



CMS Container

{ HTML Guide }

[.v 04]

CONTENTS

Introduction.....	3
CMS Container in a nutshell.....	3
Target group of this manual.....	3
Document structure.....	3
1. CMS Container basics.....	4
1.1 Overview.....	4
1.2 Templates.....	4
1.3 Portlets.....	4
2. Content independent designs.....	5
2.1 Dynamic website design.....	5
2.2 Portlet oriented design.....	5
2.3 Portlets: under the hood.....	6
2.4 Views.....	6
2.5 General design process.....	6
3. Design guidelines.....	7
3.1 Layout requirements.....	7
3.2 Text areas.....	7
3.3 Design for content.....	7
4. HTML guidelines.....	8
4.1 General requirements.....	8
4.2 Setting up stylesheets.....	8
4.2.1 Using stylesheets.....	8
4.2.2 Proper stylesheets.....	8
4.2.3 Separating colors.....	9
Glossary.....	10

INTRODUCTION

CMS Container is an open source content management system for creating and maintaining dynamic websites. The heart of the system is MMbase (<http://www.mmbase.org/>), also an open source system.

CMS Container in a nutshell

CMS Container is a portal based Content Management System and is used as a base to implement the (new) look and feel of a website.

The whole process of building a new website with CMS Container can be summarized in six steps:

1. Design a new look and feel using your graphical tools
2. Create static HTML / CSS pages (a "template")
3. Distinguish functionality in the HTML pages
4. Convert the static HTML pages into dynamic pages and create the portlets
5. Implement and test the system and add functionality when required
6. Deliver the new website

Target group of this manual

The target group of this document is the designer, who has to create the visual design and has to deliver the HTML templates and stylesheets. Not in all cases this task is done by the same person.

For the designer, only steps 1 to 3 of the process will be of importance, as the technical party will process the remaining steps. This document will only focus on steps two and three.

This document will emphasize on how to create a "content independent" design, that will enable the technical staff to easily convert the templates to dynamic pages which can be used by CMS Container. This creation process is tied to several guidelines and restrictions which should be kept in mind by the party that will create the HTML templates.

Document structure

- 1. CMS Container basics:** this chapter quickly explains the main principles of CMS Container.
 - 2. Content independent designs:** this chapter focuses on how to create content independent designs using portlets.
 - 3. Design guidelines:** this chapter lists guidelines for designing sites for CMS Container.
 - 4. HTML guidelines:** this chapter lists concrete guidelines for the HTML and stylesheets.
- Glossary:** the document ends with a glossary, which provides a basic explanation of several technical terms used in this document.

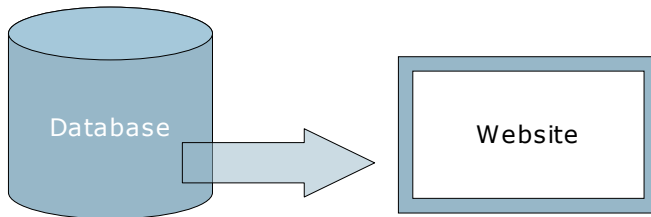
!Note Before being able to create a usable design (and template!), it's important to know how the system basically works and how portlets affect the way content is being displayed. Everyone not familiar with CMS Container is advised to read chapters 1 and 2!

1. CMS CONTAINER BASICS

Before creating a CMS Container design, it is essential to know the basics of the back-end of the system you will be designing a site for. This chapter will globally cover technical aspects that are of importance when designing a site and/or delivering the HTML for the templates.

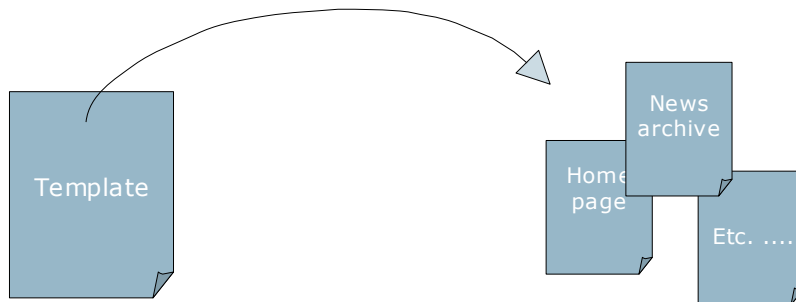
1.1 Overview

CMS Container is a Content Management System. Basically, a CMS driven website will exist of pages that contain HTML and content from a central database. This database stores all the content and objects. Editors can write articles from virtually anywhere in the world, while the webmasters can create new pages and fill them with content.



1.2 Templates

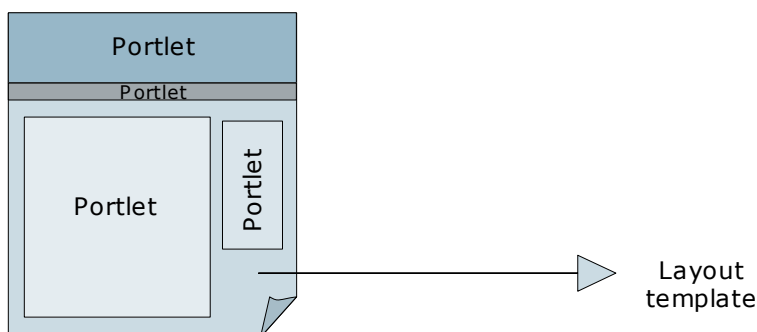
Maintaining a CMS based site eliminates the use of static HTML pages. Therefore, "templates" are used. Basically, templates are pages which contain a layout that is based upon CSS styles. Depending on your design, a site can contain several templates. But, it's also possible that only one template is defined and used for the whole site layout. One template can have more than one stylesheet attached.



1.3 Portlets

An important CMS Container feature is the use of "portlets". Portlets are basically (invisible) "blocks" in a template which can be filled with dynamic content from the database. What content will be displayed on a page and where can be chosen by the users. Portlets depend on the site layout; it's your job as a designer to specify where content can be displayed and how.

>> Creating content independent design is covered in **Chapter 2**.

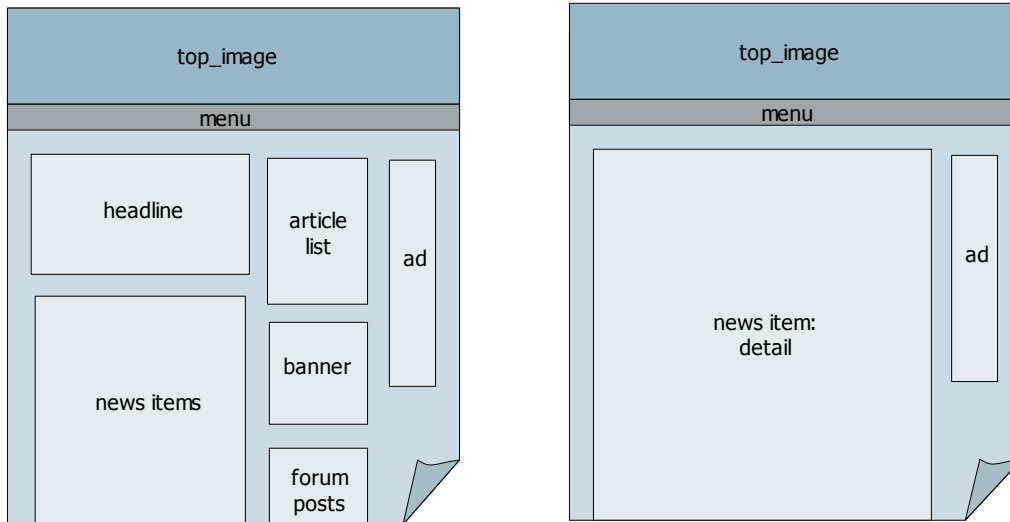


2. CONTENT INDEPENDENT DESIGNS

Designing a dynamical website is significantly different from designing a static HTML site. Dynamic sites use different ways of displaying data, which leads in most cases to efficient use of columns.

2.1 Dynamic website design

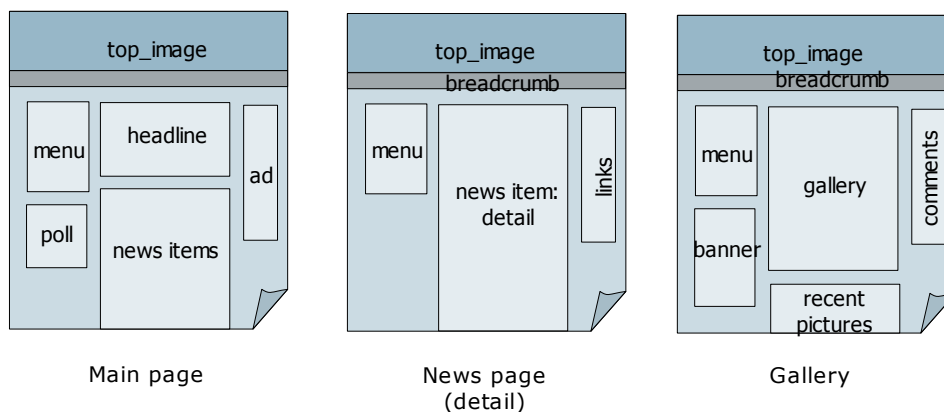
When designing a CMS Container site, you should think in “views” instead of per page. For instance, a main page can contain a summary of most of the content (using list views), while a news item has a detail view which only lists the selected news item and eventually related content (see figures below). Both of these pages have a different template; the main page uses a three column layout and the detail view two columns.



CMS Container enables you to choose if and where these items can be displayed. For instance, if you want to move the article list underneath the news items list, it's theoretically possible. The use of portlets enables this dynamical way of displaying content.

2.2 Portlet oriented design

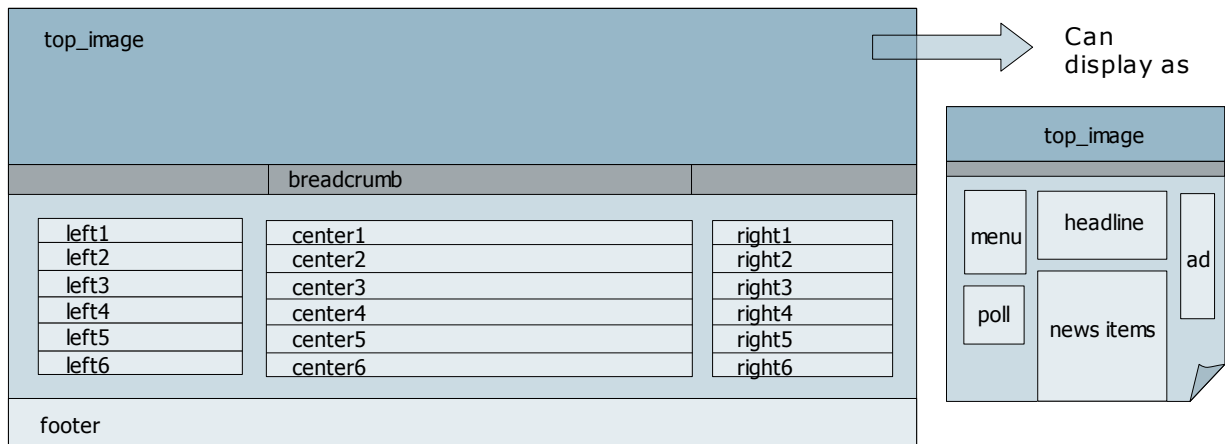
Portlets enable templates to be used for multiple purposes. This way it is possible to have one template which is used as a main page, a news page and a gallery. The figure below illustrates this principle.



As seen in the figure items use the same layout template. Still, items on the pages can be positioned everywhere. For example, on the news page it is even possible to put the links in the left most column.

2.3 Portlets: under the hood

The template for the pages shown in paragraph 2.2 will appear in the CMS like the figure below when the page is totally empty. This is the case when a new (empty) page is created. Portlets (empty or filled) are **not** visible unless this is defined in the CSS as part of the design.



A portlet can be created in each defined section (which are the named boxes). As stated before, it is possible to add the poll to left2 or right1. If a portlet is empty, nothing will be shown to the visitor.

2.4 Views

Portlets not only enable content to be positioned at different places, but also the way this content is displayed. This principle is called a "view". A good example is a news page: it is possible to show a whole list, one item in detail, the maximum number of displayed items, page numbers and so on. The figures below illustrate this principle using an existing CMS Container website.



Template with default portlets, ready to be filled



One portlet filled with news (list view)



Only two items are shown on this page



All the news items are shown with full details

2.5 General design process

Designing for CMS Container can be globally summarized in a view steps. Before actually beginning the design process you should have a clear view of all its steps:

1. Create a base design using the design guidelines (see chapter 3)
2. Define the number of unique templates needed (lay-outs) using your interaction documentation
3. Visualize your content in (Photoshop) mock-ups; try not to create mock-ups of *all* pages
4. Create HTML files + CSS stylesheets from your mock-ups using the HTML guidelines (see chapter 4). You only need to build HTML pages that have different **views**. Try not to create HTML pages of *all* pages!

3. DESIGN GUIDELINES

This chapter will focus on some concrete guidelines and requirements for the design. These are needed in order to produce an HTML template using CSS, which easily can be processed by the technical party.

3.1 Layout requirements

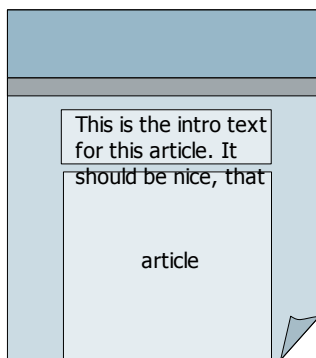
While CMS Container enables you to design from a visual perspective, it is important to keep the underlying system and its properties in mind. In general, try to create a layout that is usable for multiple purposes.

General notes concerning site design:

1. Make sure the design is made for screen purpose:
 - don't use CMYK colors
 - design for common screen resolutions
2. Use common fonts for content text (i.e. Arial, Verdana, etc.). Content text is all text which is displayed dynamically (retrieved from the database). It is strongly advised to display them in your Photoshop mock-up already (disable anti-aliasing for these types of text).
3. Make sure your design can be displayed in HTML using stylesheets. If this isn't the case, it is not guaranteed that the final result will display correctly in a browser.
4. Try to keep the overall layout (e.g. a column structure) consistent

3.2 Text areas

An important aspect concerning text retrieved from the database is its length. Since it is possible to position content anywhere possible, it is recommended to make sure enough space is defined for special text. For example, these can be intros or headers which contain dynamic text with a fixed height or graphical element as a background. The example below will try to visualize this.



In this case, an "intro text" with graphical element in the background (fixed height) is defined at the top of an article. However, as seen in the example, this space clearly has a problem as the text that is being displayed is larger than the space defined and exceeds the specified box. This clearly leads to unwanted markup of the concerning page.

To avoid similar problems, you should define enough room which can handle smaller and bigger texts, or make sure this space scales when having longer texts. Also, don't define a fixed number of rules that similar texts and headers can contain in the design.

3.3 Design for content

When creating mock-ups for pages, it is advised to design from a content perspective. For instance, when you create a front page you should try to imagine how a page visually is presented to the user. This way, the mock-up can be divided looking at its functionality when creating static HTML and stylesheets.

Try to keep the content areas identical as much as possible. This way you can reuse whole templates (or parts) for multiple purposes.

4. HTML GUIDELINES

This chapter will focus on some concrete guidelines and requirements for the HTML / stylesheet(s). These are needed in order to process the delivered templates as quick and easy as possible by the technical party.

4.1 General requirements

When delivering one or more HTML templates, make sure you meet the requirements listed below:

1. Make your XHTML W3C compliant. You can validate your XHTML using <http://validator.w3.org/>
2. Make sure your pages are cross-browser compatible using today's most popular browsers as a reference (Firefox 1.5 / 2.0 and Internet Explorer 6.0 / 7.0)
3. For accessibility, refer to the Dremfels Weg standards (see <http://www.dremfelsweg.nl/>)
4. Use a stylesheet for the overall look and feel. Try to define as much as graphical elements in the stylesheet as possible. Set up your stylesheet(s) properly.
5. Don't use images to display content text. These texts can be changed in the CMS. An exception is the use of reusable image headers.
6. Minimize the use of Javascript. This is only allowed for the use of menus. Be aware that users using non-Javascript browsers will not be able to see these menus.
7. Don't use tables for the layout; tables for statistics are allowed
8. Don't use I-frames
9. For stacked <div>'s, the z-index must be lower than 1000
10. Don't use an image as background color, unless it concerns a gradient

4.2 Setting up stylesheets

Stylesheets are used for the layout of the website. You basically deliver HTML pages + CSS styles, which the technical party will process. Since CMS Container code will directly inserted into your HTML, it is important to deliver clear code, which is set up properly structure- and/or style-wise.

4.2.1 Using stylesheets

When designing a site for CMS Container, you should try to define styles in multiple stylesheets as much as possible. CMS Container can attach one or more stylesheets to a page template.

Divide parts of the design into separate stylesheets:

- **A base stylesheet**, which holds common definitions such as font definitions, link decorations and list markups
- **Layout stylesheet(s)**, for the layout itself (e.g. column structure, blocks and areas, etc.) If you use multiple layouts, create a separate stylesheet for each layout!
- **Color stylesheet(s)**, if you use multiple color themes, for instance

4.2.2 Proper stylesheets

As stated before, be sure to set up your CSS styles properly. For syntax, we refer to common available CSS reference sources.

Special points of interest:

1. Try to put common styles in the base stylesheet as much as possible
2. Make sure menus and lists use the and tag structure
3. Use comments in your HTML and stylesheets to describe its parts
4. Use conditional comments to fix cross browser problems

!Note Above list shows only a small selection of special points of interest. This list is certainly **not** limited to this selection! More tips using proper CSS can be found in common available CSS reference sources.

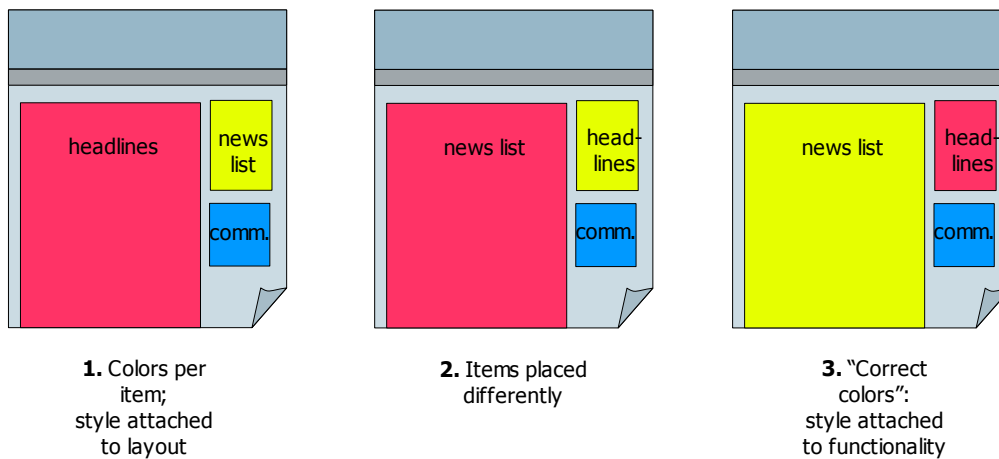
4.2.3 Separating colors

When using multiple color themes on your site, you should start by putting colors in separate stylesheets first. There are many possibilities when using colors, however, a practical example is that of parts that always keep their “own” color, independent of their position.

!Example You have a reusable template which can be used for all pages. The template contains two columns, a broad one and a slim one. The main page contains news headlines, a news list and news comments. On each page, the main subject per page is shown on the left, but all items have separate colors.

If you assign colors to parts of the layout, items will automatically inherit the defined color. In this example, the item displayed on the left side of the screen automatically gets the style assigned to its column (which is pink). However, the design defines that a “news list” should always be yellow (and not pink), so this is not correct. This can be solved by attaching the color styles to an element which describes its functionality (for instance: “news”), instead of attaching it to its position on the screen (for instance, “main column”).

The figure below will try to clarify this:



As seen, image three shows the desired result for this case.

GLOSSARY

CMS Container: an open source content management system (based on MMBase) for creating and maintaining dynamic websites

Dynamic content: content of a website which is stored, edited and retrieved in/from a central database

Dynamic website: website driven by content which is retrieved from a central database

Layout: the visual canvas in which content is placed

MMBase: an open source content management core which CMS Container is based upon (<http://www.mmbase.org/>)

Portal: a (start)page which contains blocks of functionalities

Portlet: (predefined) "blocks" on the page that can contain content

Template: (predefined) pages based on the website design that can be filled with dynamic content

View: the way content is being displayed through a portlet