

aMaze

Content Management System

User Documentation

aMaze Version 1.01beta
written by Henrik Lehtinen

Tuesday, November 15, 2005

General

About aMaze

aMaze is a simple CMS, or Content Management System. It is a server-side program that presents an easily navigable and customizable web site based on the user-built directory and file structure. The current version of aMaze works as a purely passive page renderer; that is, there is no separate admin function whereby content is created or uploaded, but all content is rather uploaded to the server by the webmaster through the usual channels (ftp, scp, etc.), and the CMS only worries about presentation.

Site structure

The default mode of aMaze follows a blog-style page presentation. The whole site usually shares a common visual theme and navigation. Beyond this, it is organized into categories and story items that are sorted, for example, in a descending order by date. At the highest level, all content is split into one or more main categories. The default category **main** is therefore at the same hierarchical level as the first user-defined categories. This is so that users can create clearly separate sections of their site, without the stories in one category appearing in others as well.

Navigation

Navigation is provided automatically by aMaze, based on the underlying directory structure. Each page contains (unless overridden):

- A navigation bar offering access to the highest level categories, including main (the “front page”).
- A breadcrumb history that provides links to every parent page up to and including the highest level category.
- A link list to all child categories of the currently active page. In addition to which, the subcategories that are displayed on the page as stories can contain links to view the story by itself.

Viewing modes

There are three different display modes in aMaze: story view, list view and file view:

Story view is the basic way to display a directory's contents. A story is displayed on a page of its own, and aMaze will look at the story directory and attempt to display every file that it finds in the manner appropriate for each file type.

List view occurs when looking at a story with sub-categories, and is used to offer reduced previews of the current story's sub-categories. Any files in the directory are displayed normally on the top of the page, then the sub-categories are presented in list view. A typical blog-style news page uses story view for the main introduction, and list view for each timestamped news article.

File view is used to display a single file (be it an image, video, archive, program code, or executable), using the appropriate method of rendering. Rendering of file types can be expanded by rendering modules, but if no method of rendering is available, the file view will simply provide a direct link to the requested file.

Every page rendered by aMaze uses customisable templates to provide basic functionality. These provide the formatting of the whole site, menus, appearance of story items, etc..

When a directory is being displayed on its own in story view, every file in it that is not

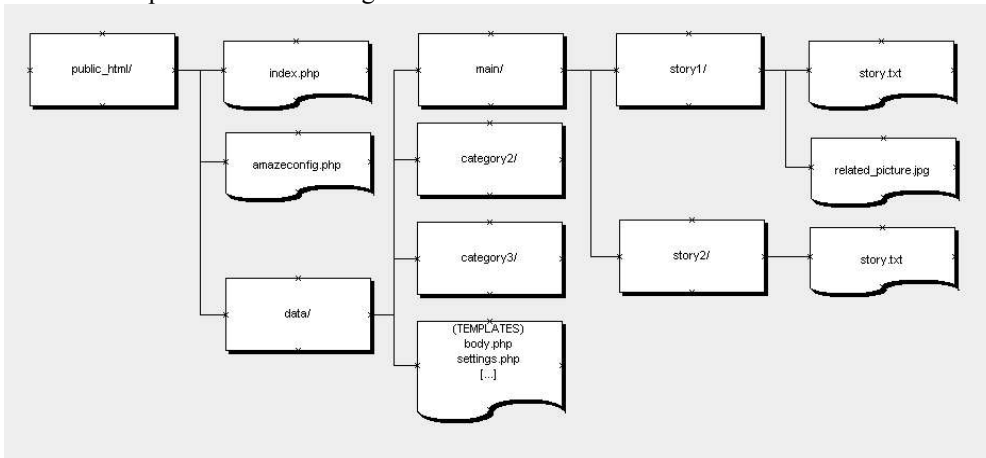
recognized as an aMaze system file is displayed in the best manner possible. Different file types have different rendering modules associated with them, as well as a sorting number. Files are shown grouped according to their extension, with the file types carrying the lowest sorting number displayed first. Within an extension, files are ordered alphabetically by file name.

When previewing a directory in list view, the same process is followed, except that there is a user-defined setting for the maximum number of items displayed. Items in list view may also use a different method of rendering from story view, depending on the file type and the rendering module associated with it.

Building your site

File structure

Here is a sample file structure diagram of an aMaze installation:



It is important to note that each story is a distinct directory. This is so that binary content can be packaged easily with the story to which it belongs. Note also that aMaze makes no distinctions between categories and story items. This allows one to add category descriptions using the normal tools for story publishing.

Each directory should have at least a **story.txt** file, which is a plain-text file with the possibility to include HTML descriptions. This is the basis for all textual content in aMaze. Story titles are derived from the directory names, with underscores ('_') replaced with spaces. If you wish to use special characters in a story title, or wish to use an especially long title, you have the option of including a **title.txt** file, with the desired title on a single line by itself, in the story directory. This will then override the directory-name as the story's title.

Content types and rendering modules

Different content types are handled by rendering modules, which are placed in the **/modules** directory. All correctly formed modules placed into the **/modules** directory are automatically installed. When aMaze starts rendering a page, it registers all supported file types and links them to the installed modules. aMaze system files are naturally ignored, and unsupported types are passed onto the **render_unknown**-module and simply displayed as direct links.

Each rendering module is passed the name and path of the file to be displayed and whether the current viewing mode is **list**, **story** or **file**. This way, rendering modules can exhibit different functionality depending on the context. For image rendering, for example, you might want to show thumbnails in list and story mode, and the full image in file mode. Similarly, for a video module, you would not want to play multiple videos at the same time. Therefore, only the file view should show the video as an embedded object. The settings for all modules are accessed by **settings.php** or **localsettings.php** files.

C u s t o m i z a t i o n

User settings

There are two different settings files, **localsettings.php** and **settings.php**. The difference between these two files is that **localsettings.php** applies for that directory only, whereas **settings.php** applies for that directory and all subsequent child directories.

The global settings for an aMaze installation are read from the file in **/data/settings.php**. Any settings files found further down in the data hierarchy will override these settings with custom functionality.

Look and feel

Customisation of the look and feel of the site is done by means of several template files. These are simple HTML snippets with special markers built-in. The default installation of aMaze comes with a complete set of template files.

These template files are at the root of the **/data/** directory, and modifying them will affect the appearance and functionality of the whole site. Template files can also be created in any position of the **/data/** hierarchy, where they will override the default settings for that category and every category below them. In addition to the HTML-based templates, which will completely replace the overridden file, the **settings.php** and **localsettings.php** templates only have to include the variables that user wants to change.

The different template files are as follows:

settings.php

Sets variables concerning behaviour, such as number of stories displayed per page, the sorting order, etc. with changes propagating downwards. Note that some settings are restricted, and can only be set in the original **/data/settings.php**. These are clearly marked in the file.

localsettings.php

Similar functionality to **settings.php**, except that the settings here do not propagate onto child categories.

body.php

Defines the enveloping HTML code of every page on the site. Includes the markers **\$categories**, **\$breadcrumbs**, **\$menu**, **\$storyspace** as inline code, which are replaced by aMaze with the appropriate functionality when the page is displayed.

catitem.php

Defines the repeating HTML look for every main category link.

catitem_active.php

Defines the HTML look for the active main category link.

menuitem.php

Defines the repeating HTML look for every sub-category link.

crumb.php

Defines the repeating HTML look for every breadcrumb link.

story.php

Is used to format story items. Uses the variable **\$story**, to display the main content, and **\$story_time** and **\$story_date** to display a timestamp in the place the user specifies.

Appendices

I. Settings files

Here is an overview of the general settings available in the settings files:

\$site_title

String variable that sets the site title as displayed in the browser's titlebar, or used in the templates.

\$category_sortby

Category sorting scheme. One of DATE, TITLE.

\$category_sortorder

Category sorting order. One of ASCENDING, DESCENDING.

\$story_sortby

Story sorting scheme. One of DATE, TITLE.

\$story_sortorder

Story sorting order. One of ASCENDING, DESCENDING.

\$story_dateformat

Formatting string for the date of the main story that is passed to the PHP date function, requesting the current date. This is separated from the time string so that the date can be easily omitted or replaced with a string such as "Today", where appropriate.

\$list_dateformat

Formatting string for the date of any listview stories.

\$story_timeformat

Formatting string for the time of the main story.

\$list_timeformat

Formatting string for the time of any listview stories.

Note that, in addition to general site settings, all the settings for rendering modules are placed in these files.

II. Default rendering modules

The basic modules included with aMaze are **render_story** and **render_image**. Here is their default behaviour:

render_story, Registers (**story.txt** | **.story**)

List mode	If \$render_text_lines is greater than 0 , displays \$render_text_lines worth of the text, starting from the beginning, then terminating in "... [more]", with a link to story mode. Otherwise, it displays the complete text.
Story mode	Displays the complete text.
File mode	Displays the complete text. (File mode for text files usually wouldn't happen, but it can be used to display a text without associated images or other files.)

render_image, Registers (.jpg | .gif | .png)

List mode	Displays image thumbnail. If no thumbnail is found, refer to variable \$render_image_showlist on whether to display the full-size image or a textual link to the image in file mode. (Thumbnails have to be provided for aMaze, and follow the form image.thumb.extension for every image.extension .)
Story mode	Displays image thumbnail. If no thumbnail is found, refer to variable \$render_image_showstory on whether to display the full-size image or a textual link to the image in file mode.
File mode	Displays full resolution image.