hyper → ContentManagementServer
CMS

Version 5.7
Template Designers Guide

2014-10-29

Table of contents

1 General

When designing templates you should care about the effective maintenance of the corporate design. It is recommended to use only a few page templates and stay flexible by using component templates and scripts instead.

2 Template Management

Templates define the design and the appearance of your website as well as the editable elements of a page or component (e.g. text, images, links, etc.). Using either the TagWizard of the template management system or code generated with an HTML-editor (e.g. Dreamweaver) you can create new templates. HTML-files can be imported into hyperCMS directly. Using the TagWizard you can create new templates online in an easy way and without programming skills.
All templates are managed by a version management system. When editing and saving a template a new version is created automatically by the system. Therefore you can undo any change in a document or just view the single versions of a template.
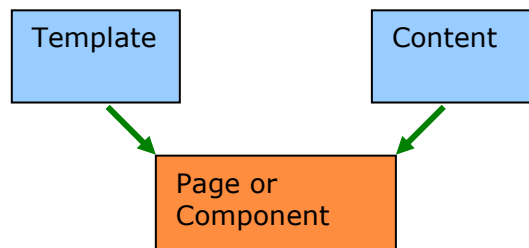

All versions of a template are being archived. Every time you save, a new version of the template is created. Therefore it is possible to view, select and use previous versions of the template.

The hyper Content Management Server distinguishes between „pages" and „components" and needs design templates for both. The functionalities of the templates, however, vary only slightly. Basically, "pages" and "components" are treated nearly the same. There are some differences in personalisation; e.g. customer trackings can only be used on pages whereas profiles are always used on components. The definition of metadata and the page title only is possible by using page templates.

Besides the above mentioned templates for pages and components there are templates for metadata, which take an exceptional position. They are used for defining the keywords for multimedia files and folders. Metadata templates therefore do not define the design but the input options of metadata and are very limited compared to templates for pages or components. HTML- and script code is ignored and without function. It is important that there is a predefined template called "default" in the system. The "default" template cannot be removed or deleted. Every time a file is uploaded or a new folder is created the default metadata template is assigned to the new object.

Please select the corresponding option from the navigation bar to gain access to the templates for pages or components. If you do so, a toolbar will be shown in your working environment. If you move the mouse pointer over one of the icons, the name and description of the action will be displayed.
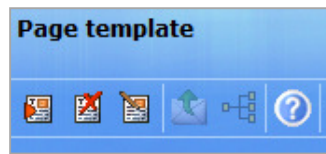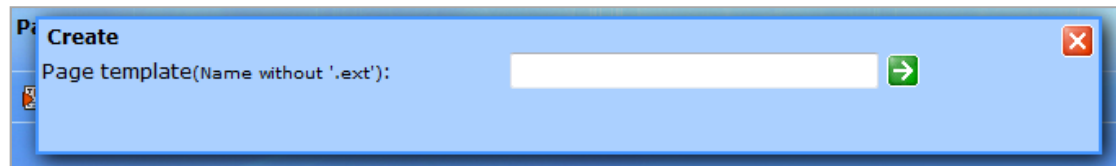

You can start the following actions:

- Creating a new template
- Uploading a file (HTML-code)
- Deleting an existing template
- Editing an existing template

3 Page-, Component- und Metadata Templates

3.1 Creating a Template

Please click on the first icon on the left of the toolbar. A description field will appear requesting a name for the new template.

Enter the name (the name must be unique for each publication). Page templates always refer to a page whereas component templates are used to create components. Template Includes on the other hand are used to amend one or various page or component templates. The usage of template includes is similar to the one of file includes. By using an include, it is possible to integrate recurrent code into any other page or component templates.
Metadata templates are used for gathering meta-information of folders and multimedia files.

After selecting the type of the template, click on „OK" and the editor will be displayed on top of your working environment. By using the editor you can now define the file extension of your page or component.
The file extension tells the web server which application to use for executing the file unless the file can directly be opened. Usual file extensions are: .asp, .php, .jsp or .html. Therefore it is important that your web server knows the file extensions and the corresponding applications to open the files. Please refer to the configuration notes of your web server.

The selection field for the application defines the template of the underlying application. Please make sure that the selected file extension corresponds with the application.

Please note the following overview of applications:

Application / Format	File Extension (usually)	Explanation
Active Server Pages	.asp	Active Server Pages (ASP) from Microsoft consists of a server-based script language related to Visual Basic.
Extensible Markup Language or Text	.xml, .txt, .xsl, etc.	The Extensible Markup Language (XML) is a data description language. The Extensible StyleSheet Language (XSL) can be used to render XML-documents. It can also manage any format based on ASCII-code like e.g. text documents. Link management, personalisation, time management as well as applications with templates can not be used because server-based program code cannot be executed. XML-documents and XSL-documents are often used by other web applications to dynamically render HTML-pages. XML-documents are also used for data exchange and are directly provided by the web server.
Hypertext Markup Language	.html, .htm	HTML is the standard language but does not allow any server-based interactivity. Link management, personalisation, time management as well as applications can not be used. HTML-documents only need a web server; e.g.: Apache or Microsoft IIS.
Java Server Pages	.jsp	Java Server Pages is a server-based programming language and is related to the computer language Java from SUN Systems. Common application servers are Tomcat, WebSphere as well as Bea Weblogic.
PHP Hypertext Preprocessor	.php, .phtml	PHP is a server-based script language. It can be used as a module on all common web servers like e.g. Apache or Microsoft IIS.

The so-called „hyperCMS TagWizard" can be found among the definition fields. You can copy existing HTML-code from your HTML-editor and paste it into the TagWizard.

On the basis of a small example we are now going to look at the practice on working with existing HTML-code in page and component templates:

The designer creates the design of a new page, which ends in HTML-code. All design-elements have been integrated in this code. The following HTML-code will show this on the basis of a small example:

Page template for creating pages

```
<html>
<head>
<title>Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body bgcolor="#FFFFFF" text="#000000">
<p>Page Title</p>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Menu1</td>
    <td>Menu2</td>
    <td>Menu3</td>
    <td>Menu4</td>
  </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td>Title</td>
  </tr>
  <tr>
    <td>Text</td>
  </tr>
</table>
</body>
</html>
```

Component Template for the main navigation bar of the page

Component Template for the content of the page

As you can see in the example above, tables in the HTML-code are removed to use them for component templates. Instead of both of the tables the hyperCMS tags for both of the components [hyperCMS:components id='menu'] and [hyperCMS:componentm id='pagecontent'] are integrated.

You should have some basic HTML-knowledge to create templates. This is why this task is usually done by web designers who generate HTML-code for pages respectively components out of their favourite HTML-editor. The generated HTML-code can then be converted into templates with the help of the hyperCMS TagWizard.

To create templates from existing HTML-code you must insert specific hyperCMS-tags. To do so, select the text or position in the HTML-code where an editable object should appear (e.g. formatted text with id='Introduction'). By clicking on one of the icons the hyperCMS tags are integrated into the HTML-code.

Be aware that an ID (Identification) has to be assigned to most of the elements. The ID can easily be defined via the form fields of the TagWizard.

To integrate media files we make use of a path variable. When publishing a page or component, the path variable is replaced by the URL (address) of the target of the publication.
%tplmedia% stands for the path of the template media repository.

The following tables describes all hyperCMS tags:

Article Tag prefix
(can be used as prefix for all text, media, link and component tags, excl. personalized component tags.)

Article	[hyperCMS:art...tag...]

Text Tag set

unformatted text	[hyperCMS:textu id='...']
formatted text	[hyperCMS:textf id='...']
Text option from text list	[hyperCMS:textl id='...' list='...']
Checkbox	[hyperCMS:textc id='...' value='...']
Date	[hyperCMS:textd id='...']
Date format (optional)	format='...'
Display name (optional)	label='...'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
Width of editorfield in pixel (optional)	width='...'
Height of editorfield in pixel (optional)	height='...'
Toolbar selection for richtext editor (optional)	toolbar='...'
constraints definitions for non-formatted text (optional)	constraint='...'
Meta-information type (optional)	infotype='meta'
Default value (optional)	default='...'
valid language value (optional)	language='...'
edit/write permission for certain user groups (optional)	groups='...'
dpi value to autoscale images (optional)	dpi='...'
Autoconvert images to the given colorspace [CMY, CMYK, Gray, HCL, HCLp, HSB, HSI, HSL, HSV, HWB, Lab, LCHab, LCHuv, LMS, Log, Luv, OHTA, Rec601YCbCr, Rec709YCbCr, RGB, scRGB, sRGB, Transparent, XYZ, YCbCr, YCC, YDbDr, YIQ, YPbPr, YUV] (optional)	colorspace='...'

Autoconvert images to the given ICC-Profile [CMYK Profiles: CoatedFOGRA27, CoatedFOGRA39, JapanColor2001Coated, JapanColor2001Uncoated, JapanColor2002Newspaper, JapanWebCoated, UncoatedFOGRA29, USSheetfedCoated, USSheetfedUncoated, USWebCoatedSWOP, USWebUncoated, WebCoatedFOGRA28 / RGB Profiles: AdobeRGB1998, AppleRGB, ColorMatchRGB] (optional)	iccprofile='...'

Media Tag set

File	[hyperCMS:mediafile id='...']
Alignment	[hyperCMS:mediaalign id='...']
Width	[hyperCMS:mediawidth id='...']
Height	[hyperCMS:mediaheight id='...']
alternative text	[hyperCMS:mediaalttext id='...']
Display name (optional)	label='...'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
Media type [audio, compressed, flash, image, text, video]	mediatype='...'
Meta-information type (optional)	infotype='meta'
use thumbnail image	thumbnail='yes'
valid language value (optional)	language='...'
edit/write permission for certain user groups (optional)	groups='...'
dpi value to autoscale images (optional)	dpi='...'
Autoconvert images to the given colorspace [CMY, CMYK, Gray, HCL, HCLp, HSB, HSI, HSL, HSV, HWB, Lab, LCHab, LCHuv, LMS, Log, Luv, OHTA, Rec601YCbCr, Rec709YCbCr, RGB, scRGB, sRGB, Transparent, XYZ, YCbCr, YCC, YDbDr, YIQ, YPbPr, YUV] (optional)	colorspace='...'
Autoconvert images to the given ICC-Profile [CMYK Profiles: CoatedFOGRA27, CoatedFOGRA39, JapanColor2001Coated, JapanColor2001Uncoated, JapanColor2002Newspaper, JapanWebCoated, UncoatedFOGRA29, USSheetfedCoated, USSheetfedUncoated, USWebCoatedSWOP, USWebUncoated, WebCoatedFOGRA28 / RGB Profiles: AdobeRGB1998, AppleRGB, ColorMatchRGB] (optional)	iccprofile='...'

Link Tag set

hyper reference	[hyperCMS:linkhref id='...']
Frame target	[hyperCMS:linktarget id='...']
Link text	[hyperCMS:linktext id='...']
Display name (optional)	label='...'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
Meta-information type (optional)	infotype='meta'
valid language value (optional)	language='...'
edit/write permission for certain user groups (optional)	groups='...'

Component Tag set

Single component	[hyperCMS:components id='...']
Multi component	[hyperCMS:componentm id='...']
Display name (optional)	label='...'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
static include (optional)	include='static'
Hide icons on edit (optional)	icon='hidden'
Meta-information type (optional)	infotype='meta'
Default value (optional)	default='...'
valid language value (optional)	language='...'
edit/write permission for certain user groups (optional)	groups='...'

Title Tag set

Page title	[hyperCMS:pagetitle]
Display name (optional)	label='...'
Default value (optional)	default='...'
valid language value (optional)	language='...'

Meta information Tag set

Author	[hyperCMS:pageauthor]
Keywords	[hyperCMS:pagekeywords]
Description	[hyperCMS:pagedescription]
content-type	[hyperCMS:pagecontenttype] [hyperCMS:compcontenttype content='...']
Language	[hyperCMS:pagelanguage]
Display name (optional)	label='...'
Default value (optional)	default='...'

| edit/write permission for certain user groups (optional) | groups='...' |

Language session setting

Session name and values	[hyperCMS:language name='...' list='...']
Display name (optional)	label='...'
Default value (optional)	default='...'
valid language value (optional)	language='...'

Personalization Tag set
(content can be personalized for customers based on the customer profile of a component)

Customer tracking	[hyperCMS:pagetracking]
Default value (optional)	default='...'
edit/write permission for certain user groups (optional)	groups='...'

Script Tag set

| Template script | [hyperCMS:scriptbegin ... scriptend] |

Database Connectivity Tag set
(DB-Connect file has to be defined)

| DB-connectivity | [hyperCMS:dbconnect file='... File ...'] |

Workflow Tag set
(workflow must be defined)

| assign workflow | [hyperCMS:workflow name='...'] |

Stylesheet references for components Tag set

| Styelsheet (optional) | [hyperCMS:compstylesheet file='...'] |

Template and file inclusion Tag set

| Template include | [hyperCMS:tplinclude file='...File ...'] |
| File include | [hyperCMS:fileinclude file='...File ...'] |

View option of the object Tag set

| Name of the view (cmsview, inlineview, preview, formedit, formmeta, formlock, template, publish) | [hyperCMS:objectview name='...'] |

Please note: Each ID of a text-, media- or component-tag set within a page, article or component must be unique. Please use the same ID for multiple and similar elements of a tag set (e.g. unformatted text). Therefore changes on one element also effect other elements with the same id; e.g. multiple usage of the tag [hyperCMS:textu id='Introduction'] within the same template.

It is not allowed to use different types of the same tag with the same ID, like e.g. [hyperCMS:textu id='Introduction'], [hyperCMS:textf id='Introduction'] and [hyperCMS:textl id='Introduction']. In this case hyperCMS will display an error message on the preview screen.

Please also note that the tag [hyperCMS:mediafile id='...'] always links to the media database. The tag [hyperCMS:linkhref id='...'] links to an object within the content structure or contains a freely defined link.

Default values

Using default values (default attribut) one can set standard values in hyperCMStags. When using component default values the path has to be set in the following way:
%comp%/publication name/location/component

Multiple components have to be divided by the separator ;.

Display name

Use the display name (label-attribute) inside a hyperCMS-tag to display a name instead of the tag ID.

Language setting

Use the language tag to define supported languages.
The following example defines 2 languages, the display name and the default language:

[hyperCMS:language name='langcode' list='DE|EN' label='Deutsch|Englisch' default='EN']

Now a session variable "langcode" exists that holds the value "DE" or "EN". Please note that you will need to define a condition for the language content block depending on the selected application language of the:

```
<?php if ($_SESSION['langcode']=="DE") { ?>
<h1>[hyperCMS:textu id='title' label='Titel' language='DE' height='25']</h1>
<p>[hyperCMS:textf id='content' label='Inhalt' language='DE']</h1>
<?php } ?>
```

```
<?php if ($_SESSION['langcode']=="EN") { ?>
<h1>[hyperCMS:textu id='title' label='Title' language='EN' height='25']</h1>
<p>[hyperCMS:textf id='content' label='Content' language='EN']</h1>
<?php } ?>
```

If you use the language-attribute inside a hyperCMS-tag, the value (language='DE') must be inside the given scope of the language-tag (list='DE|EN').

Template variables

With hyperCMS you also have the possibility to work with template variables. These variables take an exceptional position because they do not have to be connected to (PHP)-script. In fact, they are placeholders for the value of a variable and can freely be used in templates. Please note that variables are always written in small letters.

%container%	for the name of the content container of an object
%container_id%	for the ID of the content container of an object
%objecthash%	for the hash of an object
%template%	for the template file name of an object used
%publication%	for the publication where the object is located
%url_location%	for the absolute path (URL) of the location where the object is located
%abs_location%	for the absolute path in the filesystem of the location where the object is located
%object%	for the name of the object
%date%	describes the current date (format: JJJJ-MM-TT)
%view%	describes the display mode:
	publish … published
	cmsview … Editing view in EasyEdit mode
	preview … Preview
	formedit … Editing mode in form view
	formlock … form view with editing locked
	formmeta … Metadata in form view
	template … Template preview

To integrate media files we make use of a path variable. When publishing a page or component, the path variable is replaced by the URL (address) of the target of the publication:

%tplmedia%	for the absolute path (URL) of the template media repository
%url_media%	for the absolute path (URL) of the content media repository (Alternative **%media%** can be used)
%abs_media%	for the absolute path in the filesystem of the content media repository
%url_page%	for the absolute path (URL) of the page root folder
%abs_page%	for the absolute path in the filesystem of the page root folder
%url_comp%	for the absolute path (URL) of the component root folder
%abs_comp%	for the absolute path in the filesystem of the component root folder
%url_rep%	for the absolute path (URL) of the external repository root folder
%abs_rep%	for the absolute path in the filesystem of the external repository root folder
%url_ hypercms%	for the absolute path (URL) of the hyperCMS root folder
%abs_hypercms%	for the absolute path in the filesystem of the hyperCMS root folder

Please bare in mind to add a slash "/" to the end of the path variable, if you would like to complement the variable with a continuative path.

Definition of the date format when the format-attribute is used in the textd-Tag:

%a	'am' or 'pm'
%A	'AM' or 'PM'
%d	day of the month, 2 digits with leading zeroes (01 to 31)
%D	day of the week, textual, short, eg "Fri"
%F	month, textual, long; eg "January"
%h	hour, 12-hour format (01 to 12)
%H	hour, 24-hour format (00 to 23)
%g	hour, 12-hour format without leading zeros (1 to 12)
%G	hour, 24-hour format without leading zeros (0 to 23)
%i	minutes (00 to 59)
%j	day of the month without leading zeros (1 to 31)
%l	day of the week, textual, long, eg "Friday"
%L	1 if leap year, otherwise - 0
%m	month (01 to 12)
%n	month without leading zeros (1 to 12)
%M	month, textual, short, eg "Jan"
%s	seconds (00 to 59)
%t	number of days in the month (28 to 31)
%w	day of the week, numeric (0, Sunday to 6, Saturday)
%Y	year, 4 digits, eg 2007
%y	year, 2 digits, eg "07"
%z	day of the year (1 to 366)

Here are some examples of hyperCMS tags in HTML-code:

Editable meta information in a page template as well as the integration of a cascading style sheet (CSS) which is located in the template media database:

```
<title>[hyperCMS:pagetitle infotype='meta']</title>
<meta http-equiv="Content-Type" content="[hyperCMS:pagecontenttype
infotype='meta']">
<meta http-equiv="content-language" content="[hyperCMS:pagelanguage
infotype='meta']">
<meta name="Author" content="[hyperCMS:pageauthor infotype='meta'
default='Max Mustermann']">
<meta name="description" content="[hyperCMS:pagedescription
infotype='meta']">
<meta name="keywords" content="[hyperCMS:pagekeywords
infotype='meta']">
<link rel="stylesheet" hypercms_href="%tplmedia%/main.css"
type="text/css">
```

The last sentence in the example above shows that external files (usually located in the template media database) which are fixed to a template (background images, decoration images, CSS etc.) have to use the path %tplmedia%. Thus, it is guaranteed that the external CSS-file "main.css" is used in the page. The file "main.css" has to be saved in the media database for templates.

To prevent EasyEdit from link tracking the link has to be protected. This mainly has to be done when using external style sheets (CSS files). The link must not be changed by EasyEdit to correctly integrate the external file. The outcome would be a missing formatting because of the missing style sheet. Links (hyper references) are protected by changing "href" to "hypercms_href" as seen in the example above.

Integration of various image parameters:

```
<img border="0" src="[hyperCMS:mediafile id='logo' type='image']"
width="[hyperCMS:mediawidth id='logo']"
height="[hyperCMS:mediaheight id='logo']"
alttext="[hyperCMS:mediaalttext id='logo']" align="[hyperCMS:mediaalign
id='logo']">
```

The attribute „type" of the media file tag ensures that only image files can be included.

Defining the editable parameters of hyperlinks of an article:

```
<a href="[hyperCMS:artlinkhref id='partner']"
target="[hyperCMS:artlinktarget1 id='partner']">[hyperCMS:artlinktext
id='partner']</a>
```
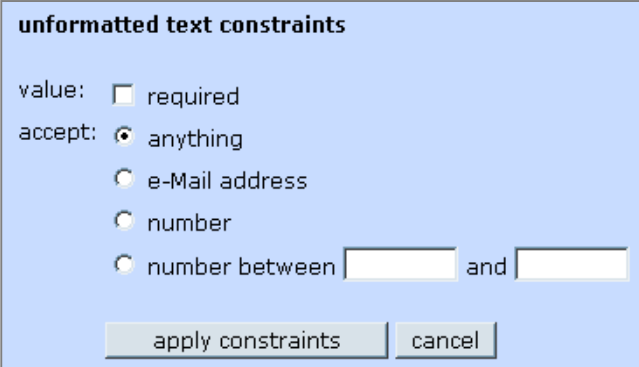
Editable unformatted text:

```
<font face="Verdana, Arial, Helvetica, sans-serif" size="1"
color="#000000">[hyperCMS:textf id='langtext']</font>
```

Unformatted texts can also be subject to conditions, e.g.:

- Value must be entered
- Value must be an email address
- Only numeric values are allowed
- Numeric values between limits

The TagWizard provides support to that effect:

Template Script:
It is possible to directly integrate applications with PHP, which are executed within the publication (published page). The whole range of functions of PHP as well as the functions of the hyperCMS libraries can be used.
With the help of template scripts it is possible to integrate applications that are executed during the publishing process. Thus, the output of the executed program code is archived within the publication.
In the manner described you can integrate dynamics into you templates, e.g. creating a directory of all pages, creating a further document in the background, and a lot more. When using scripts you are not limited.

The advantage of template scripts compared to the "normal" case, where the application is executed each time the page or component is accessed, is the performance. A template script is only executed during the publishing process and does not require any computing time when accessing the page. For details on script programming please refer to the hyperCMS Programmers Guide.

```
<font face="Verdana, Arial, Helvetica, sans-serif" size="2">
[hyperCMS:scriptbegin

// Loading content from text file
$external_data = loadfile („E:/Information/", „Imprint.txt");

// Verifying whether file has been loaded successfully
// and output of content
if ($external_data) echo „Imprint: ".$external_data;
else echo "The imprint is under construction!";

scriptend]
</font>
```

At this point we also would like to mention that there are more characteristics of tags respectively the corresponding content. Content can also be hidden.

The hyperCMS TagWizard offers 3 options:
- Do not hide content
- Hide content when publishing (onPublish='hidden')
- Hide content when editing (onEdit='hidden')

If the content generally is not hidden, no further attribute in the tag will be necessary. If the content should only be displayed in a certain area of the page and should be hidden in another area, an additional attribute will be necessary. The following example shows a PHP-application that is integrated in a template and needs adjustment inputs from the user:

```
<font face="Verdana, Arial, Helvetica, sans-serif" size="1">
[hyperCMS:textl id='Productcategory' list='Hardware|Software|Service'
onPublish='hidden']</font>

<font face="Verdana, Arial, Helvetica, sans-serif" size="2">
<?php

// Determining the value of the variable which the editor
// defined by using a text option
$Productcat = "[hyperCMS:textl id='Productcategory'
list='Hardware|Software|Service' onEdit='hidden']";

// Querying and displaying the articles from a MySQL database
$user = "username";
$password  = "password";
$database = "database";

mysql_connect ("localhost", $user, $password);

@mysql_select_db ($database) or die ("Unable to select database");

$result = mysql_query ("SELECT Name, Description, Price FROM
Products WHERE Category=$Productcat);

$num_of_rows = mysql_num_rows ($result);

if ($num_of_rows >= 1)
{
  echo "<table>";

  for ($count=1; $row=mysql_fetch_row ($result); ++$count)
  {
    echo "<tr><td>$count</td>";
    echo "<td>$row[0]</td>";
    echo "<td>$row[1]</td>";
    echo "<td>$row[2]</td></tr>";
  }

  echo "</table>";
}
else echo "Cannot find any products!";

mysql_close ();

?>
</font>
```

Because of the numerous possibilities in using hyperCMS tags in connection with HTML and JavaScript, we offer special trainings for template designers.

Furthermore it is possible to integrate PHP-code into the templates and access to the function library of hyperCMS is also available. Thus, you can also integrate very complex server-based applications into templates, such as e.g. database queries, data-extraction of xml-files, dynamic generation of sitemaps etc. Please refer to our Programmers Guide in this matter.

In the TagWizard you also find the „Help"-icon (?). Left click on the icon and you will get a short explanation of the hyperCMS tags of the templates. If you move the mouse over an icon you get a short description of the meaning of the selected symbol.

The hyperCMS TagWizard makes it possible to integrate hyperCMS tags without programming skills. With the help of this tool you can convert HTML-code into a template:

When editing a page template the hyperCMS TagWizard shows three text fields, respectively selection fields. The first field is used to define an article. Use this field to name your article (part of the ID). Each article is always associated with one or multiple page elements. For example name your article "news" and enter a text-tag with the content-ID "introduction". Thus, it is part of the article "news" carrying the prefix "art". Concurrently, you can see that the ID of the tag is composed of the article-ID and element-ID, e.g. id='news:introduction'. You can see this affiliation also by looking at the XML-content.

The ID (identification) must always be assigned to content-elements, whereas the definition of an article is not mandatory. The element features a key for identification, e.g. [hyperCMS:textf id='longtext'].
The third selection field enables the definition of meta-information. All of the selection fields insofar affect the hyperCMS tags that they assign further attributes to them.

The toolbar with its icons situated below offers the possibility to insert the hyperCMS tags into the template at the current cursor-position or paste it over the selected area of the code (text). Therewith it is possible to create templates in a simple and easy way without programming. These templates can then be used for your pages and components.

When creating component templates, the hyperCMS TagWizard offers less functions because some information is not necessary, e.g. the page title.
Page components are integrated into page- or component templates via the hyperCMS tag [hyperCMS:tplinclude file='Path']. Therewith it is possible to manage recurrent design elements independently from the template.

You can also use the tag [hyperCMS:fileinclude file='Path'] for the fixed integration of components. The indicated component will always be integrated in this location and cannot be influenced by the editor. E.g. this can be useful when integrating a menu-navigation.

Explanation of icons of the hyperCMS TagWizard:

Symbol	Explanation
	Save the template
	Save the template with preview
	Insert the hyperCMS tag for the page title
	Insert the hyperCMS tag for meta information. Please note, that afore you have to select meta information from the menu.
	Insert the hyperCMS tag for the language setting of content. You have to define a session variable name and possible session values.
	Insert the hyperCMS tag for unformatted text.
	Insert the hyperCMS tag for formatted text (WYSIWYG-editor).
	Insert the hyperCMS tag for text options. With the selection of this symbol you will be asked to enter the text options. Please use the semicolon ';' to separate the inputs from each other.
	Insert the hyperCMS tag for a selection via checkbox.
	Insert the hyperCMS tag for unformatted comments
	Insert the hyperCMS tag for formatted comments
	Insert the hyperCMS tag for media files (e.g. image file).
	Insert the hyperCMS tag for image alignment.
	Insert the hyperCMS tag for image width.
	Insert the hyperCMS tag for image height.
	Insert the hyperCMS tag for text shown on mouse-over of an image.
	Insert the hyperCMS tag for a hyperlink.
	Insert the hyperCMS tag for the link target (important when using frames).
	Insert the hyperCMS tag for the link text.
	Insert the hyperCMS tag for a single component.
	Insert the hyperCMS tag for multiple components.
	Insert PHP-code for session handling. After the selection of this symbol you will be asked to enter the variables and the corresponding values, respectively the variables that should be registered during the session.
	Insert the hyperCMS tag for the database connection.
	Insert the hyperCMS tag for the integration of a workflow for all pages or components based on the template.
	Insert the hyperCMS tag for template scripts.
	Insert the hyperCMS tag for the current date at the time of publication (JJJJ-MM-TT).
	Insert the hyperCMS tag for the path of a template media file.
	Insert the hyperCMS tag for the integration of a template include. You are asked to enter the path of the template include.
	Insert the hyperCMS tag for dynamic integration of an external file or component. You are asked to enter the path of the external file. In PHP as an application internet addresses (URLs) as well as absolute or relative paths are allowed in the file system. In JSP the file must be located in the web application root. By using slash "/" at the beginning the root is defined. In ASP only relative paths are allowed. Unfortunately, there is no way to reference on the document root of the website.

In the picture of the hyperCMS TagWizard below you can see 3 tabs. The first tab from the left shows the template, the second tab enables access to all available versions of the template and the third provides you with general information on the template.

Template	Version	Info

Template: Standard
Category: page template
Last updated: 2013-04-05 11:40
File size: 11635 bytes

Let's have a look on the versions of a template. We get a chronological list of all versions sorted by date and time beginning with the oldest version:

Template	Version	Info				
Version date		**Page template**		**Compare**	**Current**	**Delete**
2013-01-16 18:55:40		**Standard** (Source Code)		☐	○	☐
2013-01-16 18:57:58		**Standard** (Source Code)		☐	○	☐
2013-01-16 19:07:36		**Standard** (Source Code)		☐	○	☐
2013-01-16 19:08:51		**Standard** (Source Code)		☐	○	☐
2013-01-16 19:13:57		**Standard** (Source Code)		☐	○	☐
2013-04-05 11:40:03		**Standard** (Source Code)		☐	○	☐
Current version		**Standard** (Source Code)		☐	●	☐

▶ Submit changes to versions :　　　➜
▶ Compare selected versions :　　　➜

To have a look on the design of a version, just click on the template name. By doing so a new browser window will open and provide you with a preview that shows the editable elements by using placeholders.

Two versions can be compared. The difference between the old and the new version will be shown as red crossed-out (deleted) and green text (new).

It is also possible to delete template versions by activating the corresponding checkbox and clicking "OK". Thus, the selected templates are irretrievable removed from the system.

Furthermore it is possible to update an old version so that it is the current version. Simply click on the radio button for the version you want to use as your current version and click "OK". All pages and components which use this template will now access this (now current) version of the template. Please note that the new design of online pages and components will only be affected after you publish them again.

3.3 Deleting Templates

To delete a template please click on the third icon from the left in the toolbar. After that you can select a template. Please be aware that you really do not need the template anymore because it will be irretrievable removed from the system. After clicking on the icon a selection field is opened. Here you can select the template you would like to delete. This window also provides a preview of the template. If you are sure to delete the template simply click on the "OK"-button. Please note, that also all other versions of the template will be removed.

3.4 Editing Templates

To edit existing templates please click on the forth symbol on the left in the toolbar and select the requested template. The code of the template is then displayed in the TagWizdard editor and can now be edited. All changes affect the pages or components that use this template. Thus, it is possible to e.g. add additional text fields or images to the template. Existing pages are now supplemented by adding additional input options.

The same functions are also available for component templates. They are treated the same way as described above.

An important point when it comes to creating templates is the integration of media (background image, decoration images, flash movies etc.) and also external JavaScript or CSS (Cascading Style Sheets) files. Please note that these files also have to be uploaded to the system. This is done via the media database for templates which saves all external files of the templates.

3.5 Meta data mapping

Meta data mapping is only available for meta data templates.
Meta data mapping enables the mapping of meta data of files (mostly images used IPTC- and XMP-format) with the defined meta data schema in the system.

When uploading an image holding meta data, the meta data will be extracted and saved according to the mapping definition. Then the meta data is available and can be editing in the system.

A standard mapping exists already for the most used IPTC and XMP tags of media files and can be easily edited.

Meta data mapping

```
// Mapping definition: 'Meta data tag' => 'hyperCMS tag-ID'

// IPTC tags
'iptc:title' => 'Title'
'iptc:keywords' => 'Keywords'
'iptc:description' => 'Description'
'iptc:photographer' => 'Creator'
'iptc:source' => 'Copyright'
'iptc:urgency' => ''
'iptc:category' => ''
'iptc:supp_categories' => ''
'iptc:spec_instr' => ''
'iptc:creation_date' => ''
'iptc:credit_byline_title' => ''
'iptc:city' => ''
'iptc:state' => ''
'iptc:country' => ''
'iptc:otr' => ''
'iptc:headline' => ''
'iptc:source' => ''
'iptc:photo_number' => ''
```

4 Template Media Management

Template Media are a collection of different files that are used by templates. This disjunction of multimedia files that can be managed by editors prevents editors from manipulating constituents of the website's design. Only designers, who have the permission to embed media into templates, are allowed to manage template media.
Media files can be all kinds of files like images, CSS-files, video, audio, etc.
The media management stores all media files centrally und allows categorization by individual needs. It also features an advanced search function using information like category, file type and search term.

After choosing the template media in the navigator the following icons appear in the toolbar:

To see further details about the single functions just move the mouse cursor over the icons.

The following functions are available:

- Create media category
- Delete media category
- Rename media category
- Upload media file
- Delete media file
- Browse and search media files

4.1 Managing Media Categories

Using the first three icons you can create, delete and rename categories. By clicking on one of the icons additional fields are displayed on the desktop, which call for input.
After creating a category you can store media files inside it.

4.2 Uploading Files

To upload files click on the fourth icon from the left. Now two fields appear, one for selecting a category and one for browsing your local file system.
Select a target category for the file, click on the „browse"-Button to access your file system, then select the file and upload it by clicking „OK".

4.3 Removing Files

To permanently remove files from the system, click the „delete media file"-icon in the toolbar. In the following menu you can search the entire media repository by choosing a media category, file format and/or entering a filename in the search field.
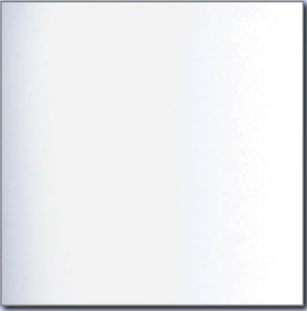If you want to display all available media files choose „all categories" in the media category menu, „all formats" in the file formats menu, leave the search field empty and click the submit button (green arrow).
As a result the list of files matching the search criteria is displayed. By clicking a file in the result list you can preview it. Images, Shockwave- and Flash-Files are displayed directly on the workspace as well as additional file information, such as file size (in bytes) and media size (height and width in pixels). Files with other formats (e.g. doc, xls, pdf,...) can be opened with the appropriate application (e.g. Word, Excel, Acrobat,...) by clicking the link below the file information.

Please note: Deleting a media file still used in a page or component can lead to display errors.

4.4 Searching files

By clicking the icon on the very right in the toolbar you open the search menu. It offers the same browsing and search functionality as described in „Delete Files" above. You can search for files by entering search criteria like media category, format or filename and browse the entire repository.

5 Legal reference / flag

5.1 Questions and suggestions

For advanced questions and suggestions, please contact the support. We are available for every question regarding our reseller- and partner-program. You can apply for an access to our enhanced Online-Demo of the hyper Content Management Servers via our support.

hyperCMS Support:
support@hypercms.com
www.hypercms.com

5.2 Imprint

Responsible for the content:

hyperCMS
Content Management Solutions GmbH
Rembrandtstr. 35/6
A-1020 Vienna – Austria

office@hypercms.com
http://www.hypercms.com

5.3 Legal information

The present product information is based on the version of the program, which was available at the time the document was composed.

The maker reserves the rights of modifications and corrections of the program. Errors and misapprehension excluded.

© 2014 by hyperCMS Content Management Solutions