

SYNEPSIS DOCXPORT™

USER GUIDE



January 2019

V4.0

[© Synopsis Consulting inc. All brand names, product names and trademarks are the property of their respective owners.]

CONTENTS

Introduction	6
Configuration FILE - APPLICATIONS.PROPERTIES.....	6
docbase.xmlchunksize	6
Replacement characters	6
docbase.catchall.foldername	6
max.os.pathlength	7
docbase.virtualDocumentsFolderEx	7
Email notification	7
#Excel export parameters.....	7
Configuring log and output – Log4j.properties.....	8
Configuring for output	8
Launching DocXport.....	8
Login	8
Selecting Cabinets, Folders or files.	9
File	9
New.....	9
Open	10
Load by Folder	10
Open vs Load by Folder, an example	10
Save As.....	11
DQL Request	11
Logout.....	11
Exit	11
Settings	11
Attributes.....	12
Show All Versions	13

Include renditions	13
Generate navigation	13
Export attributes for content less objects	13
Use Filestore Links in attributes file.....	13
Resume Mode.....	13
Tools	15
Report Collisions	15
Actions	16
Set Destination	16
Export Content.....	16
Export Permissions	16
Compress (ZIP) Output	17
Folder Map	18
Folder Properties	18
powerXport Mode	19
Entering a dql qualification.....	20
Configuring export settings	20
Command Line Export	21
The XML configuration file.....	21
The <attributes> node	21
The <exportParams> node	22
The <DQL> node	23
Launching a command line export.	23
Q&A	24
Known issues and bugs.....	24

INTRODUCTION

This manual lists and explains the different functionalities available in DocXport v4.0.

CONFIGURATION FILE - APPLICATIONS.PROPERTIES

The file **application.properties** in the <DocXport_root_dir>/config directory allows the user to set the default behavior of the application.

Note: When the application.properties file is modified, DocXport must be restarted in order for the changes to be taken into account.

DOCBASE.XMLCHUNKSIZE

Defines how many objects will be included in the xml attributes properties file before switching to the next. Each document version, with or without content is considered an object. DocXport will not switch XML file in the middle of exporting a version tree in order to keep all version of a same document in the same xml file. Therefore, some files might contain more than the defined value.

REPLACEMENT CHARACTERS

DocXport will check for and replace forbidden characters in the file path and file name of an exported file. By default, DocXport is configured to replace characters forbidden on an NTFS file system. Each character has a matching replacement character. By default, all forbidden characters are replaced by a “-”.

The following properties define the forbidden characters for file path (the characters list must be enclosed in double quotes, special characters such as the double quote must be escaped using a backslash).

```
forbidden.chars.path = ".*?\"<>|.>";
replacement.chars.path = "-----";
```

The following properties define the forbidden characters for a file name (the characters list must be enclosed in double quotes, special characters such as the double quote must be escaped using a backslash).

```
forbidden.chars.path = ".*?\"<>|.>";
replacement.chars.path = "-----";
```

DOCBASE.CATCHALL.FOLDERNAME

Defines the name of the default folder where files will be exported when the file path exceeds the limit allowed by the **max.os.pathlength** property.

MAX.OS.PATHLENGTH

Define the maximum allowed number of characters for the full path of a file for the current O/S. Default is set to 255 which is the maximum size allowed for the windows file system.

DOCBASE.VIRTUALDOCUMENTSFOLDEREX

Virtual documents are exported as folders containing the virtual document assembly. This property defines the extension which will be added to the object_name of a virtual document when it is exported to disk.

EMAIL NOTIFICATION

DocXport can be configured to send an email notification when the export is completed or has stopped. This functionality is configured using the following properties.

#To enable email notification, set the following property to true. When this property is set to false, all other email notification related properties are ignored by DocXport.

mail.enabled=false

#define the name of your company's email server.

mail.host=mail.synepsis.com

#The name that you wish to see as the sender of the email

mail.from.name=DocXport Notification

#The email address you wish to see as the sender of the email

mail.from.address=docxport@synepsis.com

Addressees. Multiple recipients can be specified using a ";" as a separator

mail.to.address=test@synepsis.com;

#If your email server requires authentication: username/password of the mail account.

mail.user=user

mail.password=pass

#EXCEL EXPORT PARAMETERS

DocXport can be configured to export parameters as an excel worksheet. The following parameters will allow the user to customize the output of the excel attributes file.

excel.chunksize=10000

Defines how many objects will be included in a single Excel attributes file before switching to the next.

excel.repeatable.separator=|

When exporting attributes to Excel, DocXport will export all repeatable attributes values to a single cell. This parameter defines which character will be inserted to separate repeatable attributes values within an Excel cell.

CONFIGURING LOG AND OUTPUT – LOG4J.PROPERTIES

DocXport uses the log4j package to log errors and traces. Two types of logs can be generated:

Errors only: Will log information only on files that have errors on export

Could not export document : /Wombat Insurance Co/Index Server workfiles/attach.trace.txt;

Verbose output: Will log an entry for each file exported.

2010-11-22 12:43:58,821;09de75d180005d92;/DocXport Demo/Section B/onDemand - Server Deployment.pdf;Export OK;

CONFIGURING FOR OUTPUT

The file **log4j.properties** in the <DocXport_root_dir> directory allows the user to set the log output of the application.

Note: When the log4j.properties file is modified, DocXport must be restarted in order for the changes to be taken into account.

To enable verbose output modify the following entry in the log4j.properties file:

log4j.logger.verbose.document.log = **ERROR**,V

to

log4j.logger.verbose.document.log = **INFO**,V

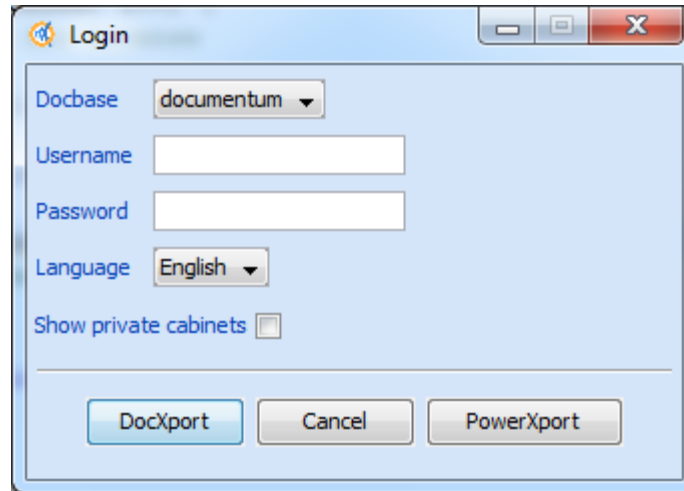
LAUNCHING DOCXPORT

To start DocXport, double-click on the DocXport.bat file in the DocXport installation directory.

LOGIN

To login into DocXport:

- Enter your docbase username (user_os_name) and password.
- Select which Docbase you would like to connect to (Docbase list is retrieved from the DocBroker).
- Select the language you would like to use for the application. Currently, English and French are available.
- Check the option *Show private cabinets* if you would like to see the private cabinets in the docbase.
- Click DocXport



SELECTING CABINETS, FOLDERS OR FILES.

DocXport allows a user to select documents for exporting using a visual interface or a DQL query. The user interface is split in two frames. On the left frame, the cabinets and folders in the currently connected docbase are displayed as a tree structure. Next to each folder, a checkbox is available. Clicking this checkbox will select all documents in this folder and in **all its sub-folders**. If there are no documents in this folder or in its subfolders, then nothing will be selected (no checkboxes will be checked even though the user clicked on it).

Clicking on the folder name will display this folder's documents on the center frame. Note that only the documents and not the sub-folders are displayed in the central frame. Next to each document in the central frame is a checkbox. Clicking on this checkbox will toggle selection for this specific document.

FILE

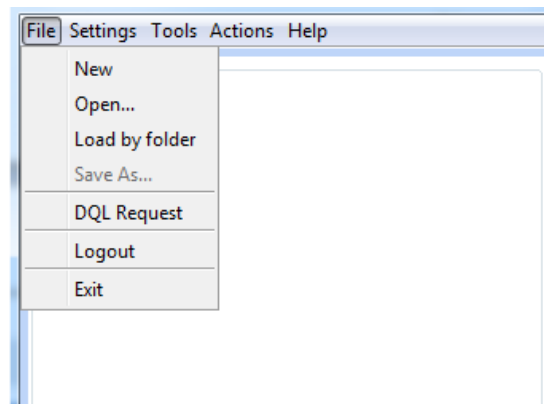


Figure 1: File menu bar items

NEW

Clears the current selection and prepares DocXport for a new extract. The current selection will be lost unless it has been previously saved.

OPEN

Opens a previously saved export configuration. All previously selected documents (with or without versions) attributes and DocXport settings will be reloaded. This functionality loads documents selection using the document's object ids.

LOAD BY FOLDER

Opens a previously saved export configuration. All previously selected documents (with or without versions) attributes and DocXport settings will be reloaded. This functionality loads documents selection using the folder ids of objects. This functionality complements the "Open" functionality. It is similar in that it loads a previous selection of documents. However, it differs in that it will load all the documents in ANY of the selected folders.

OPEN VS LOAD BY FOLDER, AN EXAMPLE

A Docbase contains a folder structure as per Table 1. An export of the folders **MOPs** and **Reports** is required. The DocXport user prepares the export by selecting the required documents and folders and saves the selection.

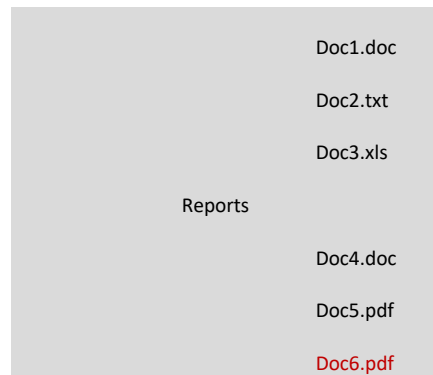
Table 1: Sample folder structure for Cabinet 1

Cabinet 1	
	MOPs
	Doc1.doc
	Doc2.txt
	Doc3.xls
	Reports
	Doc4.doc
	Doc5.pdf

Now, suppose that Doc6.pdf is added to the structure as shown in Table 2 by a Docbase user. Using the "Open" function to load the previously saved export will omit Doc6.pdf as it has not been selected specifically when the selection was made. However, using the "Load by Folder" option of DocXport, Doc6.pdf will automatically be added to the selection because it is contained in a folder where other documents are selected.

Table 2: Sample folder structure for Cabinet 1 after adding Doc6.pdf

Cabinet 1	
	MOPs



SAVE AS

Saves the current DocXport selection as an XML document which can later be reloaded.

DQL REQUEST

Allows the user to input a DQL qualification to select the documents. The user interface will be updated accordingly. However, if the “ALL” keyword is used, all versions of the documents will be selected but this won’t be visible on the interface unless the “Show all versions” option of the Edit menu is checked.

Note: Multiple DQL qualifications can be entered separated with the “GO” keyword. For example:

```
dm_document where object_name='test.doc'
GO
dm_document where object_name like 'docXport%'
GO
dm_document (all) where folder('/folder1')
```

LOGOUT

Allows the user to change the current Docbase connection. Logs the current user out of the Docbase but keeps DocXport running.

EXIT

Logs the current user out of DocXport and shuts down the application.

SETTINGS

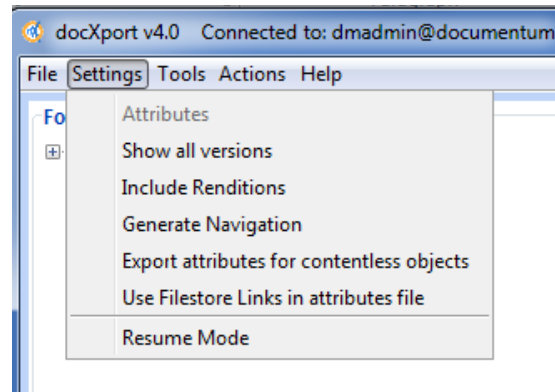


Figure 2: Settings menu bar items

ATTRIBUTES

This function is only available once at least one document has been selected. It will display a window which will allow the user to select which attributes for each of the selected object types should be included in the export. The user can:

- Select attributes to be exported for each object type
- The radio buttons at the bottom of the screen allow the user to select if the attributes should be exported to an XML file or to an Excel spreadsheet.
- When the Excel option has been selected, the buttons on the right of the screen allow the user to select the column order of the Excel attributes file.

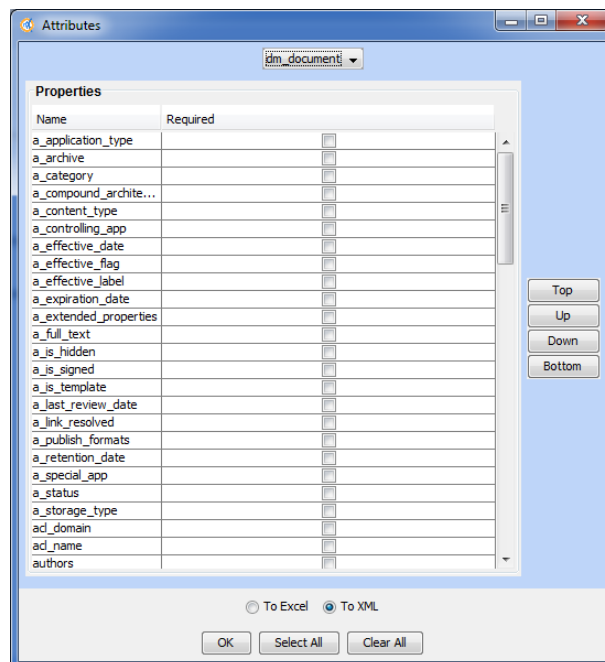


Figure 3: Attributes selection window

SHOW ALL VERSIONS

When selected, DocXport displays all versions of the documents in a folder. Otherwise, only the current (active) version of a document is shown and can be selected.

INCLUDE RENDITIONS

In Documentum systems, renditions are representations of a Document in a different file format. For example, you import a MS Word document for editing. The .doc rendition becomes the primary rendition of this document. Once you are done editing the document, you might want to import a pdf version of that document for publication. This becomes a secondary rendition of your document. More information on renditions can be found in the Documentum documentation. When the include renditions options is checked, DocXport will export all renditions of an object. In the example above, both the .doc and .pdf version of the object will be exported to the disk.

GENERATE NAVIGATION

Defines whether DocXport should generate an HTML navigation window to navigate the exported documents. If checked, DocXport will generate an HTML navigation interface where a list of exported documents, their attributes and different renditions can be browsed. Currently, this feature is limited to 250 documents. If more than 300 documents are selected for exporting, only the last X documents will be displayed in the navigation (where $X = \text{Total number of document MOD } 300$).

EXPORT ATTRIBUTES FOR CONTENT LESS OBJECTS

Documentum can store content less objects. These objects are used for storing only attributes information and have no related file. When this option is selected, DocXport will include all attributes of a content less object in the outputted attributes file. If this option is not selected and a content less object is encountered during the export, DocXport will trace the object in the error log.

USE FILESTORE LINKS IN ATTRIBUTES FILE

When this option is selected, DocXport will not copy the content files to the target destination on the disk. Instead, DocXport will include a link in the attributes file pointing to the Documentum docbase filestore (the file where Documentum stores and reads the file in the docbase). In order to use this link, the docbase's filestore(s) must be mounted on the local machine to the same relative path as it is mounted on the Documentum server. This option is particularly interesting for data migrations where one would like to avoid an intermediate copy of the file.

Note: Synopsis strongly recommends mounting the filestore in read only mode in order to avoid any involuntary modification to the docbase.

RESUME MODE

Sets DocXport in resume mode. This special mode allows DocXport to “resume” a previously failed export. In order to use this functionality:

- The Resume mode must be enabled by selecting this option in the menu.
- The document selection must be the same as in the previously interrupted export.
- The target destination must be the same as the previous interrupted export.
- The attributes files will be re-generated for ALL selected objects whether or not they were exported on the current run. If the previous existing attributes files are not deleted, DocXport will not overwrite them and the new attributes files will be written using the next number in the numbering sequence.

How it works

DocXport will go through each of the selected objects and check whether this object already exists in the target destination folder. If the object exists, DocXport will check that the last modify date on the disk is the same as the last modify date in the docbase. If it is, DocXport will NOT overwrite the file. If the last modify date has changed or if the file does not exist, DocXport will export that file to disk.

TOOLS

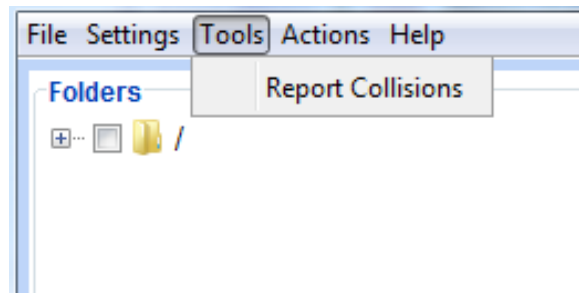


Figure 4: Tools menu bar items

REPORT COLLISIONS

The term **Collision** is used here to identify files which have the same name in the same directory in the docbase when only the current version is selected or the same name and same version number in the same directory when the export all versions is selected. This is possible in a Documentum docbase since Documentum uses the `r_object_id` attribute to uniquely identify objects. However, on most file systems, file names must be unique in a directory. In normal export mode, Docxport will resolve this problem by appending the `r_object_id` of the object to the file name when a collision occurs. This report allows the DocXport user to view which files in the docbase will “collide” with others during the export. The user can then choose to ignore these files or correct this issue directly in the docbase

ACTIONS

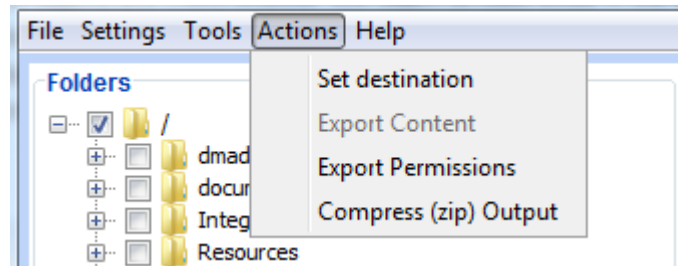


Figure 5: Actions menu items

SET DESTINATION

Displays a folder selection window to allow the user to select a target destination for the export. DocXport will use this as the Root folder for creating the exported file structure.

EXPORT CONTENT

Launches the document export. This operation will

- Export all selected documents while respecting folder structure and document names.
- Appends version numbers to document.
- Replace forbidden characters in document names or folder names with "-".
 - o The following are forbidden in windows file or folder names: \ : * ? " < > | **and will be automatically replaced during the export operation.**
- Generates an attributes file with the selected attributes for each of the selected types being exported.
- Generates an error log listing the files that could not be exported.

EXPORT PERMISSIONS

This command will export all permissions information of the selected objects. The ACLs will be exported as an XML file as per figure 6.


```

<?xml version="1.0" encoding="UTF-8" ?>
- <ACL name="dm_4500000680013535">
- <group group_name="docu">
+ <user>
- <user>
- <user_name>
  <![CDATA[ Michael Magenta  ]]>
</user_name>
- <user_os_name>
  <![CDATA[ mmagenta  ]]>
</user_os_name>
- <user_os_domain>
  <![CDATA[ synepsis  ]]>
</user_os_domain>
- <user_address>
  <![CDATA[ mmagenta@documentum.com  ]]>
</user_address>
<permit>7</permit>
<xPermit>3</xPermit>
</user>
+ <user>
+ <user>
+ <user>
</group>
+ <user>
- <user>
- <user_name>
  <![CDATA[ dm_world  ]]>
</user_name>
<user_os_name />
<user_os_domain />
<user_address />
<permit>7</permit>
<xPermit>0</xPermit>
</user>
</ACL>

```

Figure 6: Exported ACL sample output

COMPRESS (ZIP) OUTPUT

This command will create a zip file of each folder in the target root folder of the export. If no target is selected, a folder selection window will appear.

Note: Once this command is launched, there is no way of stopping the zip process.

FOLDER MAP

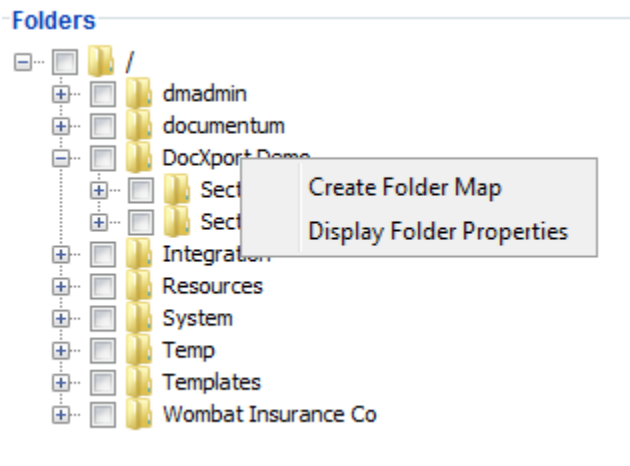


Figure 7: Folder map and folder properties function

This functionality is accessed by right clicking on a folder in the folder tree and selecting **Create Folder Map** (Figure 7). It generates an html rendered tree of all the files and folders under the selected tree node. This feature is useful for users who would like to see the detail of their folder's content for cleanup or inventory before exporting.

FOLDER PROPERTIES

This functionality is accessed by right clicking on a folder in the folder tree and selecting **Display Folder Properties** (Figure 7). It displays a summary of the objects contained in the selected folder (Figure 8).

The screenshot shows a dialog box titled '/DocXport Demo/Section B' with a close button (X) in the top right corner. The dialog contains a table with the following data:

Object Type	Count	Size (MB)
dm_document	21	14.0
dm_folder	5	0.0
Total	26	14.0

Below the table is an 'OK' button.

Figure 8: Folder properties

POWERXPORT MODE

PowerXport mode is a simplified interface for exporting documents. The output is similar to using the DocXport interface but some features are not available (Analyze, Create Navigation, Folder Map, Save, Load). This interface is optimized for exporting larger number of files (tens of thousands), as there is no visual display of the selected items. This allows the selection process to be faster.

To access the powerXport interface, enter your username/password in the login screen, select the docbase you wish to connect to and press the PowerXport button.

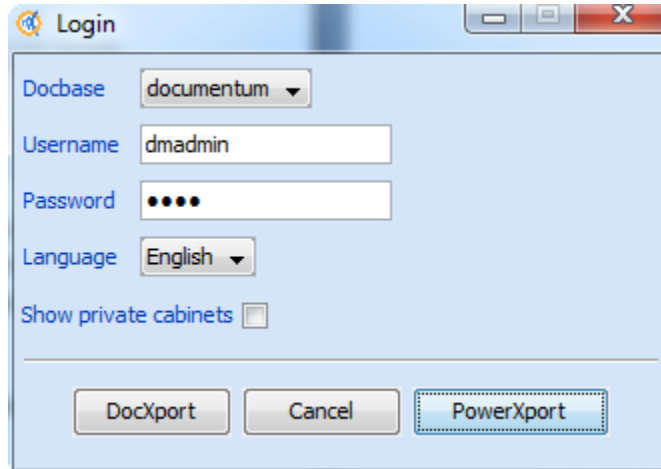


Figure 9: Starting powerXport interface

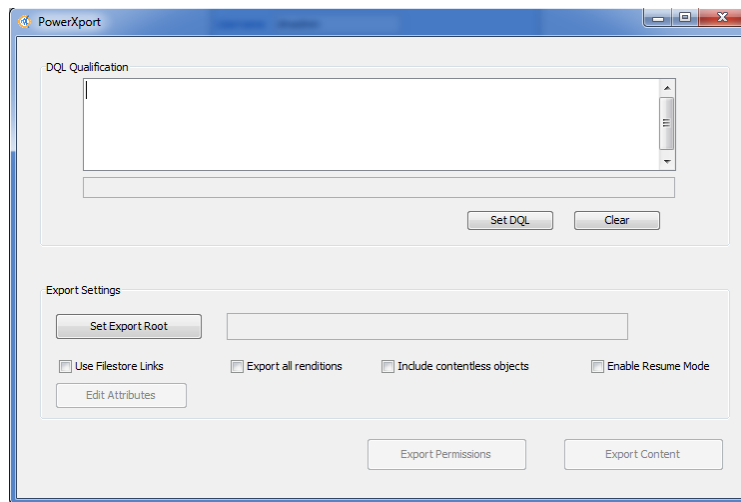


Figure 10: PowerXport interface

ENTERING A DQL QUALIFICATION

Selecting documents in the PowerXport mode is done using DQL (Documentum Query Language) qualifications. For example, if you wish to export all documents having the word “test” in the title attribute, you would enter the following DQL qualification in the DQL qualification input area:

Dm_document where title like '%test%'

Click the Set DQL button. If the qualification syntax is correct, the number of selected documents will appear under the DQL qualification input area. Otherwise, the error message will be displayed.

The Clear button will clear the DQL qualification.

CONFIGURING EXPORT SETTINGS

- In the export settings area, you can select the target directory will the exported files will be written on the disk. Clicking on the Set Export Root button will open a folder selection window. You can then select the target for your export.
- If you require only Docbase filestore links (see USE FILESTORE LINKS section above) check the *use filestore links* checkbox
- If you require all renditions for your export check the ***Export all renditions*** checkbox
- If you would like to include the attributes of content less objects, check ***Include content less objects***.
- You can specify which attributes you would like to include in the export by clicking the *Edit Attributes* button.
- If you would like to enable “RESUME” mode, check the appropriate checkbox.
- Once all the settings are in place, the buttons are activated:
 - Clicking the *Export Content* button will launch the export.
 - Clicking the *Export Permissions* button will export the permissions for the selected objects.

COMMAND LINE EXPORT

The DocXport tool can be used in command line mode. This mode allows a user to execute and export from the command line instead of using one of the two graphical user interfaces described above. In command line mode, the export configuration, the object selection and the attribute selection are done using an XML configuration file.

THE XML CONFIGURATION FILE

Figure 10 shows a sample configuration file for the command line export.

```

<?xml version="1.0"?>
- <Extract>
  - <exportParams>
    <Renditions includeRenditions="true"/>
    <ContentStoreLink isUseContentStoreLink="false"/>
    <Navigation isGenerateNav="false"/>
    <exportContentlessXML isExportContentlessXML="false"/>
    <showAllVersions isShowAllVersions="true"/>
    <PrivateCabinets isShowPrivateCabs="false"/>
    <resumeMode isResumeMode="true"/>
    <ExtractLocation target="C:\Users\tcs2010\Desktop\random"/>
    <!-- XML or EXCEL (case sensitive) -->
    <attributeFormat format="EXCEL"/>
  </exportParams>
  - <attributes>
    + <dm_document>
  </attributes>
  - <DQL>
    <Qualification>dm_document (all) where folder('/DocXport Demo/Section A',descend)
    </Qualification>
  </DQL>
  <ACL>false</ACL>
</Extract>

```

Figure 11: Sample XML export configuration file

THE <ATTRIBUTES> NODE

The <attributes> node defines which attributes will be included in the attributes output file for attributes values. It can contain one or more object type nodes, dm_document or one of its sub types. Each of these types contains sub nodes with the attributes of those types. Setting an attribute value to true means this attribute will be included in the attributes output file containing attributes values of the exported documents. See figure 12 for examples.

```

- <Extract>
- <attributes>
  - <dm_document>
    <r_lock_date>false</r_lock_date>
    <group_name>false</group_name>
    <a_extended_properties>false</a_extended_properties>
    <i_reference_cnt>false</i_reference_cnt>
    <r_component_label>false</r_component_label>
    <r_alias_set_id>false</r_alias_set_id>
    <r_modifier>false</r_modifier>
    <i_cabinet_id>false</i_cabinet_id>
    <i_retainer_id>false</i_retainer_id>
    <i_folder_id>false</i_folder_id>
    <a_is_template>false</a_is_template>
    <r_full_content_size>false</r_full_content_size>
    <ad_name>false</ad_name>
    <subject>false</subject>
    <r_composite_label>false</r_composite_label>
    <title>false</title>
    <r_has_frzn_assembly>false</r_has_frzn_assembly>

```

Figure 12: Defining required attributes for a type

THE <EXPORTPARAMS> NODE

The <exportParams> node can contain the following child nodes:

<Rendition includeRenditions="true/false"/>: If the value of this attribute is set to true, all renditions of an object are exported. If set to false, only the primary rendition is exported.

<ContentStoreLink isUseContentStoreLink="true/false"/>: If this value is set to false, the exported files will be copied to the target location. If set to true, only a reference to the file in the content store will be set in the attributes xml file (see USE FILESTORE LINKS IN XML section above).

<Navigation isGenerateNav="true/false"/>: If this value is set to True, DocXport will generate a html file to navigate through the exported files.

<exportContentlessXML isExportContentLessXML="true/false"/> Setting this value to true will force DocXport to write attributes information for dm_syobjects or subtype objects that have no content objects associated to them.

<showAllVersions isShowAllVersions="true/false"/>: if this value is set to true, only the current version of each document will be exported. If this value is set to false, all version of a document will be exported.

<PrivateCabinets isShowPrivateCabs="false"/>: this value should not be modified.

<resumeMode isResumeMode="true/false"/>: Setting this value to true will enable resume mode for the exports. (see RESUME MODE section above).

<ExtractLocation target="<fully qualified folder path"/>: Defines the target location for the export.

<attributeFormat format="EXCEL" />: Selects if the attributes output should be XML or EXCEL.

THE <DQL> NODE

The DQL node is where the object selection takes place. This node can contain one or more <Qualification> tags with dql qualification statements. When running the export, DocXport will execute each DQL qualification sequentially and add the selected documents to the export selection. (See Figure 11 for examples).

LAUNCHING A COMMAND LINE EXPORT.

Once the parameter file has been correctly configured, edit the **command_export.bat** file found in the root DocXport directory. The file contains the following command line:

```
java -Xmx1024m -Ddfc.properties.file=C:/Documentum/config/dfc.properties -cp DocXport.jar  
com.synopsis.docxport.DocXport <docbase_name_case_sensitive> <user_name> <password>  
<params_file_path>
```

The command requires the following arguments:

- 1) The docbase name (case sensitive).
- 2) The Documentum user name to connect to the docbase.
- 3) The Documentum password.
- 4) The full or relative path to the parameter configuration file.

Replace the values in the **launch_command_line.cmd** with the correct values for your docbase and system.

Note: In order for the command to launch correctly:

- 1) The java.exe executable must be found on the path of the operating system. Otherwise, the full path to the java.exe must be given in the command
- 2) The dfc.properties file path must be the correct one on the local machine where the command is executed
- 3) If the relative path is used for the parameter file, the command must be launched from within the directory where the parameter file is located.

Launch the export by executing **launch_command_line.cmd**

Q&A

1. When I select a folder using the DocXport visual interface or DQL qualification, I sometimes get less files selected than what I get using a Select count(*) from <object_type> query. Is this a bug?

This is not a bug. When using the DQL query select count(*) on folders that have files with multiple links (multiple i_folder_id attribute values), the count(*) query will count these objects multiple times. However, DocXport will only select/export the element one time. In order to count unique values under a folder, use the following query:

```
SELECT count (distinct r_object_id)
FROM dm_document (ALL)
where folder('<folder_path>', descend)
```

KNOWN ISSUES AND BUGS

DOCBASE ROOT NODE SELECTION GENERATES AN ERROR

When trying to select the root node of the Docbase, DocXport generates an error and no selection is made.

DQL QUERY VIEW IN DOCXPORT MODE DOES NOT GIVE FEEDBACK ON THE DQL ERROR.

If an erroneous DQL qualification is entered in the DQL query selection window, no error is displayed. However, the window does not disappear unless the user enters a correct query or presses the cancel button.

LINK TO FILES WITH A POUND (#) SIGN IN THE FILE PATH WILL NOT WORK IN GENERATED HTML FILE VIEWER

Documentum server allows document and folder names to contain the # character. This is a special character in html used for defining anchors. When displaying the html navigator, the web browser interprets the # sign as an anchor in the link to the file and therefore cannot display the exported file.

DQL HINTS ARE NOT SUPPORTED IN DQL INPUT

DQL hints such as **ENABLE(RETURN_TOP 5)** to limit the number of returned objects are not supported by DocXport.

JAVA ERROR WHEN LAUNCHING DOCXPORT

java.lang.UnsatisfiedLinkError: C:\Program Files\Documentum\Shared\dmcl40.dll: Can't load IA 32-bit .dll on a AMD 64-bit platform. This error occurs when a 64 bit java runtime is used to launch a Documentum application. To correct this issue, please install a 32 bit JVM.

Please report any bugs to support@synopsis.com