hyperCMS ContentManagementServer

Version 5.6

Connector Guide

2014-06-05

Inhalt

1 OpenSearch

1.1 Introduction

OpenSearch is a collection of technologies that allow publishing of search results in a format suitable for syndication and aggregation. It is a way for websites and search engines to publish search results in a standard and accessible format.

OpenSearch mainly consists of two main components:

1. OpenSearch description files: XML files that identify and describe a search engine.
2. OpenSearch response: providing open search results.

More detailed information about the OpenSearch standard could be found at http://www.opensearch.org

1.2 OpenSearch Description Document

The OpenSearch Description Document is an XML-file which describes the search-service in detail. It contains information about the service-provider, attribution, adult content, syndication right, language and much more. The most valuable information is delivered by the "Url" element, which describes an interface by which a client can make requests for an external resource, such as search results, search suggestions, or additional description documents.

1.2.1 The "Url" element

As mentioned above this element characterizes how clients can access the search-service and what kind response will be delivered. One can reach the service via URL and define the search via GET parameter. The URL and the GET parameters are described in the template attribute of the "Url" element.

Here is a simple example:

```
<Url  type="text/html"
      indexOffset="0"
      pageOffset="0"
      rel="results"
      template="http://yourdomain.com/opensearch/?action=base_search&amp;search_expression={searchTerms}&amp;startPage={startPage?}&amp;count={count?}"
/>
```

The example shows that the template attribute contains the URL to the service "**http://yourdomain.com/opensearch/**" and the GET Parameters with placeholders for their corresponding value "**?action=base_search&search_expression={searchTerms}&startPage={startPage?}&count={count?}**".

1.2.2 Fetching the OpenSearch Description Document

Calling the URL to the service with a GET Parameter "desc" with a value "1", returns the OpenSearch Description Document:

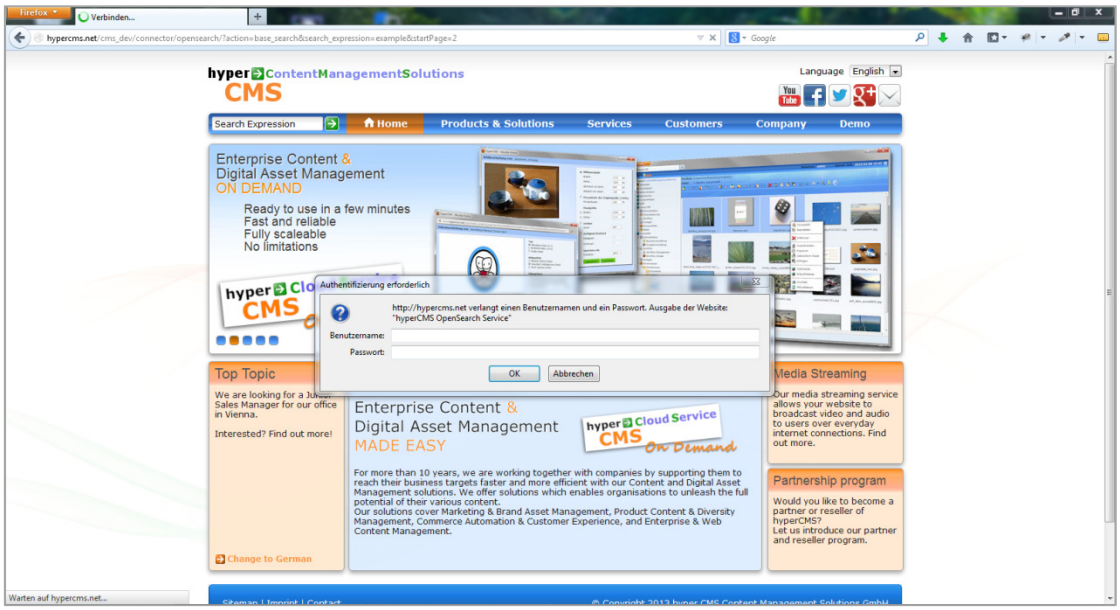
http://yourdomain.com/opensearch/?desc=1

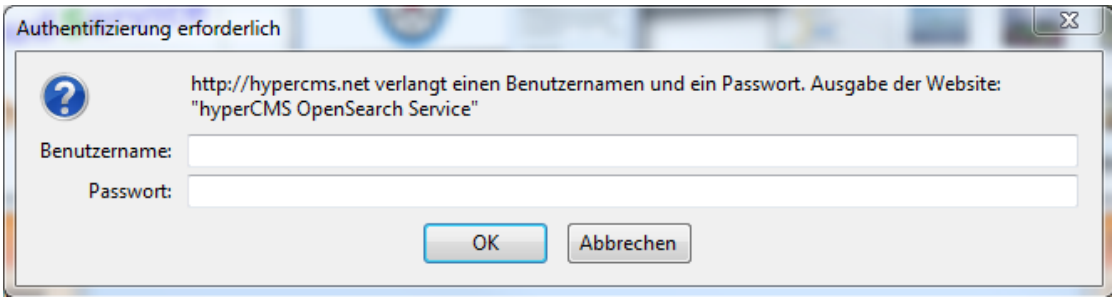
1.3 Accessing the service

As already mentioned in chapter 2.1 the search-service is reachable via a simple URL and the search-query is specified via GET parameters. Considering the "Url" element example in chapter 2.1 such a query-URL can look like this:

http://yourdomain.com/opensearch/?action=base_search&search_expression=example&startPage=2

The search-service is secured via HTTP Basic Authentication; therefore a search can be triggered only with a valid username and password combination.





1.4 Different options to use the search-service

1.4.1 Object based search

In case of known object-ID, it is possible to retrieve the corresponding object directly.

Parameters:

object_id – unique ID or hash of a hyperCMS Object.

Example:
http://yourdomain.com/opensearch/?object_id=3ed2vmtmac43f1zq

1.4.2 Container based search

The search-service also enables to query for objects of the same container via the container-ID.

Parameters:

container_id – unique ID or hash of a hyperCMS container.

Example:
http://yourdomain.com/opensearch/?container_id=0003983

1.4.3 Base search

The base search is probably the common way to search for content in hyperCMS. One can search for textual phrases, attributes of media files, modification date etc.

Parameters:

search_expression – a string, which could be a part of an object name, file path, textual phrase
Example:
http://yourdomain.com/opensearch/?search_expression=test

search_cat – if the search should only consider object names or a part of the file path, then the value of this parameter has to be "file".

Example:
*http://*yourdomain.com*/opensearch/?search_expression=test&search_cat=file*

search_textnode – an array of strings, where each string is representing a search expression and its corresponding key is standing for the text_id. The text_id indicates where the given expression has to be looked at.

Example:
*http://*yourdomain.com*/opensearch/?search_textnode[test]=test&search_textnode[test]=example*

search_dir – path to a folder which only should be considered for the search.

Example:
http://yourdomain.com/?search_expression=test&search_dir=%page%/ExamplePublication/

search_format – an array of strings, which defines what kind of objects should be considered. These can be: page, comp, audio, document, text, image, video, compressed, flash, binary

Example:
http://yourdomain.com/opensearch/?search_expression=test&search_format[]=document&search_format[]=video

date_modified – if the search should consider modification date, then the value of this parameter hast to be "yes". Then date interval has to be defined via following parameters:

- **year_from**
- **month_from**
- **day_from**
- **year_to**
- **month_to**
- **day_to**

Example:
http://yourdomain.com/opensearch/?search_expression=test&date_modified=yes&year_from=2011&month_from=3&day_from=12&year_to=2012&month_to=3&day_from=24

template – name of a template, to retrieve all objects, which are using this template.

Example:
http://yourdomain.com/opensearch/?template=BoxStandard.comp.tpl

search_imagesize – an interval of pixels, which reduces the result only to media files, where at least one side (whether horizontal or vertical) is in the given interval.

Example:
http://yourdomain.com/opensearch/?search_imagesize=500-700

search_imagewidth – exact amount of pixels a media files width has to have to be considered as a hit

Example:
http://yourdomain.com/opensearch/?search_imagewidth=1024

search_imageheight - exact amount of pixels a media files height has to have to be considered as a hit

Example:
http://yourdomain.com/opensearch/?search_imageheight=768

search_imagetype – defines which format should the searched image have. These value of the parameter can be: landscape, portrait or square.

Example:
http://yourdomain.com/opensearch/?search_imagetype=square

search_imagecolor – a color code that defines which main color should be considererd for the search. These can be: K for Black, W for White, E for Grey, R for Red, G for Green, B for Blue, C for Cyan, M for Magenta, Y for Yellow, O for Orange, P for Pink and N for Brown.

Example:
http://yourdomain.com/opensearch/?search_imagecolor=K

1.5 Pagination

The OpenSearch standard specifies parameters for implementing pagination. The hyperCMS OpenSearch service is using of them to page through the results:

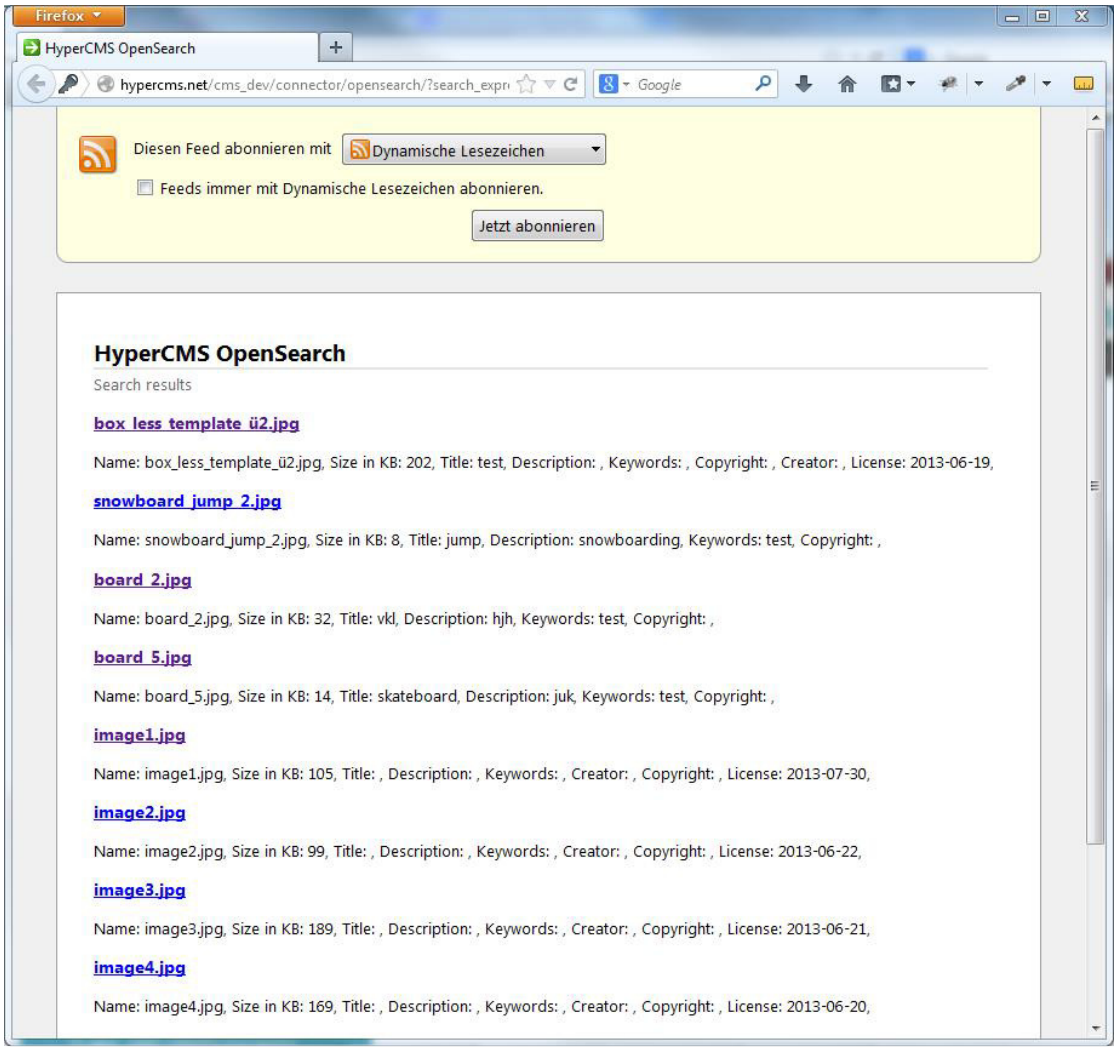
count – defines the page size – amount of elements. If not given, the default value for this parameter is 10.

startPage – defines which page should be displayed. If not given, the default value for this parameter is 1.

Example:
http://yourdomain.com/opensearch/?search_expression=test&count=5&page=2

1.6 OpenSearch Response

The hyperCMS OpenSearch service is using the RSS 2.0 format to display the search results.



2 OpenAPI

2.1 Introduction

The OpenAPI of the hyper Content Management Server allows you to create, modify, and delete objects, folders, groups and users from publications.
SOAP and WSDL is used as the technology layer for the communication between the client and server.

2.2 SOAP

A short Introduction into SOAP can be found here: http://en.wikipedia.org/wiki/SOAP

SOAP is supported in the version 1.1 and 1.2.

2.3 WSDL

A short Introduction into WSDL can be found on Wikipedia here:
http://en.wikipedia.org/wiki/Web_Services_Description_Language

The file which describes the OpenAPI functions is accessible under the following link:
<URL to hyperCMS>/connector/openapi/?wsdl.

2.4 (re)generate the WSDL

Whenever you need to generate or regenerate the WSDL file (for example: after an update to the OpenAPI) you can do that by opening the following link:
<URL to hyperCMS>/connector/openapi/?wsdl=generate
The newly generated WSDL file will be shown afterwards.

2.5 Types

There are several types defined for use with the OpenAPI.

2.5.1 download_object

Describes an object located inside the hyperCMS System. A object is either a component, multimedia asset, page or folder. This object has the following attributes:

objectpath
Complete path to the object, it has the typical hyperCMS Format (%comp% (Components) or %page% (Pages) followed by the publication name and then the exact location of the object.

type
The mime-type of the file

modifiedtime
Date of the last change to this object in the format „YYYY-MM-DD HH:SS"

downloadlink
A link to download the file. This link is only provided for multimedia assets.

wrapperlink
A link to the file which can be used to view the file inside the browser

content
Contains the data saved to an object. For multimedia files this is normally metadata, for other files these are the fields you can change when editing an object.

Example:

```
<text>
<text_id>Title</text_id>
<textuser>admin</textuser>
<textcontent><![CDATA[Test document]]></textcontent>
</text>
<text>
<text_id>Description</text_id>
<textuser>admin</textuser>
<textcontent><![CDATA[This is a test document]]></textcontent>
</text>
```

2.5.2 Group

Is used to identify a group when changing user settings.

Name: Name of the group
Publication: The publication from which the group should be used

2.5.3 Grouplist

A list of Groups.

2.5.4 Permission

A permission that is assigned to the group.

Name:
Name of the permission assigned to the group.
The following names are currently recognized by the system:

desktopglobal	persprofcreate	tplmediaupload
desktopsetting	persprofdelete	tplmediadelete
desktoptaskmgmt	persprofedit	componentglobal
desktopcheckedout	workflowglobal	compupload
desktoptimetravel	workflowproc	compdownload
userglobal	workflowproccreate	compsendlink
usercreate	workflowprocdelete	compfoldercreate
userdelete	workflowprocedit	compfolderdelete
useredit	workflowprocfolder	compfolderrename
groupglobal	workflowscript	compcreate
groupcreate	workflowscriptcreate	compdelete
groupdelete	workflowscriptdelete	comprename
groupedit	workflowscriptedit	comppublish
siteglobal	templateglobal	pageglobal
sitecreate	tpl	pagesendlink
sitedelete	tplcreate	pagefoldercreate
siteedit	tpldelete	pagefolderdelete
persglobal	tpledit	pagefolderrename
perstrack	tplmedia	pagecreate
perstrackcreate	tplmediacatcreate	pagedelete
perstrackdelete	tplmediacatdelete	pagerename
perstrackedit	tplmediacatrename	pagepublish
persprof		

If you need further explanation about the permissions, please have a look at the Administrators Guide.

Set:
Either 1 or any other value. If this value is 1 the right will be granted to this group.

2.5.5 Permissionlist

A list of permissions.

2.5.6 Folderlist

This is a list of folders to which the group has access. It consists of "%page%/" for pages, "%comp%/" for components, then the publication name followed by "/" and the subsequent folders. For example "%page%/Boarderline/05_Kontakt/" grants access to the folder 05_Kontakt in the pages of the Publication "Boarderline".

2.5.7 Textfield

A field that is saved to object.

Name: The name of the new value
Value: The value that is stored.

2.5.8 Fieldlist

A list of text fields which are stored to the specified object.

2.6 Functions

These are functions of the OpenAPI which can be used to manipulate and view objects, folder, users, and user groups. You can use any common programming language to create a SOAP client.

See the following example of a SOAP client written in PHP:

```
try
{
  // PHP Version 5.01 is required to be able to use the SoapClient class
  $hypercms = new SoapClient
("http://www.hypercms.net/cms_dev/connector/openapi/?wsdl=generate", array
('trace' => 1, 'exception' => 1));
  $sessionID = $hypercms->authenticate
('f6f9ae656204c2bc2521cbaffef060a6');
  $hypercms->folder_create ($sessionID, "%comp%/Boarderline/Test/");
  $file = "/tmp/test.mp3";
  $hypercms->upload ($sessionID, "%comp%/Boarderline/Test/test.mp3",
base64_encode (file_get_contents($file)), filesize($file), 0, 0, 0, 0);
  $download = $hypercms->download ($sessionID,
"%comp%/Boarderline/Test/test.mp3");
  $hypercms->close ($sessionID);
}
catch (Exception $e)
{
  // Do your error handling here
  var_dump ($e);
}
```

2.6.1 Authenticate

Authenticates a user in the system in order to be able to use the OpenAPI. It expects the hash of a valid user to be provided. After successful login the authenticate functions returns the session ID that is needed for other functions.

Example:

```
$sessionID = $hypercms->authenticate('f6f9ae656204c2bc2521cbaffef060a6');
```

2.6.2 Close

Closes the session with the session id you provided, so further attempts on funtion calls with this id will be rejected. As a security measure you should always close a session id you don't need any more.

Example:

```
$hypercms->close($sessionID);
```

2.6.3 Download

With this function you retrieve the download_object for the object or folder identified by the objectpath. Which in this case can either be the path to the object, or the hash code of the object / folder.

Parameters:
- sessionID
- objectpath

Example:
```
$hypercms->download(15895748, “%comp%/Boarderline/video.avi”);
$hypercms->download(15895748, “zjm3quu0ahqd3k6y”);
$hypercms->download(15895748, “%comp%/Boarderline/Test/song.mp3”);
$hypercms->download(15895748, “%comp%/Boarderline/Test/”);
```

2.6.4 Upload

This function allows you to upload a single file to the objectpath. This objectpath must lead to a non-existing file. If you upload a zip file the function can extract its content into the folder where it is uploaded into, if requested. It is also possible to create a preview image of supported files and if you upload an image you can specify if you want to resize the image and the resize percentage (value is valid between 0 and 200)

Parameter:
- sessionID
- objectpath
- mediacontent
- extract zip files
- create preview image
- resize image
- resize percentage

Example:
```
$hypercms->upload(15895748, “%comp%/Boarderline/Test/song.mp3”, <base64
encoded content>, 0, 0, 0, 0);
```

2.6.5 user_create

Creates a new user with the provided data for the specified publication. If you don't specifiy a publication or it is *Null* then the user will be created without a publication assignment. Loginname and password are used to authenticate with the system

Parameter:
- sessionID
- publication
- loginname
- password

Example:
```
$hypercms->user_create(15895748, “”, “User1”, “password”);
$hypercms->user_create(15895748, “*Null*”, “User2”, “password”);
$hypercms->user_create(15895748, “Boarderline”, “User3”, “password”);
```

2.6.6 user_edit

This function can change the password, real name, language, theme, email, signature and the group membership of the user identified by his loginname. If you change your own userdata you must provide the current password as a security measure, else you can leave it empty.
For currently supported Languages and Themes have a look at the edit user page in your system.

Parameter:
- sessionID
- loginname
- password
- real name
- language
- name of the theme
- email
- signature
- list of groups
- current password

Example:
```
$groups = array(array("publication" => "Boarderline", "name" =>
"Designer"), array("publication" => "DemoDAM", "name" => "ChiefEditor"));

$hypercms->user_edit(15895748, “TestUser”, “Testpassword1”, “Test User”,
“de”, “Standard”, “test@user.com”, “This is a Signature to test”, $groups,
"");

# If the authenticated user is TestUser
$hypercms->user_edit(15895748, “TestUser”, “Testpassword1”, “Test User”,
“de”, “Standard”, “test@user.com”, “This is a Signature to test”, $groups,
"TestingTest");
```

2.6.7 user_delete

This function deletes a user from the user database. If you provide an empty publication or *Null* as publication the user will be deleted from the system else from the publication only.

Parameter:
- sessionID
- publication
- loginname

Example:
```
$hypercms->user_delete(15895748, "*Null*", “TestUser”);
$hypercms->user_delete(15895748, "", “TestUser2”);
$hypercms->user_delete(15895748, "Boarderline", “TestUser3”);
```

2.6.8 group_create

Used to create a new group for a publication.

Parameter:
- sessionID
- publication
- name of the group

Example:
```
$hypercms->group_create(15895748, “Boarderline”, “TestGroup”);
```

2.6.9 group_edit

Changes the allowed paths and permissions associated with a group.

Parameter:
- sessionID
- publication
- group name
- folder access to pages
- folder access to components
- permissions

Example:
```
$accesslistpage = array("%page%/Boarderline/05_Kontakt/",
"%page%/Boarderline/02_Unternehmen/");

$accesslistcomp = array("%comp%/Boarderline/_denis_test/",
"%comp%/Boarderline/News/");
$permissionlist = array(array('name' => 'desktopglobal', 'set' => 1),
array('name' => 'desktopsetting', 'set' => 1));

$hypercms->group_edit(15895748, “Boarderline”, “TestGroup”,
$accesslistpage, $accesslistcomp, $permissionlist);
```

2.6.10 group_delete

Deletes a group out of a publication

Parameter:
- sessionID,
- publication
- group name

Example:
```
$hypercms->group_delete(15895748, “Boarderline”, “TestGroup”);
```

2.6.11 publish / unpublish

Publishes / Unpublishes the object identified by objectpath. If the object is a folder all files contained in it will also be published / unpublished. Its also possible to only apply this to already published objects.

Parameter:
- sessionID
- objectpath
- shall ony published elements be used

Example:
```
$hypercms->publish(15895748, “%comp%/Boarderline/Test/test.php”, 0);
$hypercms->publish(15895748, “%comp%/Boarderline/Test/”, 1);

$hypercms->unpublish(15895748, “%comp%/Boarderline/Test/test.php”, 0);
$hypercms->unpublish(15895748, “%comp%/Boarderline/Test/”, 1);
```

2.6.12 set_fields

You can change the value of fields stored with an object with this function, please be aware that you can set any fields, but they won't appear in the system if the respective template doesn't use them.

Parameter:
- sessionID
- objectpath
- list of fields
- charset for the object

Example:
```
$fields = array();
$fields[0] = new stdClass();
$fields[0]->name = 'test1';
$fields[0]->value = 'value1';
$fields[1] = new stdClass();
$fields[1]->name = 'test2';
$fields[1]->value = 'value2';
$fields[2] = new stdClass();
$fields[2]->name = 'test3';
$fields[2]->value = 'value3';
$hypercms->set_fields(15895748, “%comp%/Boarderline/Test/test.php”,
$fields, “UTF-8”);
```

2.6.13 folder_create

Creates a new folder at the specified path.

Parameter:
- sessionID
- folderpath (Path including the folder to be created)

Example:
```
$hypercms->folder_create(15895748, “%comp%/Boarderline/Test/Test2/);
```

2.6.14 folder_rename

Renames the folder at the specified path to new name which should just be the new name.

Parameter:
- sessionID
- folderpath
- new name

Example:
```
$hypercms->folder_rename(15895748, “%comp%/Boarderline/Test/Test2/“, „test3“);
```

2.6.15 folder_delete

Deletes the specified folder

Parameter:
- sessionID
- folderpath (Ordner der gelöscht werden soll)

Example:
```
$hypercms->folder_delete(15895748, “%comp%/Boarderline/Test/Test3/“);
```

2.6.16 object_create

Creates a new object based on a defined template. The template must exist and must be of the correct type (page templates for pages, component template for components). The objectpath should contain the filename without an extension because the extension is defined by the template

Parameter:
- sessionID
- objectpath
- template (Name of the template which should be used)

Example:
```
$hypercms->object_create(15895748, “%comp%/Boarderline/Test/Test2”, “Default”);
```

2.6.17 object_rename

Renames the object to another name. Object should contain the file extension of the file, but the new name must not contain any extension.

Parameter:
- sessionID
- object
- new name

Example:
```
$hypercms->object_rename(15895748, “%comp%/Boarderline/Test/Test2.php", “Test3”);
```

2.6.18 object_delete

Deletes the object and all associated data from the System. It can then only be restored from the backup

Parameter:
- sessionID
- object

Example:
```
$client->object_delete(15895748, „%comp%/Boarderline/Test/Test3.php“);
```

3 Legal reference / flag

3.1 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:
www.hypercms.com
support@hypercms.com

3.2 Imprint

Responsible for the content:

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

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

3.3 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.