

Filesystem structure

Overview

CONTENTIDOs filesystem structure seems a bit overwhelming for the novice developer but follows a quite straight forward scheme. This article describes this scheme and should help to find the files you're looking for. Files are systemwide resources.



The location of many folders can be configured to suite your needs. This can be achieved by editing several configuration files (TODO explain which). This article assumes that all folders have their default locations!

When [installing](#) CONTENTIDO you have to define where to install it. On a usual *nix box running no other sites this is probably /var/www. Lets assume this folder as **ROOT**.

After a successfull installation you will find some folders in **ROOT** which I like to explain first:

name	required	description
ROOT /cms/	yes	This folder is often called frontend or client folder. All files contained in this folder are related to a single client.
ROOT /contenido/	yes	The CONTENTIDO backend is located in this folder.
ROOT /data/	yes	Configuration files and other stuff is located in the data folder.
ROOT /docs/	no	This folder contains some documentation files. It could be deleted when read.
ROOT /setup/	no	The setup folder should be deleted immediatly after a successfull installation.
ROOT /test/	no	Could be deleted when you're not interested in developing and testing the CONTENTIDO core.

Frontend

The frontend or client folder contains files which are related to exclusively a single client. When creating additional clients you will have several of these frontend folders. The default client is named cms, but can be renamed to an arbitrary name, as long as it does not collide with existing folders in **ROOT**.

name	description
ROOT /cms/cache	Contains cached images, style sheets and JavaScript files. Images that are scaled using CONTENTIDOs image functions (TODO include link to article describing image functions) are stored in this folder using a hash as name. Module specific style sheets and JavaScripts that are used for a template are merged to a single file each and stored using the templates name.
ROOT /cms/cache/code	This folder contains the code cache (TODO add link describing the code cache functionality). When displaying an article in frontend, the PHP code of all its modules is merged into a single file, that is stored in this folder. TODO describe the naming conventions for these files.
ROOT /cms/cache/template_s_c	Smarty cache for templates that are used with the class cSmartyFrontend when using the Smarty plugin. Smarty usually precompiles its templates and stores them in this folder.
ROOT /cms/css	When developing a layout, its style sheets are often outsourced (which is really a good idea!) and should be located in this folder. This folder can have arbitrary subfolders. You can e.g. put all files for a single layout in its own folder. This folder should also be used to store any CSS framework (e.g. Blueprint et.al.) you might use.
ROOT /cms/data	This folder contains client specific files that change on a regular basis. It contains no files on its own but has a set of subfolders which are described below.
ROOT /cms/data/config	The config folder contains client specific configuration files. These are structured by defined environments as subfolders. By default there is just the production environment, but this can e.g. be extended by development, testing or staging.
ROOT /cms/data/layouts	Layouts that can be created using the CONTENTIDO backend are stored in this folder. They have each an own folder with a single HTML file, both having the same name as the layout.
ROOT /cms/data/logs	This folder is used for client specific log files. TODO explain how to enable client specific logs.
ROOT /cms/data/modules	This is probably one of the most important folders in each client folder. It contains the installed modules which have a folder structure on their own.

<code>ROOT/cms/data/version</code>	When editing style sheets, JavaScripts, layouts, modules or templates in CONTENIDO's backend, their previous content is versioned and stored in this folder.
<code>ROOT/cms/images</code>	When developing a layout, its images should be located in this folder. This folder even can have arbitrary subfolders. You can e.g. put all files for a single layout in its own folder.
<code>ROOT/cms/includes</code>	When developing modules that share functions or classes these could be located in this folder, although it might be a good idea to create another folder classes for these.
<code>ROOT/cms/js</code>	When developing a layout, its JavaScripts are often outsourced (which is really a good idea!) and should be located in this folder. This folder can have arbitrary subfolders. You can e.g. put all files for a single layout in its own folder. This folder should also be used to store any JavaScript framework (e.g. jQuery, prototype et al.) you might use.
<code>ROOT/cms/templates</code>	This folder contains cTemplate and Smarty templates. This could be either templates for content types as CMS_FILELIST or CMS_TEASER or even templates that are used by more than one module.
<code>ROOT/cms/upload</code>	All files that are uploaded using CONTENIDO's file manager or the content types CMS_IMGEDITOR or CMS_LINKEDITOR are stored in this folder. This folder can have arbitrary subfolders.
<code>ROOT/cms/xml</code>	When your modules or plugins need or create XML files, e.g. XML sitemaps or RSS feeds, these should be located in this folder.

Backend

name	description
<code>ROOT/contento/classes</code>	Contains one file for each CONTENIDO class.
<code>ROOT/contento/cronjobs</code>	Cron or pseudocron jobs are located in this folder.
<code>ROOT/contento/external</code>	External tools like CodeMirror or TinyMCE can be found here. TODO the folders backendedit and frontend have to be described.
<code>ROOT/contento/images</code>	Images that are used in the CONTENIDO backend are located in this folder.
<code>ROOT/contento/includes</code>	Contains files with functions used in the CONTENIDO backend that can also be used in plugins and modules.
<code>ROOT/contento/jar</code>	Contains the Java package org.contentido that is used for the file upload drag and drop feature.
<code>ROOT/contento/plugins</code>	CONTENIDO plugins are located in this folder.
<code>ROOT/contento/scripts</code>	This folder contains JavaScripts that are used in the CONTENIDO backend. This also includes the jQuery library.
<code>ROOT/contento/styles</code>	This folder contains CSS style sheets that are used in the CONTENIDO backend. This also includes the style sheets for jQuery.
<code>ROOT/contento/templates</code>	Templates that are used to render pages in the CONTENIDO backend.
<code>ROOT/contento/templates/_c</code>	Smarty cache for templates that are used with the class cSmartyBackend when using the Smarty plugin. Smarty usually precompiles its templates and stores them in this folder.
<code>ROOT/contento/tools</code>	Command line tools (CLI) that are used for certain CONTENIDO related tasks e.g. automatic creation of the autoloader configuration file or the detection of PHP dependencies.

<code>ROOT/contentido/xml</code>	This folder contains some XML files required for the CONTENIDO backend. <code>legend.xml</code> contains descriptions of the icon legend shown in Content Article. <code>navigation.xml</code> defines the structure of the main- and subnavigation. <code>system.xml</code> defines settings which will be displayed on the system configuration page in Administration System.
----------------------------------	--

Data

name	description
<code>ROOT/data/cache</code>	This folder is used for caching of client independant files that are created by some plugins (e.g. PIFA or PIXI) and modules. Also the CONTENIDO UpdateNotifier uses this folder for its files.
<code>ROOT/data/config</code>	Client independant configuration files are located in this folder.
<code>ROOT/data/cronlog</code>	When cronjobs are setup, these usually store *.job-files containing a timestamp in this folder to signal that and when they have been run last.
<code>ROOT/data/locale</code>	This folder contains gettext files used to localize the CONTENIDO backend.
<code>ROOT/data/logs</code>	Depending upon configuration settings some log files are maintained that can be found in this folder.
<code>ROOT/data/temp</code>	This folder is intended to be used for any temporary data. When using this folder it might be a good idea to create a subfolder for each task in order to avoid name collisions.