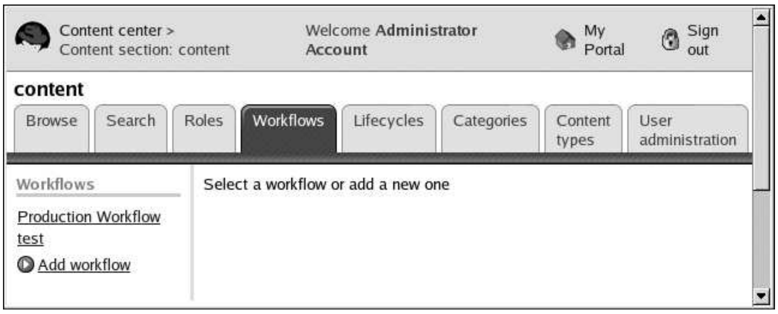


Figure 6 The Workflows tab

13.3 Workflow Administration Tasks

You can add workflows, add tasks, assign users and roles to tasks, remove users and roles, and edit and delete tasks and workflows. For more information on workflows, refer to 4.18 *Workflows*.

- *Adding a Workflow*
- *Editing a Workflow*
- *Deleting a Workflow*
- *Adding Tasks to a Workflow*
- *Assigning Users to a Task*
- *Unassigning Users from a Task*
- *Adding a Role to a Task*
- *Deleting a Role from a Task*

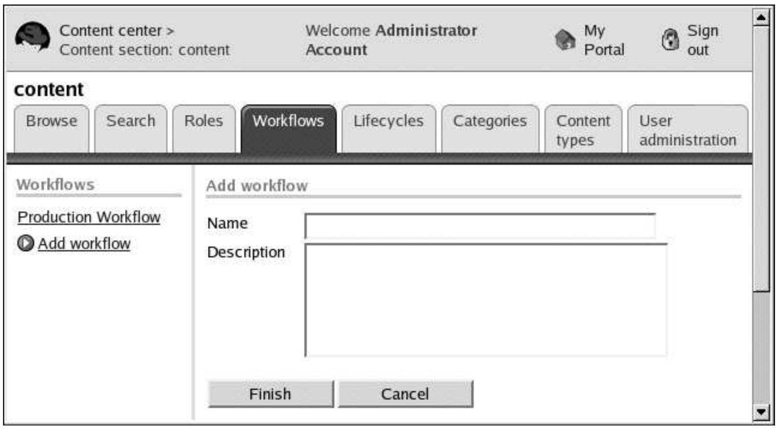
13.3.1 Adding a Workflow

To add a workflow:

- Select **Add workflow**. The **Add workflow** section is displayed.

Figure 7 Add workflow screen

- Type the name of the workflow (required).
- Type the description of the workflow (required).



- Click **Finish** to save your entries or **Cancel** to return to the previous screen.

13.3.2 Editing a Workflow

To edit a workflow:

- Select a workflow name in the left pane. The **Workflow details** and **Tasks** sections are displayed.

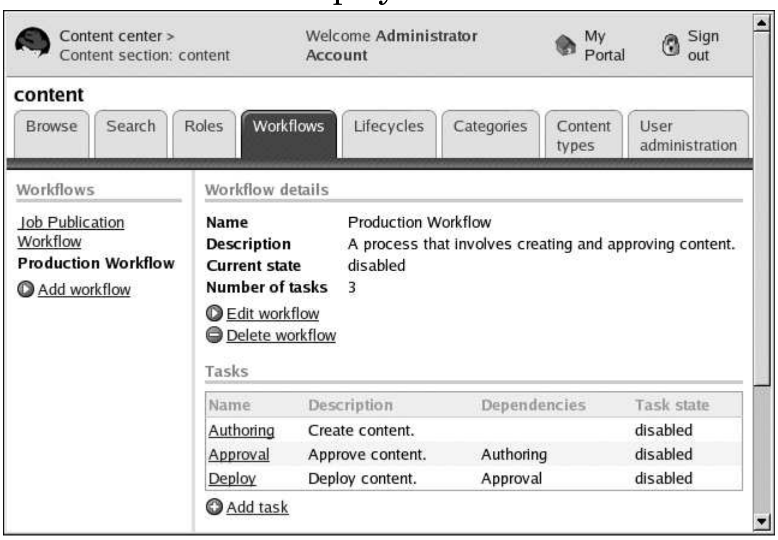


Figure 8 Workflow details screen

- Click **Edit workflow** and the **Edit workflow** section is displa yed. This screen is the same as the **Add workflow** screen. Refer to 7-2.
- Make your changes to the workflow's name or description.
- Click **Finish** to save your entries or **Cancel** to return to the previous screen.

13.3.3 Deleting a Workflow

To delete a workflow:

21. Select a workflow name in the left pane.
22. Click **Delete workflow** and the **Attention** section is displayed.
23. Click **Delete** to permanently remove the workflow or **Cancel** to return to the previous screen.

13.3.4 Adding Tasks to a Workflow

To add a task to a workflow:

- Select the workflow name in the left pane.
- Click **Add task** under the **Tasks** section. The **Add task** section is displayed.

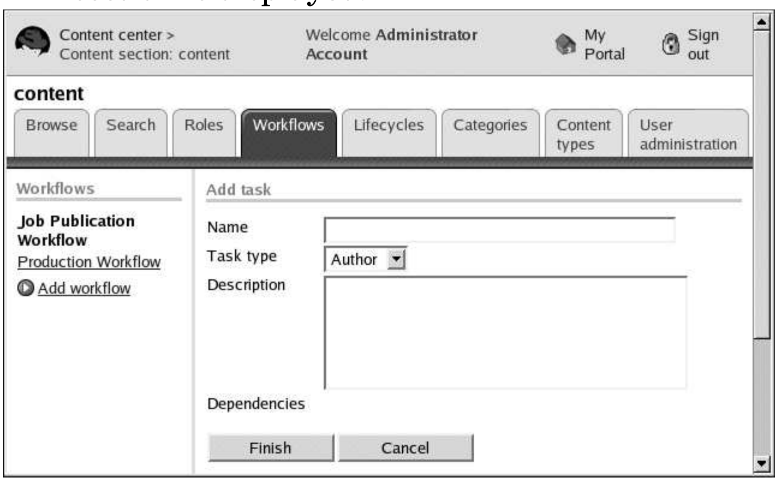


Figure 9 Add task section

- Type the name of the task (required).

- Select the task type (Author, Deploy, or Edit). The default is **Author**.
- Type a description for the task (required).
- Select any dependencies for this task. The dependencies that are displayed are based on existing tasks for this workflow. If this is the first task for the workflow, no dependencies are displayed.
- Continue adding tasks until all tasks have been added.
- When you are done, click **Finish** to save your entries or **Cancel** to return to the previous screen.

13.3.5 Editing a Task

To edit a task:

- Select a task name in the **Task** section.
- Click **Edit task** and the **Edit task** section is displayed. This screen is the same as **Add task**. Refer to 7-4.
- Make your changes to the task information.
- Click **Finish** to save your entries or **Cancel** to return to the previous screen.

13.3.6 Deleting a Task

To delete a task:

8. Select a task name in the left pane.
9. Click **Delete task** and the **Attention** section is displayed.
10. Click **Delete** to permanently remove the task or **Cancel** to return to the previous screen.

13.3.7 Assigning Users to a Task

To assign a user to a task, you must first open the task.

To open a task and assign a user:

8. Select a workflow from the Workflows list. The **Workflow details** and **Tasks** sections are displayed.

9. Select a task name from the list. The **Task details** is displayed.

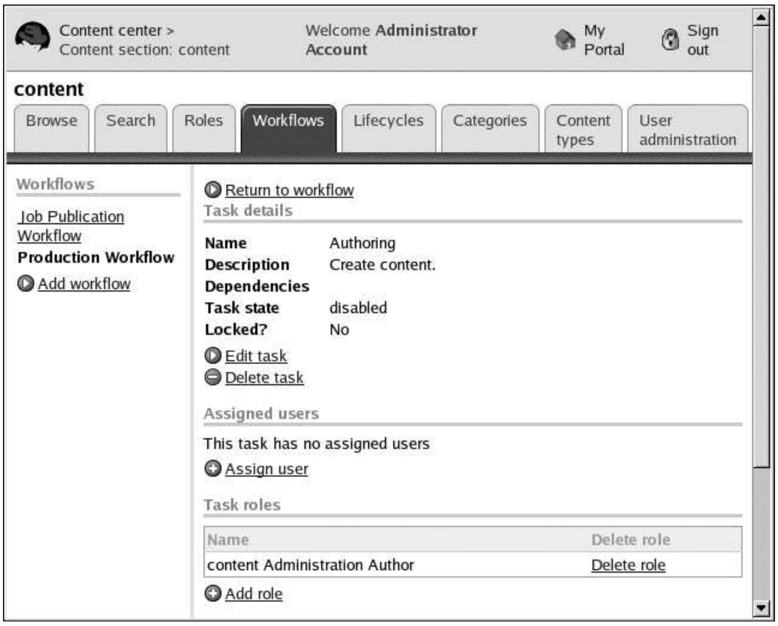


Figure 7-5. Assign User

10. Under **Assigned users**, click **Assign user**. The **Search to Add New Members** section is displayed. Refer to 5.3 *Searching for a Person or Group*.

11. When the matches are displayed, select the checkbox next to the person's name you want to assign as a user and click **Add Members** . The **Task details** section is displayed.

13.3.8 Unassigning Users from a Task

To remove a user's assignment from a task, you must first open the task.

To open a task and remove a user's assignment to a task:

5. Select a workflow from the Workflows list. The **Workflow details** and **Tasks** sections are displayed.

6. Select a task name from the list. The **Task details** section is displayed.

7. Under **Assigned Users**, click **Unassign user** next to the appropriate user name. The user is no longer assigned to the task.

13.3.9 Adding a Role to a Task

To add a role to a task, you must first open the task.

To open a task and add a role:

- Select a workflow from the Workflows list. The **Workflow details** and **Tasks** sections are displayed.
- Select a task name from the list. The **Task details** section is displayed. Refer to 7-5.
- Click **Add role**. The **Task roles** section is displayed.

Figure 7-6. Task roles section

- Select the roles that you want to assign to this task. You can select more than one role.
- Click **Finish** to save your entries or **Cancel** to return to the previous screen.

13.3.10 Deleting a Role from a Task

To delete a role from a task, you must first open the task.

To open a task and delete a role:

- Select a workflow from the Workflows list. The **Workflow details** and **Tasks** sections are displayed.
- Select a task name from the list. The **Task details** section is displayed.
- Click **Delete role**. The role is deleted from the task.

Note: No confirmation prompt is displayed.

14. Lifecycles

This chapter

- provides *conceptual information* about the lifecycle concept and implementation in the CMS application
- and discusses the settings available on the **Lifecycles tab** in a content section.

14.1 CMS Life-cycles

Lifecycles are a concept and implementation which aim at efficiency and resources savings in the efforts to keep a CCM site current.

Lifecycles define the start and end publishing dates of a content item.

Lifecycles consist of at least one phase, which controls its start delay and duration. The start delay determines the amount of time between when a user with publish privileges publishes a content item and when the item is displayed on the website. The duration determines how long the item remains on the website.

To publish items immediately, type 0 in the Days, Hours, and Minutes fields or leave them blank. To keep items on the website indefinitely, leave the three fields blank. The **Lifecycles** tab includes the phase names, description, delay, and duration.

14.2 Administration Tools

Management of lifecycles is done in the **Lifecycles** tab, which must be active in a content section.

To activate the Lifecycles tab:

1. If necessary, open a content section. Refer to the Red Hat Content Management System User Guide for more information.

2. Select the **Lifecycles** tab.

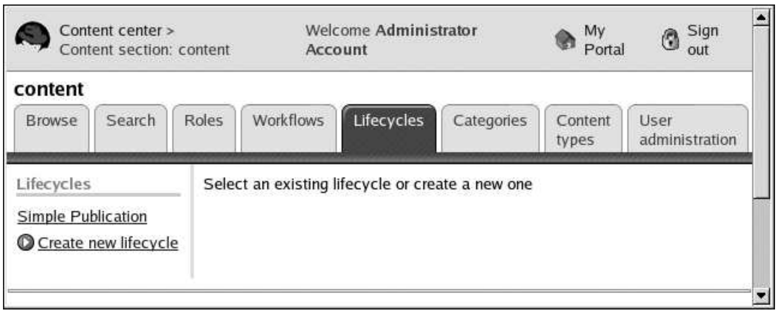


Figure 8-1. Lifecycles Tab

14.3 Administration Tasks

The screen uses a split pane, with the list of lifecycles on the left. When you select an existing lifecycle or create a new one, the Lifecycle Details are displayed on the right. Use the Lifecycle Details pane to:

1. *Creating a Lifecycle*
2. *Editing a Lifecycle*
3. *Deleting a Lifecycle*
4. *Adding a Phase*
5. *Editing a Phase*
6. *Deleting a Phase*

14.3.1 Creating a Lifecycle

To create a lifecycle:

1. Select **Create new lifecycle**. The **Create new lifecycle** section is displayed.

Figure 8-2. Create new lifecycle screen

2. Type the name of the lifecycle.

3. Type the description of the lifecycle.
4. Click **Finish** to save your entries or **Cancel** to return to the previous screen.

14.3.2 Editing a Lifecycle

To edit a lifecycle:

1. Click a lifecycle name in the left pane.
2. Click **Edit Lifecycle** and the **Edit Lifecycle** section is displayed. This screen is the same as the **Create Lifecycle** screen. Refer to 8-2.
3. Make your changes to the lifecycle's name or description.
4. Click **Finish** to save your entries or **Cancel** to return to the previous screen.

14.3.3 Deleting a Lifecycle

To delete a lifecycle:

1. Click a lifecycle name in the left pane.
2. Click **Delete Lifecycle** and the **Attention** section is displayed.
3. Click **Delete** to permanently remove the lifecycle or **Cancel** to return to the previous screen.

14.3.4 Adding a Phase

To add a phase to a lifecycle:

1. Select the lifecycle name in the left pane.
2. Click **Add phase** under the **Phases** section. The **Add phase** section is displayed.

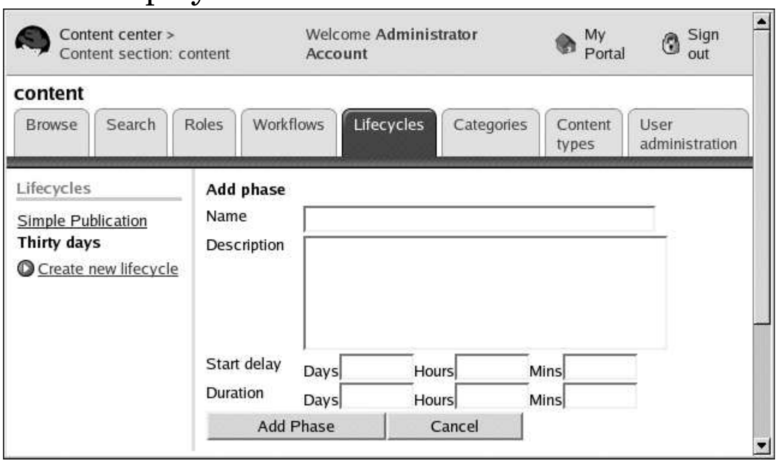


Figure 8-3. Add phase screen

3. Type the name of the phase.
4. Type a description for the phase.
5. Enter the start delay in days, hours, and minutes. Type **0** in each field or leave them blank to publish the content items immediately.
6. Enter the duration in days, hours, and minutes to enter the default expiration time of the content item. Type **0** in each field to publish the content item indefinitely (it never expires).
7. Click **Finish** to save your entries or **Cancel** to return to the previous screen.

Important

You can indicate notification alerts when you publish the content item. The notification alert needs to be after the current time, but before the expiration. The values you enter in the **Duration** fields set the default expiration and you can override them when you publish the content item.

14.3.5 Editing a Phase

To edit a phase:

1. Click a lifecycle name in the left pane.
2. Click **Edit phase** and the **Edit phase** section is displayed. This screen is the same as the **Add phase** screen. Refer to 8-3.
3. Make your changes to the phase information.
4. Click **Edit Phase** to save your entries or **Cancel** to return to the previous screen.

14.3.6 Deleting a Phase

To delete a phase:

1. Click a phase name in the **Phases** section.
2. Click **Delete phase** and the **Attention** section is displayed.
3. Click **Delete** to permanently remove the phase or **Cancel** to return to the previous screen.

15. Categories

Categories are a cornerstone of the CCM framework and a key for the information retrieval in large and complex web sites. Refer to chapter 3.4 for details.

This chapter

- provides implementation details of the category concept in the content management applications (section)
- and discusses the *settings available* on the **Categories tab** in a content section (remaining sections of this chapter).

15.1 Categories Implementation Details

Two implementations of category hierarchies:

3. *Use Context*, implemented CCM core
4. *Category Domain*, implemented in Terms as an extension of CCM core's categories implementation

Currently all installation bundles use *Category Domains* and Terms.

When you add a Use Context, you create a new organizational category. Add subcategories to create the hierarchy within the Use Context. When you select a different use context, the tree under *Categories* subtitle displays the category hierarchy associated with that use context. You assign categories (if desired) when you create content items.

> If module *Terms* is installed (which all installation bundles do by default) a *Use Context* is added in *terms* module as a new *domain*. A *domain* may be considered as a special *Use Context*.

Category information includes:

Name
: Type the name for the category. This name is displayed on the category tree and on the Authoring tab when assigning a category.

Description

Type the category description. The description is displayed on the content item's Summary tab.

Can you place objects in this category?

Select whether users can choose this category when organizing their content items. Otherwise, only subcategories can be added to the category.
You could also use this option to create a category for future use.

Is enabled?

If a category is enabled, it is displayed on the category tree. Users can only add content items to the category if you enable the previous option. This lets you create a parent category for organizational purposes, because users can only add content items to subcategories.

URL

The URL is automatically created when you type the name. The CMS converts any upper case letters in the name to lower case letters and replaces any spaces with hyphens as you type the name. Clear this field if categories are used for item URL resolution.

View and manage subcategories

Subcategories let you further subdivide a category for more organization. For example, you might have a parent category of Entertainment that includes subcategories for Music, Movies, and Television. If Entertainment is a child category of your root category, then Music is actually a grandchild of the root level and Entertainment is both a parent (of Music) and a child (of root).

For subcategories, view and manage linked categories.

By linking one subcategory to another, you create a relationship between categories in different parts of the category tree by assigning it a virtual parent. Your users can search the "true" parent or the virtual parent and find results within that subcategory, which is displayed in both places on the tree. Users will not realize that the categories are linked.

> You cannot create linked categories for the root category or for Use Contexts.

15.2 Administrations Tools

Before modifying any category information, the **Categories** tab must be active in a content section.

To activate the Categories tab:

5. If necessary, open a content section. Refer to the Red Hat Content Management System User Guide for more information.
6. Select the **Categories** tab.

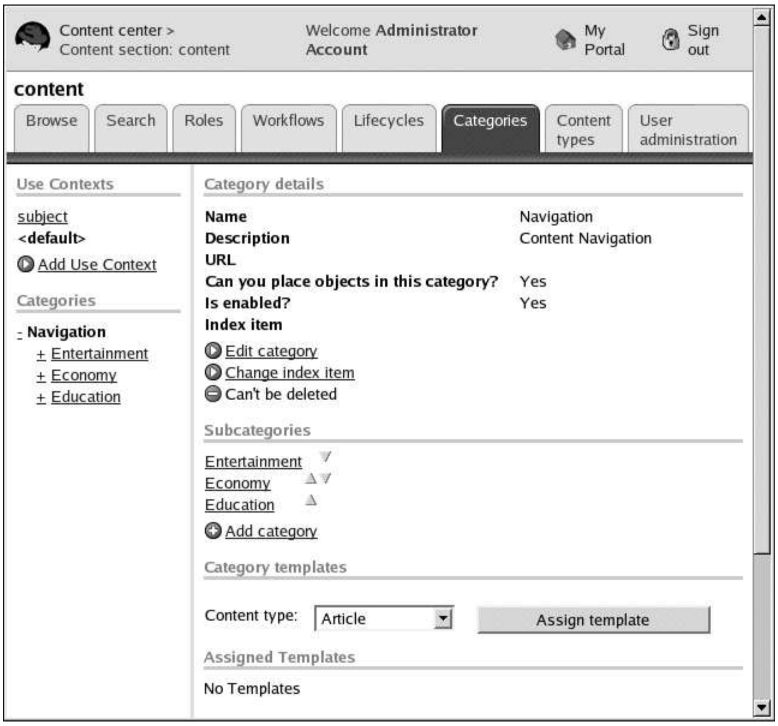


Figure 9-1. Categories Tab

The screen uses a split pane, with the category tree on the left and the category details on the right.

In the category tree, the current category is displayed in black bold text. Use the Category Details pane to see the current category information.

The *Category Details screen* also includes the category maintenance functionality. You can edit the category, change the index item, and delete the category. If you delete a parent category, all children are orphaned. You should delete all children first, then delete the parent category.

15.3 Administration Tasks

You can add use contexts (category hierarchy) and categories, edit and delete use contexts and categories, add and delete subcategories, link categories, and assign and remove category templates.

5. *Adding a New Category Tree*
6. *Adding a Category or Subcategory*
7. *Editing a Category or Subcategory*
8. *Deleting a Category or Subcategory*
9. *Assigning a Category Template*
10. *Removing an Assigned Category Template*
11. *Changing the Index Item*
12. *Adding a Linked Category*
13. *Removing a Linked Category*

15.3.1 Adding a New Category Tree

This sections describes the Use Context method for category hierarchies. By default Category domains are used and a new category tree is created using the terms administration tool. The Add Use Context method is probably not available.

To add a use context:

4. On the **Categories** tab, click **Add Use Context**.
   The **Add Use Context** screen is displayed.

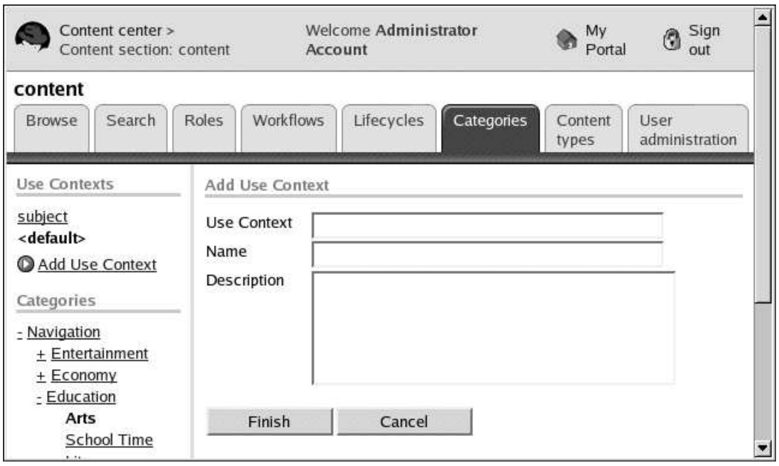


Figure 9-2. Add Use Context

5. Type the Use Context in the **Use Context** field (required).
6. Type the name for the page in the **Name** field, using all lower case letters and no spaces (required).
7. Type the description in the **Description** field (required).
8. Click **Finish** to save your entries or **Cancel** to return to the previous screen. The **Category details** screen for the Use Context is displayed.

If you want to modify the other category information, click **Edit Category**. Refer to 9.3 *Editing a Category or Subcategory*.

You can change the index item for a Use Context. Refer to 9.7 *Changing the Index Item*.

You can delete Use Contexts. Refer to 9.4 *Deleting a Category or Subcategory*.

15.3.2 Adding a Category or Subcategory

To add a category or subcategory:

8. Click a category name in the Categories tree in the left pane. The CMS will create the new category as a new level under your selection.
9. Click **Add category** and the **Add category** section is displayed.

Figure 9-3. Add category

10. Type a name in the **Name** field (required). The **URL** field will be filled in automatically.

11. Select **Yes** if users can place other objects (subcategories and content items) in this category. Select **No** if users can place only subcategories in this category.
12. Select **Yes** to enable the category so that it is displayed to users. Select **No** if it is not.
13. If your CMS uses a category-browsing user-interface, you will be able to assign an index item to the category.
14. Type a description for the category in the text box.
15. The URL defaults to the name. Change the URL as necessary, using all lower-case letters and hyphens in place of spaces. Clear this field if your site uses categorization instead of folders.
16. Click **Finish** to save your entries or **Cancel** to return to the previous screen.

15.3.3 Editing a Category or Subcategory

To edit a category or subcategory:

5. Click a category name in the Categories tree in the left pane.
6. Click **Edit category** and the **Edit category** section is displayed. This screen is the same as the **Add category** screen. Refer to 9-3.
7. Make your changes to category information.
8. Click **Finish** to save your entries or **Cancel** to return to the previous screen.

15.3.4 Deleting a Category or Subcategory

If you delete a category with subcategories, the subcategories are not deleted. However, they are orphaned, which means that they are not listed under the Categories section and you cannot perform maintenance on them.

> If terms module is installed and a category had been created as *term*, you can not delete it in the content section administration tool but you must use the terms administration GUI instead. All categories which had been created in the content section administration tool may me deleted here as well.

To delete a category or subcategory:

4. Select a category from the Categories list. If necessary, use the **+** and **-** keys to expand and collapse the list.
5. Click **Delete category** and the **Attention** section is displayed.
6. Click **Delete** to permanently remove the category or **Cancel** to return to the previous screen.

15.3.5 Assigning a Category Template

To assign a template:

1. Under the **Category templates** section, click the drop-down arrow to **Content type** and select the content type template that you want to use.
2. Click **Assign Template** . If any templates have been uploaded for the content type, they are displayed under the **Assigned Templates** section.
3. Click **Assign this template** for the template you want to assign to the category. The assigned template is displayed under **Assigned Templates**.
4. If you want to assign another template, repeat these steps

15.3.6 Removing an Assigned Category Template

To remove an assigned template:

1. Under the **Assigned Templates** section, click **Remove** next to the template you want to remove. The prompt **Are you sure you want to remove this template?** is displayed.
2. Click **OK** to remove the template or **Cancel** not to.

15.3.7 Changing the Index Item

To change an index item:

1. Select a Use Context.
2. Click **Change index item**. If at least one content item has been assigned to this Use Context, **Select index item for category** is displayed.
   Note: If no index items have been assigned to the Use Context, the only selection is **Inherit Index from Parent Category**. If the index item for the category cannot be changed, no selection is displayed.
3. Select the index item that you want to use for this category.
4. Click **Save** to save your changes or **Cancel** to return to the previous screen.

15.3.8 Adding a Linked Category

To add a linked category:

1. Display the category details for a subcategory.
   Important
   You cannot link the main category.
2. Under **Linked categories**, click **Add or remove linked categories**. The available categories are displayed.

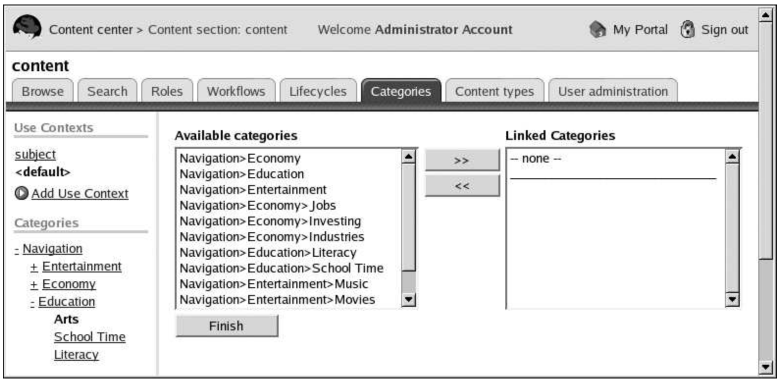


Figure 9-4. Available and Linked Categories screen

3. Select the categories on the left that you want to link and click **>>** . The selected categories are displayed under **Linked Categories**.
4. Repeat for any other categories that you want to link.

5. When you're done, click **Finished** . From this now on, when you select one of the linked categories, all linked categories are displayed as the current selected category.

15.3.9 Removing a Linked Category

To remove a linked category:

1. Display the category details for a linked subcategory.
2. Under **Linked categories**, click **Add or remove linked categories**. The available categories are displayed. This screen is the same as the **Available and Linked Categories** screen. Refer to 9-4.
3. Select the categories on the right that you want no longer want to link to and click **<<** . The selected categories are no longer displayed under **Linked Categories**.
4. Repeat for any other categories that you want to remove the links to.
5. When you're done, click **Finished**.

15.3.10 Manage and assign category templates.

If you are using category-specific templates for content presentation, you can assign a different template for each category. When you re-categorize an item, it changes the template that is used to display it.

> **Note**
> **With out-of-the-box Content Management System, you can use URL resolution, instead of folders.**

15.3.11 For subcategories, view and manage category permissions.

- Select a subcategory to see the **Permissions** section, which always displays the parent category that provides the base permissions for the subcategory. You can set custom permissions to specific users, groups, or roles. Custom permissions are always displayed.
- The possible default permissions for a category (other than the Root) are Edit, Categorize Items, Delete, and Admin, which are inherited from the parent category. The possible custom

permissions for a category (other than the root) are Edit, Categorize Items, Delete, Admin, and Remove All custom permissions.

16. Modifying Content Types

This chapter discusses how to add, edit, and delete custom content types through the user interface by using the **Content Types** tab in a content section. Refer to 4.6 *Content Type* for more information on content types.

- *Adding a Content Type*
- *Adding a Category Template*
- *Editing a Content Type*
- *Deleting a Content Type*

Before modifying any content type information, the **Content Types** tab must be active in a content center.

To activate the Content Types tab:

5. If necessary, open a content section. Refer to the Red Hat Content Management System User Guide for more information.

6. Select the **Content Types** tab. The Content type details for the first content type in the list is displayed.

Figure 10-1. Content Types Tab

16.1 Adding a Content Type

To add a content type:

7. If necessary, activate the **Content Types** tab.

8. Click **Add content type**. The **Add Content Type** screen is displayed.

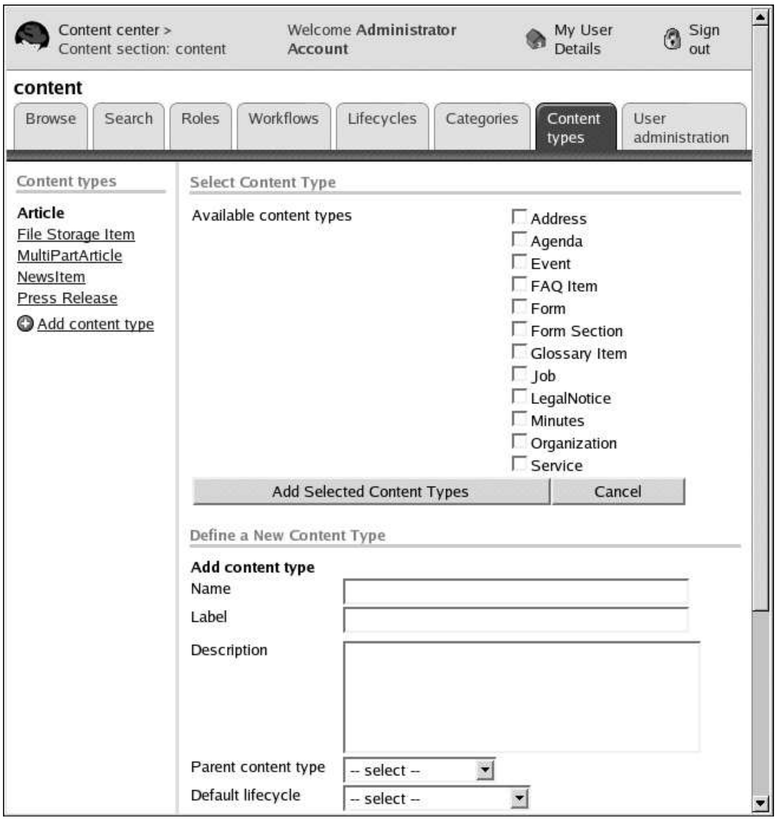


Figure 10-2. Add Content Types

You can add a new content type from the list of available content types or define a new child content type.

16.1.1 Adding a Selected Content Type

To add a selected content type:

7. Click the checkboxes for the content types you want to add.
8. Click **Add Selected Content Types**. The **Content type details** screen is redisplayed.

16.1.2 Adding a User Defined Content Type (UDCT)

To add a user defined (child) content type:

14. Type the name of the new content type in the **Name** field (required).
    You can only use letters and numbers in the name. It cannot contain any spaces or punctuation. It can use mixed capitalization (for example, **holidayEvent**).

15. Type the label for the content type in the Label field (required). The label should reflect the parent content type (for example: **Economy - Article**). It is displayed in the **Content types** list on the left. This field is required.
16. Type the description of the content type in the Description field.
17. Select the parent content type from the drop-down list, which only includes installed content types (listed on the left under **Content Types**).

    This field is optional. If you do not select a parent content type, you must enter the pieces for the authoring kit or users will not be able to add any content to the content item.
18. Select the default lifecycle from the drop-down list. If you do not select a lifecycle when creating the content type, you must select one before users can publish the content item.
19. Select the default workflow from the drop-down list. If you do not select a workflow, users must select one as they create a content item.
20. Click **Save** to save the new content type or **Cancel**.

    The **Content type details** screen is displayed. The new content type is displayed in the list at the left.

16.2 Adding a Content Type / Category Template

(You can only use category templates when you are using categories for item URL resolution instead of folders.)

To add a category template:

9. Select the **Content Types** tab in a content section.

10. Click **Add template**. The new template page is displayed.

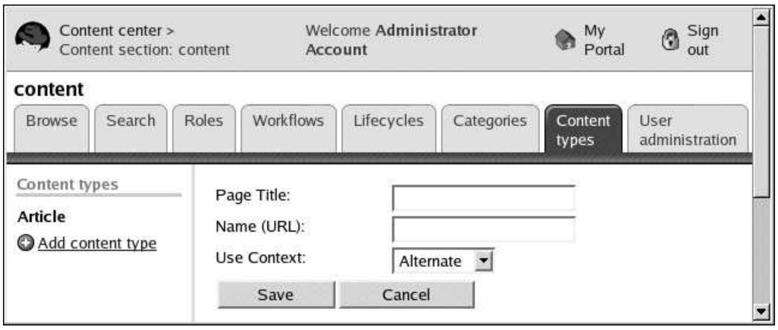


Figure 10-3. Add new template

11. Type a page title. The Name (URL) field is filled in automatically with the correctly formatted information.
12. Select a use context by clicking the drop-down arrow. The information that is displayed depends on your installation.
13. Click **Save** . The template authoring screen is displayed.

Figure 10-4. Template authoring screen

14. Click **Body text**. The body text screen is displayed.

Figure 10-5. Template authoring - body text

15. Click **Upload a file** . The upload screen is displayed.

Figure 10-6. Template authoring - upload

16. Click the drop-down arrow next to **Text Type** to select the MIME type (XSL or JSP) or accept the default to let the system recognize the MIME type from the file.

17. Click the **Browse** button to navigate to the file that you want to upload.

18. Click **Save**. The file is displayed.

19. Click the **Contents** tab. The templates are displayed in the Templates section.

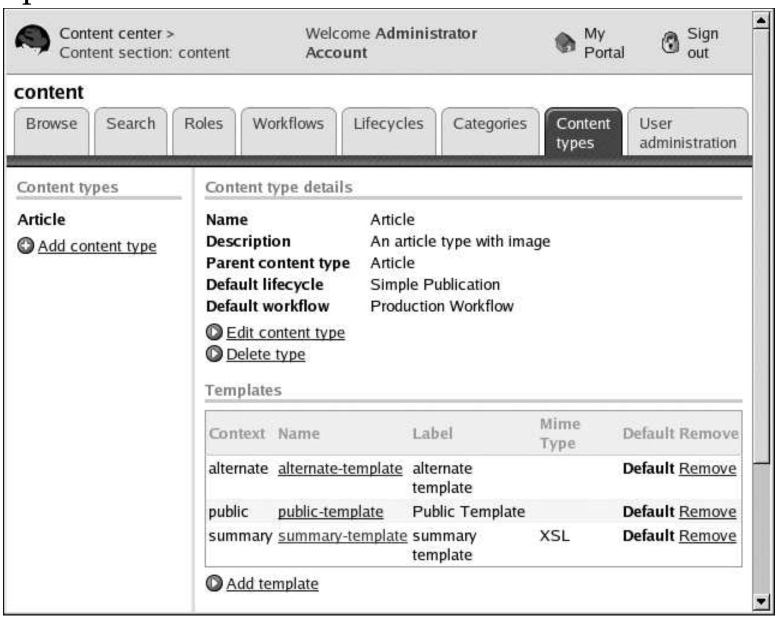


Figure 10-7. Content types tab with templates

16.3 Editing a Content Type

To edit a content type:

17. If necessary, select the Content Types tab and the content type that you want to edit.

18. Click **Edit content type**. The content type information is displayed.
19. Edit the fields (**Label**, **Description**, **Default Lifecycle**, or **Default Workflow**).
20. Click **Save** to save the edited information or **Cancel** to return to the previous screen.

16.4 Deleting a Content Type

To delete a content type:

9. If necessary, select the Content Types tab and the content type that you want to edit.
10. Click **Delete Type**. The Attention screen is displayed, with the prompt **Are you sure you want to delete this content type?**
11. Click **Delete** to delete the content type or **Cancel** to return to the previous screen.

17. User Administration

This chapter discusses the settings available on the **User Administration** tab in a content section. You can reset the user's home folder or, if you are a sitewide administrator, you can work with users and groups in the WAF User Administration Interface. Refer to the Red Hat Web Application Framework Developer Guide. Refer to 4.16 *User Administration* for more information.

- Resetting a User's Home Folder

Before using the information in this chapter, the **User Administration** tab must be active in a content section.

To activate the User Administration tab:

7. Open a content section. Refer to Red Hat Content Management System User Guide for more information. If you are already working in the content section, select the **User Administration** tab.

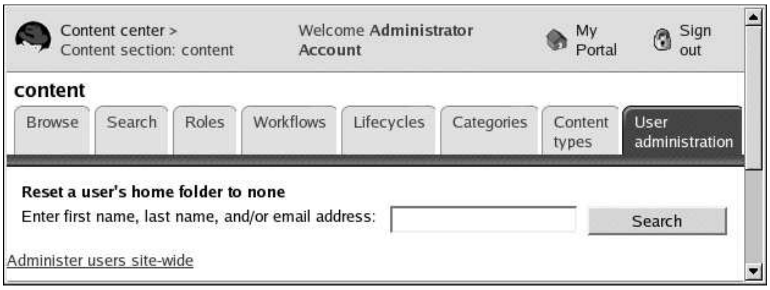


Figure 11-1. User Administration Tab

17.1 Resetting a User's Home Folder

To reset a user's home folder:

9. Find the user whose folder you want to reset by typing their first name, their last name, or their e mail address in the **Search** field and clicking **Search**.For more information on Search, refer to 4.14 *Search*.
10. When the matches for the search are displayed, select the checkmark next to the appropriate user name.
11. Click **Set no default folder for users**.

The user will no longer have a specified home folder and can use **Set as home folder** to assign a new one. Refer to the Red Hat Content Management System User Guide for more information.

18. CMS Command Line Tools

This chapter details the functionality available in CMS command line tools.

18.1 Content Import Tool

The content import tool can be used to bulk import items to the CMS. The primary use of this tool is to import the APLAWS+ generic services content.

The content import is a multi stage process. The first stage processes the folder definition file (by convention index.xml) creating the folder hierarchy and building up a mapping between folders and items. The second stage processes each item XML file, importing the item into its corresponding folder.

The content import tool requires three command line parameters. The first is the name of the folder definition files. The second is the path to the directory containing the item XML files. The third is the path to the directory containing any binary file attachments or images. To get full error reports of any problems encountered it is recommended to use the -D flag too. So a complete command line would be:

```
# ccm-run  \
com.arsdigita.london.importer.cms.ItemImportTool \
-D index.xml /path/to/item/dir /path/to/asset/dir
```

Note
The importer processes each individual item within a dedicated database transaction, and maintains a mapping between source and destination items. This allows the importer to be interrupted and restarted without importing the same item twice.

Tip
After importing a large amount (> 500 items) of content it is recommended that the DBA manually update the database optmizer statistics for the APLAWS+ schema. This will ensure continued high performance of the system.

18.2 Content Bulk Publish

The bulk publish tool will iterate over every single item in the CMS, and if an item is not already live, will publish it with the default lifecycle corresponding to its content type. The lifecycle will be set to start immediately and never expire. Any items which are already live will not be changed. The bulk publish tool can be run without any command line arguments, although to get full error reports it is recommended to use the -D flag:

```
# ccm-run com.arsdigita.london.util.cmd.BulkPublish -D
```

Note:
Each item is processed within a dedicated database transaction, so if the command is interrupted partway through, next time it is run it will continue where it left off.

Tip:
To get a progress message output for each item, edit `/etc/ccm/conf/log4j.properties` setting log level on the command to 'INFO':
`log4j.logger.com.arsdigita.london.util.cmd.BulkPublish=INFO`

18.3 Invariant Checking Tool

The invariant checking tools run a number of queries against the APLAWS+ database to ensure that no stored objects are violating semantic integrity rules. For example, it will check that all top levels items have an associated content type, or that a multi-lingual content bundle only has one item for each language. If any invariants are violated a message will be displayed on the console detailing what SQL query failed. The problem should then be investigated to identify the root cause such as a code bug, or SQL upgrade script or database corruption.

18.3.1 ACSObject Invariant Checks

The first invariant tool verifies that all ACSObjects in the database have rows in all the tables required by their object type. For example, for a ContentBundle object it will check that there are rows in acs_objects, vc_objects, cms_items and cms_bundles. It can be invoked with:

```
# ccm-run \
com.arsdigita.london.util.db.ACSObjectInvariantCheck
```

Tip:
To get a progress message output for each invariant checked, edit `/etc/ccm/conf/log4j.properties` setting log level on the command to 'INFO'.:
`log4j.logger.com.arsdigita.london.util.db.ACSObjectInvariantCheck=INFO`

18.3.2 CMS Invariant Checks

The second invariant tool verifies an application specific set of constraints. The ccm-ldn-util application defines a set of invariants for CMS. It can be invoked with:

```
# ccm-run com.arsdigita.london.util.db.InvariantCheck \
     ccm-ldn-util
```

Tip:
To get a progress message output for each invariant checked, edit `/etc/ccm/conf/log4j.properties` setting log level on the command to 'INFO':
`log4j.logger.com.arsdigita.london.util.db.InvariantCheck=INFO`

18.3.3 Search Re-index Tool

The search re-index tool will repopulate the contents of the search engine index tables. This can be useful if the rules for which attributes in an object are indexed get updated, or when switching to an alternate search engine on an already populated site.The reindex tool can be run without any command line arguments, although to get full error reports it is recommended to use the -D flag:

```
# ccm-run com.arsdigita.london.search.Reindexer -D
```

The default behaviour is to re-index all items within the CMS, however, it can be restricted to items with a specific content type by using the --object-type argument to set the name of the content type. For example, to only reindex event objects:

```
# ccm-run com.arsdigita.london.search.Reindexer -D \
     --object-type com.arsdigita.cms.contenttypes.Event
```

Tip:
To get a progress message output for each item, edit `/etc/ccm/conf/log4j.properties` setting log level on the command to

’INFO’:

```
log4j.logger.com.arsdigita.london.search.Reindexer=INFO
```

Part V TERMS Domain Administration

Advanced Categorization of Content

19. TERMS Administration Overview

Welcome to the TERMS Administrator Guide. The purpose of this guide is to:

- Introduce the Terms Application.
- Provide an administration task overview.
- Provide step-by-step instructions for the different administration tasks.

19.1 Purpose

The *TERMS module* is an extension and enhancement of the CCM core categorization service. It is designed specifically for the creation and maintenance of **classification** systems.

It adds an extended concept of a category hierarchy, the (category) *Domain* as well as an extended concept of category, the *Term*. A *Domain* is build up of multiple *Terms*.

Each Domain may be mapped to one or multiple CCM applications. So they form a central pool of category systems, whereas Core provides it's categorization service for each application of its own (and therefore a CMS *Use Context* is only valid for the specific content section).

A *Domain* adds some properties to the Use Context as used by CMS:

Domain	Use Context	Usage
Key	Name	Unique ID
URL		Link to the classification's publication
Title	Use Context	Pretty title
Description	Description	Description
Version		Classification's version and
Release Date		release date, this domain is based upon

Tabelle 19.1: Comparison of Domain and Use Context

A domain is automatically listed as Use Context in the CMS application. But the creation of a Domain must be done in TERMS.

The TERMS module adds some facilities to synchronize a Domain with an established classification system.

Just like a Domain a Term adds some properties to a category:

Term	Category	Usage
Unique ID		Manageable by classification system
Name	Name	Pretty title as publicly displayed
	Assign objects	Link to the classification's publication
	Is enabled	
Description	Description	Description
	URL	Unique part of the URL
Shortcut		Actually unused
In A-Z		Actually unused

Tabelle 19.2: Comparison of *Term* and *Category*

The TERMS module automatically creates a core service's category while you add a new *Term*, and deletes it respectively, keeping both in sync.

Conversely, this is not valid. You can add categories in the Content Center which the core service will automatically associate with an existing *Domain*, but it does not add a *Term*. This way you may locally extend and customize an existing classification scheme.

The TERMS module cooperates specifically with the NAVIGATION module. It is the TERMS module, where you select the category tree to be used as the navigation tree. This is explained in detail in the *Navigation Administration Guide*.

19.2 Administration Goal

The goal is to maintain one or multiple different classification schemes in order to make the site content accessible in an easy way for various groups of readers with different interests and perspectives. In particular, an administrator has to create a domain, add terms and arrange them in a classification scheme.

19.3 Administration Tools

TERMS provides a graphical user interface for administration. It can be reached at

```
http://[your.server.dom]/ccm/admin/terms
```

It is a dedicated administration tools, not accessible to other users. System wide administration privileges are required.

In addition, a command line interface is available. It is used to set various configuration parameters as well as to perform some special tasks as a mass import ot *Terms*.

19.4 Working with the Administration GUI

To access the administration GUI point your browser to

```
http://[your.server.dom]/ccm/admin/terms
```

If you are not logged in, a login screen is displayed and after successfully logged in the request is redirected to the terms administration

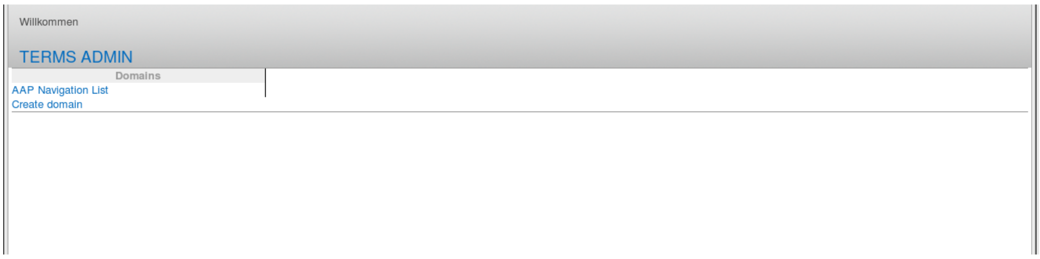


Figure 1: Terms Administration Entry Screen

entry screen.

The clean screen design lists on the left side all defined terms domains and an option to create a new domain.

19.5 Administration Tasks Overview

There are four areas of responsibility for administrators:

8. Maintain *Domains*
9. Maintain *Domains* usage
10. Maintain *Terms*
11. Build *classification systems*

19.5.1 Maintain *Domains*

This area involves the creation, modification, and probably deletion of *Domains*. During installation all installation bundles create at least one base category Domain which is configured by default for the navigation menu. It is an administrators responsibility to adjust this *Domain* to the organization's requirements and to create additional *Domains* as needed. Refer to the next chapter for further information.

19.5.2 Maintain Domains Usage

This area involves specifically the mapping of each *Domain* to one or multiple applications which use this *Domain*. During installation all installation bundles create a default mapping. It's details depend on the specific bundle. It is an administrators responsibility to adjust this *Domain* mapping to the organization's requirements and to create additional *Domains* as needed. As an example, if your organization decides to provide some content by RSS, the administrator has to create a *Domain* used to determine which content items to deliver and map this *Domain* to the RSS application.

Refer to the next chapter for further information.

19.5.3 Maintain Terms

This area involves specifically creating, editing, and deleting *Terms*. Every installation bundle creates at least some *Terms* as a starting point. Unless the installation bundle does not import an established classification scheme the administrator has to adjust the pre-installed *Terms*.

19.5.4 Build a Classification System

This area involves specifically arranging *Terms* into a classification hierarchy. Again, it is an administrators responsibility to adjust the pre-installed hierarchy to the organization's requirements.

Refer to the next chapter for further information.

References:

This chapter uses material from

12. APLAWS FedoraHosted WIKI
    https://fedorahosted.org/aplaws/wiki/AdminTermsGuide
    (last accessed May 2012)

20. Terms GUI Administration Tool

This chapter provides a step-by-step description of the various administration tasks performed within the graphical administration tool.

- Manage a category **Domain**
  - Create a new Domain
  - Modify an existing Domain
  - Delete an existing Domain
- Manage **Domain Usage**: Map a Domain to applications
- Manage **Terms**
  - Create a term
  - Edit a term
  - Delete a term
- Build a **classification system**.

20.1 Managing a Category Domain

All domain administration tasks start at the TERMS administration entry screen.

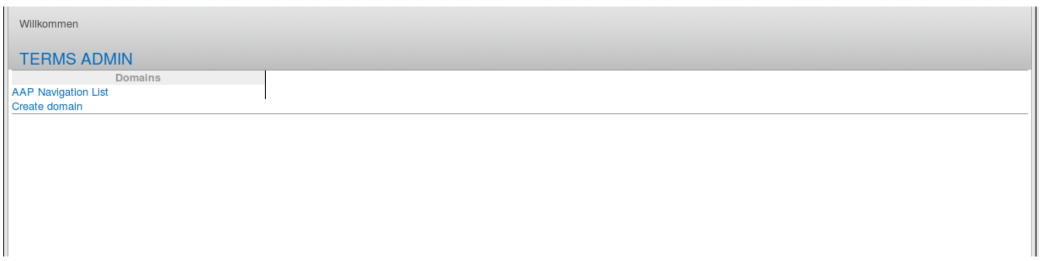


Figure 2: TERMS Administration Entry Screen

20.1.1 Create a New Domain

At the bottom of the domain listing a `Create Domain` link is provided.

> As an alternative to the GUI you can create an xml file offline, which defines the category tree, and use a CLI interface to import it. This may be more time efficient for non-trivial category trees, or if the tree has to be loaded on several Server instances.

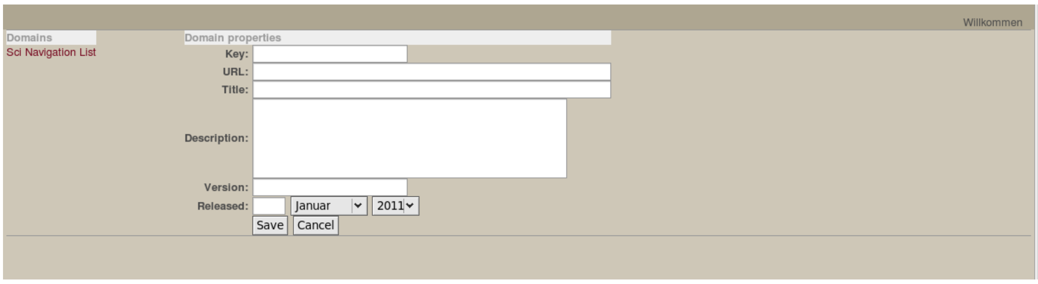


Abbildung 10

A click will presented a form:

13. **Key**: enter a unique name for the tree, such as NEW-NAVIGATION
14. **URL**: enter the URL from which the list of terms can be found. It has to be **unique**! If you do not have such a URL enter a dummy URL, as this is not actually used
15. **Title**: enter a pretty title for the tree, such as "My Council new navigation"
16. **Description**: enter a brief description of the purpose of the tree (this is optional)
17. **Version**: enter the version of the tree that you are creating; you can start with "1.00", and increase it later when you revise/modify the tree
18. **Release**: enter the official release date for this tree; this is not used internally, the tree will be usable right away

A click on "Save" will create a domain and present the domain details

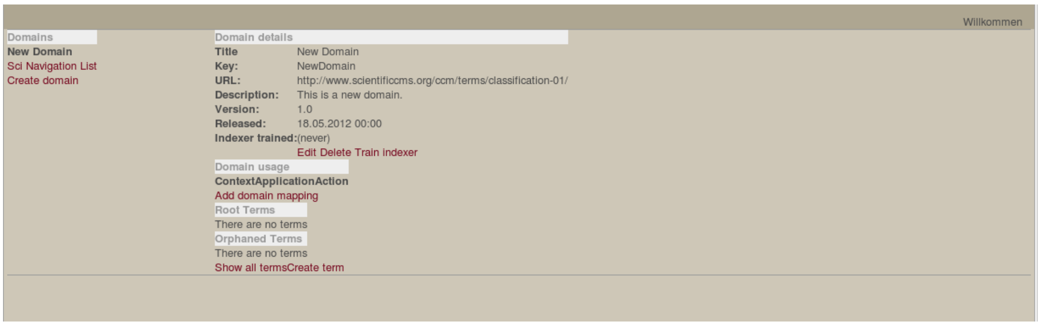


Abbildung 11

screen which lets you perform the next administration tasks.

20.1.2 Edit Domain Details

Below the *Domain* details you find a button to editing the *Domain's* details.

Abbildung 12

You can modify all the properties but the key. It is immutable.

As soon as you save your editing, the changes take effect immediately. There is no preview mode for *Domains*.

20.1.3 Delete a Domain

A click on the **Delete** button will delete the Domain immediately. Be extremely careful!

20.1.4 Train Indexer

(coming soon)

20.2 Map a Domain Usage

All Domain mapping is done from the Domain's details screen. To activate the details screen from the entry screen click on the Domain you have to work on. As an example if you click on the default Navigation Domain the screen is redrawn:

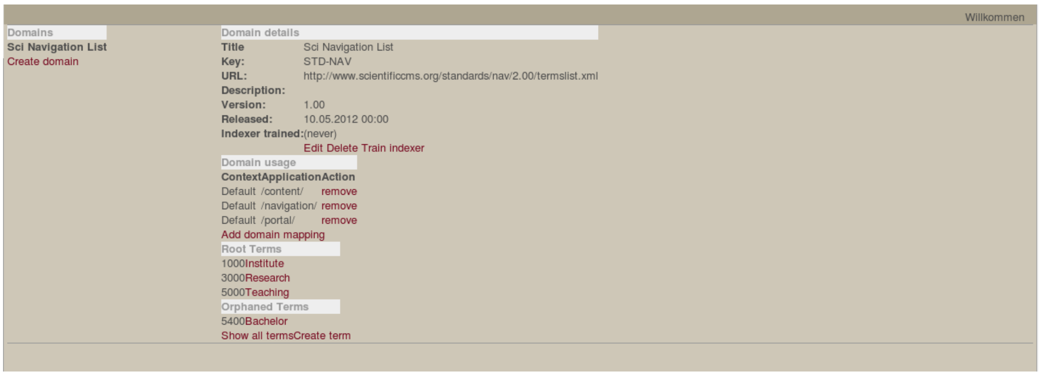


Abbildung 13

A click on the **remove** button will remove a mapping immediately.

A click on Add domain mapping opens a window where you can add a new domain mapping:

Abbildung 14

The select box lists all installed applications. Select the appropriate one.

20.3 Managing Terms

20.3.1 Creating a term

The next step is to create the terms (categories) within the domain. For each category, you need to create a term, by clicking on the "Create term" link. This will show a form:

9. **Unique ID**: enter a number, which must be unique within the category tree

10. **Name**: enter the name of the category, as it will be displayed on the website
11. **Description**: enter a brief description of the category (this is optional)
12. **Shortcut**: leave this blank (it does not seem to be used)
13. **In A-Z**: leave it to "no" (it does not seem to be used)

Then click on **Save**.

20.3.2 Edit a term

20.3.3 Delete a term

20.4 Build a classification system

Ypou build a classification system by defining the category hierarchy and category interrelations.

Go back to the terms administration screen at:

```
http://your.server/ccm/admin/terms
```

then click on your new domain in the left-hand column. You will see all your categories under the heading "Orphaned terms".

For each top-level category in your category tree:

21. click on the category name
22. click on "Set as root term"
23. for each sub-category of this category, click on "*Add narrower term*", enter the sub-category name and click on "Save", then click on the sub-category name

By now you will have defined the structure of the first- and second-level categories. If you have third-level categories, repeat the process of "Add narrower term" to place them underneath their second-level category. Then repeat this again if you have fourth-level categories, and so on.

21. Terms Command Line Administration Tools

As with all command line interface tools yo have to log in to your server and make the `WEB-INF/bin` directory the default.

21.1 Terms Import Tool

The TERMS import tool is used to import controlled lists, hierarchy definitions and mapping files both within and between TERMS domains. You can create a new *Domain* including *Terms* or you can update an existing *Domain*.

The import works on one or more files describing the *Domain*. Currently, two import file formats are supported:

- TERMS internal format
- SKOS format

21.1.1 Importing a new category tree using the TERMS format

XML files

Each category tree is defined in a set of 2 or 3 XML files: one for the categories ('terms'), one for their hierarchical structure, and (optionally) one for 'related' relationships between the categories. There may be additional XML files if you want the new category tree to be mapped to an existing one. In any case, the first thing to do is to get hold of these XML files.

The installation bundle installs some example files as well as the actually files from which the initial domains had been created to the server. You will find at least the following files at `~/WEB-INF/bin/bundle/res/bundle/categories/`

12. xample-nav-domain.xml: definition of the categories/terms
13. xample-nav-hierarchy.xml: structure of the categories
14. xample-nav-related.xml : 'related' links between categories

15. xample-nav-mapping-anav.xml: mappings between categories

The first two XML files are the most important ones; the other ones are effectively optional.

Loading the XML files

A systems administrator is required to perform this step.

In order to 'load' these files, you need to use the Importer command-line utility. This is done by running in a shell:

```
ccm-run com.arsdigita.london.terms.Importer a-file.xml
```

This needs to be done for each XML file.

So to load the example files, you would do:

```
ccm-run com.arsdigita.london.terms.Importer \
        xample-nav-domain.xml
ccm-run com.arsdigita.london.terms.Importer \
        xample-nav-hierarchy.xml
ccm-run com.arsdigita.london.terms.Importer \
        xample-nav-related.xml
ccm-run com.arsdigita.london.terms.Importer \
        xample-nav-mapping-anav.xml
```

The domain file should always be loaded first, followed by the hierarchy file. There is no need to restart the server afterwards.

21.1.2 Importing a new category tree using the SKOS format

SKOS (Simple Knowledge Organization System) provides an alternative RDF schema for defining vocabularies.

SKOS Files

Using SKOS syntax, a single file is used to define the terms, hierarchy, synonyms and relations. An example, being a subset of the AGROVOC vocabulary, is shown below:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
>
  <skos:ConceptScheme
rdf:about="http://www.fao.org/aos/agrovoc4">
  </skos:ConceptScheme>
  <skos:Concept
rdf:about="http://www.fao.org/aos/agrovoc#c_3">
    <skos:prefLabel xml:lang="en">ABA</skos:prefLabel>
    <skos:altLabel xml:lang="en">Abscisic acid</skos:altLabel>
    <skos:broader
rdf:resource="http://www.fao.org/aos/agrovoc#c_3397"/>
    <skos:broader
rdf:resource="http://www.fao.org/aos/agrovoc#c_32543"/>
  </skos:Concept>
  <skos:Concept
rdf:about="http://www.fao.org/aos/agrovoc#c_4">
    <skos:prefLabel xml:lang="en">Abaca</skos:prefLabel>
    <skos:altLabel xml:lang="en">Manila hemp</skos:altLabel>
    <skos:broader
rdf:resource="http://www.fao.org/aos/agrovoc#c_3494"/>
    <skos:related
rdf:resource="http://www.fao.org/aos/agrovoc#c_4997"/>
  </skos:Concept>
  <skos:Concept
rdf:about="http://www.fao.org/aos/agrovoc#c_5">
    <skos:prefLabel xml:lang="en">Abalones</skos:prefLabel>
    <skos:altLabel xml:lang="en">Ormers</skos:altLabel>
    <skos:scopeNote xml:lang="en">Isscaap group b-
52</skos:scopeNote>
    <skos:broader
rdf:resource="http://www.fao.org/aos/agrovoc#c_7033"/>
    <skos:related
rdf:resource="http://www.fao.org/aos/agrovoc#c_3475"/>
    <skos:related
rdf:resource="http://www.fao.org/aos/agrovoc#c_4606"/>
    <skos:related
rdf:resource="http://www.fao.org/aos/agrovoc#c_5240"/>
    <skos:related
rdf:resource="http://www.fao.org/aos/agrovoc#c_6844"/>
  </skos:Concept>
  …
</rdf:RDF>
```

Each Concept element is mapped to a single Term. Each altLabel is mapped to a new synonym (non-preferred) Term. The relationship to other Term's is reconstructed from the broader and related references. The narrower references are ignored because the broader/narrower relationship is bi-directional.

Loading the SKOS File

A systems administrator is required to perform this step. In order to 'load' these files into Aplaws, you need to use the DomainImportTool command-line utility. This is done by running in a shell:

```
ccm-run com.arsdigita.london.terms.skos.DomainImportTool \
        vocabulary.rdf
```

where vocabulary.rdf is the name of the SKOS formatted file. For example:

```
ccm-run com.arsdigita.london.terms.skos.DomainImportTool \
        ccm-ldn-terms/data/skos/agrovoc.rdf
ccm-run com.arsdigita.london.terms.skos.DomainImportTool \
        ccm-ldn-terms/data/skos/GEMET.skos.xml
```

A single command line tool is capable of importing multiple files at once, although care must be taken to ensure the domain files are imported before any mapping / hierarchy files that reference them. The tool is invoked as follows:

```
# ccm-run com.arsdigita.london.terms.Importer \
      [list of XML files]
```

Note

Each XML file is processed within a dedicated database transaction, so if the command is fails partway through import, the database will be rolled back, ensuring no partial lists are saved.

Tip

To get a progress message output during the import, edit `/etc/ccm/conf/log4j.properties` setting log level on the terms package to 'INFO'.:

`log4j.logger.com.arsdigita.london.terms=INFO`

21.2 Terms Configuration

In order to configure TERMS use the command

```
ccm set [config_option=config_value]
```

22. Terms Configuration

The following list of parameters can be used to customize *TERMS* for specific needs of a site. The can be set by using the

```
ccm set [parameter]=[value]
```

command line interface.

22.1.1 Initialization Parameter

These parameter are read by the installed system at each startup. Any modification will take effect.

com.arsdigita.london.terms.traversal_adapters

String with the full name (including path) of file containing rules for configuring information in generated XML.

Example:

```
com.arsdigita.london.terms.traversal_adapters=
/WEB-INF/resources/terms-adapters.xml
```

This is the program default value.

com.arsdigita.london.terms.default_domain

String with the default navigation domain.

Example:

```
com.arsdigita.london.terms.default_domain= NAV
```

com.arsdigita.london.terms.ajax_expand_on_all_branches

Use Ajax on all branches, or bring back entire subtree on expansion of top level root.

Example:

```
com.arsdigita.london.terms.ajax_expand_on_all_b
ranches=true
```

22.2 Loader Parameter

These parameter can be changed only for the loader to use at installation time. Thereafter they are not red by the installed system.

Currently there are no Loader parameters.

Part VI
Content Types

23. Content Types

23.1 Purpose

As explained in the introductory chapters *Content Types* are a corner post and a main characteristic of the software. Refer to chapter 3 for detailed information.

The CCM framework includes a large number of content types for various purposes.

The system differentiates between

- *Generic* Content Types
  which may be part of any distribution build upon the CCM framework
- *Domain specific* Content Types
  which are specific for a special edition of CCM. ScientificCMS provides domain specific content types for publications, research projects, and more.

Forms are a special kind of a content type. Genuinely is is a core service. It's GUI has been transformed into a special content type to make it's usage easier and more flexible. The content type itself still uses core's form-builder service.

23.2 Administration Goals

Administrators should make the use of content types as easy and efficient to use for content contributors. Some content types allow to deactivate some of their properties or to preselect some property values. These settings should be adjusted to the organisation's requirements.

23.3 Administration Tools

Most of the administrative activities use a command line interface. Properties are set by a command like

```
ccm  set  [parameterName]=[value]
```

In very seldom cases a GUI for special tasks is available.

24. Forms

24.1 Purpose

The handling of forms is encapsulated into two specialized content types: *Form* and *Form Section.* If these have been installed, you can very easily create reusable online forms in a graphical user interface.

You can use forms to collect user information (such as name and address), and specify the action that will be taken when the user clicks the Submit button.

If different forms include duplicate pieces of information, such as name and address, you can create a *Form Section*. A *Form Section* lets you create and organize pieces of forms, or *components*, that you can reuse when creating forms. For example, creating a Form Section for contact information is useful for many types of forms, because you will typically always collect the same contact information such as name, address, phone number, and so on, in many forms you create.

For most widgets, you add information in Option Value/Option Label pairs. The value is the internal value that is analysed. The label identifies the widget when the form is displayed to the user and is also passed through for analysis. You can use any value you like, but information is generally easier to analyse if the data is identical in both fields.

All *widgets* (or form elements) that will be visible to the user include a Name parameter. This prevents mistakes, because like items (such as check-boxes) are grouped together. For example, if you were to create a group of check-boxes without the same name, each option/value pair would be analysed separately. The software prevents this from happening by requiring the name of the widget when you create it.

While there is no required order for entering information into the CMS, you have to add a form or form section to your CMS before you can add it to a content item.

24.2 Administration Tasks

There are only a very few administration tasks, specifically some but rarely used configuration parameters.

Forms is designed to enable content contributors to create nice forms without help from an administrator or another technically skilled person and without any technical knowledge about HTML forms and design.

The **Design** of created forms is completely determined by the site's theme engine and therefore beyond an administrators tasks (but solely a designer's responsibility). The Pro is that a content creator without technical knowledge is able to create nice forms and must not care about design. The Con may be in some situation that a content creator is not able to influence the design even if she / he has some knowledge about Web design techniques. This is a consequence of the strict separation of content creation and presentation.

24.3 Using Form Widgets

You can add the following widgets to a form and form section. Some of the widgets are standard HTML form elements, while others are custom form elements designed specifically for the CMS.

Check box group

Check-boxes allow the user to select multiple options. For example, to ask users to tell you how they contact your support department, you might include Telephone, Website, and Email, letting them select all that apply.

Data Driven Select Box

This custom form widget displays a list of all registered users or all installed packages. More data options may be added in the future.

Date field

This custom widget adds a date field to the form or form section. Use this when you want to add a date field for the user to fill in (for example, to specify when a subscription should start).

Hidden ID Generator field

This custom widget produces a hidden form field that is populated with a unique integer value every time the form is submitted.

Hidden field

Hidden fields are used most often with different scripting languages (such as Perl and PHP) to pass information to the server when the form is analyzed. For example, if you are creating a form that uses a script on your server to validate information, you might pass values to the script that could only come from your form. (This type of script is used to prevent hackers from using scripts to send spam.)

Multiple select box

Use this widget when you want a drop-down list with a minimum of four labels to be displayed on the screen. You can set the parameters so that users can select more than one value in this widget, which makes it very similar to check-boxes. However, a multiple select box uses less space on your form and can be used to help control the layout.

Up to four labels are displayed in a multiple select box. If you enter more than four label/option pairs, a drop-down arrow is displayed so that users can see the remaining entries. If you enter fewer than four label/option pairs, the remaining lines are blank in the list. In this case, you might want to use a Single Select Box, which allows multiple lines, but only displays one on the form.

Password field

While not a secure password entry field, this widget lets you add a password-entry field to your form. For example, you could add a password that track user logs into the website by saving the value entered here in the user's cookie file. The next time the user logs into the website, the password is typed into the password field. Passwords are displayed on the user's screen as a series of asterisks.

To add a secure password to the form, you must use a scripting language such as Perl or PHP.

Radio group

Radio groups let users select one option from a group. Once they select that option, they cannot select any others, unless they want to change their answer.

Single select box

Use this widget when you want a single -line drop-down list to be displayed on the screen.

To use more than one label/option pair, click the drop-down arrow so that users can see the remaining entries. If you are entering more than four label/option pairs, you might want to use a Multiple Select Box.

Submit button

Once users finish filling in the form, they must submit it to be analysed. Typically, you add the Submit button at the bottom of the form, which will use the assigned action to process the form. Use the Actions authoring step to assign the action (confirmation email, simple email, templated email, URL redirect, or XML email). Refer to XXX *Applying Form Actions*.

Text area

Text area and text field widgets are similar in functionality, because they both let you add text boxes to your form where users can type information. They differ in size: text area lets you specify the number of lines of the text area, while a text field is only one line.

When adding a text area, you must also specify the width of the field (how many characters can be displayed on the screen) and the height (how many lines can be displayed). If the user adds more text than the text area can display, scroll bars are added so that they can view the rest.

Text field

A text field is a one-line text area.

When adding a text field, you must also specify the width of the field (how many characters can be displayed on the screen) and the length (how many characters the user can type into the field), up to 70 characters. To add a field that allows users to add more than 70 characters, use a text area. The user can use their arrow keys to see the text that is not displayed in the text box.

You must also specify the data type of the field. Refer to XXX *Applying Data Types to Parameters*.

24.4 Applying Form Actions

When you add a Submit button to a form, you can apply one of the following actions:

Confirmation email

A Confirmation Email is used to send a note to the user after they submit the form and can include links for follow-up information. For example, you would use a confirmation email to send a "thank you" note to the user for filling out your form. You specify who the email is from, the subject, and the body of the email.

Simple email

A Simple Email sends a notification to you or whoever you designate to indicate that a form was submitted. It includes a subject line (for example, "User Survey Completed") and the form data, specified as key-value pairs. For example, if the form requests the user's name, the simple email will include Name: Joe Bloggs. You specify who the email is to and the subject line.

Templated email

A Templated email sends an email that includes submitted values interpolated with the body text entered when the action was created. For example, if the administrator created an action with the body text of:

```
The user's name was ::form.title:: ::form.name:: and you can contact
them at ::form.email::
```

The resulting email's body text is:

```
The user's name was Mr Joe Bloggs and you can contact them at
joe@example.com
```

URL redirect

After users click the Submit button, you can specify the page that is displayed in their browser window. For example, you can take them back to your homepage or you could take them to a special page that provides more information, such as a "thank you" message with links to more information.

XML email

An XML email sends an email formatted as an XML document to a designated recipient, typically a remote service that can process XML emails. The results would look something like this:

```
<form-data>
  <field type="java.lang.String">
    <key>name</key>
    <value>Joe Bloggs</value>
  </field>
  <field type="java.lang.String">
    <key>title</key>
    <value>Mr</value>
  </field>
  <field type="java.lang.String">
    <key>email</key>
    <value>joe@example.com</value>
  </field>
</form-data>
```

24.5 Applying Data Types to Parameters

When you are creating a text field in a form or form section, you can apply a data type. This information is used when the form is analyzed to verify that users entered the correct type of information.

Specify one of the following data types:

Single word

Users can enter one word. They receive an error message if they enter more than one word.

Single line

Users can enter more than one word.

E-mail address

Users must type an email address that includes the @ and a domain name, or an error message is displayed.

Whole number

Users must enter a whole number (integer).

Fractional number

Users can enter a number with a decimal point.

URL

Users must enter a URL, including protocol (http://) and domain name with a valid extension (.com, .net, .org, and so on).

24.6 Other Properties of Forms and Form Sections

When entering forms, you can also specify:

Answer Required

If users do not enter or select a value, an error message is displayed.

Default Value

Enter the value to be displayed in the field. Users can change this information when they use the form.

Height

Specify the number of lines to be displayed.

Label

Specify the field's description. This information is displayed on the form.

Max length

Specify the total number of characters that users can enter in the field. If they try to type more than that, an error message is displayed.

Value

For parameters that users can select, enter the value.

Width

Specify the width of the widget that is displayed on the screen. Width is not the same as length, which is the number of characters that the user can enter.

24.7 Properties of Forms and Form Sections

The properties for forms and form sections are very similar.

24.7.1 Forms Basic Properties of Forms

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Description		

Tabelle 24.1: Basic Properties for Forms

24.7.2 Forms Additional Properties Forms

There are the following additional properties:

Property	Purpose	Comment
Controls		Form Widgets. Refer to XXX *Using Form Widgets*.
Actions		Refer to XXX *Applying Form Actions*.

Tabelle 24.2: Additional Properties for Forms

24.7.3 Forms Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

24.7.4 Form Sections Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Description		

Tabelle 24.3: Basic Properties for Form Sections

24.7.5 Form Sections Additional Properties

There are the following additional properties:

Property	Purpose	Comment
Controls		Form Widgets. Refer to XXX *Using Form Widgets*.
Actions		Refer to XXX *Applying Form Actions*.

Tabelle 24.4: Additional Properties for Forms

24.7.6 Form Sections Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

24.8 Configuration Options

The following list of parameters can be used to customize *Forms* for specific needs of a site. The can be set by using the

```
ccm set [parameter]=[value]
```

command line interface.

24.8.1 Initialization Parameter

These parameter are read by the installed system at each startup. Any modification will take effect.

There are no initialization parameters yet.

24.8.2 Loader Parameter

These parameter can be changed only for the loader to use at installation time. Thereafter they are not red by the installed system.

com.arsdigita.cms.formbuilder.FormItem.sections

> A comma separated list of content sections into which FormItem should be installed. An empty list (the default) installs into the default section.

Example:

```
com.arsdigita.cms.formbuilder.FormItem.sections=content,
surveys
```

Default value is an empty list.

com.arsdigita.cms.formbuilder.FormSectionItem.sections

A comma separated list of content sections into which FormSectionItem should be installed. An empty list (the default) installs into the default section.

Example:

```
com.arsdigita.cms.formbuilder.FormSectionItem.sections=
content,surveys
```

Default value is an empty list.

Usually both parameter should be kept in sync.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 5
- Red Hat Content Management Systems 3.1: CMS Administrator Guide, chapter 4
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 4

It supersedes these guides and provides updated information.

25. Generic Content Types

ScientificCMS comes with a set of generic content types, with are supposed to be required by any research institute. Additional content types may be installed from the add-ons package.

25.1 Agenda

25.1.1 Purpose

Use the Agenda content type to produce an online agenda that includes the launch date, agenda date and time, location, attendees, subject items, contact information, and description.

25.1.2 Status

This content type is not part of any preconfigured installation bundle and must be installed by the add-ons bundle

25.1.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Agenda Date and Time		
Location		
Attendees		
Subject Item		
Contact Info		
Description		
Creation Date		

25.1.4 Additional Properties Body Text

There are the following additional properties:

Property	Purpose	Comment
Text Type		
Text Body		

25.1.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.1.6 Configuration Options

25.2 *Article*

25.2.1 Purpose

Use the Article content type to produce an article with a captioned image.

25.2.2 Status

This content type is not part every preconfigured installation bundle.

25.2.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Description		

25.2.4 Additional Properties Body Text

There are the following additional properties:

Property	Purpose	Comment
Text Type		
Text Body		

25.2.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.2.6 Configuration Options

25.3 Event

25.3.1 Purpose

Use the Event content type to produce information about an event that includes the start and end times, date description, location, description, main contributor, event type, link to a map, and cost.

25.3.2 Status

This content type is part of several preconfigured installation bundle.

25.3.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Start Time		
End Time		
Description		
Location		
Main Contributor		
Event Type		
Link to Map		
Cost		

25.3.4 Additional Properties Body Text

There are the following additional properties:

Property	Purpose	Comment
Text Type		
Text Body		

25.3.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.3.6 Configuration Options

25.4 FAQ Item

25.4.1 Purpose

Use the FAQ Item content type to produce a list of questions and answers for a frequently asked questions section.

25.4.2 Status

This content type is part of several preconfigured installation bundle.

25.4.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Question		
Answer		

25.4.4 Additional Properties Body Text

None

25.4.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.4.6 Configuration Options

25.5 File Storage Item

25.5.1 Purpose

Use the File Storage Item content type to upload one file to the site, which users can access for reference, resources, downloads, more information, and so on. This content type an alternative for the Attach Files Authoring Step.

25.5.2 Status

This content type is part of several preconfigured installation bundle.

25.5.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Description		

25.5.4 Additional Properties File Attachment

There are the following additional properties:

Property	Purpose	Comment
File Type		
File		

25.5.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.5.6 Configuration Options

25.6 Glossary Item

25.6.1 Purpose

Use the Glossary Item content type to produce glossary terms and definitions for a glossary page on your site.

25.6.2 Status

This content type is part of several preconfigured installation bundle.

25.6.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Term		
Definition		

25.6.4 Additional Properties

None

25.6.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.6.6 Configuration Options

25.7 *Job*

25.7.1 Purpose

Use the Job content type to post job openings and requirements for your company that includes the applicant’s personal information, department, job description, and contact information.

25.7.2 Status

This content type is part of several preconfigured installation bundle.

25.7.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Term		
Launch Date		
Grade		
Closing Date		
Salary		
Body/Overview		
Ref. Number		
Department		
Job Description		
Person Specification		
Contact Details		

25.7.4 Additional Properties

None

25.7.5 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.7.6 Configuration Options

25.8 Minutes

25.8.1 Purpose

Use the Minutes content type to produce meeting minutes for easy reference that includes action items and attendees .

25.8.2 Status

This content type is part of several preconfigured installation bundle.

25.8.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Reference		
Description		
Action Item		
Attendees		
Description of Minutes		

Additional Properties

Body Text

Property	Purpose	Comment
Text Type		
Text Body		

25.8.4 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.8.5 Configuration Options

25.9 *Multi-part Article*

25.9.1 Purpose

Use the Multi-part Article content type to create an article with multiple, distinct sections, automatic navigation, and captioned images for each section.

25.9.2 Status

This content type is part of several preconfigured installation bundle.

25.9.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Description		

Additional Properties

Sections

Property	Purpose	Comment
Title		
BodyText		
Image		
Caption		

25.9.4 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.9.5 Configuration Options

25.10 News Item

25.10.1 Purpose

Use the News Item content type to create a blurb that links to a press release that includes a link to the homepage, the news date, and captioned images.

25.10.2 Status

This content type is part of several preconfigured installation bundle.

25.10.3 Basic Properties

There are the following basic properties:

Property	Purpose	Comment
Name		
Title		
Launch Date		
Description		
News Date		
Homepage		

Additional Properties

Body Text

Property	Purpose	Comment
Text Type		
BodyText		
Image Type		
Image		
Caption		

25.10.4 Optional Extended Properties

All extended properties may be available depending on the local installation's configuration.

25.10.5 Configuration Options

Part VII
Navigation Administration

26. Navigation Administration Overview

26.1 Introduction

The Navigation Module uses a category tree, one of those that are defined in the terms module, to automatically construct a hierarchical menu system (tree structure), which is presented to the user for navigation through the content of a site or a subsite. For a selected menu item it automatically collects all of those content items which are associated with the corresponding category item and presents it as a compressed list (using the basic properties “title” and “summary / lead” of each content item by default) to the visitor. For the purpose of that list an index page, based on templates, is constructed automatically, which includes a link for each content item for easy access to its full content. There are several templates, which can be used to determine the basic properties of an index page, e.g. the order of the included content items (ascending or descending order, ordered by publication date, creation date or by several other criteria).

The Navigation Module offers a very systematic and straight forward way (regarding the content) to access the content of a site in a unified view (regarding the visual aspect). It is best suited for a comprehensive but semantically structured presentation of a great amount of related information with a continuous underlying concept. As an alternative you may use the Portal Module, which offers a more flexible and adjustable way of presentation.

Administration of the Navigation Module covers the following tasks:

- configuring a category tree to be used for navigation
- customizing the behaviour of the menu system
- selecting an index page template

In some circumstances you may need to

- add Your own Index Page Templates

This task is beyond normal administration work and requires developer skills.

The system in its standard installation (i.e. one of the released installation bundles) comes preconfigured for most of the above tasks.

References:
This chapter uses material from

- APLAWS FedoraHosted WIKI
  https://fedorahosted.org/aplaws/wiki/AdminNavigationGuide
  (last accessed May 2012)

27. Configuring a Category Tree for Navigation

Any APLAWS+ user with site-wide administration permission can choose which category tree should be used for the navigation of a site (or a subsite). In some cases (when loading a category tree from XML files), he/she may need a bit of help from a systems administrator. All the work for this task has to be done in the `terms administration` panel, not in the navigation administration interface.

Before proceeding you have to decide weather you will use just another of the existing category trees or wish to use a new one. If you wish to use a new one you have to create it first, using one of the existing methods of the Terms Module.

To configure a category tree for use with Navigation you need to tell both the Navigation application, and the relevant content-section(s) to use it. This is done by defining mappings in the Terms Administration Interface.

27.1 Removing existing mappings

You first need to remove whichever mappings are already defined, so that the existing navigation category tree stops being used. To do this:

- go to http://your.server/ccm/admin/terms
- click on the current navigation domain, in the left-hand navbar
- look under the "Domain usage" header for a line with a context of "Default" and an Application of "/navigation/"; if you cannot find it, you are looking at the wrong domain;
- once you have found it, click on "remove" for each content-section which you want to switch to the new navigation tree, look for a line with a context of "Default" and an Application of "/the-section-name/", and click on remove

27.2 Adding the new mappings

Now you need to add the same mappings as you have just deleted, but using the new category tree. To do this:

- go to http://your.server/ccm/admin/terms
- click on the new navigation domain, in the left-hand navbar
- look under the "Domain usage" header and click on "Add domain mapping"
- choose "/navigation/" from the drop-down list, leave the text field blank (DO NOT ENTER "Default"), and click on "Save"
- repeat the two steps above for each content-section(s) which should use the new navigation category tree: choose the content-section from the drop-down list, and always leave the text field blank

28. Customizing the Navigation Menu System

There are two configuration parameters, which can be modified by the ccm utility, which control the amount of information shown in the menu:

```
com.arsdigita.london.navigation.category_menu_show_nephe
ws=[false|true]
com.arsdigita.london.navigation.category_menu_show_grand
_children=[false|true]
```

The default values are false.

With both parameters set to false (the default) the menu is shown in a collapsed form: for a given level all the items of the level are shown as well as the subitems of the selected one (if any) and the items of the parent level(s) (without their subitems). In a standard installation the xsl stylesheet does suppress the peer items of the parent level(s) showing the menu in a very reduced or highly condensed form.

With the nephews parameter set to true it shows the subitems for the selected item (if any) and the subitems of all its peer items of the same level. For the parent level it shows the items only, not their subitems.

With the grand_children parameter set to true it shows the subitems for the selected item and their subitems. For the peer items and the parent level it shows the items only, not their subitems.

The parameter can be combined resulting in a very extensive menu.

29. Navigation Index Page Templates

Each index page is created on the basis of a template. A template is actually a jsp file, which controls the collection of information and the structure of its presentation. It does not create the visual image of the page, the "view", which is done by a xsl style sheet. The jsp acts as a "controller", not as a creator of html code.

The Navigation Module comes with three templates as part of the distribution:

- default: collects the content items in the order they have been created. You can manually modify their order
- atoz: creates an a to z paginator and collects the content items in an alphabetical order and limited by the first character of the title which corresponds to the selected letter
- recent: collects the content items in the order they have been published, latest first

The ccm-ldn-aplaws Module adds three templates to the collection:

- aplaws-legacy,
- aplaws-portal,
- aplaws-services.

As you may expect, the installation default is the *default* template. It can be modified by a configuration parameter:

`com.arsdigita.london.navigation.default_template.`

You may use the ccm command to select another one, e.g.:

```
ccm set
com.arsdigita.london.navigation.default_template=/packag
es/navigation/templates/recent.jsp
```

You must specify the complete path inside the application container.

29.1 Selecting An Index Page Template

The Navigation Administration Interface enables the administrator to select a different template for each menu item.

- go to http://your.server/ccm/navigation/admin
- in the left-hand navbar expand the navigation tree and select the item to modify
- on the right side of the main panel look for a drop down menu with the title “Template”
- select the desired template (or “inherit from parent”) and click the save button
- a new section “Assigned templates” with the selected template is displayed
- all the menu items down in the hierarchy will inherit the selection by default. You must adjust it in a next step if required.
- To revert to the previous state select “Inherit from parent” in the drop down menu of the “Assigned templates” section and click the Change button.

29.2 Adding Your Own Index Page Templates

Templates are considered part of the software distribution and are not meant to be altered by a content administrator as a routine task.

Actually, templates are jsp files and stored in the /templates/navigation/templates directory of the web application document root directory (typically /srv/[yourName]/webapps/ROOT). The name, a short description and the complete path of the files are stored in the database as well. Both must be kept in sync! You can add a template directly to the deployed location and add a corresponding entry to the database.

But be careful in a production environment! The jsp files may be wiped out after an update of the distribution, leaving their entries in the database unchanged - and you are in trouble. You have to backup your customization and restore it later.

If you need customized templates for your site you should develope a customization package which can be handled by ccm load / ccm hostinit the usual way. You can use such a package for other customization purposes, too, e.g. category trees for your site. The module ccm-ldn-aplaws is an example of a customization package to tailor the system for governmental usage. This task is beyond normal administration work and requires developer skills.

Nevertheless, if you exactly know what you are doing and are very carefully and prudent with your administration work you can add your own jsp files to the directory and register them in the database. At your own risk proceed as follows

- copy the jsp file into the ~/webapps/ROOT/templates/navigation/templates directory, e.g. mytemplate.jsp
- use ccm-run to register the new file

```
ccm-run \
com.arsdigita.london.navigation.tools.TemplateCreator \
mytemplate myNewTemplate \
/packages/navigation/templates/mytemplate.jsp
```

The parameters for the TemplateCreator class are:

```
DISPLAYNAME DESCRIPTION FULLPATH
```

30. Navigation Command Line Tools

This chapter details the functionality available in Navigation command line tools.

30.1 Navigation Template Tool

The navigation template tool can be used to add new JSP templates for use by the navigation category pages. Unlike other tools described in this chapter, it is only intended to be used in a development environment as a short cut to reloading the entire database from scratch. For production systems, the templates should be loaded by the application's Loader or Upgrader class. The tool takes two parameters on the command line, the first is the path to the JSP template, the second is the title of the template:

```
# ccm-run com.arsdigita.london.navigation.TemplateTool \

      /packages/navigation/templates/demo.jsp Demo
```

Part VIII
Portal Workspace

31. Portal Workspace – Introduction

A portal provides a facility, which can be used to integrate a large variety of units of information into a web page (the portal), each of these units having its own program, control, and presentation logic (portlets). As a simple example you may place a list of the recent 5 news item (portlet 1) into a column on the right side, a list of articles falling into a specific category (portlet 2) into a central column, a navigation menu in the left column (portlet 3) and a list of recently published content items (portlet 4) beyond it. This is just a matter of selecting the appropriate portlet from a select box and shove it to the desired position using a web based administration interface. A portal can encompass an (theoretically) unlimited number of portlets, so a very complex web page can be created quite easily.

Some basic considerations.

31.1 Portlets

Portlets are essentially configurable components that provide a view into data. The content in a portlet can include anything from a simple Webpage to a personalized view of complex data in multiple systems. From the data it generates a piece of information called a “fragment” (markup code) which adheres to a set of rules (e.g. no <html> tags) so it easily aggregates with that from other Portlets to form a portal page.

It is the responsibility of a portlet to provide the program logic to generate the content in question upon request of the portal and to deliver the generated content back to the portal. Every portlet must have an administration mode used to configure the underlying program logic.

31.2 Portal

A portal in a technical sense is a web based application which aggregates content from different sources provided by portlets. The term is a little bit ambigous. It is used for a piece of software which implements a portlet container to control and manage portlets, the Portal Server and its concrete instances. It is also used to denote a web page which aggregates the output of several portlets, the Portal Page or a collection of those. And as if not yet enough it denotes a web site which provides a specialized entry point for a specific subject, e.g. health care, the Web Portal.

Nevertheless, it is the responsibility of a portal to pick up the included portlets, to place them onto a web page in a defined manner and to render it into coherent design. Every portal must have an administration mode used to select those portlets to be included and to provide a facility to place them onto the page in a specific position. Other important characterics are single sign-on, integration from multiple sources, and personalization for users.

Portals and portlets used to be implemented in a vendor specific way. The Java Community created a standard (JSR 168, JSR 286) which aims to allow interoperability between portals and portlets of different vendors.

31.3 Administrators Tasks

References:
This chapter uses material from

- APLAWS FedoraHosted WIKI
  https://fedorahosted.org/aplaws/wiki/AdminPortalNewGuide
  (last accessed May 2012)

32. CCM Portal Workspace - An Overview

APLAWS Portal is an implementation of a (simple) portal as well as a basic portlet infrastructure. It is specific to the requirements of APLAWS collaboration and content management and does currently not comply to JSR 286 or 168. This will change in the future.

It is one of several presentation facilities in APLAWS, other being Navigation or direct rendering using the XSLT template engine. It is provided by module ccm-ldn-portal. In addition, the module ccm-ldn-aplaws specifies the behaviour of the default homepage portal, described below TODO.

32.1 CCM Portal: Workspace and Pane

CCM Portal generates a web page presenting the data provided by portlets to the visitor. It is called a Workspace. Actually, a workspace does not directly contain portlets, but is a container which can contain one or more areas, called a panes. A pane is the actual portal in a technical sense, which displays and manages portlets.

Portal Workspace can create and manage multiple workspaces which are mounted at different addresses (customizable URLs), each containing one or more panes, actually displaying and managing the portlets. An adminstration page allows for creation of multiple workspaces which make up an interlinked workspace stack. It allows even for creation of replacement stacks associated with different APLAWS applications, e.g. discussion forums, rss feeds, a document manager. Such replacement stacks may be thought of as plugins which replace the default portal pane/portlet functionality.

The installation of APLAWS Portal provides by default one portal page, i.e. a workspace, in a popular 3 partition design, where each partition is a customizable 'pane' as described above, consisting of a single column. This page is mounted at a predefined address (see below). This arrangement can not be modified (rsp. modification requires java, xslt and css programming). This is in contrast with manually created portal pages, which allow for an arbitrary number of panes.

32.2 CCM Portlets: Independent or Application Bound ¶

There are to kinds of portlets:

- Portlet, also known as independent portlet

  These portlets have no association with an application but provide their own data on a per instance basis, or may even be written without supporting data (like a "Time Of Day" portlet for instance, that displays the system time.). They are usually simpler.

- Application Portlet

  These portlets provide a view into an, probably quite complex, application or an application’s data and are closely coupled with the associated application (i.e.not independent). They are rendered with a zoom icon that when clicked, takes the user to the URL of the application instance associated with the Portlet. Forum or Document Manager are two examples which use portlets as a view into the application.

APLAWS comes with a variety of prebuild portlets.

32.2.1 Integrated Portlets

These are independent portlets and are, as part of the portal server implementation, always available:

- Application Directory - Displays a list of portal pages / a list of applications (cf. WorkspaceDirectoryPortlet).
- Content Directory - Displays a list of content categories for navigation, including links to navigation index pages.
- Content Item - Displays the body of a single content item.
- Flash - Plays Flash using open source flash player.
- Freeform HTML - Displays a freeform area of HTML text.
- Navigation Directory - Displays a selected navigation tree in selected depth, including links to navigation index pages.
- RSS Feed - Allows an RSS feed to be fetched periodically from an external site.
- Site Login - Displays a login form or user details if logged it.

- Time of Day - Displays the current date and time.
- Workspace Directory - Displays a list of workspaces (cf. ApplicationDirectoryPortlet).

32.2.2 Application provided Portlets

Several APLAWS applications use portlets as part of the applications presentation facility:

- Forum Application
  - My Forums - displays ...
  - Recent Forum Postings - displays a list of recent forum postings
- Navigation Application
  - Content Item List - displays a list of content items, sorted according to specified criteria
  - Object List - displays a list of objects, sorted according to specified criteria

Of course, the availability of these portlets depend on the installed modules.

32.2.3 Use Cases

This allows a flexible use of APLAWS Portal for different purposes. As an example:

- For a simple application web site it may be used standalone as a frond end for an application, e.g. just a forum site, nothing else.
- For an informational web site a stack of portal pages may be used to present information sorted and filtered by various criteria.
- This kind of informational web site may be combined with one or more applications (e.g. a forum as well as a document repository) using a stack of portal pages associated with different applications.
- The default portal page may be used as a flexible start page of a web site which presents a lot of information in a straightforward, well structured way using the Navigation

presentation facility (this is the default APLAWS setup for UK LAWs).

- An even more complex and rich information web site may use the well structured presentation of information using the Navigation application in combination with a portal page for selected navigation elements for a more flexible and feature rich presentation of selected information. This can be accomplished by using a stack of portal pages associated with the Navigation application.
- Of course the latter can be combined with additional applications as in one of the previous examples.

All these different use cases are facilitated by the modular structure of the APLAWS system.

33. Accessing the Portal Workspace

During installation (precisely: while initializing and loading the database with the basic information) a default instance of ccm-portalworkspace is mounted at the address /portal/ relative to your context path and prefixed by the standard prefix of your installation. This prefix is ccm in a standard installation. So you will find your portal at http://server.your.domain/ccm/portal.

It is possible to modify that address convention by configuration parameters (cf. Configuration Options below). But this is strongly discouraged. Several parts of a standard installation rely on this address, e.g. the standard theme, where the logo in the header is linked to the standard portal address.

If you navigate to the portal address after installation, you will see the well known header and footer, and a large empty space in the center. The installation procedure does not add any portlet by default. It is the administrators responsibility to customize the portal according to the sites needs.

By default configuration the default portal page is invoked as index page if a visitor specifies just the basic address (http://your.server.domain). By modifying the parameter waf.pagemap.root a different start page may be configured.

34. Styling the Portal Workspace

Portal Workspace provides a kind of grid layout for positioning a portlet. By default the workspace (i.e. the page) is divided into 3 columns, where each column is a pane as described above. Each pane can be subdivided by the user into 1 - 5 columns, each hosting one or more portlets. The portlets in a column are arranged vertically.

The 3 column default main layout could be modified by a system administrator using a configuration parameter at system startup (cf. Configuration Options below). But you may have to adjust your theme manually. So it is basically fixed. The standard theme uses a 20% / 60% / 20% column layout. This is fixed as well.

While customizing the workspace an administrator selects a pane to work on and adds a portlet to one of its columns (you always have one column, even if you specify the number of columns as 1). You can add multiple portlets to each column and can order them top - down manually. The exact appearance of the page depends on the site's theme.

For customization go to the address /portal/edit.jsp (as usual prepend your server name and the standard prefix, in fact you may go to http://server.your.domain/ccm/portal/edit.jsp).

At the top of the screen, just below the header, you see 2 links

- Edit refers to the page you are seeing (so it es basically useless)
- Admin brings you to an advanced administration page (see Advanced Configuration Options below)

34.1 Arranging the Workspace

Right below the top two entries you see a row containing several items, left most the 3 default columns, named "Left", "Middle", and "Right" by default. Here you select which of the 3 columns you will customize (i.e. populate with portlets).

There are several options available:

- The active pane is marked by an active text entry field containing its (internal) name. You may change it to a more descriptive word (e.g. navigation area). Klick Save to store it.
- You switch to another pane by following the link under that pane's name.
- You may rearrange the panes using the arrows left of the pane names. This relocates the pane (i.e. relocates the content), not the design! If you move the left pane, which is 20% wide, into the middle position, it will be 60% wide and be displayed in the design of the center column!
- You may use the trash symbol right of a pane's name to delete this pane, effectivly switching to a 2 column layout of the content. The remaining panes are ordered from left to right, e.g. if you delete the left pane, the theme still renders a 3 column screen layout and leaves the third column empty (if you previously added a new, forth pane (see below) this pane is rendered in the third column's position).
- You specify the number of columns for the active pain using the select box on the right. You must save your selection to make it active (the other options are immediately active).

34.2 Populating a pane

Beneath the row where you specify the pane arrangement there is the content area. It can be populated with one or more portlets.

First you have to activate the pane you wish to populate. For each of its configured columns you find a drop-down list where you can select one of the available portlets. The list contains all the portlets which are registered in the system and available for the authenticated user (some portlets may be available for administrators only, see paragraph Configuration Options below). Clicking the *Add* button adds the portlet to the column.

Once portlets are added, below the display area of each portlets there is a configuration bar:

- you can use the *edit* icon to invoke the portlets configuration screen.

- you can use the up/down arrow to arrange the portlets vertically in the column.
- you can use the right/left arrow to move a portlet to another column in the pane, if applicable (you can not move a portlet to another pane, you have to delete and to recreate it instead).
- you can use the trash icon to remove the portlet from the column

4.3 Advanced customization options

Underneath the portlet content area additional customization options are available:

- The *Add pane* button can be used to create an additional pane. You can use it to create a new pane if you accidentally deleted one of the three default ones. You can use it to create a forth or fifth pane. Please note that currently, the default portal page does not allow for newly created panes to be rendered, as the portal is fixed to the aforementioned 3 partition layout. This limitation does not apply for manually created portal pages.
- The *Create Theme* button may be used to create a kind of simple theme definition. It is deprecated and should not be used. Use theme manager (ccm-themedirector) instead.
- The *Edit basic properties* can be used to edit title and description of the workspace. Both are used for internal information only. You may specify any internally useful description.

35. Advanced Portal Workspace Administration Options

35.1.1 5.1 Portal Page Maintenance ¶

If you follow the `Admin` button at top of the portal customization page (cf. above) the *Workspace Admin* page (portal/admin/index.jsp) shows up. Here you can

- Add and remove *Workspace members* who are granted access to the portal page (view). By default *Public User* is added, so the portal is viewable by any visitor.
- Add and remove *Administrators*, i.e. users or groups who are authorized to modify the portal page.
- *Assign a category* to the portal page. It can be used to include the page in lists of portal pages, generated by portlets using a category as a filter criteria.
- You can *delete* a portal page using the provided *Permanently delete this application* link. The default page can not be deleted, only any additional portal pages you created (see below).

5.2 Portal Administration

For administration of the portal server application go to the address `~/portal/admin/sitemap.jsp` (as usual prepend your server name and the standard prefix, in fact you may go to `http://server.your.domain/ccm/portal/admin/sitemap.jsp`.

The page presents a list of applications included in the installed APLAWS instance. It lists the applications title, its description, a link (URL) to the main page and a link named `View`. This link adds a *Details* view for the selected application upside the list. Here you can perform administration tasks for that application:

- `Edit` the applications basic properties, title and description
- `Create` a new child application beneath the application, the **most important option** here. It provides a tool to make those

applications available which relay on the portal presentation facility (e.g. Forum), and to create additional, independently configurable portal pages, which are mounted onto different addresses and can be used to present information (content items) following a great variety of relevance criteria or categories. For details see paragraph *Use Cases* below.

36. Portal Workspace Use Cases and Examples

We describe some use cases which are widely used. It is not an exhaustive list of ways to use CCM *Portal Workspace*.

36.1 Configure a Workspace as Start Page (Home)

- In a standard installation default behaviour
- Nothing else to do as to configure the portal according to 4.

36.2 Configure a Workspace as Index Page in Navigation Application

A portal workspace can be configured to be used as part of the *Navigation application* and to replace the *navigation index page* at a selected leaf of the navigation category tree. This adds a great deal of flexibility.

There are 2 prerequisites:

- A **template** which contains a workspace component (i.e. a <portal:workspace> element) must be available in *Navigation*. The standard distribution provides a template named `aplaws-portal.jsp` in `~/templates/ccm-navigation/navigation/`. It is shown as `APLAWS Portal` in the navigation administration screen, if you open the select box for templates. This may be modified by a sites individual *APLAWS* installation.

- **Theme support** must be available for the `<portal:workspace>` element. The standard theme as provided by *APLAWS* distribution does *currently not*!

The process involves three steps

1. Create a new workspace (portal page)
2. Configure the new workspace
3. Integrate the new workspace into navigation tree

36.2.1 Create a new workspace (portal page)

This is accomplished using the sitemap tool.

```
Go to /ccm/portal/admin/sitemap.jsp
```

You see a list of applications installed in the system. A new workspace must be created as a child application of an existing one. There are two alternative ways:

- workspace creation as child of *Navigation Control Center*, i.e. navigation application
- workspace creation as child of *Portal Homepage*, i.e. portal application

Which one to choose depends on a sites theme engine.

The next few steps are identical for both alternatives:

- Select the 'view' link against the application to create a child of This will display details & show a form for creating a child application
- Open the select box *Create new child application* and select 'Portal Workspace' from the drop down list of applications and submit using the create button.
  NB, do not select anything other than 'Portal Workspace'.
- In the form that appears, the 'url' is used to form the URL of the new workspace - this is relative to the parent application, either /ccm/navigation or /ccm/portal depending on the previous decision. So be sure to only enter the last fragment, i.e. 'CATEGORYNAME-portal'!
  NB, to avoid clashing with named based category urls you may use the category name followed by '-portal' or establish some other naming convention.
- Enter a title & description as appropriate of leave as default.
- Pick a layout for the number of columns (this can be changed later).
- Submit the form to create the actual workspace.

36.2.2 Configure the new workspace

To configure the workspace you have basically to perform the same steps as in paragraph 4.

Either go to

`/ccm/navigation/CATEGORYNAME-portal/edit.jsp`

or go to

`/ccm/portal/CATEGORYNAME-portal/edit.jsp`

depending on the previously selected alternative. You may simply follow the provided link in the sitemap tool and klick on the *Edit* link in the upper left corner of the presented page or enter the address manually.

The created workspace contains just one pane by default.

- Add / remove / relocate panes as required.
  NB, the number of panes displayed on the public site is determined by the theme not by the number of panes created here! Details should be layed down in a sites theme guide.
- Add / remove / edit portlets as required.
  NB, depending on the theme your site is using, portlets must be placed in the correct column using the appropriate controls, in order for these to be rendered by the theming mechanism. Details should be layed down in a sites theme guide.

36.2.3 Integrate the Workspace into Navigation Tree

This tasks involves 2 steps:

- Assigning the category leaf where the workspace is to replace the *Navigation* standard index page to the created workspace
- Assign an appropriate JSP template to that category leaf in *Navigation* administration

Assigning a category

- Follow the `Admin` link presented in the upper left corner of the configuration pane used in the previous step or go manually to either
  -> /ccm/navigation/CATEGORYNAME-portal/admin
  or
  -> /ccm/portal/CATEGORYNAME-portal/admin
  depending on the previous decicion.

- Below the title *Assigned Categories* select the category tree, expand it, and click on the category to be used for the workspace. You may select more than one category and use the workspace at more than one category tree leaf.

Assign an JSP template ¶

You have to assign a navigation JSP template with WorkspaceViewer component to actually display the portal.

- Go to /ccm/navigation/admin/
- Select your navigation category list from the list of available categorizations and expand to your category.
    - Leave the text field *Add template Dispatcher context* as is (it's for subsites).
    - Leave the text field *Use Context* as is (set to 'default' for the default context).
    - In select box *Template* choose "APLAWS Portal" of another appropriate template
      NB, your site may use a customized template named differently.
    - Apply the template to the portal page (*Save*)
- If your "Portalised" category has child categories, then you must go to each child and change the template to the one formely used. This is because all categories inherit the template from the parent by default. If you don't do this you won't be able to navigate to any child of the "Portalised" category.

The portal page displaying the configured portlets should now show up at /ccm/navigation/PATH/TO/YOUR/CAT/

37. Portal Workspace Configuration

The following list of parameters can be used to customize *APLAWS Portal* for specific needs of a site. The can be set by using the

```
ccm set [parameter]=[value]
```

command line interface.

37.1.1 Initialization Parameter

These parameter are read by the installed system at each startup. Any modification will take effect.

com.arsdigita.london.portal.traversal_adapters

String with the full name (including path) of file with rules for configuring information in generated XML.
Example:

```
com.arsdigita.london.portal.traversal_adapters=
/WEB-INF/resources/portal-adapters.xml
```

This is the program default value.

com.arsdigita.london.portal.default_layout

String to describe the default workspace layout by specifying the relative width of each column.
Example:

```
com.arsdigita.london.portal.default_layout=25%, 25%,
25%, 25%
```

This is the program default value.

com.arsdigita.london.portal.create_user_workspaces

Boolean value whether to create user workspaces, i.e. non-admin users should have their own custom workspaces.
Example:

```
com.arsdigita.london.portal.create_user_workspaces=false
```

This is the default program value.

com.arsdigita.london.portal.excluded_portlet_types

String containing a comma separates list of portlet types not available on this site. It prevents these type appearing in the drop down list of portlets to add to the page.
Example:

```
com.arsdigita.london.portal.excluded_portlet_types=
com.arsdigita.london.portal.portlet.ApplicationDirectory
Portlet,com.arsdigita.london.portal.portlet.TimeOfDayPor
tlet
```

com.arsdigita.london.portal.admin_only_portlet_types

String containing a comma separates list of portlet types only available to administrator, administrator of homepage, or subsite frontpage.
Example:

```
com.arsdigita.london.portal.admin_only_portlet_types=
com.arsdigita.london.portal.portlet.ApplicationDirectory
Portlet,com.arsdigita.london.portal.portlet.TimeOfDayPor
tlet
```

com.arsdigita.london.portal.portlet.freeform_html.wysiwyg_editor

Boolean value whether to use wysiwyg editor for freeform html portlet. Editor is specified by `waf.bebop.dhtml_editor`.
Example:

```
com.arsdigita.london.portal.portlet.freeform_html.wysiwy
g_editor=false
```

com.arsdigita.london.portal.workspacePartyPrivilege

String denoting which privilege is granted to the workspace party. Should be "read" or "edit".
Example:

```
com.arsdigita.london.portal.workspacePartyPrivilege=edit
```

com.arsdigita.london.portal.checkWorkspaceReadPermissions

Boolean value whether to check READ permissions when viewing Workspaces. By default we don't, which is odd. Should be set to true by administrator.
Example:

```
com.arsdigita.london.portal.checkWorkspaceReadPermission
s=false
```

37.1.2 Loader Parameter

These parameter can be changed only for the loader to use at installation time. Thereafter they are not red by the installed system.

com.arsdigita.london.portal.default_url

String containing the URL on which to mount the portal workspace.
Example:

```
com.arsdigita.london.portal.default_url=/portal/
```

This is the default program value.

com.arsdigita.london.portal.default_title

String containing the he title of the portal workspace.
Example:

```
com.arsdigita.london.portal.default_title=Portal
Homepage
```

This is the default program value.

com.arsdigita.london.portal.default_is_public

Boolean value to indicate whether or not the Portal Workspace is available to the public.
Example:

```
com.arsdigita.london.portal.default_is_public=true
```

True is the default program value.

Part IX

CCM Subsite

38. Subsite – Introduction

38.1 What is a Subsite?

Subsites are, to a web user, like completely distinct websites, while in fact they are created and managed from one single APLAWS+ instance. For example, a local authority (MyCity) can have several websites:

- www.mycity.gov.uk
- business.mycity.gov.uk
- tourism-mycity.com
- intranet.mycity.gov.uk

All four websites can be managed from a single APLAWS+ instance; www.mycity.gov.uk would be the main site of this instance, and the three other sites would be subsites.

38.2 How does it work?

Each subsite is given a specific URL to "listen" to. Each time a web user views a page on a subsite, APLAWS+ uses the URL to identify which subsite the user is looking at.

Subsites can also be configured by giving them optional parameters. These parameters effectively customise web user's experience of the subsite. Without any of these parameters, web users would see exactly the same content and layout on tourism-mycity.com than on www.mycity.gov.uk, which would be pointless. The optional parameters are:

- a navigation category tree: to customise the navigation structure of the subsite
- a portal homepage: to customise the homepage of the subsite
- a theme: to customise the graphical design of the subsite

38.3 Why use it?

Subsites allow Local Authorities to manage several sites from the same APLAWS+ instance. The only alternative would be to install several APLAWS+ instance, one for each site. The pros and cons of a subsite-based setup are:

Pros

- Cheaper hardware: only one set of servers is required
- Simpler setup and maintenance: only one APLAWS+ instance has to be installed and maintained
- Reuse of user accounts, roles, workflows: the same user data is used to manage the content for all sites, thus reducing further the maintenance work (e.g. users only need to register once, workflows created once)
- Reuse of content: the same content can be published on several subsites at once

Cons

4. Potential slowdowns: as several sites are managed via the same APLAWS+ instance, on the same server(s), very high levels of traffic can occur which may slowdown all subsites at once; the server's hardware should be specified accordingly

38.4 How does this module fit in with the rest of APLAWS+?

The subsite module is an add-on to other, usually older, modules. These other modules need to be subsite-aware in order to really benefit from the subsite functionality. This is now the case for the most important modules.

Besides, the subsite functionality benefits a lot from the using multiple content-sections. Only when several content- sections (which are a standard feature of the APLAWS+ CMS) are used, can the best be made out of the subsite functionality. This will be explained further in 2.2.3. “Setting up a subsite-specific content-section” on page 7.

38.5 Skills required to create a Subsite

38.5.1 Web Administrator

A Web Administrator should be able to perform most of the tasks involved in creating and maintaining a subsite.

38.5.2 Server Administrator

The skills of a server administrator are required for two specific tasks: setting up the URL of the subsite (see 2.2.2. “Create the subsite” on page 6), and specifying a custom navigation category tree for the subsite's content- section(s) (see 3.2. “Assigning Category Trees” on page 10).

38.5.3 Web Designer and Design Integrator

A web design and a designer integrator will be required if you want a subsite to have a custom theme (graphical design), see 3.7. “Custom XSL Configuration” on page 15.

39. Create a Subsite Step by Step

39.1 Setup the URL of the Subsite

A server administrator should get APLAWS+ to listen to the URL of the subsite you are going to create.

The subsite module distinguishes the incoming traffic by hostname, so your server administrator should configure a DNS entry for each subsite. As ever local hostname entries can be used for testing.

If you are proxying the traffic (e.g. using Squid or Apache's mod_proxy), then you will need to take care to preserve the distinction between the traffic for the different subsites. This may result in one set of hostnames for the traffic to the proxy (the subsite's "public URL"), and another set of hostnames for the internal traffic between the proxy and the APLAWS+ service (the subsite's "internal URL"). In this case the APLAWS+ subsite modules should be configured to respond to these internal subsite URLs.

39.2 Create the Subsite

Go to `/ccm/admin/subsite/`, then:

- enter a Title and Description for the subsite
- select "Create custom homepage" if you want the subsite to have its own portal homepage; otherwise select "Use main site homepage"
- in the Hostname text field, enter the subsite's *internal* URL (or if you are not using Apache as a front-end, enter directly the subsite's URL); be careful when typing this as there may be some issues with changing it later on
- if you want to use a specific theme for the subsite, select if from the "XSLT Directory" drop-down list, otherwise select "Site Wide Default"
- do no enter anything in "XSLT Directory (Other)"
- in the "Root category" drop-down list, select the category tree you want to use for the navigation of the subsite

- click on "Save"

39.3 Setting up a Subsite-Specific Content-Section

If you intend your subsite to have its own, specific content, then it is advised that you create one (or several) content- sections specifically for the subsite. This will simplify the management of the subsite, as you can then specify which content types, which workflows, and which permissions are in use just for the content of the subsite. Also, it will reduce the risk that content is created or published for the wrong subsite.

Section 3.1. “Creating a new content section” on page 8 explains how to setup a new content-section.

39.4 Setting up a Subsite Specific Search

It is a common requirement that the search page, in each subsite, returns only content published for that subsite. To achieve this, you need to setup one or several subsite-specific content-sections (see above), then you will need to customise the search XSL. 3.5. “Searching within Subsite only” on page 14 explains how to do this.

39.5 Configuring the Portal Homepage

Finally, if the subsite was set to have its own portal homepage (see section 2.2.2. “Create the subsite” on page 6), you will need to configure that homepage. 3.3. “Configuring the subsite's Portal Homepage” on page 13 explains how to do this.

40. Common Administration Tasks

40.1 Creating a new content section

This can be done using your project's enterprise.init.in file. Edit (or create) this file, and add one SectionInitializer statement. Within this statement, you can specify which roles to create, and their permissions. These can be modified at a later stage, using the APLAWS+ administrative interface.

Finally restart the application server (Tomcat or Resin). The content-section will be created by APLAWS+ during the restart. Go to the content-center to verify that your content-section has been created.

INSERT: IMAGE CONTENT-CENTER

A new 'intranet' content-section has been created .

Once this is all done, you should configure the content-section. This is done exactly the same way as you would configure the 'content' or 'form' content-sections. More specifically, you should:

- register the content-types that you want to be available for the content-section
- register which workflows you want to use
- register which lifecycles you want to use
- define/edit the roles and permissions available for the content-section

This is all done by going to /ccm/<my-new-content-section>/admin/

There is only one extra customisation step that you most probably want to do: specifying which category hierarchies to use (particularly, which navigation). See below for how to do this.

Note: the content-section is not, in itself, really subsite-specific. APLAWS+ does not have any way to know that a given content-section is meant to be used for a given subsite. It is more the case that you have to configure the content-section and the subsite properly, so that they can benefit from each other. This whole document explains you how to achieve this. If you do not follow these instructions correctly, some content will sooner or later be published on the wrong subsite.

40.2 Assigning Category Trees

You may want to configure various category trees for your subsite (and content-section).

40.2.1 Subsite Navigation Tree

When you created a new subsite, you were asked to specify which navigation categories to use (see 2.2.2. “Create the subsite” on page 6). This dictates which category tree is used to generate the left-hand navigation when browsing through your subsite. You can change this at any time by going back to the subsite administration screen, at /ccm/admin/subsite/, and editing the subsite.

INSERT IMAGE EDIT SUSITE ADMIN

Choosing the navigation category tree of a subsite

40.2.2 Content-Section Navigation Tree

If you have created a content-section specifically for your subsite, you should configure the content-section to use the same navigation categories as the subsite. This is very important, as otherwise the subsite's navigation will be broken.

Before doing this, a developer has to perform an important task. This task has to be done once and only once. It consists in running the following database query:

```
delete from cat_root_cat_object_map
where use_context is null
and object_id in (select object_id
      from acs_objects
      where object_type =
'com.arsdigita.cms.ContentSection'
      and display_name = '<my_content_section>');
```

This query should delete one row from the database. It is advised that a backup of the database is performed prior to running this query, as a matter of safety.

Once this has been done (and again, this needs to be done only once), you can configure the navigation tree of the content-section using the Terms administration interface, at `/admin/terms/`.

INSERT IMAGE: TERMS ADMIN

The Terms administration interface

On that interface, click on the navigation tree that you want to use (it should be in the list on the left of the screen).

INSERT IMAGE: TERMS EDIT

After selecting the Local Navigation tree

Then click on 'Add domain mapping'. Then, in the '--select one--' drop-down list, select your content-section, do not type anything in the text field next to it, and click on 'Save'. That's it!

INSERT TERMS ADMIN: ADD NEW DOMAON MAPPING

Adding a new domain mapping

40.2.3 Other Category Trees for the Content-Section

You can also use the Terms administration interface to configure which 'subject', 'local', 'rss', and 'services' categories you want the content-section to use, if any. To do this, you need to add mappings between the category tree and the content-section, just as you did in 3.2.2. “Content-section navigation tree” on page 11, but this time you should type the 'content' (that is: 'subject', or 'local', etc.) in the text-field next to the drop-down list.

For all these category trees, you do not need to run the database query mentioned in 3.2.2. “Content-section navigation tree” on page 11 (this was a once and only once task).

Note: if you categorise content from two different subsites (or, from the main site and one subsite) using the same 'subject' categories, this will generate 'Related links' from one subsite to the other, which you may not want. If you want the subsites to be completely separate in terms of navigation, do not use the same 'subject' categories for more than one (sub)site.

40.3 Configuring the Subsite's Portal Homepage

If you want your subsite to have its own Portal homepage, you should have selected that option when you created it (see 2.2.2. “Create the subsite” on page 6). If you haven't, you should go to the subsite administration screen, at `/ccm/admin/subsite/`, and select it now.

Now, configuring the subsite's Portal homepage is done in nearly exactly the same way as you would configure the main site's Portal homepage. The only different is the screen on which the configuration happen. While for the main site you had to go to:

```
http://main.site:port/portal/
```

to configure the subsite's portal homepage, you should go to:

```
http://sub.site:port/portal/internal.subsite.domain:port
/
```

For example, if your main site is www.mycity.com, and the subsite is www.tourism-in-mycity.com, while the internal address of the subsite is tourism.mycity.com:9090, then you should go to:

```
http://www.tourism-in-
mycity.com/portal/tourism.mycity.com:9090/
```

If you do not have an internal address for the subsite (because you are not using Apache as a caching front-end), then use something like:

```
http://www.tourism-in-mycity.com/portal/www.tourism-
in-mycity.com/
```

If you do not use the correct screen to configure the portal, you will not be able to properly configure certain portlets (such as the Content Directory portlet, which will not let you choose from the correct Navigation category tree).

40.4 AtoZ Configuration

If you want the AtoZ screen (at: `/ccm/atoz/`) to reflect the subsite's navigation tree, you need to do two things:

- ask a server administrator to do the following:
  - run in a shell:

```
ccm set
com.arsdigita.london.atoz.use_subsite_specif
ic_navigation_category=true
```

- restart the server

- configure the main site's AtoZ to contain one Category Provider, using the main site's Navigation category tree

This will have the effect of making the main site's AtoZ page display categories from the main site's navigation category tree, and of making each subsite's AtoZ page display categories from the subsite's navigation category tree.

One limitation to this is that the definition of aliased and black-listed categories is still done in only one place, that is, the main site's AtoZ Category Provider.

40.5 Searching within Subsite Only

If you want each subsite's search page to only display content specific to the subsite (i.e. no content from other subsite, nor from the main site), you will need to do 3 things:

- setup one or more content-section for each subsite (see 3.1. “Creating a new content section” on page 8)
- get each subsite to use its own theme (see 3.7. “Custom XSL Configuration” on page 15)
- in each theme, customise the XSL which renders the search widgets

The third point will definitely require the involvement of a web designer (or of anybody with XSL skills). The customisation consists in adding a hidden parameter to the search; this parameter will tell the search to only look for content within specific content-sections.

The exact change to make consists in modifying the search.xsl file of your theme, to make it look like this:

```
<form id="siteSearch" action="{$dispatcher-
prefix}/search/" method="get">
<label for="terms">Search site</label>
<input type="text" name="terms" id="terms" />
<input type="submit" value="Search" id="go" />
<input type="hidden" name="restrictToContentSections"
value="my-content-section1,my-content-section2,etc"/>
</form>
```

Where "my-content-section1,my-content-section2,etc" is a list of the content-section(s) which you have setup specifically for this subsite.

Once this change is done, republish the theme, and check that the search only returns appropriate content.

40.6 Sitemap Configuration

Once you have configured the Navigation category tree of your subsite, the Sitemap screen of this subsite (i.e. `/ccm/navigation/sitemap.jsp`) will automatically use the correct Navigation category tree.

40.7 Custom XSL Configuration

If you want your subsite to use its own set of XSL files for rendering purposes, you should make these files into a new Theme. Make sure this Theme has been publish and, either at subsite creation time (see 3.1. “Creating a new content section” on page 8), or by editing the subsite (at /ccm/admin/subsite/), select the Theme in the 'XSLT Directory' drop-down list.

40.8 Editing existing Subsite

You can edit existing subsites using the subsite administration screen, at /ccm/admin/subsite/. You should see an 'edit' link next to each subsite. Clicking on this will show you a form where you can change the configuration of the subsite.

INSERT IOMAGE: SUBSITE EDIT SCREEN

Editing a subsite

There are issues with editing the (internal) URL of subsites. Changing this will not change, for example, the URL to use if you want to configure the subsite's Portal Homepage. It is therefore recommended that you avoid changing it.

40.9 Deleting a subsite

It is not possible to delete subsite, nor content-section, at this point in time. If you want to de-activate a subsite, you can

5. delete its content in its specific content-sections

6. ask a server administrator to stop requests for the subsite's URL, or to redirect them to the main site's URL

41. Subsite Troubleshooting

Some of the functionality that you describe doesn't work!

Are you using APLAWS+ 1.0.2? If not, you should upgrade.

My new content-section doesn't get created when I restart APLAWS+

Make sure you have edited your enterprise.init.in file correctly. Make sure you have copied that file into <cdweb>/webapps/WEB-INF/classes/enterprise.init (**not** .init.in). Make sure there isn't already a content-section with the same name; all content-sections should have different names.

I get an error message when I go and configure my subsite's Portal Homepage!

Are you sure that you are viewing the correct URL? Make sure that you are using the subsite's internal domain name in the second part of the URL. See 3.3. “Configuring the subsite's Portal Homepage” on page 13.

When I configure my subsite's Portal Homepage, I cannot select the correct Navigation category tree

Make sure that you are viewing the correct URL. Do *not* use:
http://sub.site:port/portal
See 3.3. “Configuring the subsite's Portal Homepage” on page 13.

When I view a category on the main site, I can see a link to a content item from a subsite (or vice-versa)

The subsite's content item has probably been categorised using the main site's Navigation category tree. If you do not want this to happen, configure the content-section so that it doesn't use the same category trees (see 3.2. “Assigning Category Trees” on page 10).

When I view a content item on the main site, it shows a 'Related Link' to a content item from a subsite (or vice-versa)

The subsite's content item has probably been categorised using the main site's Subject category tree. If you do not want this to happen, configure the content-section so that it doesn't use the same category trees (see 3.2. “Assigning Category Trees” on page 10).

My subsite's AtoZ shows the main site's Navigation categories

Are you sure you have set the
com.arsdigita.london.atoz.use_subsite_specific_navigation_category
parameter?
See 3.4. “AtoZ Configuration” on page 14.

My subsite's Search page shows me links to content items from the main site

Make sure that the hidden search parameter has been added to your XSL. To check whether this is the case, view the HTML source of the search page, and look for "restrictToContentSections". If you cannot find this string, then something is wrong with the XSL (i.e. with the subsite's Theme).

42. Subsite Glossary

APLAWS+ instance

This is one installation and deployment of the APLAWS+ application. One instance may actually require several computers (a web cache server, one or several web servers, a database server). An instance can be used to create a main site, and none, one or many subsites.

Category tree

Category trees are lists of categories, organised in a hierarchy: one category will be called the 'root', and will be 'linked' to a few categories, each one of these categories will be, in turn, linked to more categories, and so on. In APLAWS+, the most important category tree is the Navigation category tree, which lets public users go from the Portal Homepage to top-level categories, then to sub-categories, then sub-sub-categories, and so on.

Content-section

A content-section is an area within the APLAWS+ administrative area, where content can be created, edited, workflowed, and published. Each content-section has therefore its own content, permissions, and workflows. It is recommended that each subsite has one or more dedicated content-sections, and that each content-section contains content used on only one subsite (or the main site).

Internal subsite URL

Each subsite is created by being given an internal URL (domain name and port). This URL is used by APLAWS+ to recognise which subsite each individual public user is looking at, at any time. But it may differ from the actual public URL that these users are seeing in their browsers. This is the case when Apache has been configured to cache queries, and translates the public URL into the internal URL.

Main site

Each APLAWS+ instance can only be used, initially, to manage one single site. This site is called the main site, and differs from subsites in various ways, notably that it cannot be configured using the subsite administration interface, at /ccm/admin/subsite/.

Portal

The Portal (or Portal Homepage) is the first page that public users see when they visit your site's URL (e.g. http://www.mysite.com/ccm). The Portal can be configured by administrators, through the addition of Portlets (small parts of the screen). Each subsite can have its own Portal Homepage.

Public subsite URL

When a public user looks at a subsite, his browser window will show a URL such as http://www.my.subsite.com. That URL is therefore called the public URL of the subsite. It may differ from the internal URL of the subsite defined in APLAWS+. This is the case when Apache has been configured to cache queries, and translates the public URL into the internal URL.

Theme

A theme is a set of XSL files which are used to dictate the graphical representation of a site, or subsite. New themes can be created for each subsite. Subsites (and even the main site) can easily be reconfigured to switch from one Theme to another.

Subsite

A subsite is an independent site which can be created within an APLAWS+ instance, in parallel to the main site. An APLAWS+ instance can in fact be used to create any number of subsite, as long as the hardware specifications are sufficient.

Part X
Various Helper Applications

43.

Shortcuts – Introduction

44. CCM Search Application

44.1 Introduction

The Aplaws+ search administration application can be found at:
`/ccm/search/admin/`

It allows you to perform the following tasks:

- Define search hosts for the remote site search (`/ccm/search/remote.jsp`)
- Define sponsored links for search results

44.2 Remote search

This feature allows you to search another APLAWS+ site and present the search results within your own site. it is currently not possible to limit the results to specific content sections of the remote site. It can be useful when you have an intranet and public site running on different instances of Aplaws and you want to allow your users to search the public site from the intranet.

These allow you to present the user with a specifically chosen link when the user is searching for a specifed term. It can be used for advertising, or for local authorities, it can be used to link search terms with the most relavent page on your site. For instance allowing you to always present your Refuse Collection page when a user searches for 'bins'.

44.3 Reindexing the site

Reindexing can be used on your Aplaws+ site to ensure that your search indexes are correct. It should not need to be done too often because aplaws should index content as it's published, taking care of the index in the background.

To reindex your site, run the following commands:

```
ccm stop
ccm-run com.arsdigita.london.search.Reindexer
ccm start
```

Depending upon the amount of content on the site, it could take a while while it completes this task.

45. CCM A to Z – Introduction

45.1 Purpose

A service application to provide an alphabetical list of the site's content.

45.2 Usage

By default the application presents the A-Z list at `~/ccm/atoz/`.

An administration interface is provided at `~/ccm/atoz/admin/`. Here you can configure the content to be listed by selecting an appropriate content provider.

Currently available providers are:

- ESD Toolkit Domain Provider
- Item Provider
- Site Proxy Provider
- Catgegory Provider

MORE TO COME!

46.

RSS Feed

46.1 Purpose

The ccm-rssfeed package implements an RSS-Feed provider which can be used to inform users about latest changes on an APLAWS+ website.

It is based on the CCM categorization system. A content item is added to an RSS-Feed by assoziating it to its corresponding domain category.

46.2 Accessing CCM RSS-Feed

46.2.1 RSS-Feed Deployment

Installed RSS-Feed are available at ~/channels/rss

A prerequisite for validity of the address is the existence a RSS category domain and a domain mapping or this category to the RSS-Feed application (using ccm-ldn-terms admin interface). This setup is not performed by the ccm-rssfeed package itself but has to be done either manually or as part of the setup procedure.

46.2.2 RSS-Feed Administration

An Administratin intervace is available at ~/channels/admin

Part XI
Various Collaboration Applications

47. Forum – Introduction

47.1 Purpose

47.2 Features

- Conversational/Threaded interface: yes
- Search through posts: not sure
  (Chris Gilbert - re search through posts. I suspect that at the moment, site search only retrieves cms items, and so doesn't retrieve forum postings, but I have submitted patches to widen the search to objects other than cms items. There may also be a need to implement search interfaces in order for posts to be indexed. If this has not been done yet, I will do it in the very near future. 2006-01-13)
- Watches (email notification of new messages): yes via alerts
- Ability to edit Posts: yes via security control
- Ability to add File Attachments to messages: yes, do ccm load ccm-cms-assets-fileattachment (I think this relates to any attachments).
- Permission-based Security: yes
- Customisable Interface: yes via css folder
- Role-based Administration: yes, incorporating security setup with roles
- Archiving: not sure
- Reports and Statistics: not sure
- Internationalisation support: not sure
- Database backend: think so
- XML support (for importing and exporting forum data): think so
- Support for forum moderation: yes
- Integration with existing databases of users: yes via security

47.3 Setting up

There are several ways to set up the forum module.

For an easy setup proceed as follows:

- Go to /ccm/portal/admin/sitemap.jsp.
  NB: NOT /ccm/admin/sitemap/. That page is deprecated and will eventually be replaced.
- Choose an existing application to mount your new forum underneath, for eg Portal. Click on 'View' next to portal.
- From the dropdown, choose 'Discussion Forum Application' and click Create.
- Give it a url, eg 'chat', name and description, and submit.
- Go to /ccm/portal/chat/.

48. CCM Document Repository

48.1 Purpose

48.2 Usage

49. Web Log (blog) – Introduction

49.1 Purpose

Weblog provides a Blogging facility. The package provides a portlet ("WebLog Portlet") to display and edit a blog. Each blogging entry has the structure of an CCM article: Title, lead, and body. A reader can create comments to each blogging entry.

49.2 Prerequisites

The application requires a portal to display the blog. So the site must at least use one portal page (e.g. the front page at root).

As an alternative the blog can be displayed on its own page, but can not be integrated into the navigation menue (Navigation application).

49.3 Usage

By default the blogging page shows a list of all blog entries by title and lead (abstract) and provides a "more" link the a details page which contains the complete text and all comments. This page enables the reader to add a comment.

Before a blog can be used, an instance of WebLog must be created using one of the administration facilities. By default one instance is created at admin/blog, e.g. `http://my.domain/ccm/admin/blog`.

To display the blog on a portal page, an instance of WebLog portlet must be installed on a portal page. It's installation UI provides (besides the standard properties title and description) a drop-down list to select the list provider, "/admin/blog" in case of the default installation in the above example.

For each additional, separate blog it is necessary to install an additional WebLog instance. This may be done using the UI at `~/portal/admin/sitemap.jsp` or the administration UI of portalserver's `~/ccm/administration` portal page. It is good practice to install the WebLog UI beyond admin of eventually the default content section.

Guess: An instance may not installed as root application.

Additional instances of WebLog portlet have to be installed with (one of) the additionally installed WebLog application instance(s) selected as provider.

49.4 Administrative Tasks

Part XII
Extra Applications

50. CCM FAQ Application

50.1 Purpose

FAQ is a tiny application which provides one or more lists of Frequently Asked Questions, independently from CMS and it's faq content item. It's index page provides a list view, and, if the user is logged in with appropriate permission, a link to an admin page to add, edit, and delete questions.

If a site includes the CMS module it is probably preferable to use the content type FAQ instead. It provides the facility to list FAQ's on portal pages (using one of several portlets), on a navigation page and on a standard page (e.g. using the related link facility). The list of FAQ's used by this application does not make use of the content type nor the content center administration facility. It uses a storage of it's own.

50.2 Usage

A default FAQ instance is created at ~/ccm/faq.

For each additional, separate FAQ it is necessary to install an additional instance.

50.3 Administrative Tasks

Part XIII Troubleshooting and Problem Resolving

51. Resolving Problem Debugging

This chapter details various debugging features available within APLAWS+ that are useful for diagnosing operational and development problems.

51.1 XSL / XML debugging

This section details a couple of tips which can be useful when writing XSL for APLAWS+ .

51.1.1 Downloading XSL files in a JAR

The themes application (at /ccm/admin/themes) provides links to download a JAR file containing the global base styles and individual JAR files containing per-theme styles. Sometimes it can be useful to download a JAR containing just the XSL for a particular page on the site. This can done by adding the 'output=xsl' parameter to any page URL. If waf.debug is enabled, then a link will also appear at the bottom of every page.

51.1.2 Viewing XML for a page

The XML document for any page on the site can be seen by appending the 'output=xml' parameter to any page URL. If waf.debug is enabled, then a link will also appear at the bottom of every page.

51.1.3 Viewing bebop structure

If the waf.debug parameter is enabled, then appending the 'debug=transform' parameter to any page will show a summary of the following information: XSL files applied, XML document, bebop class-name generating each XML element, the bebop component hierarchy and visibility of each component. A link to show/hide this information will also appear at the bottom of every page.

51.1.4 Fancy XSL error handling

The themes application validates all XSL before publishing, and when viewing the site in preview mode. Any errors are then reported directly in the web pages rather than then log files. If the themes application is not being used, then this behaviour can be enabled system wide by setting the waf.bebop.fancy_xsl_errors parameter to true.

51.2 Configuration Registry Browser

The WAF registry maintains the state of all configuration parameters an APLAWS+ system. Some of these are aimed at developers integrating a collection of applications and will not need to be changed, while others are for administrators to customize.

To current settings in the config registry (along with a explanatory description and example values) can be viewed online in the developer support area of the site. This is at the URL /ccm/ds/config/. To change values, the 'ccm set' command line program should be used.

51.3 Log4j code debugging

This chapter outlines how to change the log4j levels, enabling debug output to be sent to the system log files. Bear in mind that generating lots of log4j output can severly impact on the performance of the server.

51.3.1 Temporary settings

The log4j levels can be changed interactively using the WAF developer support pages at /ccm/ds/log4j. Changes made here will be reset next time the server is started, so this is recommended for production systems.

51.3.2 Permanent settings

To make permanent changes to the log4j debug settings, edit the file /etc/ccm/conf/log4j.properties. This file is read upon server startup, and when running any of the command line tools. The file has a number of common example settings commented out.

51.4 Request and query logging

This section details how to enable logging of all requests made to a server. This records the URL parameters, execution time and all SQL queries run for a request.

To turn on request logging, go to the WAF developer support pages (/ccm/ds/) and click on the 'enable request logging' link. Then go off, make a few requests to the application page to be debugged, and come back to the developer support pages. There will now be a list showing the requests just made (upto the last 50). If submitting a bug report about performance of a page, then use the 'query log' link to save a complete log of execution time per SQL query. To view a stack trace, or SQL explain plan, then follow the main link to the request info page.

> **Note**:
> Request logging is automatically disabled upon server startup to prevent an unneccessary impact on system resources / performance.

> **Warning**:
> Do not leave request logging active on a production server for longer than absolutely neccessary. Request logging significantly slows down performance of APLAWS+ and in some circumstances can lead to memory exhaustion in the JVM, requiring a server restart.

Appendix

52. Glossary

administrator

Person(s) in charge of maintaining the overall settings for CMS. Administrators create roles and users, assign permissions, and define content types, templates, workflows, and lifecycles.

alerts

Email messages sent to assigned users when published content expires (exceeds its Duration date) or when tasks are completed.

approval

Stage for a content item that indicates that it can be published. Only specific roles or users have permission to approve content items.

archiving

A method for preserving previous content items, enabling you to rollback if necessary. You can view the differences between versions.

assets

Object that you can use to add content to the published webpage, including image and files.

attributes

Fields that help to define individual content items, found under the Basic Properties Authoring Step.

author

Role assigned to users who can add content to a content item, following the process defined by the administrators. The default process usually includes selecting the item type and folder location, assigning a name and title, defining the attributes and content, and categorizing the content. The content item must be approved before it can be published.

authoring kit

User interface (UI) that lets administrators and users create specific content items. The UI can be customized programmatically to meet the needs of your company.

categories

Way to organize content items for easy searching by viewers. The CMS uses a category tree that includes categories, subcategories, and content items.

child

A subentry within CMS.

content

Information in a content item, which can include text, graphics, hyperlinks, and file attachments.

Content Center

Portal for the CMS, providing access to the different content sections, public sites, and content items.

content items

Individual "packets" of information that have their own workflow, life-cycle, and template, which are controlled by the content type. Content items are assigned default parameters for their type when users first create them.

content management

Software that enables administrators and users to organize and track information.

content repository

Database that holds all information in the CMS, allowing for content re-use, version control, and more.

content sections

Self-sufficient unit of organization for content that acts as a sub-site. Different content sections can have different administrators, users, viewers, workflows, lifecycles, categories, and content types.

content types

Defines attributes and hierarchy of content items. Administrators and programmers can create new content types, called UDCTs (user- definable content types). When administrators create new content types, they are children of the existing content types (address, agenda, article, event, FAQ item, file storage item, glossary item, job, legal notice, minutes, multipart article, news item, organization, press release, or service). Programmers can create parent and child UDCTs.

Deployment

Publishing new content items to the website.

Edit

Role for users who can make changes to an existing content item. Users who want to make edits must have Edit Items privileges.

end date

Indicates the date when the content item will be removed from the viewable area of the CMS.

finish (task)

Third (and final) step when producing a content item. Once a content item is finished, no further edits can be made and it can be published to the public site.

form

A way to organize input from viewers, such as name and address.

groups

Way to categorize users and other groups under roles. Groups must be created programmatically in the Groups Administrator interface in the Web Application Framework.

Once groups have been created, administrators can assign permissions to them, which are inherited by the group's users. Administrators can create groups of groups. See Roles.

history

List of all changes made to a content item since it was originally created. The History list lets you analyze and track changes over time, and rollback earlier versions when necessary.

home folder

Folder that is displayed when a user enters the CMS. Users get to choose their own home folder. Administrators can reset a user's home folder to **none** when necessary.

Java Server Page (JSP)

Technology for controlling the content of a webpage through the use of servlets. JSPs call a Java program that is executed by a Web server. The CMS creates JSPs for the public sites.

Information Architecture

Refers to the organization of a website's structure and content, helping users find the information they want.

instance

Installation of a separate CMS within the Content Center. Each instance of a CMS has its own permissions and privileges, workflows, lifecycles, and content items.

keywords

Words or phrases that identify the content item during searches.

lifecycles

Defines the start and end dates of a content item. The default lifecycle publishes items immediately, with no expiration date. Administrators can control the lifecycles of all content items.

lock (task)

Preventing other users from being able to make changes to the content item.

metadata

Technically, "data about data." Metadata lets you further categorize content items.

multiuser

Term applied when more than one user can access the repository at the same time, making edits to those pieces where they have the appropriate permissions and privileges.

parent

Entry in the CMS at the root level of a hierarchy.

party

Another way to refer to "users or groups."

PDF

Portable Document Format. File format created by Adobe® and viewable on different platforms with Adobe's Acrobat® Reader®.

permissions

Set to allow or disallow specific actions, or privileges, by users when creating, modifying, or publishing content items. Permissions can be assigned to a role, group (if it has been created), or to individual users. Permissions consist of a user (or party), a privilege, and an object.

privilege

Named ability to perform certain actions, such as "read", "write", or "delete."

publishing

Makes a content item publicly viewable on the website.

repository

See Content Repository.

reuse

Using an object more than once within the CMS.

roles

Defines what actions users have permission to perform, such as authoring, approving, or publishing. CMS includes four default roles:

Author, Editor, Publisher, and Manager. Administrators can create other roles as necessary.

RSS

RDF Site Summary. RSS provides a method for making Web content available for distribution or syndication to users.

sections

Isolated instances of CMS that can be running at the same time. Sections provide a way to further subdivide content within a company.

security

Way to preserve the integrity of the information in the CMS by allowing only certain users to access specific areas, create or edit new content items, and more.

start date

Indicates the date when the content item will be displayed to the Viewers of the CMS.

stylesheets

Files that describe how the information in the content item is to be displayed in a browser.

tasks

Individual steps that are assigned to specific roles or users as part of the workflow of a content item.

templates

Settings that control layout, styles, and more to preserve consistency across the website.

unlock (task)

Releasing a content item so that others might make changes to it.

users

Members assigned specific tasks and roles.

version control

Tracking previous editions of a content item, allowing rollback to previous versions.

viewers

Those who are allowed to see the end results of the CMS. Administrators can specify eligible viewers as Public or Registered Users, or specify any groups that have been added. Content is only visible to those who are members of the Viewer Role. By default, the CMS assigns all administrators to the Viewers Role.

workflows

Defined series of steps, or tasks, that are required to create a content item, which allows administrators to assign specific content item types to specific roles or users. Administrators can create new workflows with additional or different steps to satisfy the needs of your organization.

XSL

Extensible Stylesheet Language, used to describe how XML (Extensible Markup Language) documents should be displayed in a browser.

References:
This chapter uses material from

- APLAWS+ Project: Administration Guide For: CMS Administrator Guide, chapter 3
- Red Hat Content Management Systems 3.1: CMS Administator Guide, chapter 2
- Red Hat Content Management System rickshaw: CMS Administrator Guide, chapter 2

It superseeds these guides and provides updated information.