hyper CMS Content & Digital Asset Management Server

Version 10.2

Import/Export Guide

2026-01-05

Index

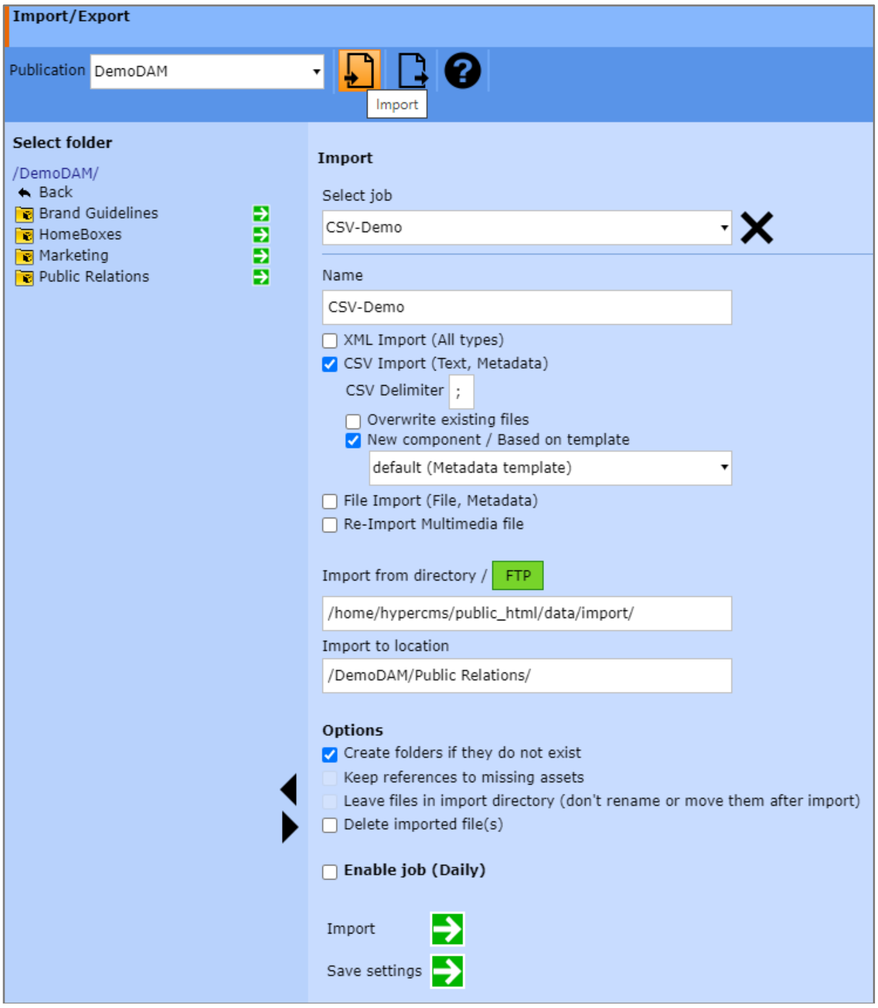
1 Introduction

The hyper Content & Digital Asset Management Server allows the import and export of objects (pages, components, multimedia assets) via a defined interface. Content from external systems can be automatically imported in the system and exported from the system.
The graphical user interface for the import and export can be found inside the Administration item of the Navigator, in case the Connector module is installed.

2 Import

2.1 General

The import is provided by the REST API in "hypercms/connector/import", which is located in the connector module. The import can be triggered by a job at regular intervals or on demand.
The names of the imported files are irrelevant since all the files in the import directory will be analyzed and imported. However, if you use XML files for the import, it is important that the file names are different and have the file extension .xml.
In case a container ID is provided in the XML import file, the existing object with the same ID will be overwritten, if the object exists already on the same location in the system.
Besides the REST API there is also a graphical user interface for the automated import.



Import/Export

Publication: DemoDAM

Import

Select folder

/DemoDAM/
- Back
- Brand Guidelines
- HomeBoxes
- Marketing
- Public Relations

Import

Select job: CSV-Demo

Name: CSV-Demo

□ XML Import (All types)
☑ CSV Import (Text, Metadata)
CSV Delimiter: ;
□ Overwrite existing files
☑ New component / Based on template
default (Metadata template)
□ File Import (File, Metadata)
□ Re-Import Multimedia file

Import from directory / FTP
/home/hypercms/public_html/data/import/

Import to location
/DemoDAM/Public Relations/

Options
☑ Create folders if they do not exist
□ Keep references to missing assets
□ Leave files in import directory (don't rename or move them after import)
□ Delete imported file(s)

□ **Enable job (Daily)**

Import

Save settings

Name:
You need to provide a name for the import job if you want to save the import parameters.

XML Import:
The import of content requires the storage of XML files in a provided directory. The standard directory for the import is /data/import. This allows you to import contents exported from another system into your own system.

importtype … “xml“ for the import of XML files

CSV Import:
The text content of CSV files can be imported. If no CSV delimiter has been specified the system will determine it automatically based on the files content. Usually the delimiter is ; or ,. If the content of existing objects or assets should be replaced, you need to check the option “Overwrite existing files”. If you want to create new objects (components) based on a certain component template, you need to check the option “New component / Based on template” and select a template.

importtype … “csv“ for the import of CSV files
delimiter … delimiter character used in the CSV files
updateobjects … true/false, update the content of existing objects
createobjects … true/false, create new objects
template … component template name to be used for new objects

File import:
Import of all files and folders from the specified import directory.

importtype … “file“ for the import of files

Re-Import Multimedia file:
If multimedia files were previously exported, whereby the data was kept in the system (see the Export chapter), these can also be re-imported.
The symbolic links in the repository will be replaced by the original files.
Note that this import does not analyze subdirectories. The exact specification of the directory is therefore important.

importtype … “symlink“ for the import of previously exported files (symbolic links will be replaced by the file)

Import from directory:
importdir ... file system path to the import directory, default is data/import or FTP server path

Import to location:
The import of folder structures and files can also be done without the use of XML files. In this case, the import process scans all files and folders in the specified path and imports them as folders or assets in the system. All imported files are analyzed and indexed.

location … location for import of folders and assets, only mandatory for importtype “csv” and “file”

Create folders if they do not exist:
A parameter can be used to force non-existent subdirectories to be created automatically during the import. If the parameter will be activated you need to be careful since the import of a subfolder and its metadata cannot take place if the folder exists already. This missing information of a folder can only be imported later on by providing the container name for the target system.

createfolders ... true/false, create folders automatically, default value = false

Keep references to missing assets:
If links to non-existing components will be imported, the system can prevent the import of these relationships. Since the missing object might be imported later on during the import process, ignoring this error and importing the broken relations might be useful. This option will not be used when importing CSV files.

linkignore ... true/false, ignore links, default value = false

Leave files in the import directory:
If this option is selected, the system uses the files in the source directory and remembers the reference to the file. This makes sense if the files are shared and therefore can't be removed from the import directory. It should be noted, however, that the files in the source directory may neither be renamed nor moved, as this invalidates the reference to the file.
This option does not apply when importing CSV content since these are not considerer as multimedia files.

leavefiles ... true / false, default value = false

Delete imported files:
If this option is selected, the files in the source directory will be removed after the successful import.

deletefiles ... true / false, default value = false

Password protection:
If a password has been set in the main configuration file of the system config.inc.php to protect the execution of the import and export, an additional parameter is required in order to provide the password, see $mgmt_config['passcode'].

passcode ... password for the import/export REST service

Example of an import request via REST:
https://domain.com/hypercms/connector/import/?passcode=mypasscode&importtype=xml&linkignore=true

2.2 XML Import

2.2.1 Structure of a XML file

The stored xml files for the import need to fulfill the following definition:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<object>
        <hyperCMS>
                <contentcontainer></contentcontainer>
                <contentuser>username</contentuser>
                <contentlocation>%comp%/Publication/folder/</contentlocation>
                <contentobject>objectname</contentobject>
                <contenttemplate>templatename.comp.tpl</contenttemplate>
                <contentpublished>true</contentpublished>
                <contentworkflow>Publikation.Workflowname.xml</contentworkflow>
        </hyperCMS>
        <head>
                <pagetitle>title</pagetitle>
                <pageauthor>name of the author</pageauthor>
                <pagedescription>description</pagedescription>
                <pagekeywords>keyword list</pagekeywords>
                <pagecontenttype>text/html; charset=iso-8859-1</pagecontenttype>
                <pagelanguage>language shortcut (e.g. en)</pagelanguage>
                <pagerevisit>20 days</pagerevisit>
                <pagetracking>private.track.dat</pagetracking>
        </head>
        <textcollection>
                <text>
                        <text_id>summary</text_id>
                        <textuser>username</textuser>
                        <textcontent>This is the summary!</textcontent>
                        </text>
                        <text>
                        <text_id>detailed</text_id>
                        <textuser>username</textuser>
                        <textcontent><![CDATA[<font size="2">This is the detailed
                        content!]]></textcontent>
                </text>
        </textcollection>
        <mediacollection>
                <media>
                        <media_id>image</media_id>
                        <mediauser>username</mediauser>
                        <mediafile>Publication/Manual_hcm0325418.pdf</mediafile>
                        <mediaobject>%comp%/Publication/Docs/Manual.pdf</mediaobject>
                        <mediaalttext>User Manual</mediaalttext>
                        <mediaalign></mediaalign>
                        <mediawidth></mediawidth>
                        <mediaheight></mediaheight>
                </media>
        </mediacollection>
        <linkcollection>
                <link>
                        <link_id>link</link_id>
                        <linkuser>username</linkuser>
                        <linkhref>http://www.hypercms.net</linkhref>
                        <linktarget>_blank</linktarget>
                        <linktext>www.hypercms.net</linktext>
                </link>
        </linkcollection>
        <componentcollection>
                <component>
```

```
                        <component_id>component</component_id>
                        <componentuser>username</componentuser>
                        <componentcond>profilename</componentcond>
                        <componentfiles>%comp%/Publication/folder/component1.jsp|%comp%/Publication/folder/component2.jsp</componentfiles>
                </component>
        </componentcollection>
        <articlecollection>
                <article>
                        <article_id>article</article_id>
                        <articletitle>News</articletitle>
                        <articledatefrom></articledatefrom>
                        <articledateto></articledateto>
                        <articlestatus>active</articlestatus>
                        <articleuser>username</articleuser>
                        <articletextcollection>
                        </articletextcollection>
                        <articlemediacollection>
                        </articlemediacollection>
                        <articlelinkcollection>
                        </articlelinkcollection>
                        <articlecomponentcollection>
                        </articlecomponentcollection>
                </article>
        </articlecollection>
</object>
```

2.2.2 Parameters

In order to create a new object (page, component, folder or multimedia asset), the following information is required:

```
<hyperCMS>
        <contentcontainer>0014020.xml</contentcontainer>
        <contentuser>username</contentuser>
        <contentlocation>%comp%/Publication/folder/</contentlocation>
        <contentobject>objectname</contentobject>
        OR
        <contentfolder>foldername</contentfolder>
        <contenttemplate>templatename.comp.tpl</contenttemplate>
        <contentpublished>true</contentpublished>
        <contentworkflow>Publikation.Workflowname.xml</contentworkflow>
</hyperCMS>
```

Mandatory inputs:
<contentuser> … user name
<contentlocation> … location of the new object; %page% is the root of pages; %comp% is the root of components; followed by the name of the publication; each folder needs to be seperated by a slash /
<contentobject> … name of the object without file extension OR
<contentfolder> … name of the folder (folder BEFORE object, if both parameters exist)
<contenttemplate> … name of the related template

These inputs are required so the new object can be assigned to a publication and can be created in the right place based on a given template.

Password protection:
<contentpasscode> … If a password was set for the use of import/export in „config.inc.php“, the correct password need to be provided.

Optional inputs are:
<contentcontainer> … Indication of an existing container, the content of the existing container will be overwritten. If the entry is empty, a new container will be generated.
<contentdate> … date in format yyyy-mm-dd hh:mm
<contentpublish> … Should the content be published (true/false or date/no-entry)
<contentworkflow> … Assign object to an existing workflow

The section or XML-node <head> is needed to assign metadata to a page. This XML node can be left empty, if no information should be added.

Followed by the XML-nodes <textcollection>, <mediacollection>, <linkcollection>, <componentcollection> and <articlecollection>.
These nodes present the actual contents of an object and are also the editable elements that the system recognizes. These XML nodes are optional and can be left blank, but this means that an object is created without any content.

The XML node for <mediacollection>presents mainly a reference to a multimedia asset. The multimedia asset itself must be available in the system, see the location in the node <mediaobject>.
If the system can locate the associated multimedia asset, the node will be automatically filled <mediafile>. If the multimedia asset does not exist, the XML node <media> will not be imported.

```
<media>
        <media_id>image</media_id>
        <mediauser>username</mediauser>
        <mediafile>Publication/Manual_hcm0325418.pdf</mediafile>
        <mediaobject>%comp%/Publication/Docs/Manual.pdf</mediaobject>
        <mediaalttext>User Manual</mediaalttext>
        <mediaalign></mediaalign>
        <mediawidth></mediawidth>
        <mediaheight></mediaheight>
</media>
```

The same will be applied for the XML node <componentcollection>. Again, there is a check whether the references of the node <componentfiles> point to an existing objects.

```
<componentcollection>
        <component>
        <component_id>component</component_id>
        <componentuser>username</componentuser>
        <componentcond>profilename</componentcond>
        <componentfiles>%comp%/Publication/folder/component1.jsp|%comp%/Publication/folder/component2.jsp</componentfiles>
        </component>
</componentcollection>
```

2.2.3 Templates

It is important that the assigned template holds the elements with the correct ID, so the imported content can be viewed and edited in the system.
All contents are always imported, since the template can be extended or replaced at any time and therefore the content can be displayed and edited.

2.2.4 Import of multimedia assets

When importing files (multimedia assets), the system needs to have read permissions in order to access the files in the given location. Therefore the file does not have to be located in the "data/import" directory.

In order to import the files without XML, the following information needs to be available to the export script:
location ... path, e.g. %comp%/publication/folder/

If XML files are used for the import:
In the XML file, the metadata can be stored in the <textcollection>. Again, a metadata template must be available in order to allow the user to edit the data.

In order to reference to the media file, the following entry (bold) is necessary:

```
<hyperCMS>
	<contentcontainer></contentcontainer>
	<contentuser>username</contentuser>
	<contentlocation>%comp%/Publication/folder/</contentlocation>
	<contentobject>filename</contentobject>
	<contenttemplate>default.meta.tpl</contenttemplate>
	<contentmediafile>image.jpg</contentmediafile>
	<contentpublish></contentpublish>
	<contentworkflow></contentworkflow>
</hyperCMS>
```

The name of the multimedia asset <contentobject> does not have to be the same name of the imported file <contentmediafile>.

After the successful import of the data (XML files) taken from the directory „data/import“ the files will be removed. The multimedia files will be removed as well if they are located in the directory „data/import“ or another location of the file system.

2.3 CSV Import

If you want to import text content or metadata, you can provide CSV (Comma Separated Values) files in the specified import directory. It is important that the CSV file provides the objectpath (folders and file name), the name of the object or the Container ID as the first column title and the content IDs used in the templates for the content identifier as column titles. Most likely you will have the file name and not the unique container ID that the system will create whenever a new object is created.

This is an example of the metadata or text content definition in a template. This has been defined by a user with permissions to create and edit templates in the system.
Keep in mind that each content or field is defined by an ID, e.g. id='Title'. You can provide the "Title" by typing it in the corresponding field or by the import of the content stored in a CSV file.

Definitions in the template:
*[hyperCMS:textu **id='Title'** label='Title' infotype='meta']*
*[hyperCMS:textu **id='Description'** label='Description' infotype='meta']*
*[hyperCMS:textk **id='Keywords'** label='Keywords' infotype='meta']*

The CSV file must have the following structure:

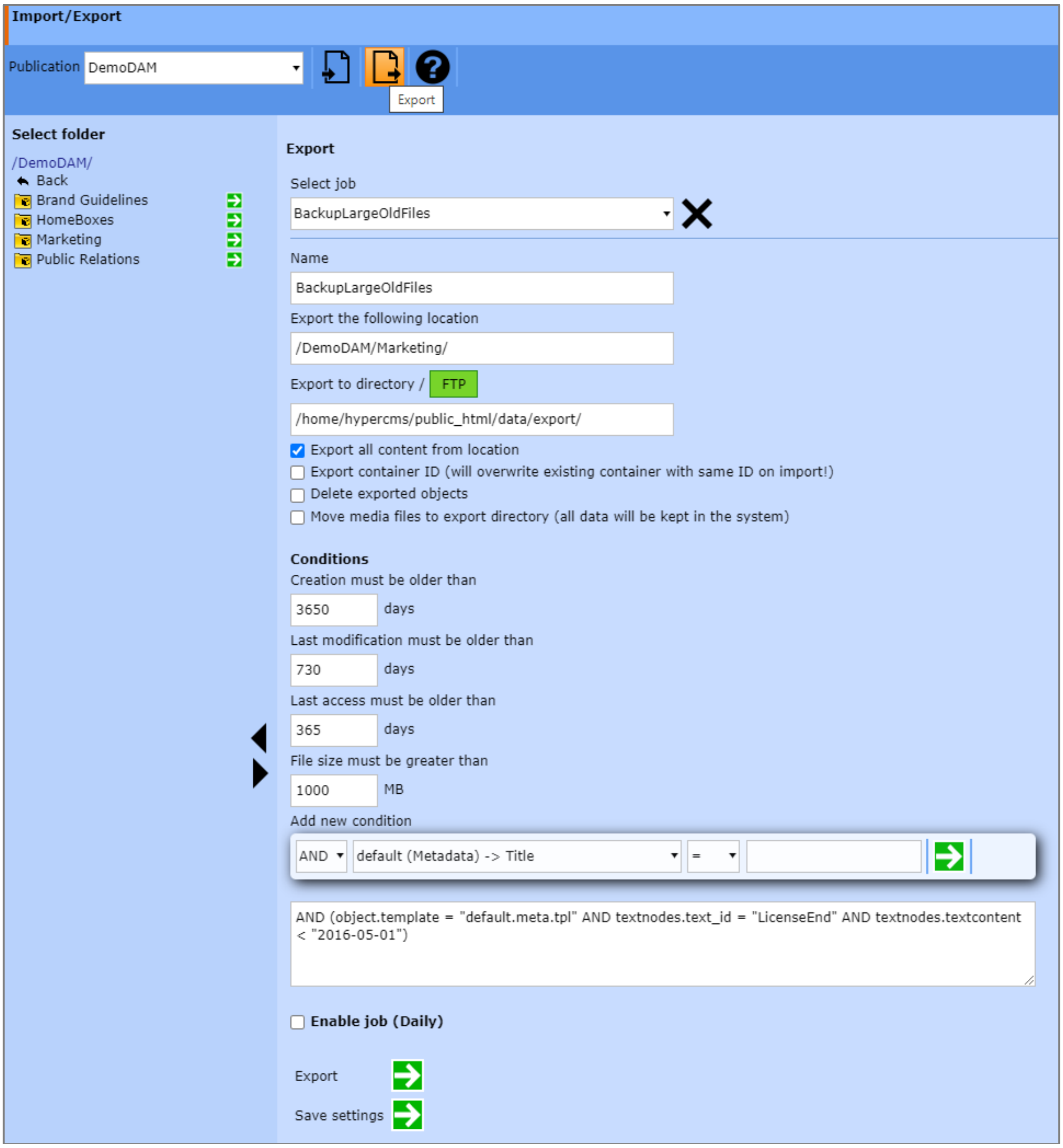
```
Name;Title;Description;Keywords
"flowers.xlsx";"Flowers";"A list of all flowers;Flowers, Nature"
"/roses/red.jpg";"Red roses";" Roses, Red, Flowers, Field, Nature"
"/roses/yellow.jpg";"Yellow roses";"Roses, Yellow, Flowers, Field, Nature"
```

The column holding the object path or only the name must be named "objectpath" or "name" (case-insensitive).
The column holding the container ID (alternative to the object path or name) must be named "containerid", "container-id" or "container_id" (case-insensitive).

3 Export

The export of content will export XML files and multimedia files in the export directory. The export is performed by the REST API in "hypercms/connector/export", which is located in the connector module. The export can be initialized by a job at regular intervals or on demand.
Besides the REST API the connector module of the system offers a graphical user interface in order to define export jobs.

Important: Only the latest version of the object will be exported, all other versions will not be exported.



Publication: DemoDAM

/DemoDAM/
- Brand Guidelines
- HomeBoxes
- Marketing
- Public Relations

Export

Select job: BackupLargeOldFiles

Name: BackupLargeOldFiles

Export the following location: /DemoDAM/Marketing/

Export to directory / FTP: /home/hypercms/public_html/data/export/

☑ Export all content from location
☐ Export container ID (will overwrite existing container with same ID on import!)
☐ Delete exported objects
☐ Move media files to export directory (all data will be kept in the system)

Conditions

Creation must be older than 3650 days

Last modification must be older than 730 days

Last access must be older than 365 days

File size must be greater than 1000 MB

Add new condition: AND | default (Metadata) -> Title | =

AND (object.template = "default.meta.tpl" AND textnodes.text_id = "LicenseEnd" AND textnodes.textcontent < "2016-05-01")

☐ **Enable job (Daily)**

In order to export the content of an object, the following information needs to be available to the export script:
location ... path, e.g. %comp%/publication/folder/
object ... name of the object (optional)

Definition of the export directory:
exportdir ... file system path to the export directory, default is data/export

Should all object of a given location be exported, an additional setting can be used:
all ... true/false, default value = false, is not working if contentcontainer is provided

This setting can be applied to folders and objects. If a folder is specified, all content inside the folder will be exported. If an object is provided, all content from the same location of the object to be exported.

Should the container name be exported, an additional setting can be used:
preserve ... true/false, default value = false, if the container name should be exported

If the container name will be exported and the exported xml file will be re-imported again, the available container will be overwritten.

Should only the multimedia files exported and symbolic links to the files be created:
symlink ... true/false, default value fault = false

This option will only export the multimedia files and will create symbolic links instead of the files in the repository. So large multimedia files that can be stored on a less costly storage. However, also the storage used should provide the files in a reasonable time.
If the original version of the object will be requested (by function downloadfile) the multimedia file will be automatically imported again.

Should the objects be deleted after the export, an additional setting can be used:
delete ... true/false, default value = false

The following conditions can be set:
createdays ... only objects with older create date than the provided value in days, will be exported
editdays ... only objects with older last modified date than the provided value in days, will be exported
accessdays ... only objects with older last access date than the provided value in days, will be exported
filesize ... only objects with a file size greater than the provided value in MB, will be exported

The free definition of conditions based on metadata or the content of objects is also possible. You can use the editor to create new conditions without limits. The editor for new conditions creates SQL code in the text box below the editor. This code can also be removed or edited if necessary.
The drop-down-boxes of the editor displays operators as well as the Text-ID of the hyperCMS tag of a template in the form of: template-name[template-type] -> Text-ID

Password protection:
If there is a password defined in "config.inc.php", the correct password must be provided using the parameter "passcode", see $mgmt_config['passcode'].

passcode ... password for the import/export REST service

Example of a REST export request:
https://domain.com/hypercms/connector/export/?passcode=mypasscode&location=%comp%/publication/folder/&all=true

All subfolders and objects will be exported. The content containers and multimedia files are stored in the export folder with the container name (XML file with container ID).
When exporting multimedia assets the files are also stored in the directory "data/export". Both the content container and the file will be exported.
The reference to the file is saved in the content container. The file name of the media file includes the name of the container ID in the form: name_hcm0014876.jpg

4 Legal reference / flag

4.1 Questions and suggestions

For advanced questions and suggestions, please contact the support.

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

4.2 Imprint

Responsible for the content:

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

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

4.3 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.