Glasswall SDK
Command Line Interface (CLI) Guide

SDK Version 2.0 (Early Release)

Document Version 0.6

Copyright and Contact Details

If there are any questions related to this document, these should be addressed to:

Glasswall Solutions Limited

e-mail: support@glasswallsolutions.com

Document History

Issue Date	Version	Description
31/10/19	0.1	Initial Draft
31/10/19	0.2	Feedback Edits
01/11/19	0.3	More Feedback Edits
08/11/19	0.4	Addition of Setup
14/11/19	0.5	Addition of Using Glasswall and edit of Setup
15/11/19	0.6	Feedback Edits

Contents

1. Introduction

This is a guide to using Glasswall CLI in the command prompt for windows or terminal window for linux with an explanation of the parameters and using the configuration files.

Technical terminology used within this document is aligned with that used in the Glasswall Product Description document and readers are directed to look there for any Glasswall speicifc terms to find their meaning.

2. Using Glasswall

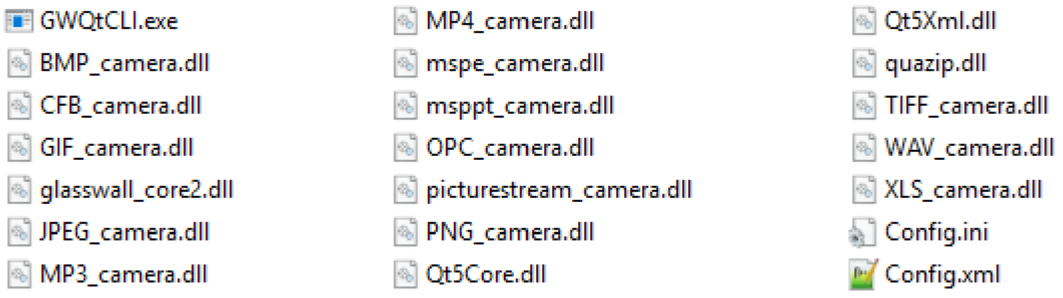
2.1. What to to use Glasswall CLI for

The best use of Glasswall CLI is as a test or evaluation tool. The results provided can be examined and compared to give you a clear understanding of any issues or unwanted content within the files. Any invalid files will have the issues reported in the XML documents to show why they were non-conforming. You will also be able to see if any disallowed content was present that made the file non-conforming and decide if you want to rerun it in sanitise mode to remove that content.

2.2. What Glasswall CLI is not

The Glasswall CLI is not suitable to be embedded within production processing systems. It has been designed as a test aid so does not have the robustness of the main Glasswall libraries.

3. Step by Step Setup



GWQtCLI.exe
BMP_camera.dll
CFB_camera.dll
GIF_camera.dll
glasswall_core2.dll
JPEG_camera.dll
MP3_camera.dll
MP4_camera.dll
mspe_camera.dll
msppt_camera.dll
OPC_camera.dll
picturestream_camera.dll
PNG_camera.dll
Qt5Core.dll
Qt5Xml.dll
quazip.dll
TIFF_camera.dll
WAV_camera.dll
XLS_camera.dll
Config.ini
Config.xml

1. Copy all library files into one folder
2. Copy the CLI to the same folder

3. Copy the sample config files to the same folder (the contents should now look like the image above)
4. Open the terminal
5. Use pushd to set the present working directory to the folder containing all the above files
6. **In Linux only** the following is required for security tagging fucntions to work; In ther terminal window enter: *export LD_LIBRARY_PATH="Core2 library location"*.
7. Run glasswall using the input/output or config file options described below
8. Once this is complete you can use the results to check for any unwanted content, the xml files will contain information about each file and any issues found within it if it was non conforming.

4. Parameters

Only the highlighted single key parameters work on Linux. Where values strings are supplied with paramters then the value must be in quotes if it contains spaces.

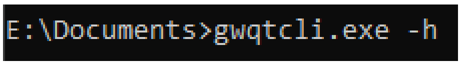
Parameter	**Description**
-?, -h, --help	Displays the help information with the list of parameters and what they are used for.
-v, --version	Displays the Glasswall version information.
-i, --inputdirectory	Used to specify the location of the folder containing the files to be processed which is given after the parameter. E.g. -i "e:\cli\test data\oxml".
-o, --outputdirectory	The location you want the processed files to go to. This can either be an existing folder using a file path, a new folder in a specified location or a new folder within the current directory. The value is supplied after the parameter, e.g –o "e:\cli\output\results".
-c, --xmlConfig	This directs the application to the content management file containing the options for how to manage the file content. Usage –c "C:\cli\config.mxml".

-s, --storageMode	Selecting the storage mode, default is File to File. 0 = file to file, 1 = file to memory and 2 =memory to memory, e.g –s 3.
-x, --exportimport	Sets the import export option, either import or export. Export will export any embedded content and import will reimport the safe files that were exported E.g. –x export.
-n, --config	This directs the application to the configuration INI file containing your chosen settings instead of using the parameters in the terminal, e.g. –n “c:\cli\config.ini”. Note: cli expect the –n and –c paramters to be used together.
-m, --concurrent	Tells Glasswall to run in multithreaded mode.
-r, --recurse	Directs the application to recursively process all files in the subfolders of the specified input directory.
-d, --determineftype	This function has been deprecated.
-t, --securTags	This is used to tell Glasswall to insert the security tags specified in the supplied XML file containing the tags, e.g. –t “e:\cli\tagging.in”.
-u, --securRetrieveTags	This retrieves the security tags that Glasswall finds in the files of the specified input directory and outputs the tags to the directory specified following this parameter, e.g. –u e:\cli\tagsout”

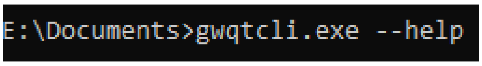
5. Examples

5.1. Help

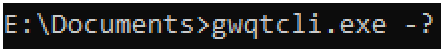
Each of the below inputs gives you the following output.



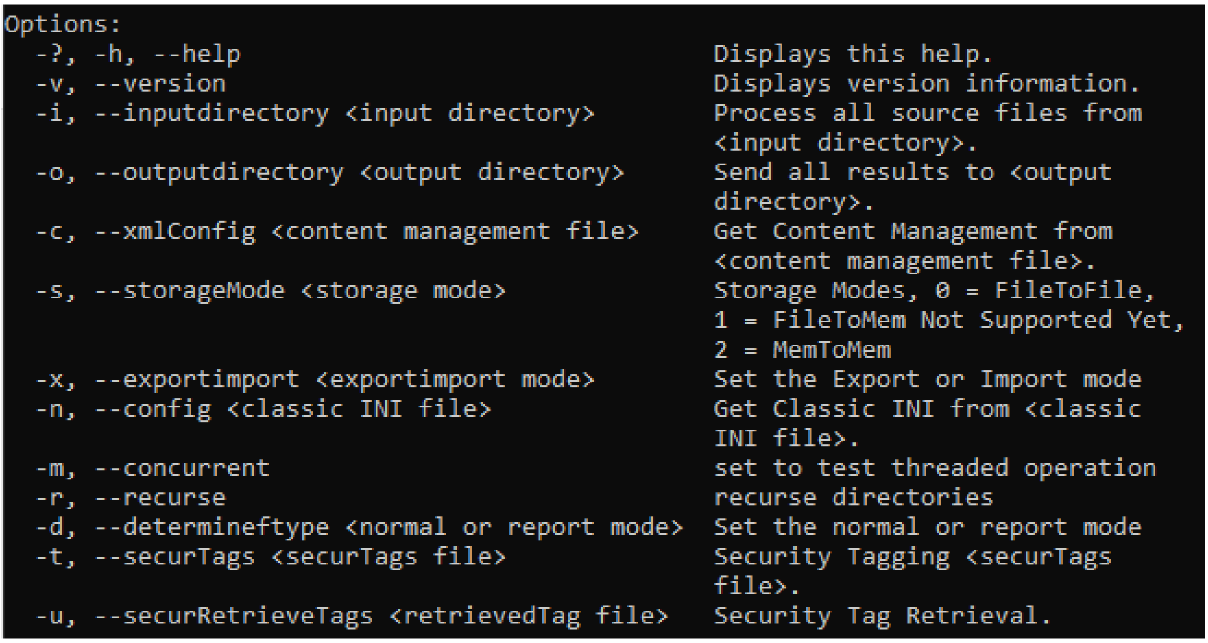
```
E:\Documents>gwqtcli.exe -h
```



```
E:\Documents>gwqtcli.exe --help
```



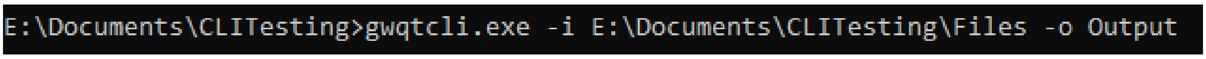
```
E:\Documents>gwqtcli.exe -?
```



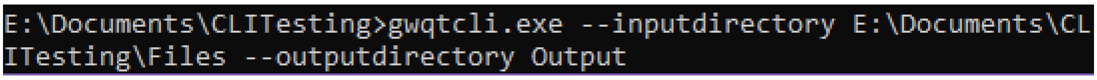
```
Options:
  -?, -h, --help                              Displays this help.
  -v, --version                               Displays version information.
  -i, --inputdirectory <input directory>      Process all source files from
                                              <input directory>.
  -o, --outputdirectory <output directory>    Send all results to <output
                                              directory>.
  -c, --xmlConfig <content management file>   Get Content Management from
                                              <content management file>.
  -s, --storageMode <storage mode>            Storage Modes, 0 = FileToFile,
                                              1 = FileToMem Not Supported Yet,
                                              2 = MemToMem
  -x, --exportimport <exportimport mode>      Set the Export or Import mode
  -n, --config <classic INI file>             Get Classic INI from <classic
                                              INI file>.
  -m, --concurrent                            set to test threaded operation
  -r, --recurse                               recurse directories
  -d, --determineftype <normal or report mode> Set the normal or report mode
  -t, --securTags <securTags file>            Security Tagging <securTags
                                              file>.
  -u, --securRetrieveTags <retrievedTag file>  Security Tag Retrieval.
```

5.2. Using Input and Output

Option 1:



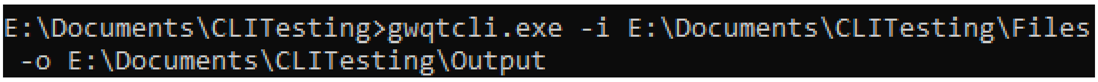
```
E:\Documents\CLITesting>gwqtcli.exe -i E:\Documents\CLITesting\Files -o Output
```



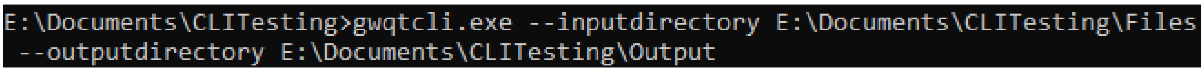
```
E:\Documents\CLITesting>gwqtcli.exe --inputdirectory E:\Documents\CL
ITesting\Files --outputdirectory Output
```

Using this option would process the files in the Files folder and create an Output folder inside the CLI Testing folder containing the files created by Glasswall.

Option 2:



```
E:\Documents\CLITesting>gwqtcli.exe -i E:\Documents\CLITesting\Files
 -o E:\Documents\CLITesting\Output
```



```
E:\Documents\CLITesting>gwqtcli.exe --inputdirectory E:\Documents\CLITesting\Files
 --outputdirectory E:\Documents\CLITesting\Output
```

Using this option would process the files in the Files folder and put the processed files in the already existing Output folder or create the output folder in the specified location if it doesn’t already exist.

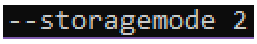
5.3. Storage Mode

Storage Modes:

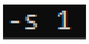
Default or 1: File to File

2: File to Memory (not currently implemented and runs as 1)

3: Memory to Memory



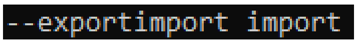
```
--storagemode 2
```



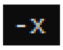
```
-s 1
```

Either of the above options will set the storage mode to your chosen setting, this parameter can be used along with the input and output parameters and the processed content will be put into the output folder specified.

5.4. Import/Export



```
--exportimport import
```

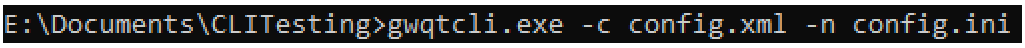


```
-x
```

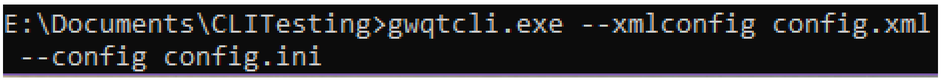
Either of the above options will set the import/export parameter to your chosen setting. This parameter can be used along with the input and output parameters and the processed content will be put into the output folder specified.

The format of the export can be controlled using a setting in the content management XML config file, it can be set to SISL(default) or XML. See example in 5.5.

5.5. Config Files



```
E:\Documents\CLITesting>gwqtcli.exe -c config.xml -n config.ini
```



```
E:\Documents\CLITesting>gwqtcli.exe --xmlconfig config.xml
 --config config.ini
```

XML Config File:

The XML config file is used to decide how content is managed within each file type. The default setting for each content type is sanitise which means the files will have this content removed or passed down to be processed by sub cameras during the process, if any content type or file type is missing from this document they will be sanitised. The other options are allow and disallow, if you allow content this means it will not be removed from the file. If you disallow content the whole file will be rejected and classed as non-conforming. Below is a sample config.xml file. The highlighted section changes the export format to XML.

```
<?xml version="1.0" encoding="utf-8"?>
<config>

<pdfConfig>
<watermark>Glasswall Approved</watermark>
<metadata>sanitise</metadata>
<acroform>sanitise</acroform>
<javascript>sanitise</javascript>
<external_hyperlinks>sanitise</external_hyperlinks>
<actions_all>sanitise</actions_all>
<embedded_files>sanitise</embedded_files>
<non_conforming_images>disallow</non_conforming_images>
</pdfConfig>
```

```
<wordConfig>
<embedded_files>sanitise</embedded_files>
<metadata>sanitise</metadata>
<macros>sanitise</macros>
<external_hyperlinks>sanitise</external_hyperlinks>
</wordConfig>

<pptConfig>
<embedded_files>sanitise</embedded_files>
<metadata>sanitise</metadata>
<macros>sanitise</macros>
<external_hyperlinks>sanitise</external_hyperlinks>
</pptConfig>

<xlsConfig>
<embedded_files>sanitise</embedded_files>
<metadata>sanitise</metadata>
<macros>sanitise</macros>
<external_hyperlinks>sanitise</external_hyperlinks>
</xlsConfig>

<sysConfig>
<interchange_type>xml</interchange_type>
<interchange_pretty>true</interchange_pretty>
</sysConfig>

</config>
```

INI Config File:

The INI config file can be an INI or TXT file, it contains most of the parameters in the terminal, you can set the report mode, storage mode, input location and output location. The only details that are processed by QTCLI are the highlighted lines in the example below. The remainder are there for backwards compatibility with Glasswall Classic CLI.

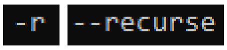
```
[GWConfig]
processMode                 = 8
reportMode                  = 0
fileStorageMode             = 1
fileType                    = *

inputLocation               = E:\Documents\CLITesting\Files
useSubfolders               = 1


outputLocation              = output
createOutputFolders         = 1
nonConformingDirName        = NonConforming
managedDirName              = Managed


writeOutput                 = 1

logFileSize                 = 0
logFileProcessTime          = 1
```

5.6. Multithreading

`-m` `--concurrent`

The above parameters will tell the application to run in multithreaded mode, this involves processing files concurrently. This can reduce the time it takes to process a set of files because multiple files can be processed at the same time up to the ability of your computer to handle.

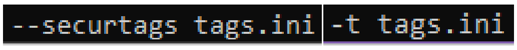
5.7. Recurse



```
-r --recurse
```

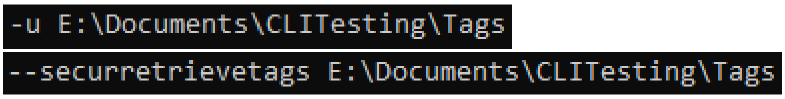
The above parameters will tell the application to process all files within the subfolders of the provided input directory. For example if you have 5 subfolders containing files within the folder Test they would all be processed through the application.

5.8. Security Tags and Retrieval



```
--securtags tags.ini -t tags.ini
```

The above parameters will insert the security tags provided in the tags.ini file. See example below.



```
-u E:\Documents\CLITesting\Tags
--securretrievetags E:\Documents\CLITesting\Tags
```

The above parameters will retrieve the security tags from the files it is processing and save the output in the folder provided. Below is an example of the file required for inserting tags and the output of retrieved tags.

```
<digitalsignature>
<s1>20191012123106</s1>
<s2>112.31.04.102</s2>
<s3>9C-35-5B-5F-4C-D7</s3>
<s4>gw.localhost</s4>
<s5>20191012083106</s5>
<s6>20191012092106</s6>
<s7>20191015123106</s7>
<s8>Fred Smith</s8>
<s9>Fred Smith</s9>
<s10>Tina Gray</s10>
<s11>RegisterOutputMem Mem TODO</s11>
</digitalsignature>
```