hyper ➜ Content & Digital Asset Management Server
CMS

Version 10.2
Template Designers Guide

2026-01-05

Index

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, component, or asset, like text, images, links.
You can create new templates using either the TagWizard of the template management system or code generated by an HTML-editor of your choice. HTML code can be copied and pasted into the TagWizard. Using the TagWizard you can create new templates without programming skills.
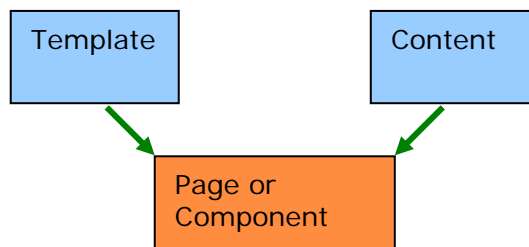
All templates are managed by the version control. When editing and saving a template a new version will be created automatically. This way you can undo any change in a document (rollback to previous versions), compare the code, or view a version of a template.

The hyper Content Management Server uses „pages" and „components" for building websites and therefore there are design templates for pages and components. 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 is commonly defined and used in 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 metadata schema for multimedia files and folders. Metadata templates therefore do not define the design but the input options for metadata and are limited compared to templates for pages or components. HTML code is ignored and has no use.
It is important that there is a predefined template named "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, in case there is no other template assigned to the parent folder.

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.
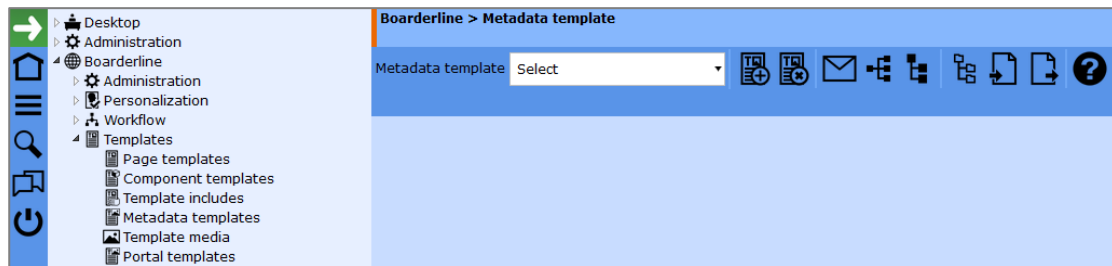

You can start the following actions, depending on the template category and if you have permission to access the features:

- Select and edit an existing template
- Create a new template
- Delete an existing template
- Setting up the license notification (only for metadata templates)
- Configuration of the metadata mappings (only for metadata templates)
- Configuration of the metadata/content hierarchy browsing (only for metadata templates)
- Display and edit the taxonomy
- Import of the taxonomy based on a CSV file
- Export of the taxonomy based on a CSV file

3 Page-, component- and metadata templates

3.1 Create 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.
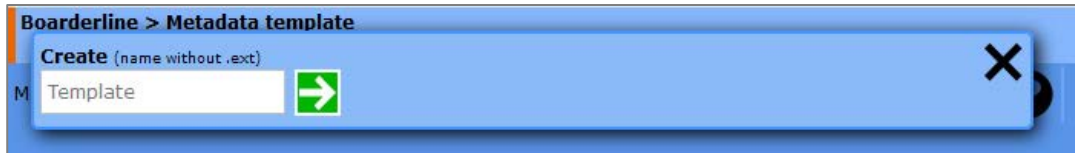
Enter the name (the name must be unique for each publication). Page templates always refer to a page whereas component templates are used to create components. Template Includes on the other hand are used to amend one or various page or component templates. The usage of template includes is similar to the one of file includes. By using 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 (assets).

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 installation package includes several examples of templates, see hypercms/install/templates/.

Metadata templates are very easy to create because they only define the fields. In each publication, the template "default" is automatically created. It includes the following definitions:

```
[hyperCMS:help id='Help' value='en:This is a help text;de:Dies ist ein
Hilfetext']
[hyperCMS:textu id='Title' label='en:Title;de:Titel' infotype='meta'
height='20']
[hyperCMS:textu id='Description' label='en:Description;de:Beschreibung'
infotype='meta' height='40']
[hyperCMS:textk id='Keywords' label='en:Keywords;d:Schlüsselwörter'
infotype='meta'  file='%taxonomy%/default/all/0/5']
[hyperCMS:textu id='Copyright' label='Copyright' infotype='meta'
height='20']
[hyperCMS:textu id='Creator' label='en:Creator;de:Ersteller' infotype='meta'
height='20']
[hyperCMS:textd id='License' infotype='meta' label='en:License valid
till;de:Lizenz gültig bis']
[hyperCMS:textl id='Quality' label='Quality' infotype='meta' height='20'
mediatype='image' list='|Print|Web']
[hyperCMS:textl id='YoutubeCategories' label='YouTube categories'
infotype='meta' height='25' mediatype='video' list='Autos &
Vehicles{2}|Comedy{23}|Education{27}|Entertainment{24}|Film &
Animation{1}|Gaming{20}|Howto & Style{26}|Music{10}|News &
Politics{25}|Nonprofits & Activism{29}|People & Blogs{22}|Pets &
Animals{15}|Science & Technology{28}|Sport{17}|Travel & Events{19}']
[hyperCMS:commentf id='Comment' label='en:Comment;de:Kommentar'
infotype='meta' height='100']
[hyperCMS:componentm id='Related' label='en:Related assets;de: Verwandte
Assets']
[hyperCMS:geolocation infotype='meta']
```

Note that each tag must have a unique technical "id" while the "label" is the user-visible identifier of the field. For multilingual field names, the language abbreviation can be placed in front of each text and the individual language specifications can be separated with a semicolon, e.g.:
label='en:Description;de:Description'

The languages or ISO 639-1 codes supported by the system are defined in the main configuration file "config.inc.php", see setting $mgmt_lang_shortcut

The meaning of the tags and their attributes can be found in the following chapter.

3.1.1 hyperCMS Tags

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…]

Help Tag set

Help/Information (Form view)	[hyperCMS:help id='…' value='…']
Width of editorfield in pixel (optional)	width='…'
Height of editorfield in pixel (optional)	height='…'
Access permission for certain user groups (optional)	groups='…'

Text Tag set

unformatted text	[hyperCMS:textu id='…']
formatted text	[hyperCMS:textf id='…']
Text option from text list	[hyperCMS:textl id='…' list='…' file='…']
Checkbox	[hyperCMS:textc id='…' value='…']
Date, Date format (optional)	[hyperCMS:textd id='…' format='%Y-%m-%d']
Keywords with optional mandatory or open list	[hyperCMS:textk id='…' list='…' file='…' onlylist='yes/no' display='taxonomy']
electronic signature	[hyperCMS:textS id='…']
Unformatted comment	[hyperCMS:commentu id='…']
Formatted comment	[hyperCMS:commentf id='…']
Display name (optional)	label='…'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
Read-only content	readonly='readonly'
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='…'
Prefix and suffix will be added to content if it is not empty	prefix='…' suffix='…'
Search and replace	replace='search=>replace|search=>replace'
valid language value (optional)	language='…'
Access permission for certain user groups (optional)	groups='…'
Media type [audio, binary, component, compressed, flash, image, text, video, watermark] (optional)	mediatype='…'
Media path type [file, url, uri, download, wrapper] (optional)	pathtype='…'
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 [ICC Profiles: hypercms/library/ICC_Profiles] (optional)	iccprofile='…'
Preview window (textu)	preview='url'

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'
Read-only content	readonly='readonly'
Media type [audio, binary, component, compressed, flash, image, text, video, watermark] (optional)	mediatype='…'
Meta-information type (optional)	infotype='meta'
Path type [file, url, uri, download, wrapper] (optional)	pathtype='…'
use thumbnail image	thumbnail='yes'
valid language value (optional)	language='…'
Access 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 [ICC Profiles: hypercms/library/ICC_Profiles] (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'
Read-only content	readonly='readonly'
Disable link in edit mode	disable='disable'
Meta-information type (optional)	infotype='meta'
valid language value (optional)	language='…'
Access 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'
Read-only content	readonly='readonly'
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='…'
Access permission for certain user groups (optional)	groups='…'
Media type [audio, binary, component, compressed, flash, image, text, video, watermark] (optional)	mediatype='…'

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='…'
Access permission for certain user groups (optional)	groups='…'
Read-only content	readonly='readonly'

Geolocation

Geolocation	[hyperCMS:geolocation infotype='meta']
Display name (optional)	label='…'
Content cannot be edited (optional)	onEdit='hidden'
hide content (optional)	onPublish='hidden'
Read-only content	readonly='readonly'
Access 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='…'
Access permission for certain user groups (optional)	groups='…'

Script Tag set

Template script (PHP/API)	[hyperCMS:scriptbegin … scriptend]
JavaScript (form-views)	[JavaScript: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

Stylesheet (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,   [hyperCMS:objectview name='…']
formedit, formmeta, formlock, template, publish)

Please note: Each ID of a text-, media- or component-tag set within a page, article or component must be unique. You can use the same ID for multiple elements of a tag set (e.g. unformatted text). Changes on one element also effect all 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 an asset. The tag [hyperCMS:linkhref id='…'] links to a page within the page structure or contains a freely defined link.

Default values
Using default values (default attribute) one can set standard values in hyperCMS tags. When using default values for components, the path has to be set in the following way: %comp%/publication-name/location-path/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.

3.1.2 Language setting

Use the language tag to define supported languages for editors.
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_DE' label='Titel' language='DE'
height='25']</h1>
<p>[hyperCMS:textf id='content_DE' label='Inhalt' language='DE']</h1>
<?php } ?>

<?php if ($_SESSION['langcode']=="EN") { ?>
<h1>[hyperCMS:textu id='title_EN' label='Title' language='EN'
height='25']</h1>
<p>[hyperCMS:textf id='content_EN' 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').

3.1.3 Template variables

You also have the possibility to work with template variables. These variables are placeholders for ceratin values and can 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
%container_content% for the XML document of the content container of an object
%objecthash% for the hash of an object
%object_id% for the ID 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
%locationname% for the readable name of the location where the object is located
%object% for the file name of the object
%objectname% for the readable name of the object
%date% describes the current date (format: JJJJ-MM-TT)
%user% for the logged in user name in the current session
%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 (assets) we can make use of a path variable. When publishing a page or component, the path variable is for instance replaced by the URL (address) of the target of the publication:

%url_tplmedia% for the absolute path (URL) of the template media repository (Alternative **%tplmedia%** can be used)
%abs_tplmedia% for the absolute path in the filesystem 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

The document root of pages and components of the publication target can be provided as well:

%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 keep 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)

3.1.4 Examples of the use of tags hyperCMS

3.1.4.1   Metadata in pages

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

```
<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 line in the example above shows that the external CSS file (located in template media) have to use the path %tplmedia%. Thus, it is guaranteed that the external CSS-file "main.css" is integrated into 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 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.

3.1.4.2   Image

Integration of various image parameters:

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

The attribute „mediatype" of the media file tag ensures that only image files can be included.
The attribute "pathtype" can be used to define the type of reference to the media file:

- file … absolute path in file system
- url … URL
- uri … URI (URL without the protocol and domain/host name)
- download … download link forces media file download (provides access statistics)
- wrapper … wrapper link provides media file in browser (provides access statistics)

3.1.4.3 Link

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>
```

3.1.4.4 Text
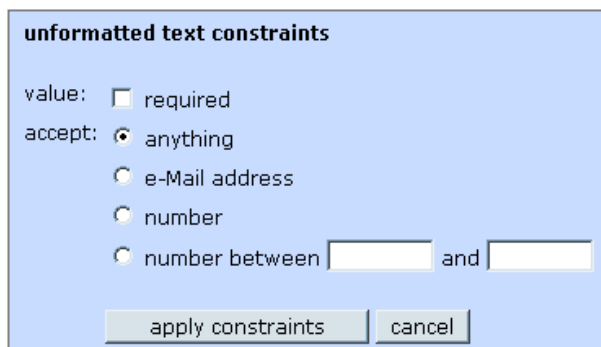
Editable unformatted text:

```
<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:

3.1.4.5   Keywords

Mandatory keywords field using the taxonomy as source for the keywords:

```
[hyperCMS:textk id='Keywords' infotype='meta' height='75'
file='%taxonomy%/default/en/0/4' onlylist=1 constraint='R']
```

This tag defines a metadata field that only allows keywords from the taxonomy as source. The keywords input field automatically presents the possible keywords defined in the taxonomy while typing.
Optionally, the taxonomy structure can also be displayed using the attribute display='taxonomy'. Here the keywords are selected by checkboxes.
The following syntax for the attribute „file" can be used that will also work for list-fields.

Usage of the taxonomy in the file-attribute:

```
Domain/Publication/Language-code/Taxonomy-ID/Taxonomy-sublevels
```

Domain … %taxonomy%
Publication …. Name of the publication in order to select a specific taxonomy or „default" for the standard taxonomy
Language-code …. 2-digit language code or „all" for all languages
Taxonomy-ID …. Line number of the taxonomy keyword or 0 for all keywords
Taxonomy-sublevels …. Number of keyword sublevels of the selected taxonomy keyword

Usage of a folder path in the file-attribute:
Returns all subfolder names as selectable values.

```
%comp%/Publication/Folder/
```

%comp% …. Page %page% or Component/Asset %comp%
Publication … Name of the Publication
Folder … Name of the folder or location path that defines the root

Use a path to a file or service in the file-attribute:
All values stored in the file or provided by a service (REST) can be used.
Note that the values must be sperated by , or ;.

Bsp: /home/values/names.txt
Bsp: https://www.domain.com/services/names.php?q=company

3.1.4.6   List fields

List field using a hardcoded list of values with 1 as default value:

```
[hyperCMS:textl id='Numbers' infotype='meta'  list='1|2|3' default='1']
```

Also list fields can use the same file-attribute as keywords, so values from teh taxonomy, a folder structure or from an external source can be integrated.

3.1.4.7   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 API 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 rendered within the published object/file.
In the manner described you can integrate dynamics into you templates, e.g. creating a directory of all pages, creating a document in the background, and a lot more.

The advantage of template scripts compared to the dynamic execution of code in PHP, ASP, or JSP files, 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 for publishing (onPublish='hidden')
- Hide content for editing (onEdit='hidden')

If the content 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 requires input from a user/visitor to start a query:

```
[hyperCMS:textl id='Productcategory' list='Hardware|Software|Service'
onPublish='hidden']

<div id="products">
<?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!";
                                                                          m
ysql_close ();
?>
</div>
```

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

With templates you can also integrate any server-based applications into templates, such as database querie, dynamic generation of sitemaps, and so on. Please see the Programmers Guide for more details about how to use the hyperCMS API.

3.1.4.8   JavaScript for forms

JavaScript or JQuery can be used on the client side to control form elements. The code can be integrated directly into the template and is executed when a form view is called.
For instance, this can be used to control fields by using the values of other fields as a trigger.

The system provides predefined JavaScript functions for field control. The following syntax must be used for the ID in order to control the fields:
tagname_tagID ... for standard tags
tagname_articleID_elementID .... for tags of articles

Function checkFieldValue (id, value)
Checks if a field has a certain value. All form elements are supported. The function returns true if matched or false otherwise. Also a substring of the value will result to true.

Function lockField (id)
Locks a field for editing.

Function unlockField (id)
Unlocks a field for editing.

Function hideField (id)
Locks and hide a field.

Function showField (id)
Unlocks and displays the field.

Example of field controls:
If the checkbox with the ID "Lock" is activated by a user of the group "Administrator", then all other fields are locked for editing.

```
[JavaScript:scriptbegin

function changelock ()
{
  if (checkFieldValue ('textc_Lock', 'Yes'))
  {
    lockField ('textu_Title');
    lockField ('textu_Description');
    lockField ('textk_Keywords');
    lockField ('textd_License-Begin');
    lockField ('textd_License-End');
    lockField ('geolocation');
    lockField ('componentm_Related');
  }
  else
  {
    unlockField ('textu_Title');
    unlockField ('textu_Description');
    unlockField ('textk_Keywords');
    unlockField ('textd_License-Begin');
    unlockField ('textd_License-End');
    unlockField ('geolocation');
    unlockField ('componentm_Related');
  }
}

// on load
changelock ();

// on change
$('#textc_Lock').change(function() {
  changelock ();
});

scriptend]

[hyperCMS:textc id='Lock' value='Yes' infotype='meta'
groups='Administrator']
[hyperCMS:textu id='Title' infotype='meta' constraint='R' height='25']
[hyperCMS:textu id='Description' infotype='meta' constraint='R' height='75']
[hyperCMS:textk id='Keywords' infotype='meta' height='75'
file='%taxonomy%/default/en/0/5']
[hyperCMS:geolocation infotype='meta']
 [hyperCMS:textd id='License-Begin' infotype='meta' label='License valid
from']
[hyperCMS:textd id='License-End' infotype='meta' label='License valid till']
[hyperCMS:componentm id='Related' infotype='meta']
```

3.1.5 hyperCMS TagWizard

The hyperCMS TagWizard makes it possible to integrate hyperCMS tags into yout emplaztes without programming skills. With the help of this tool you can convert any HTML code into a template.
In the TagWizard you will find a ?-icon (Help). Left click on the icon and you will get a short explanation of the hyperCMS tags that can be used. If you move the mouse over an icon you get a short description of the meaning of the selected symbol.

When editing a page template, the hyperCMS TagWizard shows two text fields respectively a checkbox for meta information at the top. 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 checkbox for metainformation declares a tag to be a meta information. So an editorcan edit the metainformation apart from the standard content displayed in a page.

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 (e.g. 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 a component or a metadata template, the hyperCMS TagWizard offers less actions.
Page components can be integrated into page- or even other 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:filepinclude file='Path'] for the 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 navigation element.

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 geo location of the object
	Insert the hyperCMS tag for a help-text or information (only visible in the form view)
	Insert the hyperCMS tag for the language setting of content. You have to define a session variable name and possible session values.
TXT	Insert the hyperCMS tag for unformatted text.
TXT	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 the date picker
	Insert the hyperCMS tag for keywords
	Insert the hyperCMS tag for unformatted comments
	Insert the hyperCMS tag for formatted comments
	Insert the hyperCMS tag for the electronic signature
	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.
31	Insert the hyperCMS tag for the current date at the time of publication (JJJJ-MM-TT).
PATH	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	Information
Template	Detail	
Category	page template	
Last updated	2017-01-19 16:23	
File size	3977 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	Information	Compare	Current	Delete
Version date		**Page template**			
2015-09-16 16:34:59		Detail (Source Code)	☐	○	☐
2016-08-03 12:47:45		Detail (Source Code)	☐	○	☐
2016-08-03 12:47:57		Detail (Source Code)	☐	○	☐
2016-08-03 12:49:30		Detail (Source Code)	☐	○	☐
2016-08-03 12:49:49		Detail (Source Code)	☐	○	☐
2016-08-03 12:50:21		Detail (Source Code)	☐	○	☐
2016-08-04 12:51:24		Detail (Source Code)	☐	○	☐
2016-08-04 14:42:26		Detail (Source Code)	☐	○	☐
2016-11-09 20:31:16		Detail (Source Code)	☐	○	☐
2017-01-19 16:11:32		Detail (Source Code)	☐	○	☐
Current version		Detail (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.
It is possible to update an old version so that it becomes 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.

3.2 Delete a template

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.

3.3 Edit a template

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 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.

3.4 License notification

In order to inform users about the expiration of a copyright license or other events, the license notification can be used. Several automatic notifications based on the id of a date tag, the used date format and the desired frequency of the notification via e-mail can be defined.

3.5 Meta data mapping

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

When uploading an image holding metadata, the meta data will be extracted and saved according to the mapping definition. Then the metadata is available and can be editing in the system.
A standard mapping exists already for the most used EXIF, ID3, IPTC and XMP tags of media files and can be easily edited. The XMP format is based on XML and can map multiple metadata schemas such as Dublin Core (DC) and Adobe Photoshop.

A special feature can be used for the Adobe Lightroom XMP tag lr:hierarchicalSubject. This tag includes both field names and their values. If you want to import the content pairs into the system, the value "automatic" must be used or retained. Please note that the metadata template has to contain the fields with the appropriate ID in order to see and also edit the information.

```
lr:hierarchicalSubject => automatic
```

After changes in the mapping definition and saving it, the media files can be re-indexed in order to save the new content from the files in the system.

4 Metadata/Content hierarchy

In order to enable users to browse objects by their metadata/contents a hierarchy can be defined. The hierarchy can hold up to 10 different Text-Ids on different hierarchy levels. Up to 5 different levels can be used. However, it is recommended to use not more than 3 levels for an improved browsing experience of the objects. You can define one or more hierarchy definitions. All of them will be presenting to users with page or asset access for browsing. Save the hierarchy definition so it can be used by all users.
If you want to support the label in multiple languages, please use the ISO 639-1 language code before the label, e.g.: `en:Title; de:Titel; fr:Titre`

The hierarchy will be presented to users with two different folder types/colours:
Yellow folder … Content category based on the Text-ID of the content
Blue folder … Content/Keyword

5 Taxonomy

A taxonomy or classification scheme is a uniform, hierarchical model, for the classification of objects.
The system provides a standard taxonomy in several languages. However, it can be modified for each publication in order to create an individual taxonomy.
The taxonomy definition files are located in data/include and have the following naming scheme: Publication.taxonomy.csv

It is CSV file that can be opened and edited using MS Excel or any text editor that supports the character encoding UTF-8.
When creating a new taxonomy, the structure of the content in the file is important. The top level of the terms uses the level number 1, each additional sub-level must be provided with level numbers for structuring.

You can create the file in MS Excel to define the taxonomy (please use the proper 2-digit language abbreviation according to the example file) or you can build it using the taxonomy editor of the system. The editor enables you to create new entries, set the level for each entry, and automatically translate the taxonomy keywords into other languages.

5.1 Taxonomy editor

Boarderline > Metadata template

Metadata template: Select

Taxonomy

Selected languages

☑ Arabic	☐ Bulgarian	☐ Chinese (Simplified)	☐ Danish	☐ Dutch
☑ English	☐ Finnish	☑ French	☑ German	☑ Hindi
☑ Hungarian	☑ Italian	☑ Japanese	☑ Korean	☑ Malay
☑ Norwegian	☑ Polish	☑ Romanian	☑ Russian	☑ Slovak
☑ Slovenian	☑ Spanish	☑ Swedish	☑ Turkish	☑ Urdu

Translate: Arabic > Bulgarian

Level	ID	Arabic	English	French	German	Hindi
1	1	النوع	Type	Type	Typ	प्रकार
2	2	خاتمة	Abstract	Résumé	Zusammenfassung	गूढ़
3	3	الخلفية	Background	Contexte	Hintergrund	पृष्ठभूमि
3	4	الرسم	Graphic	Graphic	Grafik	ग्राफिक
3	5	الخلفية	Wallpaper	Fond d'écran	Hintergrundbild	वॉलपेपर
2	6	المغامرة	Adventure	L'aventure	Abenteuer	साहस
2	7	الحيوان	Animal	Animal	Tier	पशु
3	8	المواضيع الحيوانات	Animal Themes	Thèmes des animaux	Tier Themen	पशु विषयवस्तुएं
2	9	الهندسة المعمارية	Architecture	Architecture	Architektur	वास् तुकला
3	10	المعمار والمباني	Architecture and Buildings	Architecture et Bâtiments	Architektur und Gebäude	वास् तुशिल् प और भवनों
3	11	مشيد	Built Structure	Structure intégrée	Stücklistenaufbau	निर्मित संरचना
2	12	فن	Art	Art	Kunst	कला
2	13	الخلفية	Background	Contexte	Hintergrund	पृष्ठभूमि
2	14	الأعمال	Business	Entreprise	Geschäft	व्यवसाय
2	15	الصراحة	Candid	Candid	Frei	स्वीकार करना
2	16	القطع	Cutout	Découpe	Ausschnitt	Cutout
2	17	نسخة الفضاء	Copy Space	Espace de copie	Kopie	अंतरिक्ष में कॉपी करें
2	18	مفهوم	Concept	Concept	Konzept	की अवधारणा
2	19	قريب	Close-up	Close-up	Nahaufnahme	के निकट
2	20	التعليم	Education	L'éducation	Bildung	शिक्षा

1 - 20

Text-ID Search restriction (use ; as delimiter): Title;Keywords

Save settings ➡
Apply Changes ➡
Delete Taxonomy ➡
Restore (Standard Taxonomy) ➡

The ID number is used as identification of the same keyword in different languages.
If you want to limit the analysis of the content for the taxonomy to certain texts or fields, you can use the text IDs defined in the templates and use semicolons as separators, see also an example of a tag in a template:

```
[hyperCMS:textk id='Keywords' label='Keywords' infotype='meta']
```

If changes in the taxonomy have been made, the system will read and re-index the content based on the new taxonomy definition only when you apply the changes. The corresponding system API command/function is createtaxonomy. It should be noted that the process may take longer if the taxonomy is changed and a large number of objects and contents are affected.
The existing taxonomy can be removed for the publication. In this case the relationship between the objects and taxonomy will be removed as well.
The default taxonomy can also be restored if necessary.

5.2 Import of a taxonomy

A taxonomy can be imported based on a CSV file.
The specifications or column definitions in CSV format must be taken into account.
The export of the existing taxonomy can serve as a template for this.
The first row contains the column definitions, such as "level" and a 2-digit language code, such as "en". All other rows contain the data.

Here is an example of the CSV data for a multilingual taxonomy:

```
level;de;en;es;fi;fr
1;Type;Type;Tipo;Tyyppi;Type
2;Summary;Abstract;Resumen;Tiivistelmä;Résumé
3;Background;Background;Antecedents;Tausta;Contexts
3;Graphic;Graphic;Gráfico;Graafinen;Graphic
```

The CSV file can be uploaded and imported manually. But it can also be imported automatically as a daily job.
You can refer to a file on a file server by specifying a path or to a web service (GET request), which provides you with CSV data in the response.
The "Job" checkbox must be activated so that the import is carried out monthly, weekly, or daily.
Optionally, the imported CSV file can also be removed from the file server to prevent the same data from being imported again and again.

5.3 Export of taxonomy

The existing definition of the taxonomy can be exported and downloaded as a CSV file.
The exported CSV file can be edited in MS Excel and can be imported into the system.

6 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

6.1 Managing media categories

Media categories are used to easily structure files needed for the design and functionality of websites, such as images, CSS-, JS-files.
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.
Within a category, media files can be stored. It should be mentioned that a media file name must be unique. The same name for different files is not allowed. This is to prevent various editors who have the authority of a file upload, e.g. put the same picture in the database several times. Redundancy is avoided.

If you have created a category for the first time (think of a structure that makes sense for the website, for example, in terms of topics, organizational units, etc.) you can save or upload media files within this category.

6.2 Upload files

To upload a file, click on the fourth icon. Two fields now appear, one for selecting a category and the second for selecting a file on your local file system. So select a category in which you want to record the file and then select a file by clicking on Browse. A window will open, giving you access to your local file system. Select the desired file from the local file system and confirm your selection. Now the window closes again and via "OK" the file is loaded onto the server.

6.3 Remove files

To permanently remove files from the system, click the „delete media file"-icon in the toolbar. Please note that deleting a template media file that is still used in a page or component can lead to display errors.
If you want to remove a file, click on the corresponding icon. After that, a search environment appears on the workstation that makes it easier for you to find files. You can now search for all files by selecting "All Categories" and "All Formats" and not entering a search term. After starting the search with "OK" you get a list of all files in the database.
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).
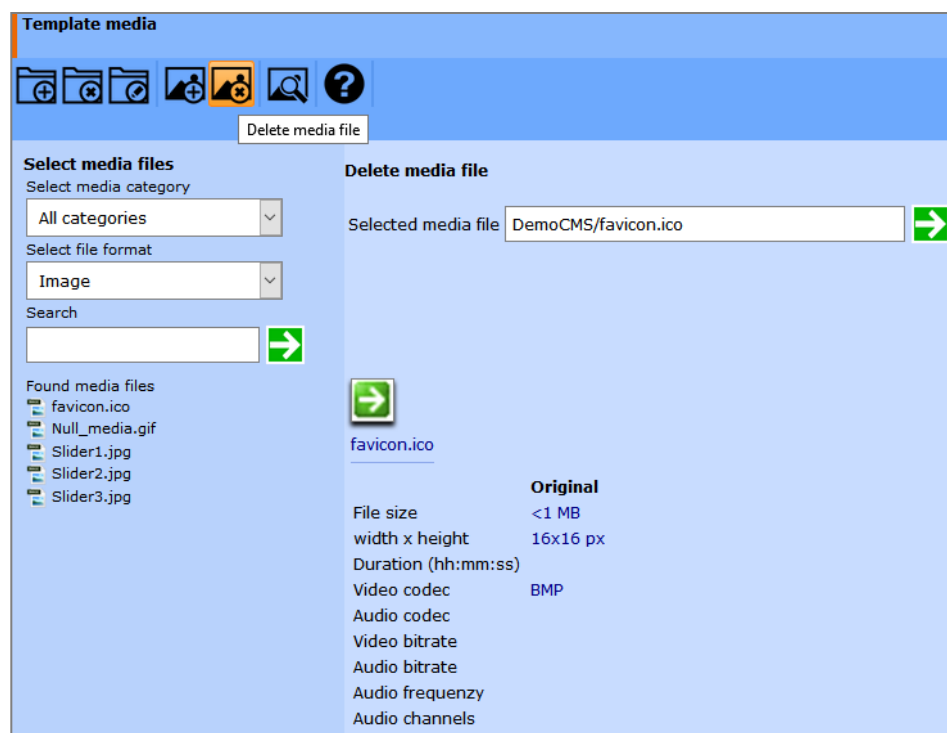
6.4 Search for files

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 the previous chapter. You can search for files by entering search criteria like media category, format or filename and browse the entire repository.
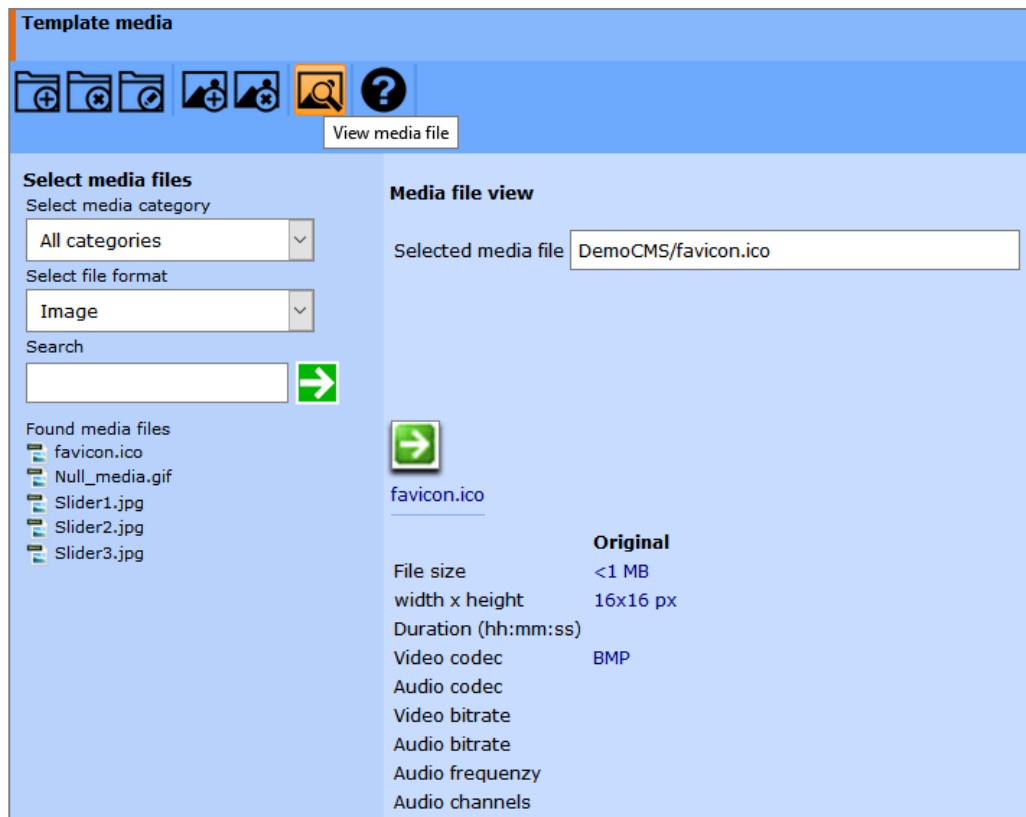
7 Portal templates

To activate the portal templates, the "Portal Access Link" must be activated for the publication, see also Administrator Guide.

Portal templates are used to define portals for the public access of visitors without prior login in the system. This way information portals with download areas for employees, partners or customers can be easily set up and managed.
The portal templates make it easy to create your own designs for the system interface. The design can be easily defined by the upload of a logo, an individual background image and an individual color scheme. The new portal design can be assigned to a publication or users. It should be noted that the restriction on download formats for the user will be effective. This may not always be desirable. Therefore, portal templates can also be used without an assigned user, so that the users who use the portal template retain all their permissions.

The portal requires a user account to be assigned to it for the purpose of authorization and the use of functions. This account must have the appropriate permissions, see Administrator Guide.
If the publication uses the taxonomy or metadata hierarchies for navigation, these navigation options can be activated or deactivated.
With the help of the access link of the portal, the portal and its contents can be made available to every visitor without registration. The access link can be sent in an e-mail or linked with your own domain.

Just like all other templates, portal templates can be created, edited and removed using the corresponding icons in the toolbar.
All settings can be made in the left pane after creating a new portal template.
After saving you will see the preview of the portal in the right area.

The portal templates allow the following design settings:
- transparent …. Transparent theme that uses the background image as the main design and color element
- Text and Icons… If you don't use a transparent theme, you can choose between light and dark text and icons, with the background color also changing accordingly to preserve contrast
- Main-Color … select a primary color for the selected non-transparent theme
- Hover-Color … Select a color of buttons (hover status) for the selected non-transparent theme, such as used in toolbars
- Logo (icon only) … Logo for the main navigation bar
- Logo (wider logo) … Logo for the Home screen
- Wallpaper … Background wallpaper for the home screen or a transparent design theme
- Customize toolbar … Position of the main navigation bar on the left side or on top

The following access settings can be defined for public access of the portal using the Portal link:
- User … assign a user to the portal for access permissions
- Navigate … Assign navigation options to the portal
- Download formats … Assign download formats to the portal
- Access link … The access link for the portal will be displayed after the template has been saved
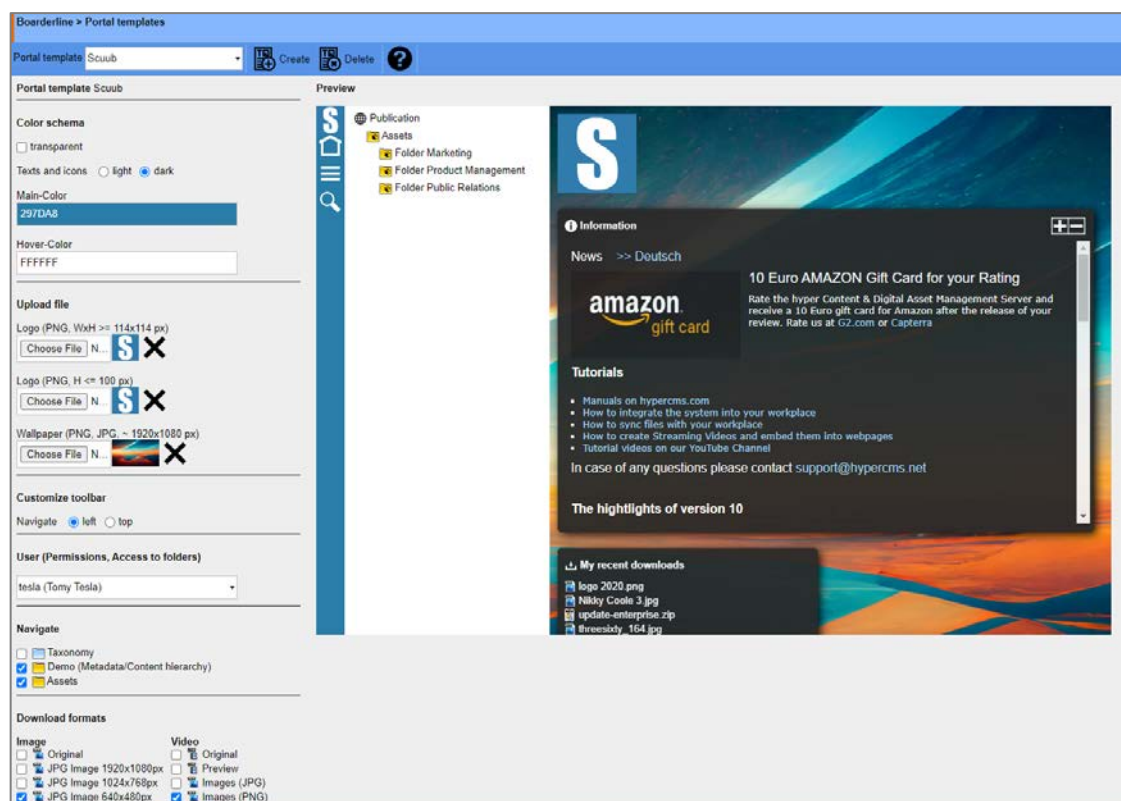
The defined portal design can be assigned to all publications and their users in the main configuration file of the system, see setting $mgmt_config['theme'] in the configuration file hypercms/config/config.inc.php.
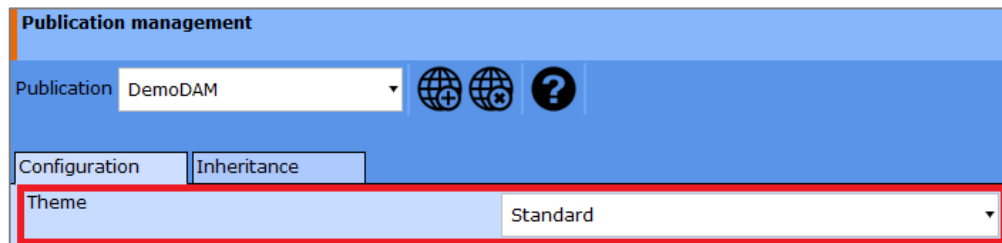The portal design can also be assigned to all users of a publication in the publication settings. This means that the user loses the right to choose his own design. However, the user retains all of his other rights and only uses the design. See setting "Theme" of the publication.

If no design has been defined in a publication, the user can select this himself in his personal settings. Here, too, the portal design is available to him.

8 Legal reference / flag

8.1 Questions and suggestions

For advanced questions and suggestions, please contact the support.

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

8.2 Imprint

Responsible for the content:

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

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

8.3 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.

© 2026 by hyperCMS Content Management Solutions