hyper ContentManagementServer

CMS

Version 5.8

Programers Guide

2015-12-22

Inhalt

1 Introduction

The following chapters deal with the function libraries of the hyper Content & Digital Asset Management Server and thus provide the documentation of the API (Application Programming Interface).
All libraries are located within the hyperCMS installation in the folder "function" and can be integrated and used in the respective scripts or templates. This can be used, for example, to create dynamic pages (applications) using the XML content repository.
If you run your application on a physically separated server, it is important that the function libraries are also available on the publication server. In this case you need to have access to the corresponding files on the publication server as well.

2 hyperCMS XML-Content-Repository

The XML content repository includes all XML Content Container and thus provides all content in native XML. The structure (schema) within an XML content container is dynamically generated based on the template used and has the following appearance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<container>
  <hyperCMS>
    <contentcontainer>0000023.xml</contentcontainer>
    <contentxmlschema>object/page</contentxmlschema>
    <contentorigin>%page%/Publication/testpage.php</contentorigin>
    <contentobjects>%page%/Publication/testpage.php|%page%/ Publication/linkedcopy_of_testpage.php
|</contentobjects>
    <contentuser>demouser</contentuser>
    <contentdate>2002-11-26</contentdate>
    <contentpublished>2002-11-26</contentpublished>
   <contentstatus>active</contentstatus>
  </hyperCMS>
  <head>
    <pagetitle>test</pagetitle>
    <pageauthor>Mr. Content</pageauthor>
    <pagedescription>just a small demonstration</pagedescription>
    <pagekeywords>demo of XML</pagekeywords>
    <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
    <pagelanguage>de</pagelanguage>
    <pagerevisit></pagerevisit>
  </head>
  <textcollection>
    <text>
      <text_id>headline</text_id>
      <textuser>demouser</textuser>
      <textcontent>fgfdgfdg</textcontent>
    </text>
    <text>
      <text_id>summary</text_id>
      <textuser>demouser</textuser>
      <textcontent><![CDATA[This is a
      <STRONG><EM>summary</EM></STRONG>]]></textcontent>
    </text>
  </textcollection>
  <mediacollection>
    <media>
      <media_id>logo</media_id>
      <mediauser>otheruser</mediauser>
      <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
      <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
      <mediaalttext>demoimage</mediaalttext>
      <mediaalign></mediaalign>
      <mediawidth>200</mediawidth>
      <mediaheight>100</mediaheight>
    </media>
  </mediacollection>
  <linkcollection>
    <link>
      <link_id>verweis</link_id>
      <linkuser>demouser</linkuser>
      <linkhref>http://localhost/index.php</linkhref>
      <linktarget>_blank</linktarget>
      <linktext>click me</linktext>
    </link>
  </linkcollection>
  <componentcollection>
    <component>
      <component_id>teasers</component_id>
      <componentuser>otheruser</componentuser>
      <componentcond>$customer == "private"</componentcond>
      <componentfiles>%comp%/Publication/teaser_1.php|%comp%/Publication/teaser_2.php|</componentfiles>
    </component>
    <component>
      <component_id>banner</component_id>
      <componentuser>demouser</componentuser>
      <componentcond></componentcond>
      <componentfiles>%comp%/banner.php</componentfiles>
    </component>
  </componentcollection>
  <articlecollection>
    <article>
      <article_id>news</article_id>
      <articletitle>Top News</articletitle>
      <articledatefrom>2002-10-01</articledatefrom>
      <articledateto>2002-11-01</articledateto>
      <articlestatus>active</articlestatus>
      <articleuser>demouser</articleuser>
      <articletextcollection>
        <text>
          <text_id>news:headline</text_id>
          <textuser>demouser</textuser>
          <textcontent>News from Scene</textcontent>
        </text>
      </articletextcollection>
      <articlemediacollection>
      </articlemediacollection>
      <articlelinkcollection>
      </articlelinkcollection>
      <articlecomponentcollection>
      </articlecomponentcollection>
    </article>
    <article>
      <article_id>special</article_id>
      <articletitle>Special Info</articletitle>
      <articledatefrom>2002-01-01</articledatefrom>
      <articledateto>2002-01-01</articledateto>
      <articlestatus>inactive</articlestatus>
      <articleuser>otheruser</articleuser>
      <articletextcollection>
        <text>
          <text_id>special:informations</text_id>
          <textuser>otheruser</textuser>
          <textcontent><![CDATA[<STRONG><FONT color=#cc0033>What is really going on behind the
Scene</FONT></STRONG>... find it out]]></textcontent>
        </text>
      </articletextcollection>
      <articlemediacollection>
      </articlemediacollection>
      <articlelinkcollection>
      </articlelinkcollection>
      <articlecomponentcollection>
      </articlecomponentcollection>
    </article>
  </articlecollection>
</container>
```

After a review of the content container, a structure can be seen, which is composed of the following main elements for content storage:

- hyperCMS specific information
- Meta-information
- Text
- Media (images or other multimedia files)
- Links
- Components
- Articles

The entire content is made up of these basic elements whose information is stored within XML tags.
Articles include the elements text, media and links as well. The entire contents of a page or component can be obtained from the associated content containers.

2.1 hyperCMS specific information

The data collected in this XML node represent primarily relevant information for the management of the container.

```
<hyperCMS>
  <contentcontainer>0000023.xml</contentcontainer>
  <contentxmlschema>object/page</contentxmlschema>
  <contentorigin>%page%/testpage.php</contentorigin>
  <contentobjects>%page%/testpage.php|%page%/linkedcopy_of_testpage.php |</contentobjects>
  <contentuser>demouser</contentuser>
  <contentdate>2002-11-26</contentdate>
  <contentpublished>2002-11-26</contentpublished>
 <contentstatus>active</contentstatus>
</hyperCMS>
```

Description:

contentcontainer	Name of the Content Container (unique for all publications)
contentxmlschema	Schema of the object: page (page) or component (comp)
contentorigin	Object (page or component) that led tot he creation of the Content Containers
contentobjects	All objects which use the Container
contentuser	Object owner (user)
contentdate	Date of the last changes of the Containers
contentpublished	Date of the last publishing of the object based on the Content Container
contentstatus	Status can be "active" if an object using the Cotainer exists. If all objects using the Container have been removed, the status will be set to "deleted". The Container therefore holds the last published information, but it can not be used anymore.

2.2 Meta-Information

The standard meta-information of a HTML page is described in this XML node.

```
<head>
  <pagetitle>test</pagetitle>
  <pageauthor>Mr. Content</pageauthor>
  <pagedescription>just a small demonstration</pagedescription>
  <pagekeywords>demo of XML</pagekeywords>
  <pagecontenttype>text/html; charset=UTF-8</pagecontenttype>
  <pagelanguage>de</pagelanguage>
  <pagerevisit></pagerevisit>
</head>
```

Description:

pagetitle	Page title
pageauthor	Author
pagedescription	Description of the content of a page
pagekeywords	List of keywords oft he page
pagecontenttype	Content-Type (character set) of the page or component
pagelanguage	Language shortcut of the page
pagerevisit	Search engine revisit of the page

2.3 Text

This XML-node stores the text.

```
  <text>
    <text_id>headline</text_id>
    <textuser>demouser</textuser>
    <textcontent>fgfdgfdg</textcontent>
  </text>
```

Description:

text_id	Text identification
textuser	Text owner (last changes of the Text by a user)
textcontent	text content

2.4 Media

This XML-node describes an included media file.

```
<media>
  <media_id>logo</media_id>
  <mediauser>otheruser</mediauser>
  <mediafile>Publication/demo_hcms0000033.jpg</mediafile>
  <mediaobject>%page%/Publication/Multimedia/demo.jpg</mediaobject>
  <mediaalttext>demoimage</mediaalttext>
  <mediaalign></mediaalign>
  <mediawidth>200</mediawidth>
  <mediaheight>100</mediaheight>
</media>
```

Description:

media_id	Media identification
mediauser	Media owner (last changes of the Media by a user)
mediafile	included multimedia file and declaration of the Publication
mediaobject	Location of the multimedia file
mediaalttext	Alternative text of the multimedia file
mediaalign	Alignment of the multimedia file
medawidth	Displayed width of the multimedia file
mediaheight	Displayed height of the multimedia file

2.5 Links

This XML-node describes the link to a page or file.

```
<link>
  <link_id>link</link_id>
  <linkuser>demouser</linkuser>
  <linkhref>http://localhost/index.php</linkhref>
  <linktarget>_blank</linktarget>
  <linktext>click me</linktext>
</link>
```

Description:

link_id	Link identification
linkuser	Link owner (last changes of the Link by a user)
linkhref	Reference (Link) to a page or file
linktarget	Target of the reference (name of the target frame)
linktext	Text describing the link

2.6 Components

This XML-node describes the reference to Components.

```
<component>
  <component_id>teasers</component_id>
  <componentuser>otheruser</componentuser>
  <componentcond>$customer == "private"</componentcond>
  <componentfiles>%comp%/teaser_1.php|%comp%/teaser_2.php|</componentfiles>
</component>
```

Description:

component_id	Component identification
componentuser	Component owner (last changes of the Component reference by a user)
componentcond	assigned customer Profile to the Component
componentfiles	Reference (Component-link) to a single or multiple Components

2.7 Articles

This XML-node describes the article information.

```
<article>
  <article_id>news</article_id>
  <articletitle>Top News</articletitle>
  <articledatefrom>2002-10-01</articledatefrom>
  <articledateto>2002-11-01</articledateto>
  <articlestatus>active</articlestatus>
  <articleuser>demouser</articleuser>
  <articletextcollection>
  </articletextcollection>
</article>
```

Description:

article_id	Article identification
articletitle	Title of the Article
articeldatefrom	Publishing date of the Article
articeldateto	End date of publishing the Article

articlestatus	Publishing settings of the Article:
	active = always published/displayed
	inactive = never published/displayed
	timeswitched = scheduled publishing/display

articleuser	Article owner (last changes of the Article by a user)
articlecollection	Holds all content of the Article

3 Function libraries

3.1 Including a library

The inclusion of a configuration or library requires that you know the absolute or relative path to the library. By using the function "require" or "require_once" and specifying the path to the library file all functions contained in the library will be available. Once the library is included, all functions can be used in the script.
To use the hyperCMS functions, the file "hypercms_api.inc.php" needs to be included. This file contains all functions required for programming.

```
// absolute path on MS Windows
require_once ("C:/inetpub/wwwroot/hypercms/function/hypercms_api.inc.php");

// relative path on MS Windows or Linux/UNIX
require_once ("function/hypercms_api.inc.php");
```

3.2 Loading the configuration

3.2.1 Content Management Server

To use the main configuration of hyperCMS the appropriate configuration file must be loaded. The main configuration will be loaded when including the hyperCMS API. However you can also load it in your script.
Using the variable $site for the identification of a publication, the publication can be loaded as well. The hyperCMS config file is located in "hypercms/config" and is named "config.inc.php". The publication config files are located in hyperCMS Data directory in the directory "data/config". Its filename holds the name of the publication as well as the ending "inc.php.", example: site.inc.php.

```
// Inlcude the main config file (please set the correct path):
require_once ("C:/inetpub/wwwroot/hypercms/config.inc.php");

// Include publication management config file
// Attention: Please use valid_publicationname to verify the name before including the file
if (valid_publicationname ($site))
{
  require_once ($mgmt_config['abs_path_data']."config/".$site.".conf.php");
}
```

The config files can be opened and read. Each parameter is described therein and is available for use in programs. Therefore, please take a look at the configuration to learn more about the parameters and their names.

If you want to set a specific language language, the variable $lang need to be set. $lang contains the language code, which is defined in the main configuration file "hypercms/config/config.inc.php".

```
// Set the language for messages in functions, German (de)
$lang = "de";
```

Since you want to use the hyperCMS API you need to include the hyperCMS API loader.

```
// Include the hyperCMS API:
require_once ($mgmt_config['abs_path_cms']."/function/hypercms_api.inc.php");
```

Now you can start using the API functions. For instance loading the content container of an object using various methods:

```
// Loading the page
$pagedata = loadfile ("%page%/MyPublication/home/", "index.php");

// Reading the name of the content container
$contentcontainer  = filepointer ($pagedata, "content");

// Loading the live content container from the content repository
$containerdata = loadcontainer ($contentcontainer, "published", $user);

// Or even more simple by using the direct path to the object
$containerdata = getobjectcontainer ("MyPublication", "%page%/MyPublication/home/",
"index.php", $user);
```

The functions will also load the publication specific configuration in case it is not provided. Since many features require the settings of a publication, it is advisable to include the configuration before you plan any actions.

3.2.2 Publication Server

Note that the configuration of the publication server (publication target) is stored separately in an INI file. If you will need the publication target settings, you must load and parse the INI file. After that you can access the settings as an array.
The INI file of the publication target is located in the external repository in the directory "repository/config". The file name corresponds to the name of the publication with the file extension ".ini".

```
// Load and parse the INI file using PHP
$publ_config = parse_ini_file ("C:/inetpub/wwwroot/repository/config/Mandant_1.ini");

// Access the settings oft he publication target
echo "This is the document root of the publication:".$publ_config[abs_publ_page];
```

3.3 Global variables

Many functions use global variables that are stored in the configuration and are available to functions as global. You should therefore take care that those global variable names of hyperCMS are not changed in your scripts.
The following list shows all global variables of hyperCMS, which must not be changed with in your own scripts:

```
$mgmt_config
$lang
$lang_name
$lang_shortcut
$lang_codepage
$lang_shortcut_default
```

Many global variables of hyperCMS are useful for use in hyperCMS scripts and PHP scripts, these are only available if the corresponding configuration has been loaded, or a hyperCMS script (used only during the publication process) is in use . Since this happened in the preview as well when publishing pages and components, these variables can be used in hyperCMS scripts. For dynamic applications that are executed each time a visitor accesses a page or component, the configuration must be integrated directly in the template, if hyperCMs variables are required.

Content Management Server:

$lang	language shortcut according to config.inc.php
$mgmt_config['url_path_cms'**]**	URL of the hyperCMS root directory according to config.inc.php
$mgmt_config['abs_path_cms'**]**	absolute path to the hyperCMS root directory according to config.inc.php
$mgmt_config['url_path_page'**]**	URL of the document root of the publication in the management system
$mgmt_config['abs_path_page'**]**	absolute of the document root of the publication in the management system
$mgmt_config['url_path_comp'**]**	URL of the component root directory of the publication in the management system
$mgmt_config['abs_path_comp'**]**	absolute path of the component root directory of the publication in the management system

Publication Server:
hyperCMS scripts can access variables at any time. The values are stored in the array $publ_config, but are also optionally available without the array. If the script/application will be executed at each access of a page or component on the publication target, the configuration file must be loaded separately.

$publ_config['url_publ_page'**]**	URL of the document root of the publication target
$publ_config['abs_publ_page'**]**	absolute path of the document root of the publication target
$publ_config['url_publ_comp'**]**	URL of the document root of the publication target
$publ_config['abs_publ_comp'**]**	absolute path of the document root of the publication target

Optional (deprected):

$url_publ_page	URL of the document root of the publication target
$abs_publ_page	absolute path of the document root of the publication target
$url_publ_comp	URL of the document root of the publication target
$abs_publ_comp	absolute path of the document root of the publication target

Vorlagenvariablen

There is also the possibility to use hyperCMS template variables in templates. These variables are a special feature, since they don not need to be used in hyperCMS script. Rather, they are placeholder for the value of a variable and can be used in any template.
This neutral form of the variables should primarily be used in templates, providing a more technology-neutral usage.
Please pay attention to the lower case of all variables!

%container% provides the name of the content containers of an object.
%template% provides the file name oft he used template of the object.
%object% provides the name oft he object.
%date% provides the actual date in the format YYYY-MM-DD.

For the integration of media files a path variable can be used. This path variable will be replaced by the URL (address) of publication target when publishing a page or component:

%media% provides the URL of the content media repository.
%tplmedia% provides the URL of the tempate media repository.

Also the document roots of pages and components of the publication target can be provided:

%url_page% provides the root URL of the pages document root.
%abs_page% provides the path to root directory of the pages document root.
%url_comp% provides the root URL of the components document root.
%abs_comp% provides the path to root directory of the components document root.

If you are using the hyperCMS APIs, it is often advisable to use the place holders %page% and %comp% to access the document root of pages and components. This path variables can be used only on the management side.
It should be noted that the variable is always paired with the publication name to form the root directory, eg:

%page%/besttrade/ …. Pages document root of the publication "besttrade"

%page%/Publikationsname/ provides the path to root directory of the pages document root.
%comp%/Publikationsname/ provides the path to root directory of the components document root.

3.4 Object operation library

This library contains all functions for the manipulation of objects (pages, components or files). You should only use these functions to access objects that are managed by the system.

3.4.1 createfolder

Syntax:
createfolder ($site, $location, $foldernew, $user)

Description:
Creates a new folder.

Example:
$result = createfolder ("besttrade", "%page%/besttrade/", "company", "brown");

Input-Parameters:

$site	Name of the publication
$location	absolute path (location of the new folder)
$foldernew	Name of the new folder
$user	User name

globale Input-Parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (has the folder been created successfully)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[folder]	Name of the folder

3.4.2 deletefolder

Syntax:
deletefolder ($site, $location, $folder, $user)

Description:
Removes an existing folder. The folder is removed only if it contains no more objects. All objects must therefore be removed by using the function deleteobject.

Example:
$result = deletefolder ("besttrade", "%page%/besttrade/", "company", "brown");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the folder)
$folder	Name of the folder
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (has the folder been removed successfully)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[folder]	Name of the existing folder is not successful, otherwise empty

3.4.3 renamefolder

Syntax:
renamefolder ($site, $location, $folder, $foldernew, $user)

Description:
Renames an existing folder.

Example:
$result = renamefolder ("besttrade", "%page%/besttrade/", "company", "news", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the folder)
$folder	old folder name
$foldernew	new folder name
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (Could the folder be renamed successfully)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[folder]	Name of the folder

3.4.4 createobject

Syntax:
createobject ($site, $location, $object, $template, $user)

Description:
Creates a new page or component based on a template. Please note that the location ($location) defines the category of the object (page/component) as well. This implies further that it the value of the parameter $template must provide a valid page or component template.

Example:
$result = createobject ("besttrade", "%page%/besttrade/", "index", "page_main", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the object)
$object	Name oft he new object (page or component)
$template	Name oft he page or component template (name of the template or template file name)
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File name of the page or component
$result[objectname]	Name of the page or component
$result[objecttype]	File-type or file extension of the file

3.4.5 deleteobject

Syntax:
deleteobject ($site, $location, $object, $user)

Description:
Removes an existing page, file or component.

Example:
$result = deleteobject ("besttrade", "%page%/besttrade/", "sales.doc", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[publication]	Name of the publication where the object is located
$result[location]	absolute path (location of the Object)
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file

3.4.6 renameobject

Syntax:
renameobject ($site, $location, $object, $objectnew, $user)

Description:
Renames an existing page, file or component.

Example:
$result = renameobject ("besttrade", "%page%/besttrade/", "sales.doc", "best.doc", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	old name of the object
$objectnew	new name of the object (without file extension)
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[publication]	Name of the publication where the object is located
$result[location]	absolute path (location of the Object)
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file

3.4.7 cutobject

Syntax:
cutobject ($site, $location, $object, $user)

Description:
Cut an existing page, file or component.

Example:
$result = cutobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File Name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file
$result[clipboard]	temporary entry in the clipboard (can be passed as global variable $clipboard to the function pasteobject, so reading the temporary file is not necessary)

3.4.8 copyobject

Syntax:
copyobject ($site, $location, $object, $user)

Description:
Copy an existing page, file or component.

Example:
$result = copyobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file
$result[clipboard]	temporary entry in the clipboard (can be passed as global variable $clipboard to the function pasteobject, so reading the temporary file is not necessary)

3.4.9 copyconnectedobject

Syntax:
copyconnectedobject ($site, $location, $object, $user)

Description:
Connected copy an existing page, file or component sharing the same content container.

Example:
$result = copyconnectedobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file
$result[clipboard]	temporary entry in the clipboard (can be passed as global variable $clipboard to the function pasteobject, so reading the temporary file is not necessary)

3.4.10 pasteobject

Syntax:
pasteobject ($site, $location, $user)

Description:
Paste Einfügen an existing page, file or component.

Example:
$result = pasteobject ("besttrade", "%page%/besttrade/", "Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$user	User name
$clipboard	temporary entry in the clipboard (can be passed as global variable $clipboard to the function pasteobject, so reading the temporary file is not necessary)

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[publication]	Name of the publication where the object is located
$result[location]	absolute path (location of the Object)
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file

3.4.11 lockobject

Syntax:
lockobject ($site, $location, $object, $user)

Description:
Locking of one or more existing pages or components based on the same content containers for the exclusive use of a user.

Example:
$result = lockobject ("besttrade", "%page%/besttrade/", "index.php","Miller");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file

3.4.12 unlockobject

Syntax:
unlockobject ($site, $location, $object, $user)

Description:
Unlocking of one or more existing pages or components based on the same content containers for the exclusive use of a user.

Example:
```
$result = unlockobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message
$result[object]	File name of the page, file or component
$result[objectname]	Name of the page, file or component
$result[objecttype]	File-type or file extension of the file

3.4.13 publishobject

Syntax:
publishobject ($site, $location, $object, $user)

Description:
Publishing a page or component. All connected copies of the object and its content container will be published as well. If a workflow is in use and does not permit the publsihing, the object will not be published.

Example:
```
$result = publishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message

3.4.14 unpublishobject

Syntax:
unpublishobject ($site, $location, $object, $user)

Description:
Unpublishing a page or component. Link and task management will be executed automatically. All connected copies of the object and its content containers will be unpublished as well.

Example:
```
$result = unpublishobject ("besttrade", "%page%/besttrade/", "index.php", "Miller");
```

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

global input parameters:
The following global input parameters need to be passed to the function:

$lang	Language setting or language shortcut, e.g. „en“, „de“

Output:

Array	Array $result holds the following information:
$result[result]	True/False (result of the action)
$result[add_onload]	JavaScript code for the onLoad event
$result[message]	Message regarding the result of the action or error message

3.4.15 getlinkedobject

Syntax:
getlinkedobject ($site, $location, $object, $cat)

Description:
This function extracts all objects that have a reference to the given object. This may be page or component links. If the object is a page, then all objects which have a page link to the object will be determined. If the object is a component, all objects which have a component link to the object will be determined.

Example:
$result = getlinkedobject ("besttrade", "%page%/besttrade/", "index.php", "page");

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$cat	optional: Objekt category [page, comp]

Output:

Array	Array $result holds the following information:
$result	False (action was not successful)
$result[publication]	Name of the publication where the object is located
$result[location]	absolute path (location of the Object)
$result[object]	Name of the object
$result[category]	Object category [page, comp]

3.4.16 getconnectedobject

Syntax:
getconnectedobject ($site, $container)

Description:
This function determines all objects that are based on the same content container.
The name of the content container of an object can be by extracted by the function "getfilename”.

Example:
$result = getconnectedobject ("besttrade", "0000127.xml");

Input parameters:

$site	Name of the publication
$container	Name of the content container

Output:

Array	Array $result holds the following information:
$result	False (action was not successful)
$result[publication]	Name of the publication where the object is located
$result[location]	absolute path (location of the Object)
$result[object]	Name of the object
$result[category]	Object category [page, comp]

3.4.17 getobjectcontainer

Syntax:
getobjectcontainer ($site, $location, $object, $user)

Description:
This function loads the content containers (XML string) of a particular object. The object can be a page, file, component or folder.
The desired content can be extracted from the XML-based container using the function “getcontent” or “selectcontent”.

Example:
```
$xmldata = getobjectcontainer ("besttrade", "%page%/Home/", "index.php", "demouser");
```

Input parameters:

$site	Name of the publication
$location	absolute path (location of the Object)
$object	Name of the object
$user	User name

Output:

XML-String	Content of the content container
False	An error occured

3.4.18 loadcontainer

Syntax:
loadcontainer ($container)

Description:
This function loads the content container (XML string) by its name or by its ID (in this case, the current working container will be loaded by default).
The desired content can be extracted from the XML-based container using the function “getcontent” or “selectcontent”.

Example:
```
// Load the published container
$xmldata1 = loadcontainer  "00012345.xml");
// Load the working container
$xmldata2 = loadcontainer ("00012345");
```

Input parameters:

$container	Name or ID (including zeros) of the container

Output:

XML-String	Content of the content container
False	An error occured

3.4.19 savecontainer

Syntax:
savecontainer ($container, $xmldata)

Description:
This function stores the content containers (XML string) by its name or by its ID (in this case, the current working container will be loaded by default).

Example:

```
// Save the published container
$result = savecontainer ("00012345.xml", $xmldata);
// Save the working container
$result = savecontainer ("00012345", $xmldata);
```

Input parameters:

$container	Name or ID (including zeros) of the container
$xmldata	Content of the content containers as XML string

Output:

True	The function was executed successfully
False	An error occured

3.5 File Pointer library

This function library allows you to determine or set the XML content container of an. You must have this object (page or component) loaded previously, e.g. via http or via the file system using the function “loadfile”.

3.5.1 getfilename

Syntax:
getfilename ($filedata, $tagname)

Description:
The input of the function is the content of a page, file object or component and the desired tag name, either content, template or media. After that the file name of the associated content container, template, or multimedia file will be returned.

Example:
```
// Load a page
$filedata = loadfile ("%page%/myPublication/home/", "index.php");

// Load the content container
$contentcontainer  = filepointer ($filedata, "content");
```

Input parameters:

$filedata	Content
$tagname	Tag name [content, template, media]

Output:

Filename	The function was executed successfully and returns the file name of the content Container, template or multimedia file
False	An error occured or the object is not managed by the system

3.5.2 setfilename

Syntax:
setfilename ($filedata, $tagname, $value)

Description:
The input of the function is the content of a page, file object or component, the tag name and a value for the file parameter of the tag. The return value oft he function will be True if writing the value has been successful or False otherwise.

Example:
```
$result = setfilename ($filedata, "template", "fullpage.page.tpl");
```

Input parameters:

$filedata	Content of the page, file object or component
$tagname	Tag name [content, template, media]
$value	(File)name of the content container, multimedia file or template

Output:

True	The function was executed successfully
False	An error occured

3.6 File operation library

The following functions for file operations should never be used to load or save objects (pages, components or files).
However you can use them to load and save XML content container, if you intend to develop extensions or applications.

3.6.1 loadfile

Syntax:
loadfile ($abs_path, $filename)

Description:
This function loads the content of a file. The absolute path and the filename itself must be provided as input parameters. The function waits usually up to 3 seconds to load locked files. If the user parameter $user is set, the function can also read locked files of the given user.

Example:
$data = loadfile ("%page%/myPublication/home/", "index.php");

Input parameters:

$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name

Output:

File content	The function was executed successfully and returns the content of the file
False	An error occured

3.6.2 savefile

Syntax:
savefile ($abs_path, $filename, $filedata)

Description:
This function saved content in files. The absolute path of the file name, and the content that will be written to the file needs to be passed as parameters. If the file is locked, it will not be saved and False will be returned.

Example:
$result = savefile ("%page%/myPublication/home/", "index.php", "text content");

Input parameters:

$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name
$filedata	Content that will be saved in the file

Output:

True	The function was executed successfully
False	An error occured

3.6.3 loadlockfile

Syntax:
loadlockfile ($user, $abs_path, $filename)

Description:
This function allows to load the content of a file like the function ”loadfile”, but it is also triggers a locking mechanism for the file.
The function should only be used when the data will be saved again using the function “savelockfile”. This ensures that no other write access by other users can take place.
The user, the absolute path and the filename itself must be passed as a parameter to load and lock the file. To save and unlock the file the function “savelockfile” must be used.

Example:
$data = loadlockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input parameters:

$user	User name of the user who locked the file
$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name

Output:

File content	The function was executed successfully und liefert den Inhalt der Datei
False	An error occured

3.6.4 savelockfile

Syntax:
savelockfile ($user, $abs_path, $filename, $filedata)

Description:
The function “savefile” saved data and unlocks previously opened files using “ loadlockfile”. For this purpose, the user, the absolute path, the file name, and the content that needs to be written to the file must be passed as parameters.

Example:
savelockfile ("Miller", "%page%/myPublication/home/", "index.php", "file content");

Input parameters:

$user	User name of the user who locked the file
$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name
$filedata	Content that will be saved in the file

Output:

True	The function was executed successfully
False	An error occured

3.6.5 lockfile

Syntax:
lockfile ($user, $abs_path, $filename)

Description:
The function “lockfile” locks a file for a specific user, so its available for the exclusive use. For this purpose, the user, the absolute path and the file name must be passed as a parameters.

Example:
lockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input parameters:

$user	User name of the user who locked the file
$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name

Output:

True	The function was executed successfully
False	An error occured

3.6.6 unlockfile

Syntax:
unlockfile ($user, $abs_path, $filename)

Description:
The function “unlockfile” unlocks files that have been previously locked by “lockfile” or opened by “loadlockfile”. For this purpose, the user, the absolute path and the file name must be passed as a parameters.

Example:
unlockfile ("Miller", "%page%/myPublication/home/", "index.php");

Input parameters:

$user	User name of the user who locked the file
$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name

Output:

True	The function was executed successfully
False	An error occured

3.6.7 deletefile

Syntax:
deletefile ($location, $file, $recursive)

Description:
With “deletefile” files and (empty) folders can be deleted. The absolute path, the file or directory name, and a parameter "recursive", which is either (0) or (1), need to be passed. If recursive is set to 1 the entire contents of the directory will be processed, including subdirectories and their files, using the value 0 only the file or directory (if empty) will be removed.

Example:
deletefile ("%page%/myPublication/home/", "index.php", 0);

Input parameters:

$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$file	file name
$recursive	0 or 1, if subdirectories should removed recursively as well

Output:

True	The function was executed successfully
False	An error occured

3.6.8 appendfile

Syntax:
append ($abs_path, $filename, $filedata)

Description:
With “appendfile” content can be added to a file. The function does not overwrite existing data of a file, it appends the data at the file end. For this the absolute path, the file name, and the content that needs to be written to the file must be passed as parameters.

Example:
appendfile ("%page%/myPublication/home/", "index.php", "© 2003 ...");

Input parameters:

$abs_path	absolute path of the file, %page% and %comp% can be used as the root elements of the path
$filename	file name
$filedata	Content that will be appended to the file

Output:

True	The function was executed successfully
False	An error occured

3.7 Edit content library

The following functions allow yout to read and write content from XML content container. You can optionally query the contents of the container with other technologies that can deal with XML. However, the Edit Content library offers a very simple and performant way of doing this.

3.7.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Description:
Set a specific value oft he XML declaration (1.row).

Example:
$xmldata = setxmlparameter ($xmldata, "encoding", "UTF-8");

Input parameters:

$xmldata	XML string that should be manipulated
$parameter	Name of the tag that should be manipulated
$value	Value saved in the tag

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.7.2 getcontent

Syntax:
getcontent ($xmldata, $tag)

Description:
Retrieves the XML content from the content container that is located inside the tags $tag. An array containing all content or childs found will be returned.

Example:
```
// Get all text-childs from the content container
$text_array = getcontent ($xmldata, "<text>");

// Show all text-childs
foreach ($text_array as $text) echo $text;
```

Input parameters:

$xmldata	XML string holding the content
$tag	Name of the tag holding the information or child nodes

Output:

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occured

3.7.3 getxmlcontent

Syntax:
getxmlcontent ($xmldata, $tag)

Description:
Retrieves the XML content from a content container that is located inside the tags $tag and leaves in contrast to the function “getcontent” the XML tags in the return value (array). An entire node (well-formed) will therefore be returned.
An array containing all content and childs found will be returned and can be stored and used in a variable of type array.

Example:
```
$text_array = getxmlcontent ($xmldata, "<text>");
foreach ($text_array as $text) echo $text;
```

Input parameters:

$xmldata	XML string holding the content
$tag	Name of the tag holding the information or child nodes

Output:

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occured

3.7.4 selectcontent

Syntax:
selectcontent ($xmldata, $parenttag, $childtag, $childvalue)

Description:
Retrieves the XML content defined by the tag $parenttag from the content container, where the childtag $childtag has a certain value $value.
An array with all items found will be returned and can be stored and used in a variable of type array.

Example:
Extract of a content container:

```
.....
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
.....

// Get all text-childs with id=summary
$text_array = selectcontent ($xmldata, "<text>", "<text_id>", "summary");

// Extract the summary from the found content
foreach ($text_array as $text)
{
  $summary = getcontent ($text, "<textcontent>");
}
```

Input parameters:

$xmldata	XML string holding the content
$parenttag	Name of the tag holding the information or child nodes
$childtag	optional: XML tag that encloses the information that must be of a certain value
$childvalue	optional: Value of the condition, the wildcard character * can be used at the beginning and/or end of the term and is a wildcard for any further characters.

Output:

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occured

3.7.5 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $parenttag, $childtag, $childvalue)

Description:
Retrieves the XML content defined by the tag $parenttag from the content container, where the childtag $childtag has a certain value $value. In contrast to the function “getcontent” the parent tags are included in the return value (array).
An array with all items found will be returned and can be stored and used in a variable of type array.

Example:
Extract of a content container:

```
…..
<text>
  <text_id>summary</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
…..

// Get all text-childs with id=summary
$text_array = selectxmlcontent ($xmldata, "<text>", "<text_id>", "summary");

// Extract the summary from the found content
foreach ($text_array as $text)
{
  $summary = getcontent ($text, "<textcontent>");
}
```

Input parameters:

$xmldata	XML string holding the content
$parenttag	Name of the tag holding the information or child nodes
$childtag	optional: XML tag that encloses the information that must be of a certain value
$childvalue	optional: Value of the condition, the wildcard character * can be used at the beginning and/or end of the term and is a wildcard for any further characters.

Output:

Array	Array holding all found values, the first value can be accessed using the first array element (Array[0])
False	An error occured

3.7.6 deletecontent

Syntax:
deletecontent ($xmldata, $tagname, $condtag, $condvalue)

Description:
Removes the entire XML content defined by the tag $tagname. For the selection of a certain child the appropriate XML childtag $condtag and the enclosed information as $condvalue as condition can be passed.

Example:
Extract of a content container:

```
…..
<text>
  <text_id>condition</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is my summary!</textcontent>
</text>
…..
```

```
$xmldata = deletecontent ($xmldata, "<text>", "<text_id>", "bedingung");
```

Input parameters:

$xmldata	XML string holding the content
$parenttag	Name of the tag holding the information or child nodes that should be removed
$condtag	optional: XML tag that encloses the information that must be of a certain value
$condvalue	optional: Value of the condition

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.7.7 setcontent

Syntax:
setcontent ($xmldata, $parenttagname, $tagname, $contentnew, $condtag, $condvalue)

Description:
An XML string is passed and within a certain parent node ($parenttagname) a certain parameter ($condtag) must exists and must have a certain value ($condvalue). If the condition is satisfied, the value of the parameter $tagname will be replaced by the new value $contentnew.

Example:
Extract of a content container:

```
…..
<text>
  <text_id>condition</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is should set!<textcontent>
</text>
…..
```

```
$xmldata = setcontent ($xmldata, "<text>", "<textcontent>", "This is my new value!",
"<text_id>", "condition");
```

Input parameters:

$xmldata	XML string holding the content
$parenttagname	optional: XML parent tag
$tagname	optional: XML child tag, that value should be replaced (if the condition is met)
$contentnew	new value for the XML child tag $tagmame
$condtag	optional: XML tag that encloses the information that must be of a certain value
$condvalue	optional: Value for the condition

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.7.8 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Description:
All XML strings $xmlnode will be replaced by a new XML string $xmlnodenew in $xmldata. This method is faster than „setcontent” when the updated XML node has already been extracted from the container.

Example:
Extract of a content container:

```
…..
<text>
  <text_id>condition</text_id>
  <textuser>editor1</textuser>
  <textcontent>This is old content!<textcontent>
</text>
…..
```

```
$xmldata = updatecontent ($xmldata, "<textcontent>This is old content!<textcontent> ",
"<textcontent>This is my new content!<textcontent>");
```

Input parameters:

$xmldata	XML string holding the content
$xmlnode	XML string to be replaced (node or substring of $xmldata)
$xmlnodenew	optional: new XML string, if empty, the existing XML string will be removed.

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.7.9 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $tagname)

Description:
Inserts an XML string (child node) before the end tag of the given XML parent tag. The modifiied XML string will be returned.

Example:
Extract of a content container:

```
…..
  <articletextlist>
    <text>
      <text_id>art1:summary</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
---------- The new child node will be inserted here ------------
    <text>
      <text_id>art1:longtext</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
----------------------------------------------------------
  </articletextlist>
…..
```

```
$xmldata = insertcontent ($xmldata, $insertxmldata, "<articletextlist>");
```

Input parameters:

$xmldata	XML string holding the content
$insertxmldata	XML string that will be inserted
$tagname	optional: Include xml string before the end tag of the given XML parent tag

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.7.10 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $grandtagname, $condtag, $condvalue, $parenttagname, $tagname, $contentnew)

Description:
Within a parent node a child node will be added, provided that a certain value in the overlying grandparent node meets the condition. In the child node a value can be set as well. The modified XML string will be returned.

Example:
Extract of a content container:

```
…..
<article>
  <article_id>art1</article_id>
  <articletitle></articletitle>
  <articledatefrom></articledatefrom>
  <articledateto></articledateto>
  <articlestatus>active</articlestatus>
  <articleuser></articleuser>
  <articletextlist>
    <text>
      <text_id>art1:summary</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
---------- The new child node will be inserted here ------------
    <text>
      <text_id>art1:longtext</text_id>
      <textuser>editor1</textuser>
      <textcontent>This is my summary!</textcontent>
    </text>
----------------------------------------------------------
  </articletextlist>
</article>
…..
```

```
$xmldata = addcontent ($xmldata, $sub_xmldata, "<article>", "<article_id>", "art1",
"<articletextlist>", "<text_id>", "art1:longtext");
```

Input parameters:

$xmldata	XML string holding the content
$sub_xmldata	XML string that will be inserted
$grandtagname	XML child tag where the $sub_xmldata should be embedded
$condtag	optional: XML tag that encloses the information that must be of a certain value
$condvalue	optional: Value for the condition
$parenttagname	optional: XML child tag, where $sub_xmldata should be embedded
$tagname	optional: Child tag of the embedded XML string where a value will be set
$contentnew	optional: Value for the $tagname

Output:

XML-String	Return of the manipulated XML string
False	An error occured

3.8 Meta Data Generator library

This function library allows you to automatically create keyword lists and a description from a given content. This can be used to automatically generate and fill in metadata for pages. Also meta data from multimedia files can be extracted and stored in the container of an object.

3.8.1 getkeywords

Syntax:
getkeywords ($text, $language, $charset)

Description:
The function requires the text content to be passed as input. All keywords are determined from the text and returned as a keyword list.

Example:
$keywords = getkeywords ("This is just a short text.", "en", "UTF-8");

Input parameters:

$text	Content als String
$language	optional: Language [en, de], default is "en"
$charset	optional: Character set, default is "UTF-8"

Output:

Keywords	Comma seperated list of keywords
False	An error occured

3.8.2 getdescription

Syntax:
getdescription ($text, $charset)

Description:
The function requires the text content to be passed as input. A brief description from the given text will be extracted and returned.

Example:
$keywords = getdescription ("This is just a short text.", "UTF-8");

Input parameters:

$text	Content as string
$charset	optional: Character set, default is "UTF-8"

Output:

Keywords	Short description of the content
False	An error occured

3.8.3 injectmetadata

Syntax:
injectmetadata ($site, $location, $object, $mediafile, $mapping, $user)

Description:
The function requires the path to the object and, optionally, the filename of the object or the multimedia file, and a mapping in order to save the meta data. The content from the meta data will be saved in the in the appropriate text ID of the container of the object, based on the given mapping.
WARNING: Existing data of the container will be overwritten!

Example:
```
// Mapping Definition (Meta Data Name -> Text-ID)
// Dublin Core
$mapping['dc:title'] = "Title";
$mapping['dc:subject'] = "Keywords";
$mapping['dc:description'] = "Description";
$mapping['dc:creator'] = "Creator";
$mapping['dc:rights'] = "Copyright";
// Adobe PhotoShop
$mapping['photoshop:SupplementalCategories'] = "Categories";
// Image Resolution defines Quality [Print, Web]
$mapping['hcms:quality'] = "Quality";

$result = injectmetadata ("Publication", "%comp%/test/", "image.jpg", "", $mapping, "Miller");
```

Input parameters:

$site	Name of the publication
$location	Absolute path to the object (Location of the object)
$object	optional: Name of the object (multimedia object file)
$mediafile	or optional: Name of the multimedia file
$mapping	Mapping array (Meta Data Name -> Text-ID)
$user	User name

Output:

True	Meta data has been saved successfully
False	An error occured

3.9 Notifications library

This function library sends automated messages to a user based on limits of a certain value of a particular field.
The user receives a pre-formatted message with information (links) to all objects that are within the search area (date upper and lower limit).

3.9.1 licensenotification

Syntax:
licensenotification ($site, $cat, $folderpath, $text_id, $date_begin, $date_end, $user)

Description:
The function returns all objects due to the specified search range (location and date limits) and sends an e-mail to a specific user with links to all the affected objects.

Example:

```
// set language for mail message
$lang = "en";

// send mail to Miller
$result = licensenotification ("Demo-DAM", "%comp%/images/", "comp", "valid_date",
"2012-09-01", "2012-09-30", "Miller");
```

Input parameters:

$site	Name of the publication
$cat	Object category [page, comp]
$folderpath	Location for the defintion of the search area
$text_id	Text ID of the XML node that need to be analyzed
$date_begin	Start date for the seach (YYYY-MM-DD)
$date_end	End date for the seach (YYYY-MM-DD)
$user	User name

Output:

True	Mail wurdwas send successfully
False	An error occured

4 Components and applications

If applications are integrated into components and variables need to be passed from a page to a component, you need to pay attention to the following:
The components must be integrated via the file system (not via HTTP).
All variables to be passed to the component need to be defined in the component as global.

Example:
A page passes a variable to a component.

Code example of a page:

```
<html>
<head>
<title>page</title>
<head>
<body>
<?php $test="This is just a test!"; ?>
[hyperCMS:components id='component']
</body>
</html>
```

The code of the component must be as followed:

```
<p>
<?php
global $test;
echo $test;
?>
</p>
```

In the example, the variable $test and its value "This is just a test!" will be passed to the component and will be displayed in the presentation of the component.

5 Database Connectivity

The Database Connectivity of the hyper Content Management Server allows the connection of different databases for the storage and retrieval of content. Relational databases are widely used as an external content repository.
For this purpose, a corresponding hyperCMS tag for the Database Connectivity need to be present in each template, which then refers to a DB-Connect file.
In this file functions are stored that hyperCMS will call, provided that the template points to the function file.
The contents are read from the database and will be displayed to the editor. If the editor modifies the content, it will be written to the database again. For read and write access different databases can be accessed as well. The functions in the DB Connect File offer only the shell or standardized interface to hyperCMS that needs to be filled by the programmer.

The subject of database integration is complex and need to be treated individual, since existing databases and their information must be integrated. hyperCMS does not provide any ER model or commits itself to specific database vendors. In general it can be said that all the possibilities of PHP can be exploited in order to connect to various data sources.

Besides the necessary parameters for queries to relational databases, the entire content conatiner is passed as an XML string. This allows documents or content from the content repository to be stored as a node in XML databases as well.

You therfore decide from which sources you read and save data. With PHP you have a powerful language that gives you access to all major databases.

More information regarding the functions of PHP can be found here: http://www.php.net

5.1 Creating a Database Connectivity

If you want to create a Database Connectivity, you need to make a copy of the the file “db_connect_default.php”. This file can be found in the root directory for storing the management data under the following path: /data /db_connect/
The copy of the file should be named according to the database you want to connect with.

Open the file and gain insight into the functions. In the source code you will also find a description of the functions and parameters passed as well as the output parameters.

The following example will show a read access to a MySQL database in order to extract a text based content. We assume that in a table "TextContent" the content will be presented by the primary key "container_id" and "text_id" as well as the text content "text" itself and the text-type "type". The user and the article ID is not stored separately, this is also not necessary for the uniqueness of the content, because the ID of the content container as well as the ID of the element already provides the primary key.

```
// ========================= db connect  ============================
// this file allows you to access a database using the full PHP functionality.
// you can read or write data from or into a database:

// ======================== read from database ==========================
// the following parameter values are passed to each function for
// retrieving data from the database:
// name of the site: $site [string]
// name of the content container: $container_id [string] (is unique
// inside hyperCMS over all sites)
// content container: $container_content [XML-string]
// identification name: $id [string]

// -------------------------------- text -------------------------------------
// if content is text
function db_read_text ($site, $content_id, $container_content, $id, $art_id, $user)
{
  //------------------------------------------------------------------------
  // input variables: $id [string], optional: $artid [string], $user [string
  // return value: $text [array]
  //              the array must exactly look like this:
  //              $text[text], optional: $text[type]
  //              constraints/accepted values for article type, see note below
  // note: special characters in $text are escaped into
  //       their html/xml equivalents.
  //       you can decide between unformatted, formatted and
  //       optional text using $type:
  //       unformatted text: $text[type] = textu
  //       formatted text: $text[type] = textf
  //       text option: $text[type] = textl
  //------------------------------------------------------------------------

  $user = "username";
  $password  = "password";
  $database = "database";

  // connect to database
  mysql_connect ("localhost", $user, $password);
  @mysql_select_db ($database) or die ("Unable to select database");

  // fire SQL-query
  $result = mysql_query ("SELECT Text, Type FROM TextContent WHERE
          container_id=$container_id AND text_id=$id);

  // count returned rows, must be 1 if unique
  $num_of_rows = mysql_num_rows ($result);

  // get the result into an array namend $row
  if ($num_of_rows == 1)
  {
    $row = mysql_fetch_row ($result);

    // set values
    $text[text] = $row[0];
    $text[type] = $row[1];
  }
  else $text = false;

  // close connection
  mysql_close ();

  // return result
  return $text;
}
```

6 Event System

The hyper Content & Digital Asset Management Server provides an event system that allows automated execution of actions when events in the system will be executed. This can be used, for example, to automate manual processes.
Events are usually started by the user by selecting an action, e.g. publishing a page. If he corresponding event is enabled, the event "onpublishobject" will be called after successful execution of the publication process of the page. The functions defined therein will therfore be executed.
The events of the event system can be defined in the "hypercms_eventsys.inc.php" file. This file is located in the internal repository in the folder "data/event system". In this file there are also other important instructions that must be followed during the execution of events.

The event system is valid within the management system for all publications. The system is part of the hyperCMS API and is thus performed on each invocation of the API functions.

Events can be activated and deactivated, so that the use of its defined events can be easily controlled in the "hypercms_eventsys.inc.php" file.
There is a distinction between PRE and POST events. The PRE event will be fired before the actual execution of the called action, while the POST event is called after the successful execution of the action.

Example:
When publishing an object the page “index.php” located at the same position should be automatically published as well, since the page “index.php” is used to generate an overview using a hyperCMS script all objects of the same folder.

```
// ----------------------------------- on publish object POST event -------------------------------------
function onpublishobject_post ($site, $cat, $location, $object, $user)
{
  // load configuration
  include_once ("config.inc.php");

  // hide the event used in your action (1) otherwise execute event (0)
  $eventsystem[hide] = 1;

  // insert your program code here
  $result = publishobject ($site, $location, "index.php", $user);

  // return true if successful
  if ($result[result] == true) return true;
  else return false;
}
```

7 List of hyperCMS API Functions

The documentation of all API functions in the current version is also available on our website hypercms.com. You can view the documentation in its current version directly in the browser. Click on the ?-Icon in the template editor to access the help with all hyperCMS tags and API functions.

8 hyperCMS API Function Reference

8.1 Main API Functions

8.1.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Input parameters:
$xmldata ... XML content container
$parameter ... paramater name
$value ... paramater value

Output:
XML content container / false on error

Description:
set parameter values in XML declaration (e.g. encoding):
encoding="UTF-8"

8.1.2 getcontent

Syntax:
getcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname.

8.1.3 geticontent

Syntax:
geticontent ($xmldata, $starttagname)

Input parameters:

$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are however always case-sensitive!)
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname

8.1.4 getxmlcontent

Syntax:
getxmlcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.1.5 getxmlicontent

Syntax:
getxmlicontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.1.6 selectcontent

Syntax:
selectcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.1.7 selecticontent

Syntax:
selecticontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue

8.1.8 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname

$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag
$startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.1.9 selectxmlicontent

Syntax:
selectxmlicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag
$startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue

8.1.10 deletecontent

Syntax:
deletecontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:

<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.1.11 deleteicontent

Syntax:
deleteicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.1.12 setcontent

Syntax:
setcontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is a parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.13 seticontent

Syntax:
seticontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.14 setcontent_fast

Syntax:
setcontent_fast ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
function designed for link management, extremely fast but with limitations (only CASE-Sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if

condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.1.15 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Input parameters:
$xmldata ... XML content container
$xmlnode ... XML node to be replaced
$xmlnodenew ... new XML node

Output:
XML content container / false on error

Description:
updates a given xml string $xmlnode in $xmldata with the content $xmlnodenew.
this method provides a faster way to update xml nodes when the node was selected before.

8.1.16 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
```
....................
......................
<tagname> <- list start
.....................
.....................
insertxmldata <- insertxmldata
</tagname> <- list end
....................
```
insert $insertxmldata string at the end of all child between the parent $tagname

8.1.17 inserticontent

Syntax:
inserticontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$insertxmldata ... XML node to be inserted in starttagname
$starttagname ... tag name of the parent XML node

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
…………………
……………………
<tagname> <- list start
…………………
…………………
insertxmldata <- insertxmldata
</tagname> <- list end
…………………
insert $insertxmldata string at the end of all child between the parent $tagname

8.1.18 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
…………………
…………………
………………… }
<tagname>contentnew</tagname> } <- sub_xmldata
………………… }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml node to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.1.19 addicontent

Syntax:

addicontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
....
....
.... }
<tagname>contentnew</tagname> } <- sub_xmldata
.... }
</parenttagname> <- list end
</grandtagname>
$xmldata = data string to be parsed
$sub_xmldata = xml subschema to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.2 Get API Functions

8.2.1 getserverload

Syntax:
getserverload ()

Input parameters:

Output:
Returns the average system load (the number of processes in the system run queue) over the last minute and the number of CPU cores

8.2.2 getsession

Syntax:
getsession ($variable, $default="")

Input parameters:
$variable ... variable name

$default ... default value (optional)

Output:
value

8.2.3 getrequest

Syntax:
getrequest ($variable, $force_type=false, $default="")

Input parameters:
$variable ... variable name
$force_type ... must be of certain type
[numeric,array,publicationname,locationname,objectname,url,bool] (optional)
$default ... default value (optional)

Output:
value

Description:
return a value from POST, GET or COOKIE, or a default value if none set

8.2.4 getrequest_esc

Syntax:
getrequest_esc ($variable, $force_type=false, $default="", $js_protection=false)

Input parameters:
$variable ... variable name
$force_type ... must be of certain type
[numeric,array,publicationname,locationname,objectname] (optional)
$default ... default value (optional)
$js_protection ... remove characters to avoid JS injection [true,false] (optional)

Output:
value

Description:
return a escaped value tp prevent XSS from POST, GET or COOKIE, or a default value if none set

8.2.5 getuserip

Syntax:
getuserip ()

Input parameters:

Output:
IP address of client / false on error

Description:
retrieves IP address of the client/user.

8.2.6 getlanguageoptions

Syntax:

getlanguageoptions ()

Input parameters:

global input parameters:
$mgmt_config

Output:
array with 2-gigit language code as key and language name in English as value / false on error

8.2.7 getlanguagefile

Syntax:
getlanguagefile ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config

Output:
language file name

8.2.8 getcodepage

Syntax:
getcodepage ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config
$hcms_lang_codepage

Output:
code page (character set)

8.2.9 getcalendarlang

Syntax:
getcalendarlang ($lang="en")

Input parameters:
$lang ... language code (optional)

global input parameters:
$mgmt_config

Output:
supported language code for calendar

8.2.10 getescapedtext

Syntax:

getescapedtext ($text, $charset="", $lang="")

Input parameters:
$text ... text as string
$charset ... character set of text
$lang ... 2-digit language code

global input parameters:
$mgmt_config
$hcms_lang_codepage
$hcms_lang

Output:
HTML escaped text

Description:
if the destination character set is not supported by language set the text need to be HTML escaped.

8.2.11 getobjectcontainer

Syntax:
getobjectcontainer ($site, $location, $object, $user)

Input parameters:
$site ... publication [string]
$location ... location [string]
$object ... object [string]
$user ... user [string]

global input parameters:
$mgmt_config

Output:
Content Container [XML]/false

Description:
loads the content container of a given object (page, component, folder)

8.2.12 getcontainer

Syntax:
getcontainer ($containerid, $type)

Input parameters:
$containerid ... container name or container ID
$type ... container type [published

global input parameters:
$mgmt_config

Output:
Contant Container [XML]/false

Description:
obsolete function used as shell for loadcontainer function without loading locked containers

8.2.13 getcontainername

Syntax:
getcontainername ($container)

Input parameters:
$container ... container name (e.g. 0000112.xml.wrk) or container ID

global input parameters:
$mgmt_config

Output:
Array with file name of the working content container (locked or unlocked!) and username if locked

8.2.14 getlocationname

Syntax:
getlocationname ($site, $location, $cat, $source="path")

Input parameters:
$site ... publication name
$location ... location path (as absolute path or converted path)
$cat ... category [page,comp]
$source ... source for name [path,name]

global input parameters:
$mgmt_config
$lang
$hcms_lang_codepage

Output:
location with readable names instead of file names / false on error

8.2.15 getthemelocation

Syntax:
getthemelocation ($theme="")

Input parameters:
$theme ... theme name (optional)

global input parameters:
$mgmt_config

Output:
path to theme / false

Description:
returns the absolute path (URL) to the theme (css and images).

8.2.16 getcategory

Syntax:
getcategory ($site="", $location)

Input parameters:

$site ... publication name (optional)
$location ... location path

global input parameters:
$mgmt_config
$publ_config

Output:
category ['page
comp'] / false on error

Description:
evaluates the category ['page, comp'] of a location.

8.2.17 getpublication

Syntax:
getpublication ($path)

Input parameters:
$path ... converted location path

Output:
publication name

Description:
extract the publication name of a location path.

8.2.18 getlocation

Syntax:
getlocation ($path)

Input parameters:
$path ... location path

Output:
location (without object or folder)

Description:
extract the location excluding object or folder of a location path.

8.2.19 getobject

Syntax:
getobject ($path)

Input parameters:
$path ... location path

Output:
object or folder name

Description:
extract the object or folder of a location path.

8.2.20 getmediacontainername

Syntax:
getmediacontainername ($file)

Input parameters:
$file ... file name

Output:
container name / false on error

Description:
extract the container name from a multimedia file name by using the hcm-ID

8.2.21 getmediafileversion

Syntax:
getmediafileversion ($container)

Input parameters:
$container ... container name or container ID

global input parameters:
$mgmt_config
$user

Output:
media file name / false on error

Description:
extracts the name from the multimedia file by container name or ID in order to get the media file of older content versions.
if the result is false, there is no older media file version.

8.2.22 getobjectid

Syntax:
getobjectid ($objectlink)

Input parameters:
$objectlink ... converted object path or pathes separated by |

Output:
object ID

Description:
converts object path to object ID

8.2.23 getobjectlink

Syntax:
getobjectlink ($objectid)

Input parameters:
$objectid ... converted object ID or IDs separated by |

Output:

converted object link

Description:
converts object ID to object path

8.2.24 getcontainerversions

Syntax:
getcontainerversions ($container)

Input parameters:
$container ... container ID or container name

global input parameters:
$mgmt_config

Output:
array of all versions (array[version-extension] = file-name) / false

8.2.25 gettemplateversions

Syntax:
gettemplateversions ($site, $template)

Input parameters:
$site ... publication name
$template ... template name

global input parameters:
$mgmt_config

Output:
array of all versions (array['YYYY-MM-DD HH:MM:SS'] = file-name) / false

8.2.26 getfileinfo

Syntax:
getfileinfo ($site, $file, $cat="comp")

Input parameters:
$site ... publication name (optional)
$file ... file name incl. extension
$cat ... category [page,comp] (optional)

global input parameters:
$mgmt_config

Output:
array/false

Description:
defines file properties based on the file extension and returns file info in an array:
$result['file']: file name without hypercms management extension
$result['name']: readable file name without hypercms management extension
$result['filename']: file name without file extensions
$result['icon']: file name of the file icon
$result['icon_large']: file name of the large file icon
$result['type']: file type

$result['ext']: file extension incl. dot in lower case
$result['published']: if file published = true else = false

8.2.27 getobjectinfo

Syntax:
getobjectinfo ($site, $location, $object, $user="sys", $container_version="")

Input parameters:
$site ... publication name
$location ... location
$object ... object name
$user ... user name (optional)
$container_version ... container version (optional)

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
get's all file pointers (container, media, template) and object name from object file and collects info from container version if provided

8.2.28 getfilesize

Syntax:
getfilesize ($file)

Input parameters:
$file ... converted path to file or directory

global input parameters:
$mgmt_config

Output:
result array with file size in kB and file count / false on error

8.2.29 getmimetype

Syntax:
getmimetype ($file)

Input parameters:
$file ... file name incl. extension

global input parameters:
$mgmt_config

Output:
mime_type

Description:
gets the mime-type of the file of its extension.
if file has a version file extension the next file extension will be used.

8.2.30 getfiletype

Syntax:
getfiletype ($file_ext)

Input parameters:
$file_ext ... file extension or file name

global input parameters:
$mgmt_config
$hcms_ext

Output:
file type to be saved in database based on file extension

8.2.31 getvideoinfo

Syntax:
getvideoinfo ($mediafile)

Input parameters:
$mediafile ... path to video file

global input parameters:
$mgmt_config
$mgmt_mediapreview

Output:
video file information as result array / false on error

8.2.32 getbrowserinfo

Syntax:
getbrowserinfo ()

Input parameters:

Output:
client browser + version as array

8.2.33 getcontentlocation

Syntax:
getcontentlocation ($container_id, $type="abs_path_content")

Input parameters:
$container_id ... container id
$type ... type [url_path_content

global input parameters:
$mgmt_config

Output:
location of the container file / false on error

Description:
gets the content location based on the given container id

a split up of folders is necessary since the number of directories is limited by the filesystem, e.g. Linux ext3 is limited to 32000.

8.2.34 getmedialocation

Syntax:
getmedialocation ($site, $file, $type)

Input parameters:
$site ... publication name
$file ... multimedia file name (including hcm-ID)
$type ... type [url_path_media

global input parameters:
$mgmt_config
$publ_config

Output:
location of the multimedia file / false on error

Description:
gets the media repsitory location from $mgtm_config Array.
the function supports up to 10 media repositories.
any other rules for splitting the media files on multiple devices could be implemented as well.

8.2.35 getlockedfileinfo

Syntax:
getlockedfileinfo ($location, $file)

Input parameters:
$location ... location to file
$file ... file name

global input parameters:
$mgmt_config

Output:
Array holding file name incl. lock extension and user name / false on error

Description:
finds the locked file and returns the name and user as array

8.2.36 getusersonline

Syntax:
getusersonline ()

Input parameters:

global input parameters:
$mgmt_config

Output:
Array of online user names / false

8.2.37 getchatstate

Syntax:
getchatstate ($register=true)

Input parameters:
$register ... register stat in session [true/false] (optional)

global input parameters:
$mgmt_config

Output:
state of chat / false on error

8.2.38 getimagelib

Syntax:
getimagelib ()

Input parameters:

global input parameters:
$mgmt_imagepreview

Output:
name of image library used [GD
ImageMagick] / false on error

8.2.39 getfilename

Syntax:
getfilename ($filedata, $tagname)

Input parameters:
$filedata ... file content
$tagname ... hyperCMS tag name in page or component

Output:
file name

Description:
extracts the file name of the hyperCMS content and template pointer tags

8.2.40 gethypertag

Syntax:
gethypertag ($filedata, $tagname, $offset=0)

Input parameters:
$filedata ... file content [string]
$tagname ... full/partly hyperCMS tag name (with or without hyperCMS:) [string]
$offset ... offset position [integer]

Output:
full hyperCMS tag array [array]/false on error

Description:

finds the hyperCMS tag start and end position
and returns an array of the whole tags including all information.
offset value must be integer value and is used to skip search
for hyperCMS tag till offset position of filedata.

8.2.41 gethypertagname

Syntax:
gethypertagname ($tagdata)

Input parameters:
$tagdata ... full hyperCMS tag

Output:
full hyperCMS tag name/false on error

Description:
reads the name of the hyperCMS tag.

8.2.42 gethtmltag

Syntax:
gethtmltag ($filedata, $tag)

Input parameters:
$filedata ... file content
$tag ... full hyperCMS tag (or other identifier)

Output:
full html tag/false on error

Description:
finds the first html tag start and end position of a nested hyperCMS tag
and returns the whole tag including all information.
works also if other script tags are nested in the HTML-tag.
this function is not case sensitive!

8.2.43 gethtmltags

Syntax:
gethtmltags ($filedata, $tag)

Input parameters:
$filedata ... file content
$tag ... full hyperCMS tag or other identifier in html tag

Output:
string from html tag start to end tag/false on error

Description:
finds the nearest html tag start and end position of a nested hyperCMS tag
and returns the whole tag including all information.
this functions works also for html-tag pairs like <a href></a>, <div></div> and so on.

8.2.44 getattribute

Syntax:

getattribute ($string, $attribute, $secure=true)

Input parameters:
$string ... string including attributes
$attribute ... attribute name
$secure ... secure attribute value reg. XSS (optional)

Output:
attribute value/false on error

Description:
get the value of a certain attribute out of a string (...attributname=value....)

8.2.45 getoption

Syntax:
getoption ($string, $option)

Input parameters:
$string ... string including options
$option ... option name

Output:
option value/false on error

Description:
get the value of a certain option out of a string (-c:v value -ar 44100)

8.2.46 getcharset

Syntax:
getcharset ($site, $data)

Input parameters:
$site ... publication
$data ... data from template or content container [string]

global input parameters:
$mgmt_config

Output:
array with content-type and charset / false on error

Description:
extract the content-type definition and the character set from the template (1st priority), content container (2nd priority) or publication settings (3rd priority)

8.2.47 getartid

Syntax:
getartid ($id)

Input parameters:
$id ... string including id

Output:
article id/false on error

Description:
extract article id out of the id.

8.2.48 getelementid

Syntax:
getelementid ($id)

Input parameters:
$id ... string including id

Output:
element id/false on error

Description:
extract element id out of the id

8.2.49 getfirstkey

Syntax:
getfirstkey ($array)

Input parameters:
$array ... array

Output:
array key of first element in array if $value is not empty / false on error

8.3 Set API Functions

8.3.1 setsession

Syntax:
setsession ($variable, $content="", $write=false)

Input parameters:
$variable ... temporary hyperCMS variable name or array
$content ... value as string or array (optional)
$write ... write session data for load balancer [true,false] (optional)

Output:
true / false on error

8.3.2 setarticle

Syntax:
setarticle ($site, $contentdata, $contentfile, $arttitle, $artstatus, $artdatefrom, $artdateto, $artuser, $user)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$arttitle ... article title array
$artstatus ... article status array
$artdatefrom ... article beginn date array
$artdateto ... article end date array

$artuser ... user array or string
$user ... user name

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.3 settext

Syntax:
settext ($site, $contentdata, $contentfile, $text, $type, $art, $textuser, $user, $charset="", $addmicrotime=false)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$text ... text array
$type ... type array or string of text [u,f,l,c,d]
$art ... article array or string [yes,no]
$textuser ... text user array or string
$user ... user name
$charset ... character set of text content
$addmicrotime ... add microtime to ID [true,false] used for comments

global input parameters:
$mgmt_config
$publ_config

Output:
updated content container (XML)
false on error

8.3.4 setmedia

Syntax:
setmedia ($site, $contentdata, $contentfile, $mediafile, $mediaobject_curr, $mediaobject, $mediaalttext, $mediaalign, $mediawidth, $mediaheight, $art, $mediauser, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$mediafile ... media arrays (some are optional)
$mediaobject_curr ... article array or string [yes,no]
$mediaobject ... content user array or string
$mediaalttext ... user name
$mediaalign ... chracter set of text content
$mediawidth
$mediaheight
$art
$mediauser
$user
$charset

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.5 setpagelink

Syntax:
setpagelink ($site, $contentdata, $contentfile, $linkhref_curr, $linkhref, $linktarget, $linktext, $art, $linkuser, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$linkhref_curr ... current link array
$linkhref ... new link array
$linktarget ... link target array
$linktext ... link text array
$art ... article array or string [yes,no]
$linkuser ... content user array or string
$user ... user name
$charset ... chracter set of text content

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.6 setcomplink

Syntax:
setcomplink ($site, $contentdata, $contentfile, $component_curr, $component, $condition, $art, $compuser, $user)

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$component_curr ... component arrays (some are optional)
$component ... article array or string [yes,no]
$condition ... content user array or string
$art ... user name
$compuser
$user

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

8.3.7 sethead

Syntax:
sethead ($site, $contentdata, $contentfile, $headcontent, $user, $charset="")

Input parameters:
$site ... publication name
$contentdata ... container (XML)
$contentfile ... container name
$headcontent ... content array
$user ... user name
$charset ... chracter set of text content

global input parameters:
$mgmt_config

Output:
updated content container (XML)
false on error

Description:
if content is general meta information

8.3.8 setfilename

Syntax:
setfilename ($filedata, $tagname, $value)

Input parameters:
$filedata ... file content
$tagname ... hyperCMS tag name in page or component [content
$value ... template

Output:
filedata/false on error

Description:
sets or creates the file name of the hyperCMS content file, template file, media file or file name pointer

8.4 Connect API Functions

8.4.1 ftp_userlogon

Syntax:
ftp_userlogon ($server, $user, $passwd, $ssl=false)

Input parameters:
$server ... FTP servername or IP
$user ... user name
$passwd ... password
$ssl ... SSL [true,false] (optional)

global input parameters:
$mgmt_config

Output:

true / false on error

Description:
this function connects and performs logon to an FTP server

8.4.2 ftp_userlogout

Syntax:
ftp_userlogout ($conn_id)

Input parameters:
$conn_id ... FTP connection

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function disconnects from an FTP server

8.4.3 ftp_getfile

Syntax:
ftp_getfile ($conn_id, $remote_file, $local_file, $passive=true)

Input parameters:
$conn_id ... FTP connection
$remote_file ... path to file on FTP server
$local_file ... passive mode [true,false] (optional)
$passive

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function gets a file from the FTP server

8.4.4 ftp_putfile

Syntax:
ftp_putfile ($conn_id, $local_file, $remote_file, $passive=true)

Input parameters:
$conn_id ... FTP connection
$local_file ... path to local file
$remote_file ... path to file on FTP server
$passive ... passive mode [true,false] (optional)

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function puts a file to the FTP server

8.4.5 ftp_filelist

Syntax:
ftp_filelist ($conn_id, $path=".", $passive=true)

Input parameters:
$conn_id ... FTP connection
$path ... path to remote directory (optional)
$passive ... passive mode [true,false] (optional)

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
this function gets a file/directory listing of the FTP server

8.4.6 createsharelink_facebook

Syntax:
createsharelink_facebook ($site, $url)

Input parameters:
$site ... URL to share
$url

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.7 createsharelink_twitter

Syntax:
createsharelink_twitter ($site, $url, $text)

Input parameters:
$site ... URL to share
$url ... message to share
$text

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.8 createsharelink_googleplus

Syntax:

createsharelink_googleplus ($site, $url)

Input parameters:
$site ... URL to share
$url

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.9 createsharelink_linkedin

Syntax:
createsharelink_linkedin ($site, $url, $title, $summary, $source)

Input parameters:
$site ... URL to share
$url ... title
$title ... summary (optional)
$summary ... source (optional)
$source

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.4.10 createsharelink_pinterest

Syntax:
createsharelink_pinterest ($site, $image_url, $title, $description)

Input parameters:
$site ... image URL to share
$image_url ... title
$title ... description (optional)
$description

global input parameters:
$mgmt_config

Output:
Share URL / false on error

8.5 Security API Functions

8.5.1 rootpermission

Syntax:
rootpermission ($site_name, $site_admin, $permission_str)

Input parameters:
$site_name ... publication name
$site_admin ... publication admin

$permission_str ... permission string from group

global input parameters:
$rootpermission
$mgmt_config

Output:
global permission array/false

Description:
deseralizes the permission string and and returns the root permission array.

8.5.2 globalpermission

Syntax:
globalpermission ($site_name, $permission_str)

Input parameters:
$site_name ... publication
$permission_str ... permission string from group

Output:
global permission array/false

Description:
deseralizes the permission string and returns the global permission array.

8.5.3 localpermission

Syntax:
localpermission ($site_name, $permission_str)

Input parameters:
$site_name ... publication
$permission_str ... permission string from group

Output:
local permission array/false

Description:
deseralizes the permission string and returns the local permission array.

8.5.4 accessgeneral

Syntax:
accessgeneral ($site, $location, $cat)

Input parameters:
$site ... publication
$location ... location (path to folder)
$cat ... object category ['page

global input parameters:
$mgmt_config
$hiddenfolder
$siteaccess

Output:

true/false

Description:
checks general access to certain system folders, publications and returns true if access is granted

8.5.5 accesspermission

Syntax:
accesspermission ($site, $location, $cat)

Input parameters:
$site ... location (path to folder)
$location ... object category ['page
$cat ... comp']

global input parameters:
$pageaccess
$compaccess
$hiddenfolder
$hcms_linking
$mgmt_config

Output:
group with access permissions as array / false on error

Description:
evaluates page and component access permissions and returns group(s).

8.5.6 setlocalpermission

Syntax:
setlocalpermission ($site, $group_array, $cat)

Input parameters:
$site ... publication
$group_array ... group name array
$cat ... object category [page,comp]

global input parameters:
$localpermission

Output:
local permission array / false on error

Description:
sets local permissions of a user group for a specific publication

8.5.7 checkpublicationpermission

Syntax:
checkpublicationpermission ($site, $strict=true)

Input parameters:
$site ... publication name
$strict ... strictly limited to siteaccess only without inheritance [true/false] (optional)

global input parameters:

$mgmt_config
$siteaccess

Output:
"direct" for direct access via group permission / "inherited" for access through inheritance / false

Description:
checks access to a publication based on the site access and inheritance settings

8.5.8 checkadminpermission

Syntax:
checkadminpermission ()

Input parameters:

global input parameters:
$adminpermission

Output:
true/false

Description:
checks super admin permission

8.5.9 checkrootpermission

Syntax:
checkrootpermission ($name)

Input parameters:
$name ... permission name

global input parameters:
$rootpermission

Output:
true/false

Description:
checks root permission

8.5.10 checkglobalpermission

Syntax:
checkglobalpermission ($site, $name)

Input parameters:
$site ... publication name
$name ... permission name

global input parameters:
$globalpermission

Output:
true/false

Description:
checks global permission for a publication

8.5.11 checklocalpermission

Syntax:
checklocalpermission ($site, $group, $name)

Input parameters:
$site ... publication name
$group ... user group name
$name ... permission name

global input parameters:
$$localpermission

Output:
true/false

Description:
checks local permissions of a user group for a specific publication

8.5.12 userlogin

Syntax:
userlogin ($user, $passwd, $hash="", $objref="", $objcode="", $ignore_password=false, $locking=true)

Input parameters:
$user ... username
$passwd ... password
$hash ... hash code of user
$objref ... object reference for hcms linking (object ID)
$objcode ... object code for hcms linking (crypted object ID)
$ignore_password ... ignore passwordcheck needed for WebDAV or access link [true/false]
$locking ... lock IP after 10 failed attempts to login [true/false]

global input parameters:
$mgmt_config
$eventsystem
$hcms_lang_codepage
$hcms_lang
$lang

Output:
result array

Description:
login of user by sending user and password using the variables: $sentuser, $sentpasswd
this procedure will register the user in the hypercms session and in the php session.
the procedure will return true or false using the variable $result.

8.5.13 registerinstance

Syntax:
registerinstance ($instance, $load_config=true)

Input parameters:

$instance ... instance name
$load_config ... load main config of instance [true/false] (optional)

global input parameters:
$mgmt_config

Output:
true/false

8.5.14 createchecksum

Syntax:
createchecksum ($permissions="")

Input parameters:
$permissions ... array or empty

Output:
MD5 checksum

8.5.15 writesession

Syntax:
writesession ($user, $passwd, $checksum)

Input parameters:
$user ... user name
$passwd ... password
$checksum ... checksum

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
writes hyperCMS specific session data of a user

8.5.16 writesessiondata

Syntax:
writesessiondata ()

Input parameters:

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
serializes and writes all session data of a user

8.5.17 createsession

Syntax:
createsession ()

Input parameters:

global input parameters:
$mgmt_config

Output:
true

Description:
Checks if session data of a user is available. This function does access session variables directly!

8.5.18 killsession

Syntax:
killsession ($user="", $destroy_php=true)

Input parameters:
$user ... user name for hyperCMS session (optional)
$destroy_php ... destroy php session [true,false] (optional)

global input parameters:
$mgmt_config

Output:
true

Description:
destroys session data of user

8.5.19 checkdiskkey

Syntax:
checkdiskkey ($users="", $site="")

Input parameters:
$users ... user count (optional)
$site ... publication names (use | as seperator) (optional)

global input parameters:
$mgmt_config

Output:
true/false

Description:
checks the disc key of the installation

8.5.20 checkpassword

Syntax:
checkpassword ($password)

Input parameters:
$password ... password a string

global input parameters:
$mgmt_config
$lang

Output:
true if passed / error message as string

Description:
this function checks the strength of a password and return the error messages or true.

8.5.21 loguserip

Syntax:
loguserip ($client_ip, $user="sys")

Input parameters:
$client_ip ... client IP address
$user ... user logon name (optional)

global input parameters:
$mgmt_config

Output:
true / false on error

8.5.22 checkuserip

Syntax:
checkuserip ($client_ip, $user="", $timeout="")

Input parameters:
$client_ip ... client IP address
$user ... user logon name (optional)
$timeout ... timeout in minutes (optional)

global input parameters:
$mgmt_config

Output:
true if IP is not locked / false if IP is locked or on error

8.5.23 checkuserrequests

Syntax:
checkuserrequests ($user="sys")

Input parameters:
$user ... user name (optional)

global input parameters:
$mgmt_config

Output:
true / false if a certain amount of reguests per minute is exceeded

Description:
provides security for Cross-Site Request Forgery

8.5.24 checkusersession

Syntax:
checkusersession ($user="sys", $CSRF_detection=true)

Input parameters:
$user ... user name (optional)
$CSRF_detection ... include CSRF detection [true,false]

global input parameters:
$mgmt_config

Output:
true / html-output followed by termination

Description:
checks if session data of user is correct. This function does access session variables directly!
requires config.inc.php

8.5.25 allowuserip

Syntax:
allowuserip ($site)

Input parameters:
$site ... publication name

global input parameters:
$mgmt_config

Output:
true / false

Description:
checks if the client IP is in the range of valid IPs.
requires config.inc.php

8.5.26 valid_objectname

Syntax:
valid_objectname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.27 valid_locationname

Syntax:
valid_locationname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.28 valid_publicationname

Syntax:
valid_publicationname ($variable)

Input parameters:
$variable ... variable (string or array)

Output:
variable / false on error

Description:
test if an expression includes forbidden characters (true) or doesnt (false) to prevent directory browsing

8.5.29 html_encode

Syntax:
html_encode ($expression, $encoding="", $js_protection=false)

Input parameters:
$expression ... variable as string or array
$encoding ... conversion of all special characters based on given character set or to ASCII (optional)
$js_protection ... remove characters to avoid JS injection [true,false] (optional)

Output:
html encoded value as array or string / false on error

Description:
converts a string into the html equivalents (also used for XSS protection).
supports multibyte character sets like UTF-8 as well based on the ASCII value of the character.

8.5.30 html_decode

Syntax:
html_decode ($expression, $encoding="")

Input parameters:
$expression ... variable as string or array
$encoding ... conversion of all special characters based on given character set (optional)

Output:
html decoded value as array or string / false on error

Description:
this function decodes all characters which have been converted by html_encode.

8.5.31 scriptcode_encode

Syntax:
scriptcode_encode ($content)

Input parameters:
$content ... content as string

global input parameters:
$mgmt_config

Output:
escaped content as string / false on error

Description:
this function escapes all script tags.
this function must be used to clean all user input in the CMS by removing all server side scripts tags.

8.5.32 scriptcode_extract

Syntax:
scriptcode_extract ($content, $identifier_start="<?", $identifier_end="?>")

Input parameters:
$content ... content as string
$identifier_start ... identifier of script begin
$identifier_end ... and end

Output:
script code as array / false on error or if noting was found

Description:
this function extracts the script code of a given content.

8.5.33 scriptcode_clean_functions

Syntax:
scriptcode_clean_functions ($content, $type=3, $application="PHP")

Input parameters:
$content ... content as string
$type ... cleaning level type from none = 0 to strong = 3 (no cleaning = 0
$application ... basic set of disabled functions = 1

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
this function removes all dangerous PHP functions.

8.5.34 url_encode

Syntax:
url_encode ($variable)

Input parameters:
$variable ... variable as string or array

global input parameters:
$mgmt_config

Output:
urlencoded value as array or string / false on error

Description:
this function encodes all characters.

8.5.35 url_decode

Syntax:
url_decode ($variable)

Input parameters:
$variable ... variable as string or array

global input parameters:
$mgmt_config

Output:
urldecoded value as array or string / false on error

Description:
this function decodes all characters which have been converted by url_encode or urlencode (PHP).

8.5.36 shellcmd_encode

Syntax:
shellcmd_encode ($variable)

Input parameters:
$variable ... variable as string or array

Output:
encoded value as array or string / false on error

Description:
this function encodes/escapes characters to secure the shell comand.

8.5.37 hcms_crypt

Syntax:
hcms_crypt ($string, $start=0, $length=0)

Input parameters:
$string ... string to encode
$start ... start position
$length ... length for string extraction

global input parameters:
$mgmt_config

Output:
encoded string / false on error

Description:
unidrectional encryption using crypt, MD5 and urlencode

8.5.38 hcms_encrypt

Syntax:
hcms_encrypt ($string, $key="", $crypt_level="", $encoding="url")

Input parameters:
$string ... string to encode
$key ... key of length 16 or 24 or 32 (optional)
$crypt_level ... crypt strength level [weak,standard,strong] (optional)
$encoding ... encoding [base64,url,none] (optional)

global input parameters:
$mgmt_config

Output:
encoded string / false on error

Description:
encryption of a string. only strong encryption is binary-safe!

8.5.39 hcms_decrypt

Syntax:
hcms_decrypt ($string, $key="", $crypt_level="", $encoding="url")

Input parameters:
$string ... hash-string to decode
$key ... key of length 16 or 24 or 32 (optional)
$crypt_level ... crypt strength level [weak,standard,strong] (optional)
$encoding ... encoding [base64,url,none] (optional)

global input parameters:
$mgmt_config

Output:
decoded string / false on error

Description:
decryption of a string. only strong encryption is binary-safe!

8.5.40 createtimetoken

Syntax:
createtimetoken ($lifetime=0, $secret=4)

Input parameters:
$lifetime ... token lifetime in seconds (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
token / false on error

8.5.41 checktimetoken

Syntax:
checktimetoken ($token, $secret=4)

Input parameters:
$token ... token
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
true / false

8.5.42 createtoken

Syntax:
createtoken ($user="sys", $lifetime=0, $secret=4)

Input parameters:
$user ... user name (optional)
$lifetime ... token lifetime in seconds (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
token / false on error

8.5.43 checktoken

Syntax:
checktoken ($token, $user="sys", $secret=4)

Input parameters:
$token ... token
$user ... user name (optional)
$secret ... secret value (optional)

global input parameters:
$mgmt_config

Output:
true / false

8.5.44 createuniquetoken

Syntax:
createuniquetoken ($length=16)

Input parameters:
$length ... token length (optional)

global input parameters:
$mgmt_config

Output:
token as string / false

8.5.45 rand_secure

Syntax:
rand_secure ($min=1000, $max=999999999999)

Input parameters:
$min ... min and max value as integer (optional)
$max

Output:
secure random number / false

8.6 Media API Functions

8.6.1 createthumbnail_indesign

Syntax:
createthumbnail_indesign ($site, $location_source, $location_dest, $file)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name

global input parameters:
$mgmt_config

Output:
new file name / false on error (saves only thumbnail media file in destination location
only jpeg format is supported as output)

Description:
creates a thumbnail by extracting the thumbnail from an indesign file and transferes the generated image via remoteclient.
note for good results, InDesign Preferences must be set to save preview image and at extra large size.

8.6.2 createthumbnail_video

Syntax:
createthumbnail_video ($site, $location_source, $location_dest, $file, $frame)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name
$frame ... frame of video in the seconds or hh:mm:ss[.xxx]

global input parameters:
$mgmt_config
$mgmt_mediapreview

Output:
new file name / false on error (saves only thumbnail media file in destination location only jpeg format is supported as output)

Description:
creates a thumbnail picture of a video frame

8.6.3 createmedia

Syntax:
createmedia ($site, $location_source, $location_dest, $file, $format="", $type="thumbnail", $force_no_encrypt=false)

Input parameters:
$site ... publication
$location_source ... path to source dir
$location_dest ... path to destination dir
$file ... file name
$format ... format (file extension w/o dot) (optional)
$type ... type of image/video/audio file [thumbnail(for thumbnails of images),origthumb(thumbnail made from original video/audio),original(to overwrite original video/audio file),any other string present in $mgmt_imageoptions/$mgmt_mediaoptions] (optional)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error (saves original or thumbnail media file in destination location for thumbnail only jpeg format is supported as output)

Description:
creates an new image from original or creates a thumbnail and transferes the generated image via remoteclient

8.6.4 convertmedia

Syntax:

convertmedia ($site, $location_source, $location_dest, $mediafile, $format, $media_config="", $force_no_encrypt=false)

Input parameters:
$site ... publication name
$location_source ... path to source dir
$location_dest ... path to destination dir
$mediafile ... file name
$format ... target format (file extension w/o dot) of destination file
$media_config ... media configuration to be used (optional)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error

Description:
converts and creates a new image/video/audio or document from original. this is a wrapper function for createmedia and createdocument.

8.6.5 convertimage

Syntax:
convertimage ($site, $file_source, $location_dest, $format="jpg", $colorspace="RGB", $iccprofile="", $width="", $height="", $slug=0, $units="px", $dpi=72, $quality="")

Input parameters:
$site ... publication name
$file_source ... path to source image file
$location_dest ... path to destination dir
$format ... format (file extension w/o dot) of destination file (optional)
$colorspace ... colorspace of new image
[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)
$iccprofile ... width in pixel/mm/inch (optional)
$width ... height in pixel/mm/inch (optional)
$height ... slug in pixel/mm/inch (optional)
$slug ... units for width
$units ... height and slug [px,mm,inch] (optional)
$dpi ... dpi (optional)
$quality ... image quality (1 to 100)

global input parameters:
$mgmt_config
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions

$mgmt_imageoptions
$mgmt_maxsizepreview
$mgmt_mediametadata
$hcms_ext

Output:
new file name / false on error

Description:
converts and creates a new image from original. the new image keeps will be resized and cropped to fit width and height.
this is a wrapper function for createmedia.

8.6.6 rotateimage

Syntax:
rotateimage ($site, $filepath, $angle, $imageformat)

Input parameters:
$site ... publication
$filepath ... path to source media file
$angle ... rotation angle
$imageformat ... destination image format [jpg,png,gif]

global input parameters:
$mgmt_config

Output:
new image file name / false on error

Description:
rotates an image (must be jpg, png or gif) using GD library. not used if ImageMagick is available.

8.6.7 getimagecolors

Syntax:
getimagecolors ($site, $file)

Input parameters:
$site ... publication
$file ... media file name

global input parameters:
$mgmt_config

Output:
result array / false on error

Description:
uses the thumbnail image to calculate the mean color (red, green, blue), defines the colorkey (5 most commonly used colors) and the image type (landscape, portrait, square)

8.6.8 getimagecolorkey

Syntax:
getimagecolorkey ($image)

Input parameters:
$image ... image resource

global input parameters:
$mgmt_config

Output:
color key of image / false on error

Description:
extracts the color key for an image that represents the 5 mostly used colors.
K...black
W...white
E...grey
R...red
G...green
B...blue
C...cyan
M...magenta
Y...yellow
O...orange
P...pink
N...brown

8.6.9 hex2rgb

Syntax:
hex2rgb ($hex)

Input parameters:
$hex ... image color as hex-code

Output:
RGB-color as array / false on error

8.6.10 rgb2hex

Syntax:
rgb2hex ($red, $green, $blue)

Input parameters:
$red ... image color in RGB
$green
$blue

Output:
hex-color as string / false on error

8.6.11 createdocument

Syntax:
createdocument ($site, $location_source, $location_dest, $file, $format="", $force_no_encrypt=false)

Input parameters:
$site ... publication
$location_source ... path to source location
$location_dest ... path to destination location

$file ... file name
$format ... destination file format (extension w/o dot)
$force_no_encrypt ... force the file to be not encrypted even if the content of the publication must be encrypted [true,false] (optional)

global input parameters:
$mgmt_config
$mgmt_docpreview
$mgmt_docoptions
$mgmt_docconvert
$mgmt_maxsizepreview
$hcms_ext
$hcms_lang
$lang

Output:
new file name / false on error

Description:
creates a new multimedia file of given format at source destination using UNOCONV and saves it as thumbnail file at the desitnation location

8.6.12 unzipfile

Syntax:
unzipfile ($site, $zipfilepath, $location, $filename, $user)

Input parameters:
$site ... publication
$zipfilepath ... path to source zip file
$location ... path to destination location
$filename ... name of file for extraction
$user ... user

global input parameters:
$mgmt_config
$mgmt_uncompress
$mgmt_imagepreview
$mgmt_mediapreview
$mgmt_mediaoptions

Output:
true/false

Description:
unpackes ZIP file and creates media files in destination location

8.6.13 zipfiles

Syntax:
zipfiles ($site, $multiobject_array, $destination="", $zipfilename, $user, $activity="")

Input parameters:
$site ... publication
$multiobject_array ... array with path to source zip files
$destination ... destination location (if this is null then the $location where the zip-file resists will be used)
$zipfilename ... name of ZIP-file

$user ... user name
$activity ... activity that need to be set for daily stats [download] (optional)

global input parameters:
$mgmt_config
$mgmt_compress
$pageaccess
$compaccess
$hiddenfolder
$hcms_linking
$globalpermission
$setlocalpermission

Output:
true/false

8.6.14 px2mm

Syntax:
px2mm ($pixel, $dpi=72)

Input parameters:
$pixel ... pixel
$dpi ... dpi (optional)

Output:
pixel / false

Description:
convert mm to pixel

8.6.15 px2inch

Syntax:
px2inch ($pixel, $dpi=72)

Input parameters:
$pixel ... pixel
$dpi ... dpi (optional)

Output:
inch / false

Description:
convert pixel to inches

8.6.16 inch2px

Syntax:
inch2px ($inch, $dpi=72)

Input parameters:
$inch ... pixel
$dpi ... dpi (optional)

Output:
pixel / false

Description:
convert inches to pixel

8.6.17 vtt2array

Syntax:
vtt2array ($vtt)

Input parameters:
$vtt ... VTT string

Output:
array / false

Description:
converts VTT string to array

8.7 Metadata API Functions

8.7.1 getkeywords

Syntax:
getkeywords ($text, $language="en", $charset="UTF-8")

Input parameters:
$text ... text as string
$language ... supported language [de,en]
$charset

global input parameters:
$mgmt_config

Output:
keywords sperated by
/false on error

Description:
generates a keyword list for meta information. supports german and english stop words lists.

8.7.2 getdescription

Syntax:
getdescription ($text, $charset="UTF-8")

Input parameters:
$text ... text as string
$charset

Output:
cleanded description of provided text /false on error

Description:
generates a keyword list for meta information. supports german and english stop words lists.

8.7.3 getgooglesitemap

Syntax:

getgooglesitemap ($site, $dir, $url, $getpara=array(), $permalink=array(), $chfreq="weekly", $prio="", $ignore=array(), $filetypes=array('cfm','htm','html','xhtml','asp','aspx','jsp','php','pdf'), $show_freq=true, $show_prio=true)

Input parameters:
$site ... publication anme
$dir ... directory path
$url ... URL to directory
$getpara ... GET parameters to use for new versions of the URL as array (optional)
$permalink ... permanent links text-ID to use for location as array (optional)
$chfreq ... frequency of google scrawler [never,weekly,daily] (optional)
$prio ... priority [1 or less] (optional)
$ignore ... ignore file names as array (optional)
$filetypes ... allowed file types as array (optional)
$show_freq ... include frequenzy tag [true,false] (optional)
$show_prio ... include priority tag [true,false] (optional)

global input parameters:
$mgmt_config
$publ_config

Output:
xml sitemap / false on error

Description:
generates a google sitemap xml-output.

8.7.4 getmetadata

Syntax:
getmetadata ($location, $object, $container="", $seperator="\n", $template="")

Input parameters:
$location ... location
$object ... object (both optional if container is given)
$container ... container name or container content (optional)
$seperator ... seperator of meta data fields [any string,array] (optional)
$template ... publication name/template name to extract label names (optional)

global input parameters:
$mgmt_config

Output:
string with all meta data from given object based on container

8.7.5 copymetadata

Syntax:
copymetadata ($file_source, $file_dest)

Input parameters:
$file_source ... path to source file
$file_dest ... path to destination file

global input parameters:
$user
$mgmt_config

$mgmt_mediametadata

Output:
true / false

Description:
copies all meta data from source to destination file using EXIFTOOL

8.7.6 extractmetadata

Syntax:
extractmetadata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata

Output:
result array / false on error

Description:
extracts all meta data from a file using EXIFTOOL

8.7.7 xmlobject2array

Syntax:
xmlobject2array ($obj, $namespace="")

Input parameters:
$obj ... XML as object
$namespace ... namespace as array (optional)

Output:
result array / false

Description:
function to convert an xmlobject to an array, provided by xaviered at gmail dot com

8.7.8 id3_getdata

Syntax:
id3_getdata ($file)

Input parameters:
$file ... path to audio file

global input parameters:
$mgmt_config
$hcms_ext

Output:
result array / false on error

Description:
requires getID3 library since EXIFTOOL cannot write ID3 tags so far

8.7.9 id3_writefile

Syntax:
id3_writefile ($file, $id3, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to audio file
$id3 ... ID3 tag array
$keep_data ... keep existing ID3 data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata
$hcms_ext

Output:
true / false on error

Description:
writes ID3 tags into audio file for supported file types and keeps the existing ID3 tags

8.7.10 id3_create

Syntax:
id3_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
ID3 tag array / false on error

Description:
defines ID3 tag array based on the media mapping of a publication.

8.7.11 xmp_getdata

Syntax:
xmp_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false on error

8.7.12 xmp_writefile

Syntax:
xmp_writefile ($file, $xmp, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to image file
$xmp ... XMP tag array
$keep_data ... keep existing XMP data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata
$hcms_ext

Output:
true / false on error

Description:
writes XMP tags into image file for supported file types and keeps the existing XMP tags

8.7.13 xmp_create

Syntax:
xmp_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
XMP tag array / false on error

Description:
defines XMP tag array based on the media mapping of a publication.

8.7.14 geo2decimal

Syntax:
geo2decimal ($deg, $min, $sec, $hemi)

Input parameters:
$deg ... geo location in degree
$min ... minutes
$sec ... seconds
$hemi ... hemisphere [N,O,S,W]

Output:
decimal result / false

8.7.15 exif_getdata

Syntax:
exif_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false

8.7.16 iptc_getdata

Syntax:
iptc_getdata ($file)

Input parameters:
$file ... path to image file

global input parameters:
$user
$mgmt_config
$hcms_ext

Output:
result array / false

8.7.17 iptc_getcharset

Syntax:
iptc_getcharset ($tag)

Input parameters:
$tag ... iptc tag that holds character set information

Output:
charset as string / false on error

Description:
Copied from MediaWiki!
Warning, this function does not (and is not intended to) detect all iso 2022 escape codes.
In practise, the code for utf-8 is the only code that seems to have wide use. It does detect that code.
According to iim standard, charset is defined by the tag 1:90.
in which there are iso 2022 escape sequences to specify the character set.
the iim standard seems to encourage that all necessary escape sequences are
in the 1:90 tag, but says it doesn't have to be.
This is in need of more testing probably. This is definitely not complete.
however reading the docs of some other iptc software, it appears that most iptc software only recognizes utf-8. If 1:90 tag is not present content is
usually ascii or iso-8859-1 (and sometimes utf-8), but no guarantee.
This also won't work if there are more than one escape sequence in the 1:90 tag

or if something is put in the G2, or G3 charsets, etc. It will only reliably recognize utf-8. This is just going through the charsets mentioned in appendix C of the iim standard.

8.7.18 iptc_maketag

Syntax:
iptc_maketag ($record=2, $tag, $value)

Input parameters:
$record ... type of tag (e.g. 2)
$tag ... code of tag (e.g. 025)
$value ... value of tag

Output:
binary IPTC tag / false on error

Description:
convert the IPTC tag into binary code.

8.7.19 iptc_writefile

Syntax:
iptc_writefile ($file, $iptc, $keep_data=true, $movetempfile=true)

Input parameters:
$file ... abs. path to image file
$iptc ... IPTC tag array
$keep_data ... keep existing IPTC data of file [true,false] (optional)
$movetempfile ... move tempoarary file from unecrypted to encrypted [true,false] (optional)

global input parameters:
$user
$mgmt_config
$mgmt_mediametadata

Output:
true / false on error

Description:
writes IPTC tags into image file for supported file types and keeps the existing IPTC tags

8.7.20 iptc_create

Syntax:
iptc_create ($site, $text)

Input parameters:
$site ... publication name
$text ... text array (from content container)

global input parameters:
$mgmt_config

Output:
IPTC tag array / false on error

Description:
defines IPTC tag array based on the medai mapping of a publication.

8.7.21 createmapping

Syntax:
createmapping ($site, $mapping)

Input parameters:
$site ... publication name
$mapping ... mapping definition

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
prepares the PHP mapping array from the provided mapping definition and saves media mapping file

8.7.22 getmapping

Syntax:
getmapping ($site)

Input parameters:
$site ... publication name

global input parameters:
$mgmt_config

Output:
mapping code for display / false

Description:
loads the mapping file of the provided publication.

8.7.23 setmetadata

Syntax:
setmetadata ($site, $location="", $object="", $mediafile="", $mapping="", $user)

Input parameters:
$site ... publication name
$location ... location path (optional)
$object ... object name (optional)
$mediafile ... media file name (optional)
$mapping ... mapping array (meta data tag name -> text-id
$user ... optional)

global input parameters:
$eventsystem
$mgmt_config
$hcms_ext

Output:
true/false

Description:

saves meta data of a multimedia file using a provided mapping in the proper fields of the content container.
if no mapping is given a default mapping will be used.

8.8 Link API Functions

8.8.1 link_db_restore

Syntax:
link_db_restore ($site="")

Input parameters:
$site ... publication name (optinal)

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
this function restores a given or all link management index files

8.8.2 link_db_load

Syntax:
link_db_load ($site, $user)

Input parameters:
$site ... site
$user ... user

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function loads and locks the link management database
each record of the link management database has the following design:
xml-content container :| absolute path to 1-n objects :| 1-m links used by 1-n objects
important: the link management database has to saved or closed after loading it.

8.8.3 link_db_read

Syntax:
link_db_read ($site)

Input parameters:
$site ... site

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function loads the link management database for reading without locking

8.8.4 link_db_close

Syntax:
link_db_close ($site, $user)

Input parameters:
$site ... site
$user ... user

global input parameters:
$mgmt_config

Output:
true/false

Description:
closes and unlocks the link management database.

8.8.5 link_db_save

Syntax:
link_db_save ($site, $link_db, $user)

Input parameters:
$site ... link database array
$link_db ... site
$user ... user

global input parameters:
$mgmt_config

Output:
true/false on error

Description:
this function saves und unlocks the link management database

8.8.6 link_db_update

Syntax:
link_db_update ($site, $link_db, $attribute, $contentfile, $cat, $link_curr, $link_new, $option)

Input parameters:
$site ... publication name
$link_db ... link database [2 dim. array]
$attribute ... attribute ['object'
$contentfile ... 'link']
$cat ... content container [optional] [string]
$link_curr ... link category [optional] ['comp'
$link_new ... 'page']
$option ... current link [optional

global input parameters:

$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function inserts, updates and removes objects and their links from the link management database (add or update a link)
depending on which link is left empty:
link_curr = "": add new link (just one link matching given category!)
link_new = "": delete current link in use (just one linkm matching given category!)
link_curr & link_new are not empty: update current link with the new one

8.8.7 link_db_insert

Syntax:
link_db_insert ($site, $link_db, $contentfile, $cat, $object)

Input parameters:
$site ... publication name
$link_db ... link database [2 dim. array]
$contentfile ... content container
$cat ... link category ['comp
$object ... page']

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false

Description:
this function inserts a new record in the link management database
optionally the created object can be also inserted

8.8.8 link_db_delete

Syntax:
link_db_delete ($site, $link_db, $contentfile)

Input parameters:
$site ... link database [2 dim. array]
$link_db ... content container
$contentfile

global input parameters:
$mgmt_config

Output:
link database [2 dim. array] or true / false on error

Description:
this function deletes a record in the link management database

8.8.9 link_db_getobject

Syntax:
link_db_getobject ($multiobject)

Input parameters:
$multiobject ... link database attribut (references to objects seperated by |)

global input parameters:
$mgmt_config

Output:
objects [array] / false on error

Description:
this function splits the object string into an array of objects.

8.8.10 link_update

Syntax:
link_update ($site, $container, $link_old, $link_new)

Input parameters:
$site ... publication name
$container ... container name
$link_old ... old link (converted)
$link_new ... new link (converted)

global input parameters:
$user
$mgmt_config

Output:
true/false

Description:
this function updates the link of the published and working content container and link file

8.8.11 getlinkedobject

Syntax:
getlinkedobject ($site, $location, $page, $cat)

Input parameters:
$site ... publication
$location ... location
$page ... object (name and extension)
$cat ... category [page

global input parameters:
$mgmt_config

Output:
objects which link to the given object [array] or true / false

Description:
this function gets all objects which link to the given object.
works with pages (page links) and components (component links) if link management is enabled.

8.8.12 getconnectedobject

Syntax:
getconnectedobject ($container, $type="work")

Input parameters:
$container ... container name
$type ... container type [work,published,version] (optional)

global input parameters:
$mgmt_config
$user

Output:
connected objects[array]

Description:
this function gets all objects which use the same content container and are therefore connected.

8.8.13 extractlinks

Syntax:
extractlinks ($textcontent, $identifier)

Input parameters:
$textcontent ... text content as string
$identifier ... link identifiert ("href" for hyperreferences

global input parameters:
$mgmt_config

Output:
object links [array] / false on error

Description:
this function extracts all links based on it's identifier from a text and returns an array of all links

8.9 Plugin API Functions

8.9.1 plugin_getdefaultconf

Syntax:
plugin_getdefaultconf ()

Input parameters:

Output:
default value as array

8.9.2 plugin_readmenu

Syntax:
plugin_readmenu ($xml, $pluginFolder)

Input parameters:

$xml ... plugin xml as string
$pluginFolder ... plugin directory

global input parameters:
$mgmt_config

Output:
menu point array used by navigator

Description:
Reads Menupoints and menugroups from the xml data
be carefull with nesting, getcontent is used here and you can't nest groups inside of groups as a subpoint!
pluginFolder contains the folder this plugin is located in, that is needed for the icons and the links
returns an Array containing every group and menupoint with their configuration

8.9.3 plugin_parse

Syntax:
plugin_parse ($oldData=array())

Input parameters:
$oldData ... mgmt_plugin as array (optional)

global input parameters:
$mgmt_config

Output:
mgmt_plugin as array

Description:
Reads the plugin configurations from the file system.
Checks the folder defined in mgmt_config and searched for plugins and their configurations files.
It either takes needed values from the configuration, from the $oldData or defaultConfiguration.

8.9.4 plugin_generatedefinition

Syntax:
plugin_generatedefinition ($arrayName, $array)

Input parameters:
$arrayName ... name of array holding the plugin definitions
$array ... configuration array

global input parameters:
$mgmt_config

Output:
plugin array / false on error

Description:
Generates the Array definition used in php for $array with the name of $arrayName
Run recursively through the array and supports boolean, numeric and string types for the key and value
$arrayName -> Name of the Array, $array the array containing the values and keys

8.9.5 plugin_saveconfig

Syntax:
plugin_saveconfig ($configuration)

Input parameters:
$configuration ... configuration as array

global input parameters:
$mgmt_config

Output:
true / false on error

Description:
Saves the plugin configuration $configuration into the configuration file
The configuration file is located in the data/config directory and is named plugin.conf.php

8.9.6 plugin_generatelink

Syntax:
plugin_generatelink ($plugin, $page, $control=false, $additionalGetParameters=false)

Input parameters:
$plugin ... plugin name
$page ... plugin page (relative reference to the plugins main page)
$control ... control (relative reference to the plugins control page)
$additionalGetParameters ... additional GET parameters

global input parameters:
$mgmt_config

Output:
plugin link

Description:
Generates a link to be used when linking to other pages inside of a plugin.

8.10 User Interface API Functions

8.10.1 toggleview

Syntax:
toggleview ($view)

Input parameters:
$view ... view [detail,small,medium,large]

global input parameters:
$mgmt_config

Output:
true / false

Description:
sets explorer objectlist view.

8.10.2 togglesidebar

Syntax:
togglesidebar ($view)

Input parameters:
$view ... view [true,false]

global input parameters:
$mgmt_config

Output:
true / false

Description:
enables or disables sidebar

8.10.3 setfilter

Syntax:
setfilter ($filter_set)

Input parameters:
$filter_set ... set of filtera as array with keys [comp,image,document,video,audio] and value [0,1]

global input parameters:
$mgmt_config

Output:
true / false

Description:
set filter settings for object view in session.

8.10.4 objectfilter

Syntax:
objectfilter ($file)

Input parameters:
$file ... file name

global input parameters:
$mgmt_config
$hcms_ext

Output:
true / false

Description:
if an object / file name is passing the filter-test.

8.10.5 showtopbar

Syntax:
showtopbar ($show, $lang="en", $close_link="", $close_target="", $individual_button="",

$id="bar")

Input parameters:
$show ... message
$lang ... language code (optional)
$close_link ... close button link (optional)
$close_target ... link target (optional)
$individual_button ... individual button (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
top bar box / false on error

Description:
shows the standard top bar with or without close button

8.10.6 showtopmenubar

Syntax:
showtopmenubar ($show, $menu_array, $lang="en", $close_link="", $close_target="", $id="bar")

Input parameters:
$show ... message
$menu_array ... menu as array [key=name
$lang ... value=properties/events]
$close_link ... language code (optional)
$close_target ... close button link (optional)
$id ... link target (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
top bar box / false on error

Description:
shows the menu top bar with or without close button

8.10.7 showmessage

Syntax:
showmessage ($show, $width=580, $height=70, $lang="en", $style="", $id="hcms_messageLayer")

Input parameters:
$show ... message
$width ... width in pixel (optional)
$height ... height in pixel (optional)
$lang ... language code (optional)
$style ... additional style definitions of div-layer (optional)

$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

Output:
message box / false on error

Description:
shows the standard message box with close button

8.10.8 showinfopage

Syntax:
showinfopage ($show, $lang="en")

Input parameters:
$show ... message
$lang ... language code (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message on html info page / false on error

Description:
shows a full html info page

8.10.9 showinfobox

Syntax:
showinfobox ($show, $lang="en", $sec=4, $style="", $id="hcms_infoLayer")

Input parameters:
$show ... message
$lang ... language code (optional)
$sec ... display for seconds (optional)
$style ... additional style definitions of div-layer (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message in div layer / false on error

Description:
shows infobox for a few seconds

8.10.10 showsharelinks

Syntax:
showsharelinks ($link, $lang="en", $style="", $id="hcms_shareLayer")

Input parameters:
$link ... link to share
$lang ... language code (optional)
$style ... additional style definitions of div-layer (optional)
$id ... ID of div-layer (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
message in div layer / false on error

Description:
shows share links

8.10.11 showmetadata

Syntax:
showmetadata ($data, $lang="en", $class_headline="hcmsRowData2")

Input parameters:
$data ... meta data as array
$lang ... hierarchy level
$class_headline ... CSS-class with background-color for headlines (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang_codepage
$hcms_lang

Output:
result as HTML unordered list / false on error

8.10.12 showobject

Syntax:
showobject ($site, $location, $page, $cat="", $name="")

Input parameters:
$site ... publication name
$location ... location
$page ... object name
$cat ... category [page,comp] (optional)
$name ... object name (optional)

global input parameters:
$mgmt_config
$hcms_charset
$hcms_lang

$lang

Output:
html presentation / false

8.10.13 showmedia

Syntax:
showmedia ($mediafile, $medianame, $viewtype, $id="", $width="", $height="", $class="hcmsImageItem")

Input parameters:
$mediafile ... mediafile (publication/filename)
$medianame ... name of mediafile for display
$viewtype ... view type [template
$id ... preview
$width ... preview_download
$height ... preview_no_rendering]
$class ... ID of the media tag

global input parameters:
$mgmt_config
$mgmt_mediapreview
$mgmt_mediaoptions
$mgmt_imagepreview
$mgmt_docconvert
$hcms_charset
$hcms_lang_codepage
$hcms_lang
$lang
$pdfjs_path = $mgmt_config['url_path_cms']."javascript/pdfpreview/web/viewer.html?file="

Output:
html presentation / false

Description:
this function requires site, location and cat to be set as global variable in order to validate the access permission of the user

8.10.14 showcompexplorer

Syntax:
showcompexplorer ($site, $dir, $location_esc="", $page="", $compcat="multi", $search_expression="", $search_format="", $mediatype="", $lang="en", $callback="", $scalingfactor="1")

Input parameters:
$site ... publication name
$dir ... current explorer location
$location_esc ... object location (optional)
$page ... object name (optional)
$compcat ... component category [single,multi,media] (optional)
$search_expression ... search expression (optional)
$search_format ... search format [object,document,image,video,audio] (optional)
$mediatype ... media-type [audio,video,text,flash,image,compressed,binary] (optional)
$lang ... callback of CKEditor (optional)
$callback ... saclingfactor for images (optional)
$scalingfactor

global input parameters:
$user
$mgmt_config
$siteaccess
$pageaccess
$compaccess
$rootpermission
$globalpermission
$localpermission
$hiddenfolder
$html5file
$temp_complocation
$hcms_charset
$hcms_lang

Output:
explorer with search / false on error

Description:
creates component explorer incl. search form

8.10.15 showeditor

Syntax:
showeditor ($site, $hypertagname, $id, $contentbot="", $sizewidth=600, $sizeheight=300, $toolbar="Default", $lang="en", $dpi=72)

Input parameters:
$site ... publication name
$hypertagname ... hypertag name
$id ... hypertag id
$contentbot ... content
$sizewidth ... width
$sizeheight ... height of the editor
$toolbar ... toolbar set
$lang ... language
$dpi ... dpi for scaling images

global input parameters:
$mgmt_config
$publ_config

Output:
rich text editor code / false on error

Description:
shows the rich text editor

8.10.16 showinlineeditor_head

Syntax:
showinlineeditor_head ($lang)

Input parameters:
$lang ... language

global input parameters:

$mgmt_config
$hcms_charset
$hcms_lang

Output:
rich text editor code for html head section / false on error

Description:
shows the rich text editor code (JS, CSS) for include into the html head section

8.10.17 showinlinedatepicker_head

Syntax:
showinlinedatepicker_head ()

Input parameters:

global input parameters:
$mgmt_config

Output:
date picker code for html head section / false on error

Description:
shows the date picker code (JS, CSS) for include into the html head section

8.10.18 showinlineeditor

Syntax:
showinlineeditor ($site, $hypertag, $id, $contentbot="", $sizewidth=600, $sizeheight=300, $toolbar="Default", $lang="en", $contenttype="", $cat="", $location_esc="", $page="", $contentfile="", $db_connect=0, $token="")

Input parameters:
$site ... publication name
$hypertag ... hypertag
$id ... hypertag id
$contentbot ... content
$sizewidth ... width
$sizeheight ... height of the editor
$toolbar ... toolbar set
$lang ... language
$contenttype ... content-type
$cat ... category[page,comp]
$location_esc ... converted location
$page ... object name
$contentfile ... container name
$db_connect ... DB-connect file name
$token ... security token

global input parameters:
$mgmt_config
$publ_config
$hcms_charset
$hcms_lang

Output:
message box/false on error

Description:
shows the rich text inline editor

8.10.19 showvideoplayer

Syntax:
showvideoplayer ($site, $video_array, $width=320, $height=240, $logo_url="", $id="", $title="", $autoplay=true, $fullscreen=true, $loop=false, $muted=false, $controls=true, $iframe=false, $force_reload=false)

Input parameters:
$site ... videoArray (Array) containing the different html sources
$video_array ... width (Integer) Width of the video in pixel
$width ... height (Integer) Height of the video in pixel
$height ... logo_url (String) Link to the logo which is displayed before you click on play (If the value is null the default logo will be used)
$logo_url ... id (String) The ID of the video (will be generated when empty)
$id ... title (String) The title for this video
$title ... autoplay (Boolean) Should the video be played on load (true)
$autoplay ... default is false
$fullscreen ... enableFullScreen (Boolean) Is it possible to view the video in fullScreen (true)
$loop ... play loop (optional) [true,false]
$muted ... muted/no sound (optional) [true,false]
$controls ... player controls (optional) [true,false]
$iframe ... use video in iframe (optional) [true,false]
$force_reload ... reload video sources to prevent the browser cache to show the same video even if it has been changed [true,false] (optional)

global input parameters:
$mgmt_config

Output:
HTML code of the video player / false on error

Description:
generates a html segment for the video code we use.

8.10.20 showvideoplayer_head

Syntax:
showvideoplayer_head ($secureHref=true, $fullscreen=true)

Input parameters:
$secureHref ... secure hyperreferences by adding 'hypercms_'
$fullscreen ... is it possible to view the video in fullScreen [true,false]

global input parameters:
$mgmt_config

Output:
head for video player / false on error

8.10.21 showaudioplayer

Syntax:
showaudioplayer ($site, $audioArray, $width=320, $height=320, $logo_url="", $id="", $autoplay=false, $loop=false, $controls=true, $force_reload=false)

Input parameters:
$site ... publication name
$audioArray ... audio files as array (Array)
$width ... ID of the tag (optional)
$height ... autoplay (optional) [true,false]
$logo_url ... play loop (optional) [true,false]
$id ... player controls (optional) [true,false]
$autoplay
$loop
$controls
$force_reload

global input parameters:
$mgmt_config

Output:
String Code for the HTML

Description:
Generates a html segment for the video code we use. False on error.

8.10.22 showaudioplayer_head

Syntax:
showaudioplayer_head ($secureHref=true)

Input parameters:
$secureHref ... secure hyperreferences by adding 'hypercms_'

global input parameters:
$mgmt_config

Output:
head for video player

8.10.23 debug_getbacktracestring

Syntax:
debug_getbacktracestring ($valueSeparator, $rowSeparator, $ignoreFunctions=array())

Input parameters:
$valueSeparator ... separator for arguments
$rowSeparator ... separator for a Row on screen/file
$ignoreFunctions ... functionnames to be ignored

Output:
debug message

Description:
Returns the current backtrace as a good readable string
ignores debug and debug_getbacktracestring

8.10.24 showAPIdocs

Syntax:
showAPIdocs ($file, $return="html")

Input parameters:
$file ... path to API file
$return ... return result as HTML or array [html,array] (optional)

global input parameters:
= array()

Output:
HTML output of documentation / false on error

Description:
generates the documentation of an API file

8.11 Template Engine API Functions

8.11.1 checklanguage

Syntax:
checklanguage ($language_array, $language_value)

Input parameters:
$language_array ... language array with all valid values
$language_value ... language value of attribute in hyperCMS tag

Output:
true if language array holds the given language value / false if not found

8.11.2 checkgroupaccess

Syntax:
checkgroupaccess ($groupaccess, $ownergroup)

Input parameters:
$groupaccess ... group access string from hyperCMS group-tag attribute
$ownergroup ... owner groups as array

Output:
true if current ownergroup has access or invalid input / false if not

8.11.3 transformlink

Syntax:
transformlink ($viewstore)

Input parameters:
$viewstore ... view of object

global input parameters:
$site
$location_esc
$page
$ctrlreload
$mgmt_config

Output:
view with transformed links inside publication for easyedit mode

8.11.4 followlink

Syntax:
followlink ($site, $follow)

Input parameters:
$site ... publication name
$follow ... link to follow

global input parameters:
$mgmt_config

Output:
prepared input (location plus page) for easyedit mode (buildview) / false on error

8.11.5 errorhandler

Syntax:
errorhandler ($source_code, $return_code, $error_identifier)

Input parameters:
$source_code ... source code
$return_code ... return code
$error_identifier ... error identifier

Output:
error message and view of the code with line identifiers

8.11.6 viewinclusions

Syntax:
viewinclusions ($site, $viewstore, $hypertag, $view, $application, $charset="UTF-8")

Input parameters:
$site ... view of object
$viewstore ... hypertag to create view of inlcuded objects
$hypertag ... view parameter
$view ... application
$application ... character set used (optional) view-parameter explanation: $view = "template or any other word" -> the standard text (in table) will be included for the view $view = "preview" -> preview of the content of the included file $view = "publish" -> view the content of the included file as ist is (for publishing)
$charset

global input parameters:
$user
$mgmt_config
$location
$hcms_lang
$lang

Output:
view on the content including the content of included objects

8.11.7 buildview

Syntax:

buildview ($site, $location, $page, $user, $buildview="template", $ctrlreload="no", $template="", $container="", $force_cat="", $execute_code=true)

Input parameters:
$site ... publication name
$location ... location
$page ... object
$user ... user
$buildview ... view parameter (optional)
$ctrlreload ... reload workplace control frame and add html & body tags if missing [yes,no] (optional)
$template ... template name (optional)
$container ... container name (optional)
$force_cat ... force category to use different location path [page,comp] (optional)
$execute_code ... execute_code [true/false] (optional)

global input parameters:
$container_collection
$eventsystem
$db_connect
$mgmt_config
$siteaccess
$adminpermission
$setlocalpermission
$token
$mgmt_lang_shortcut_default
$hcms_charset
$hcms_lang_name
$hcms_lang_shortcut
$hcms_lang_codepage
$hcms_lang_date
$hcms_lang
$lang

Output:
result array with view of the content / false on error

Description:
buildview parameter may have the following values:
$buildview = "formedit": use form for content editing
$buildview = "formmeta": use form for content viewing only for meta informations (tag-type must be meta)
$buildview = "formlock": use form for content viewing
$buildview = "cmsview": view of page based on template, includes hyperCMS specific code (buttons)
$buildview = "inlineview": view of page based on template, includes hyperCMS specific code (buttons) and inline text editing
$buildview = "publish": view of page for publishing based on template without CMS specific code (buttons)
$buildview = "preview": view of page based on template for preview (inactive hyperlinks) without CMS specific code (buttons)
$buildview = "template": view of template based on template for preview (inactive hyperlinks) without CMS specific code (buttons)

8.11.8 buildsearchform

Syntax:
buildsearchform ($site, $template, $ownergroup="")

Input parameters:
$site ... publication name
$template ... template name
$ownergroup ... group access as array

global input parameters:
$user
$mgmt_config
$mgmt_lang_shortcut_default
$hcms_charset
$hcms_lang_name
$hcms_lang_shortcut
$hcms_lang_codepage
$hcms_lang_date
$hcms_lang
$lang

Output:
form view

8.11.9 buildbarchart

Syntax:
buildbarchart ($paper_name, $paper_width=600, $paper_height=300, $paper_top=10, $paper_left=40, $x_axis, $y1_axis, $y2_axis="", $y3_axis="", $paper_style="", $bar1_style="", $bar2_style="", $bar3_style="", $show_value=false)

Input parameters:
$paper_name ... name/id of paper
$paper_width ... width of paper in pixel
$paper_height ... height of paper in pixel
$paper_top ... top space in pixel
$paper_left ... left space in pixel
$x_axis ... x-axis values as array
$y1_axis ... y1-axis values as array
$y2_axis ... y2-axis values as array (optional)
$y3_axis ... y3-axis values as array (optional)
$paper_style ... paper CSS style
$bar1_style ... 1st bar chart CSS style
$bar2_style ... 2nd bar chart CSS style
$bar3_style ... 3rd bar chart CSS style
$show_value ... show y-value in bar [true,false]

global input parameters:
$lang
$mgmt_config

Output:
bar chart view / false on error

8.12 XML API Functions

8.12.1 setxmlparameter

Syntax:
setxmlparameter ($xmldata, $parameter, $value)

Input parameters:
$xmldata ... XML content container
$parameter ... paramater name
$value ... paramater value

Output:
XML content container / false on error

Description:
set parameter values in XML declaration (e.g. encoding):
encoding="UTF-8"

8.12.2 getcontent

Syntax:
getcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname.

8.12.3 geticontent

Syntax:
geticontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are however always case-sensitive!)
<tagname>content</tagname>
extracts the content between the given $starttagname xml-tags.
only this function will decode special characters (&, <, >) in the content and removes CDATA.
getcontent will only decode values if they are non-xml and non_html. so content inside child nodes
including tags won't be decoded.
wild card character "*" can be used at the end of $starttagname

8.12.4 getxmlcontent

Syntax:

getxmlcontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.12.5 getxmlicontent

Syntax:
getxmlicontent ($xmldata, $starttagname)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>content</tagname>
extracts the content together with the $starttagname xml tags
this function will NOT decode special characters like function getcontent!
wild card character "*" can be used at the end of $starttagname

8.12.6 selectcontent

Syntax:
selectcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname

wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.12.7 selecticontent

Syntax:
selecticontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at the end of $starttagname
wild card character "*" can be used at begin and end of $condvalue

8.12.8 selectxmlcontent

Syntax:
selectxmlcontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$starttagname ... tag name of requested XML node
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
<tagname>
....
<condtag>condvalue</condtag>
....
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue
Be Aware: $startcondtag must be a child of $starttagname !!!

8.12.9 selectxmlicontent

Syntax:
selectxmlicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata … XML content container
$starttagname … tag name of requested XML node
$startcondtag … tag holding the conditional value inside the given starttagname
$condvalue … conditional value

Output:
result array with the content of the requested XML node (tag) / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
…….
<condtag>condvalue</condtag>
…….
</tagname>
extracts the content between the given $starttagname xml tags where the child xml tag $startcondtag
value is equal with the target value $condvalue
wild card character "*" can be used at begin and end of $condvalue

8.12.10 deletecontent

Syntax:
deletecontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata … XML content container
$starttagname … tag name of requested XML node
$startcondtag … tag holding the conditional value inside the given starttagname
$condvalue … conditional value

Output:
XML content container / false on error

Description:
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.12.11 deleteicontent

Syntax:
deleteicontent ($xmldata, $starttagname, $startcondtag, $condvalue)

Input parameters:
$xmldata … XML content container
$starttagname … tag name of requested XML node
$startcondtag … tag holding the conditional value inside the given starttagname
$condvalue … conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<tagname>
<condtag>condvalue</condtag>
</tagname>
deletes the whole xml content including <tagname>
wild card character "*" can be used at begin and end of $condvalue

8.12.12 setcontent

Syntax:
setcontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is a parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.13 seticontent

Syntax:
seticontent ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag, $condvalue)

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.14 setcontent_fast

Syntax:
setcontent_fast ($xmldata, $startparenttagname, $starttagname, $contentnew, $startcondtag="", $condvalue="")

Input parameters:
$xmldata ... XML content container
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value

Output:
XML content container / false on error

Description:
function designed for link management, extremely fast but with limitations (only CASE-Sensitive!)
<parenttagname>
<condtag>condvalue</condtag>
<tagname>contentnew</tagname>
</parenttagname>
$xmldata = data string to be parsed
$startparenttagname = name of the tag that is the parent node of starttagname (necessary if condition has been set!)
$starttagname = name of the tag (child node)
$contentnew = the content that will be inserted between the child tags $starttagname
$startcondtag = child xml tag where condition will be set
$condvalue = value of the condition
wild card character "*" can be used at begin and end of $condvalue

8.12.15 updatecontent

Syntax:
updatecontent ($xmldata, $xmlnode, $xmlnodenew)

Input parameters:
$xmldata ... XML content container
$xmlnode ... XML node to be replaced
$xmlnodenew ... new XML node

Output:
XML content container / false on error

Description:
updates a given xml string $xmlnode in $xmldata with the content $xmlnodenew.
this method provides a faster way to update xml nodes when the node was selected before.

8.12.16 insertcontent

Syntax:
insertcontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata … XML content container
$insertxmldata … XML node to be inserted in starttagname
$starttagname … tag name of the parent XML node

Output:
XML content container / false on error

Description:
…………………
………………….
<tagname> <- list start
…………………
…………………
insertxmldata <- insertxmldata
</tagname> <- list end
…………………
insert $insertxmldata string at the end of all child between the parent $tagname

8.12.17 inserticontent

Syntax:
inserticontent ($xmldata, $insertxmldata, $starttagname)

Input parameters:
$xmldata … XML content container
$insertxmldata … XML node to be inserted in starttagname
$starttagname … tag name of the parent XML node

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)
…………………
………………….
<tagname> <- list start
…………………
…………………
insertxmldata <- insertxmldata
</tagname> <- list end
…………………
insert $insertxmldata string at the end of all child between the parent $tagname

8.12.18 addcontent

Syntax:
addcontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
```
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
......................
......................
...................... }
<tagname>contentnew</tagname> } <- sub_xmldata
...................... }
</parenttagname> <- list end
</grandtagname>
```
$xmldata = data string to be parsed
$sub_xmldata = xml node to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

8.12.19 addicontent

Syntax:
addicontent ($xmldata, $sub_xmldata, $startgrandtagname, $startcondtag, $condvalue, $startparenttagname, $starttagname, $contentnew)

Input parameters:
$xmldata ... XML content container
$sub_xmldata ... xml node to be inserted
$startgrandtagname ... grandparent tag name
$startcondtag ... tag holding the conditional value inside the given starttagname
$condvalue ... conditional value
$startparenttagname ... parent tag name
$starttagname ... tag name of XML node for the new content
$contentnew ... new XML node to be inserted

Output:
XML content container / false on error

Description:
CASE-Insensitive version (XML parser are always case-sensitive!)

```
<grandtagname>
<condtag>condvalue</condtag>
<parenttagname> <- list start
......................
......................
...................... }
<tagname>contentnew</tagname> } <- sub_xmldata
...................... }
</parenttagname> <- list end
</grandtagname>
```

$xmldata = data string to be parsed
$sub_xmldata = xml subschema to be inserted
$startgrandtagname (optional) = name of the grand xml tag of parent xml tag where (article)
$startcondtag (optional) = xml tag inside the parent xml tags where condition will be set
$condvalue (optional) = value of the condition
$startparenttagname (optional) = name of the parent xml tag where the xml subschema should be added (list)
$starttagname (optional) = name of the tag (child)
$contentnew (optional) = the content that will be inserted between the child tags

10Legal reference / flag

10.1 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
http://www.hypercms.com

10.2 Imprint

Responsible for the content:

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

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

10.3 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 accepted.