



FORENSIC EXPLORER

Command Line

User Manual

Published: 28-Oct-24 at 11:32:13



Chapter Contents

Chapter 1 - Introduction.....	5
Introducing Forensic Explorer CLI Interface (CLI)	7
Chapter 2 – Installation & Activation	9
2.1 Purchase	11
2.2 Licensing	11
2.3 Activation.....	11
2.4 Maintenance.....	14
2.5 Confirming Activation Status	14
Chapter 3 - Overview	15
3.1 Overview.....	17
Chapter 4 – Installation.....	19
4.1 Installation	21
4.2 Setup of the Windows Environment.....	22
Chapter 5 – CLI Execution.....	27
5.1 CLI Execution.....	29
Chapter 6 – CLI Tools.....	33
6.1 CLI Tools.....	35
Chapter 7 – TXML Processing File	37
7.1 TXML Processing File	39
7.2 XML Terminology.....	39
7.3 Forensic Explorer TMXL Elements	39
7.4 Natural Order of Command Tasks	40
7.5 TCommandTask_Parallel	41

Appendix 1 – Sample Batch Files	43
Appendix 2 – TXML Command Tasks.....	51
Appendix 3 - Forensic Explorer File Driver	65
Appendix 4 – CLI Error Codes	70

Chapter 1 - Introduction

In This Chapter

CHAPTER 1 - INTRODUCTION

Introducing Forensic Explorer CLI Interface (CLI)	7
---	---

INTRODUCING FORENSIC EXPLORER CLI INTERFACE (CLI)

Forensic Explorer is a tool for the analysis and presentation of electronic evidence. Primary users of this software are those involved in civil or criminal investigations.

The **Forensic Explorer CLI** is a stand-alone version of Forensic Explorer that enables commands to be executed from the Windows Command Line. The benefits of CLI processing are:

- **Repeatability:** Repeat exact processing tasks for each execution.
- **Expandability:** Run concurrent instances. Expand processing capabilities with server installations.
- **Customizable:** Customize processing tasks using XML and scripts.
- **Speed:** Significant faster processing speed than GUI.

Cases created with the **Forensic Explorer CLI** can be opened with the **Forensic Explorer GUI** version.

Chapter 2 – Installation & Activation

In This Chapter

CHAPTER 2 – INSTALLATION & ACTIVATION

2.1	Purchase	11
2.2	Licensing	11
2.3	Activation.....	11
2.4	Maintenance.....	14

2.1 PURCHASE

To purchase a **Forensic Explorer CLI** license, contact sales@getdata.com. Separate license key/s for the CLI are provided with your purchase and can be applied to a Wibu Codemeter USB hardware activation dongle.

2.2 LICENSING

A Forensic Explorer CLI license is independent of the Forensic Explorer GUI version. A license is required for each concurrent instance of the CLI. For example:

- If a single Forensic Explorer CLI instance is run, and 10 jobs are process sequentially, 1 license is needed.
- If 10 concurrent instances of the Forensic Explorer CLI are executed, 10 licenses are needed to activate each concurrent instance.

GetData Forensics can provide a license solution for your organization. This can be individual licensing, site licensing, or global licensing. Contact sales@getdata.com for assistance.

2.3 ACTIVATION

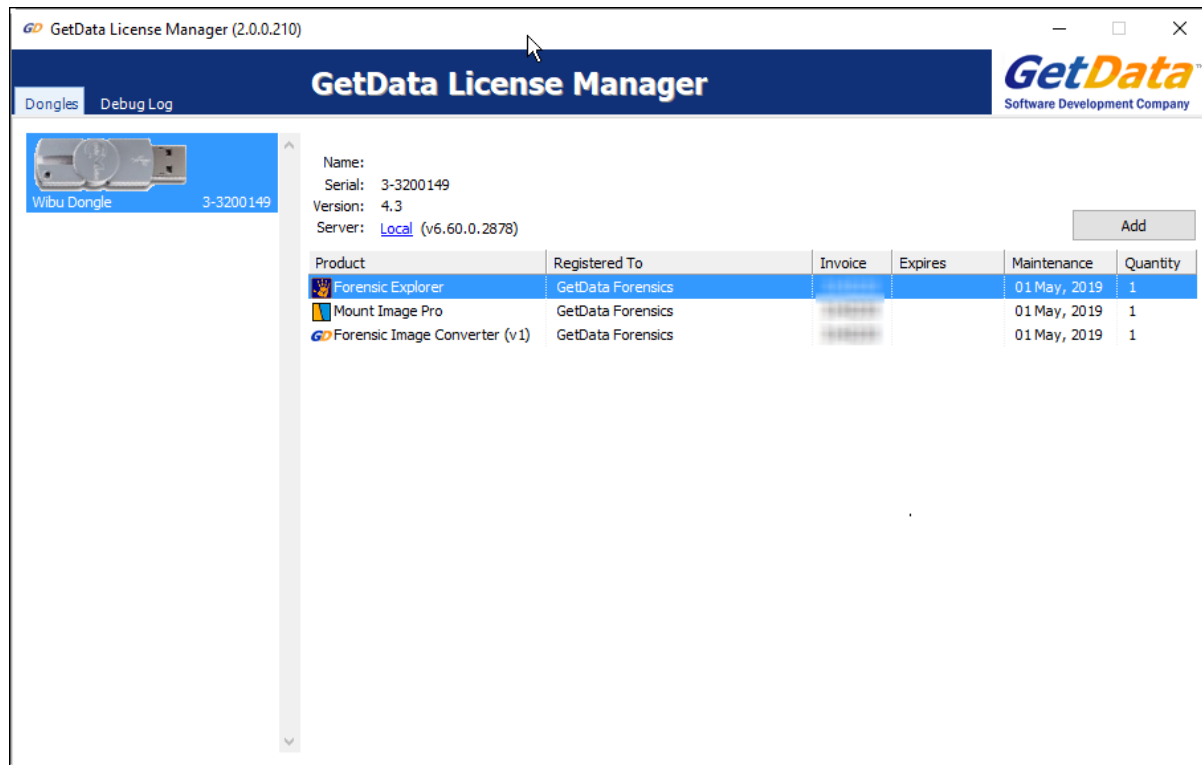
The **Forensic Explorer CLI version** is activated using a license file located on a **Wibu CodeMeter hardware dongle**. The dongle can be inserted locally, or on a network server.

For a new purchase which includes the Forensic Explorer CLI version, a dongle will be supplied already containing all necessary licenses.

To **add** a Forensic Explorer CLI license/s to an **existing dongle**:

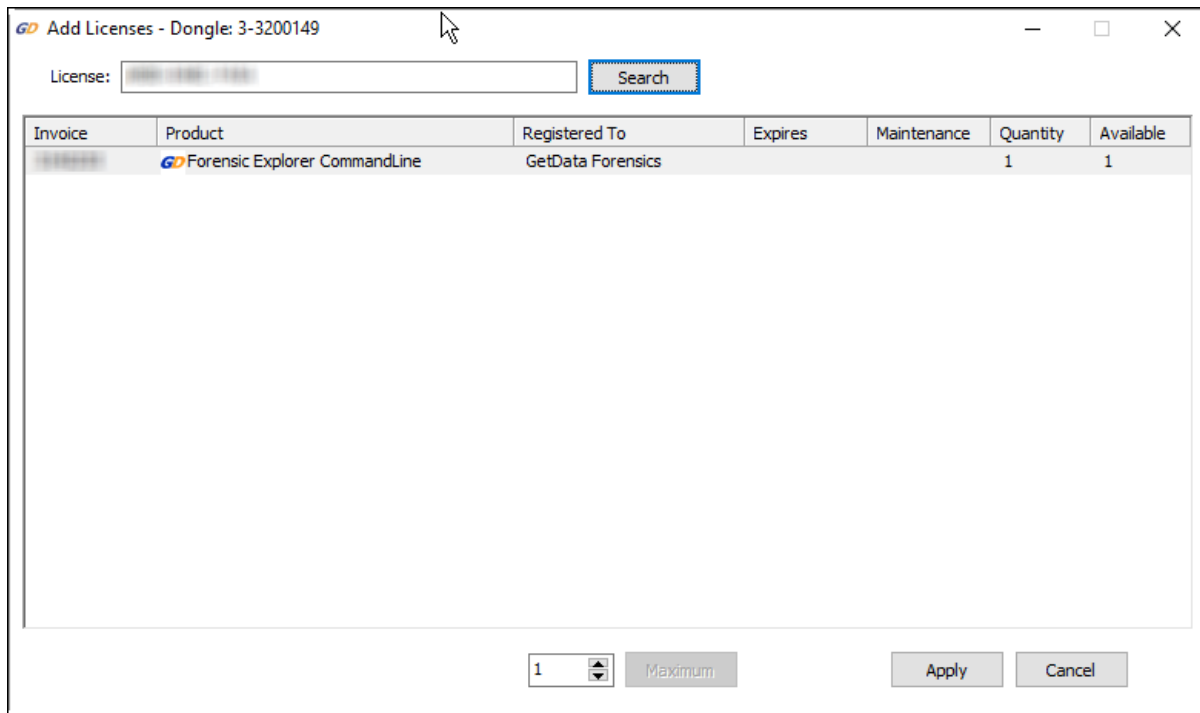
1. On a computer that has **internet access**, **insert your Forensic Explorer Wibu dongle** into a USB port. Remove any other Wibu dongles that you may have for other products.
2. Run the **GetData License Manager** located in the installation folder of Forensic Explorer. The default location is: **C:\Program Files\GetData\Forensic Explorer vX\License Manager.exe** or download from <http://download.getdata.com/support/LicenseManager.exe>
3. The GetData License Manager will **detect your Wibu dongle**, as shown in Figure 1 below. An existing Forensic Explorer dongle should show a license for:
 - a. Forensic Explorer,
 - b. Mount Image Pro, and
 - c. Forensic Image Converter.

Figure 1: GetData License Manager



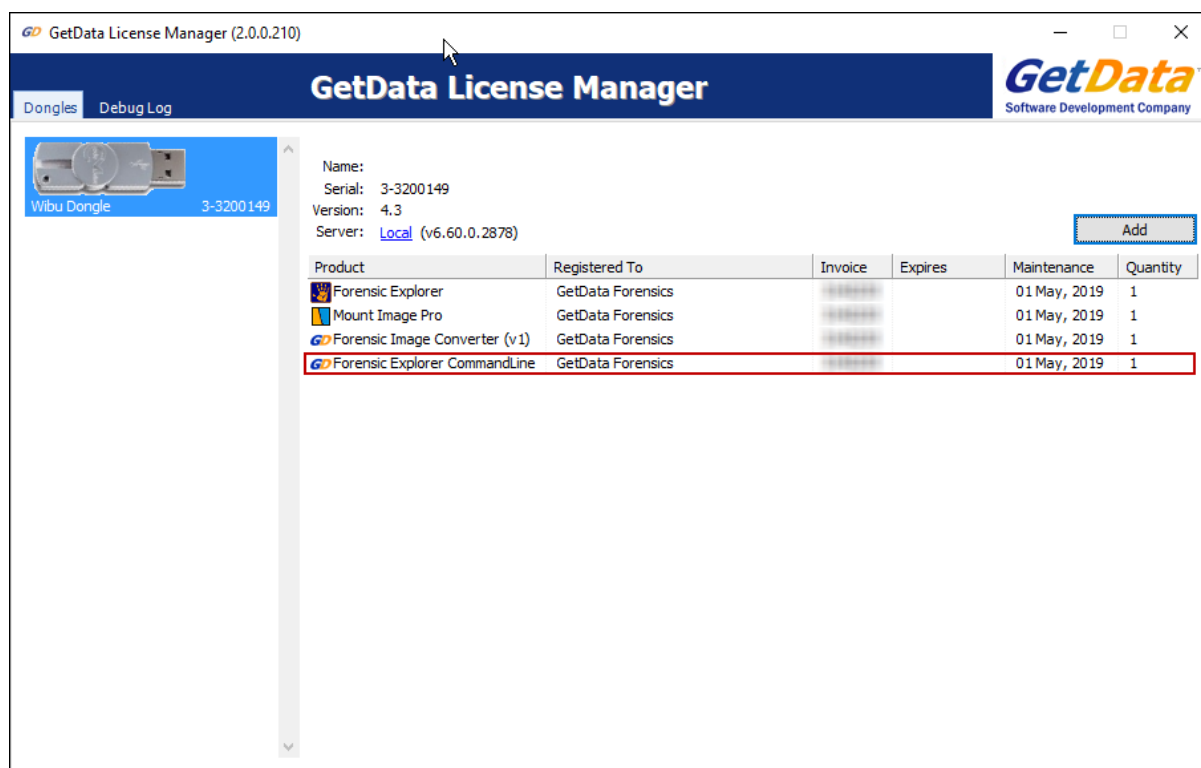
4. Click the **Add** button, shown in Figure 2 below to add a new license/s. Search for the license key provided with your purchase. The license will display showing the quantity and availability (i.e., not already written to a dongle).
5. A license is required for each instance of the command line that is launched. A dongle may be programmed with multiple licenses. To do this, use the number at the bottom of the screen, or click the **Maximum** button to apply all available licenses to the dongle.

Figure 2: GetData License Manager, Add Licenses



6. Once the license/s is successfully applied it should display in the main window of the GetData License Manager, as shown in Figure 3 below:

Figure 3: GetData License Manager, added CLI license.



2.4 MAINTENANCE

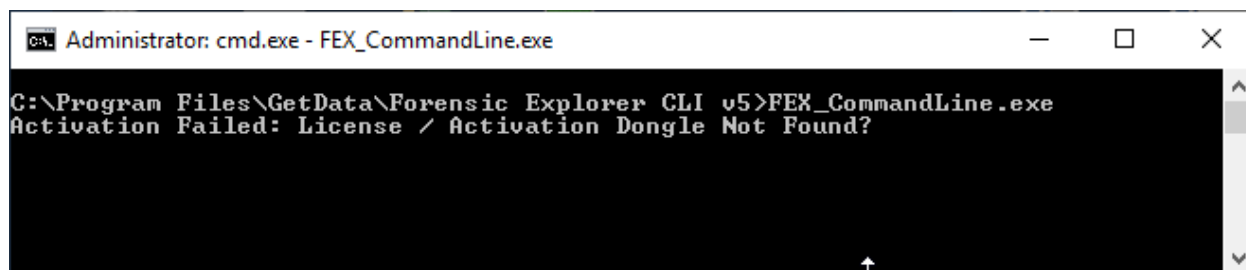
To purchase maintenance for the **Forensic Explorer CLI**, contact sales@getdata.com.

To apply maintenance, follow the procedure described above to add the maintenance key to the dongle. Do not delete any key prior to adding the maintenance key. Successful application will be reflected in the new **Maintenance** date shown the GetData License manager main screen.

2.5 CONFIRMING ACTIVATION STATUS

To confirm Forensic Explorer CLI activation status, type the Forensic Explorer CLI help command **FEX_CommandLine.exe** into the DOS window. If the software is **not** activated an error message shown in Figure 4 below will be displayed:

Figure 4: Confirming activation status (shows a physical dongle that does not have a license)



Chapter 3 - Overview

In This Chapter

CHAPTER 3 - OVERVIEW

3.1	Overview.....	17
-----	---------------	----

3.1 OVERVIEW

Cases created with the **Forensic Explorer CLI version**, or the **GUI version** have the same underlying structure. They are interchangeable, in that cases created in the CLI version can be opened by the GUI version and visa-versa.

The Forensic Explorer CLI version consists of two primary components:

1. The Command Line Arguments

The command line arguments are used to setup and manage the case. For example:

- Create or open a case;
- Attribute an investigator to the case;
- Add evidence;
- Launch a processing TXML file (see below);
- Save the case.

2. The processing TXML file

The TXML file is used to execute specific processing tasks for a case. These are the same processing tasks found when running the GUI version. For example:

- Verify the hash of a forensic image in the case;
- Run Signature Analysis;
- Carve for files;
- Hash files;
- Hash match;
- Keyword search;
- Filter files (e.g., JPG);
- Create a .L01 of specific files (e.g., JPG);
- Export files to an external folder (e.g., JPG); etc.

Chapter 4 – Installation / Setup

In This Chapter

CHAPTER 4 – INSTALLATION

- 4.1 Installation21
- 4.2 Setup of the Windows Environment.....22
 - 4.2.1 Administrator Access22
 - 4.2.2 CLI / PowerShell22
 - 4.2.3 Add Forensic Explorer CLI to the Windows Path.....23

4.1 INSTALLATION

Setup of the Forensic Explorer CLI version may vary according to the needs of your organization and the tasks for which it is used. Setup can range from a single user on a laptop, to a blade server spawning multiple virtual machines with a custom management console. This manual is intended only as a basic setup guide. Please contact support@getdata.com if further information or assistance is required.

The **Forensic Explorer CLI** is distributed as:

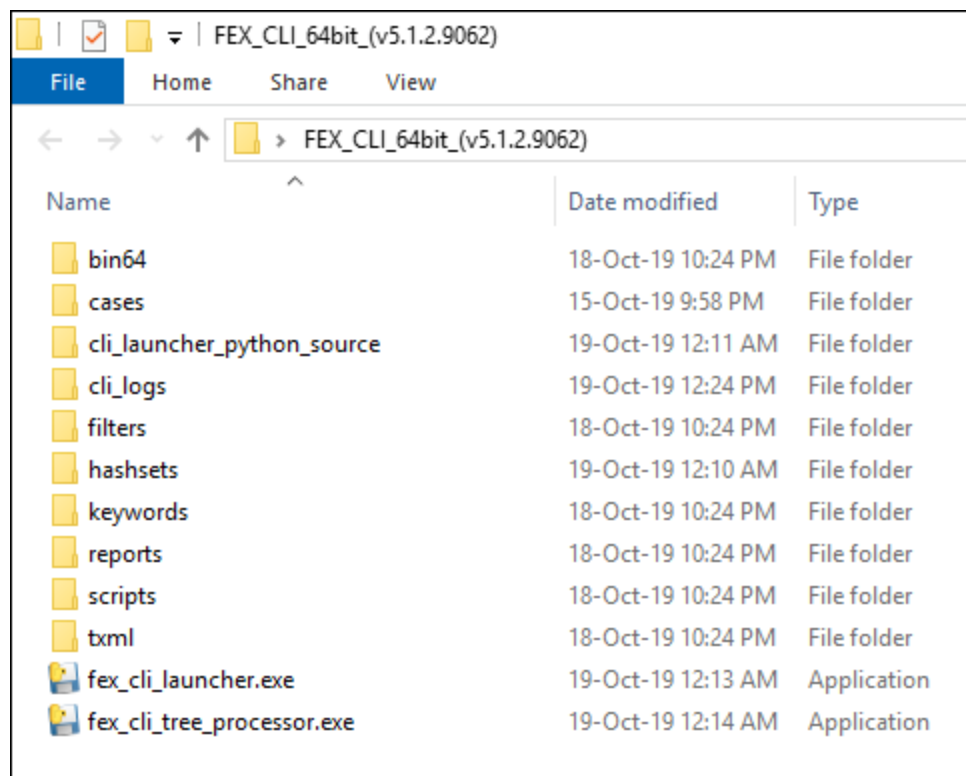
- A stand-alone ZIP file (portable version). The ZIP file contains all necessary components to run the CLI.
- A window installation file. The default installation path is: *C:\Program Files\GetData\Forensic Explorer CLI v5*.

It is not a requirement that the Forensic Explorer GUI version also be installed.

Important: It is recommended that new users install and keep the Forensic Explorer CLI version separate from the GUI version. Many of the filters and scripts in the GUI version have **GUI elements** that **will not execute** in the CLI.

The CLI installation ZIP file can be extracted to any location. The following folders are installed:

Figure 5: Forensic Explorer CLI (ZIP portable version shown)



bin64	The folder containing the FEX CLI program files, including FEX_CommandLine.exe
cases	An empty folder that can be used to store created cases
cli_launcher_python_source	Python source code for the fex_cli_launcher.exe and fex_cli_tree_processor.exe

cli_logs	An empty folder that can be used to store FEX CLI logs
filters	Filters (.pas) that can be executed by the FEX CLI (called in the TXML processing file)
hashsets	Sample hash sets
keywords	Sample keywords
reports	Sample reports
scripts	Scripts (.pas) that can be executed by the FEX CLI (called in the TXML processing file)
txml	XML file which sets out the FEX CLI processing tasks
fex_cli_launcher.exe	Python executable used to launch individual FEX CLI jobs
fex_cli_launcher.json	A JSON file used to store settings from the previous launch of <code>fex_cli_launcher.exe</code>
fex_cli_tree_processor.exe	The python executable that is the subject of this user guide
fex_cli_tree_processor.json	A JSON file used to store settings from the previous launch of <code>fex_cli_tree_processor.exe</code>

4.2 SETUP OF THE WINDOWS ENVIRONMENT

4.2.1 ADMINISTRATOR ACCESS

It is **recommended** to launch Forensic Explorer CLI with **administrator privileges**.

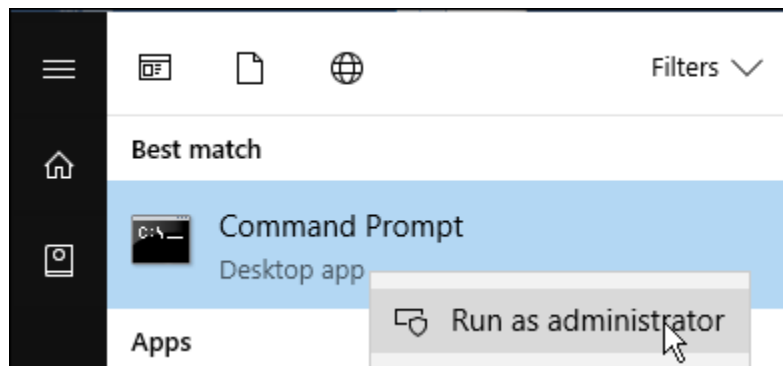
If not running as administrator, careful consideration should be given to ensure Forensic Explorer CLI has access privileges to all required paths (including forensic image files, case and output folders).

4.2.2 CLI / POWERSHELL

Forensic Explorer CLI and run under Windows CLI or Windows PowerShell.

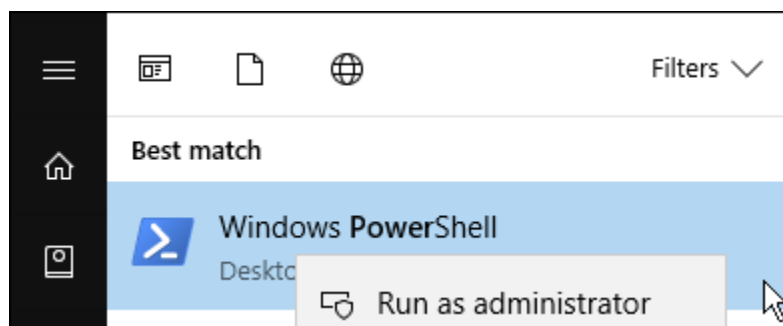
WINDOWS CLI

To launch a Windows CLI window, type CLI in the Windows search bar. To run as **administrator**, right click on the **Command Prompt** search results and select **Run as administrator** from the drop-down menu, as shown in Figure 6:

Figure 6: Launch Windows CLI as administrator (Windows 10 shown)

WINDOWS POWERSHELL

To launch a Windows PowerShell, type PowerShell in the Windows search bar. To run as **administrator**, right click on the **Windows PowerShell** search results and select **Run as administrator** from the drop-down menu, as shown in Figure 7:

Figure 7: Launch Windows PowerShell as administrator (Windows 10 shown)

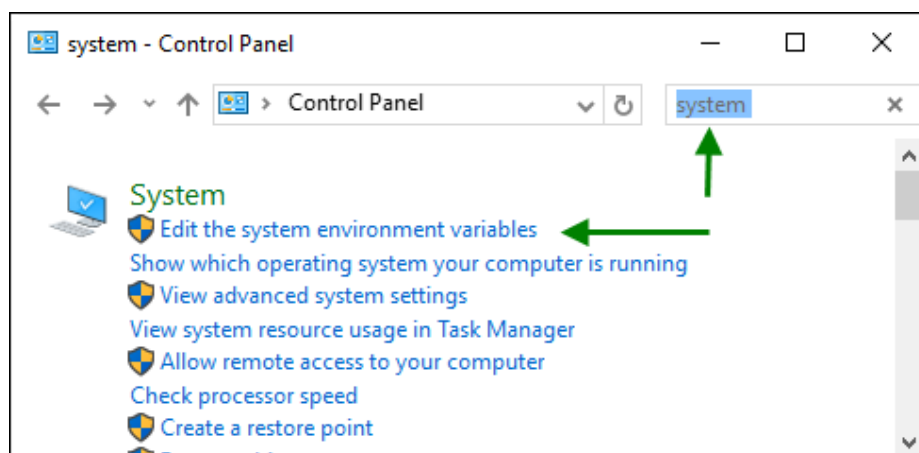
4.2.3 ADD FORENSIC EXPLORER CLI TO THE WINDOWS PATH

Frequent users of **Forensic Explorer CLI** may choose to add the program into the **Windows Path Environment Variable**. Once added the Forensic Explorer CLI can be run from any folder in the CLI Lint prompt without the need for typing the full installation path.

To add Forensic Image to the Windows Path Environment Variable:

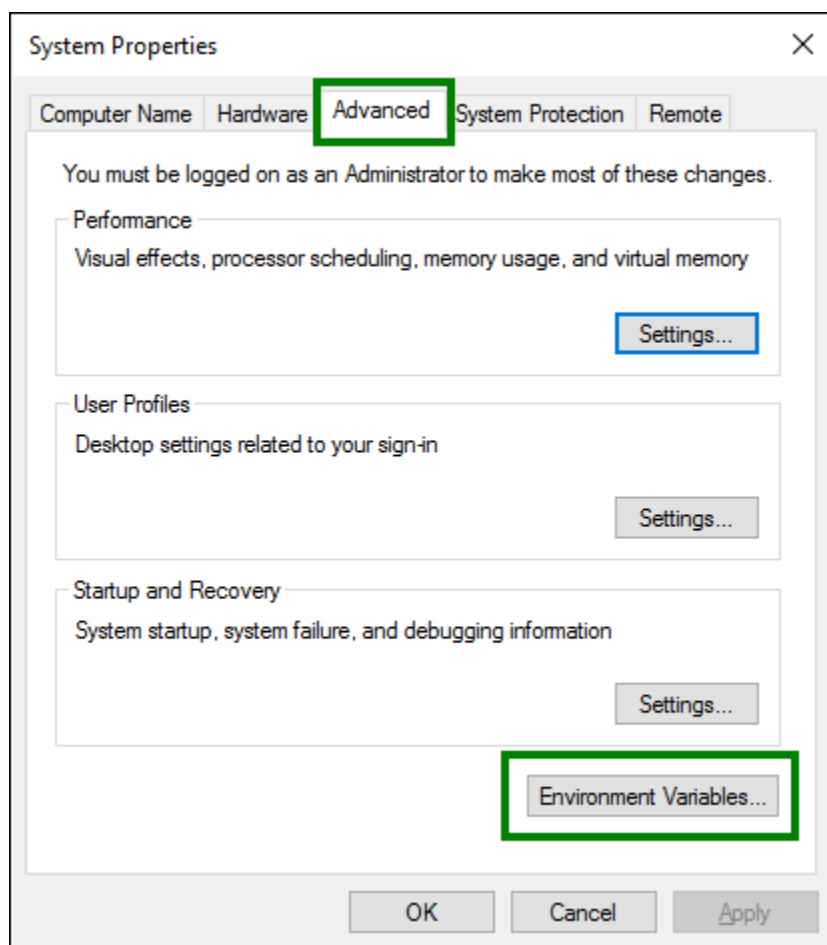
1. Open the **System Properties** window by:
 - a. Typing: **sysdm.cpl**; or
 - b. Open the Control Panel, search for **system** and select the **Edit the system environment variables** option shown in Figure 8 below:

Figure 8: Windows 10 Control Panel



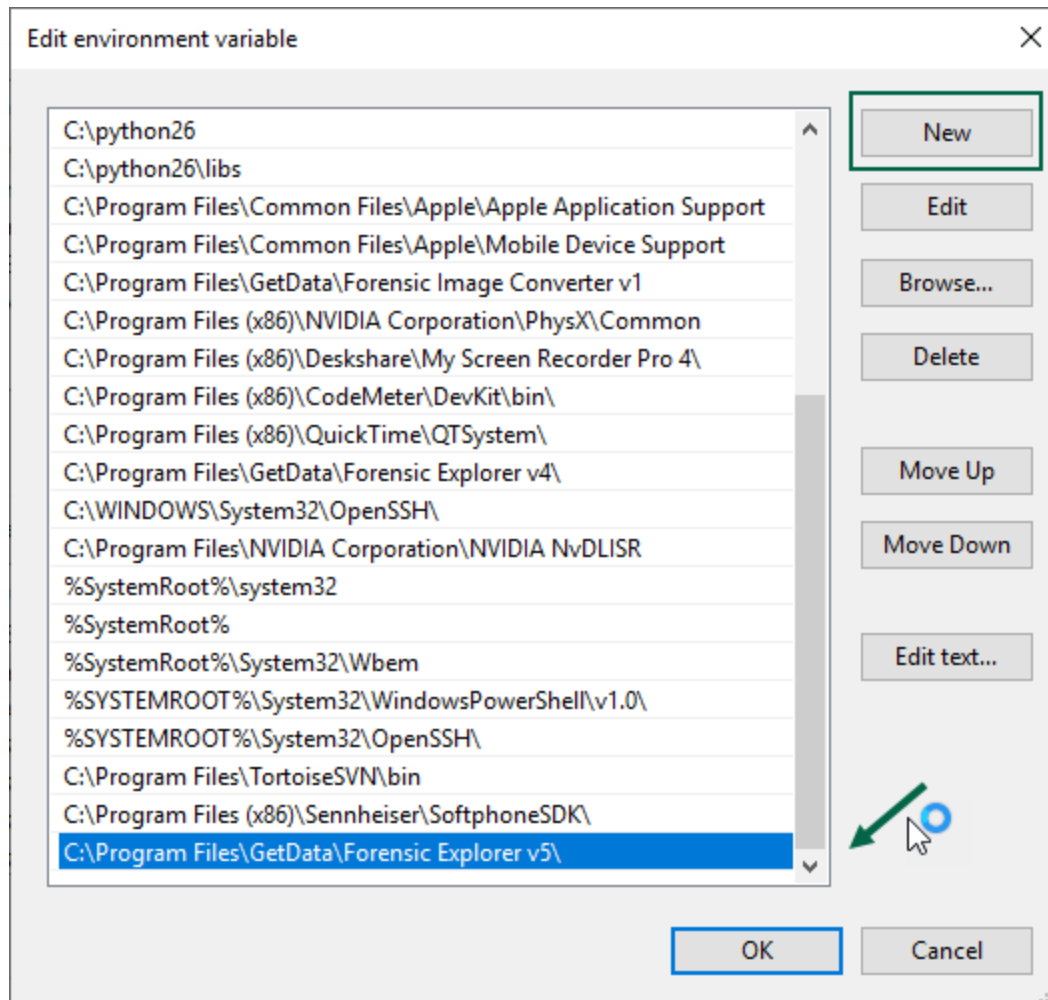
In the System Properties window select the **Advanced** tab then the **Environment Variables** button, as shown in Figure 9 below:

Figure 9: System Properties (Windows 10 shown)



In the **Environment Variables** window, in the **System Variables** box, select **Path**, then press the **Edit** button. In the **Edit environment, variable** window, click the **New** button and add the Forensic Explorer CLI path: **C:\Program Files\GetData\Forensic Explorer v5** as shown in Figure 10 below:

Figure 10: Adding the Forensic Image Converter path to the Environment variables.



Once the variable has been added, close any existing command windows. Open a new DOS window to a folder other than in the installation folder and type the command line: **FEX_CommandLine.exe /?**. The CLI help screen shown in **Error! Reference source not found. Error! Reference source not found.** in will display.

Chapter 5 – CLI Execution

In This Chapter

CHAPTER 5 – CLI EXECUTION

- 5.1 CLI Execution.....29
 - 5.1.1 FEX_CommandLine.exe Help29
 - 5.1.2 Required CLI switches29
 - 5.1.3 Optional CLI Switches.....30
 - 5.1.4 CLI execution30

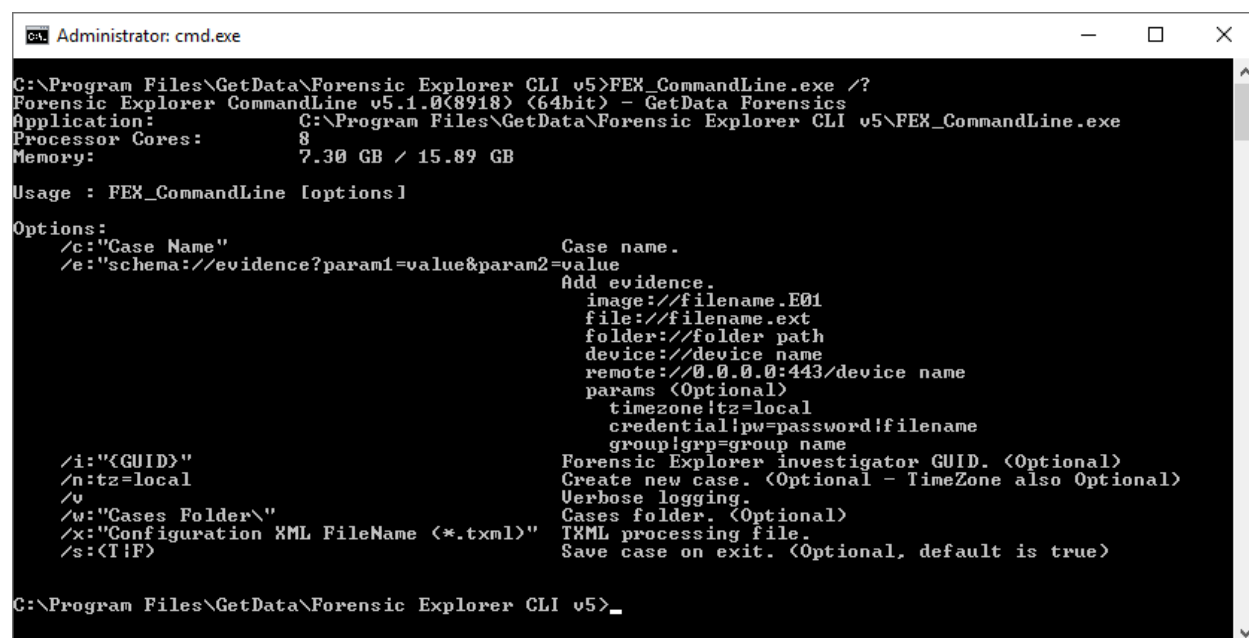
5.1 CLI EXECUTION

In this manual the **Windows CLI shell** is shown with Forensic Explorer CLI version added to the **Windows Path Variable** to enable execution of the commands from any folder. To add to the Windows Path Variable, follow the instructions in 4.2.3 above.

5.1.1 FEX_COMMANDLINE.EXE HELP

To access the help menu type: **FEX_CommandLine.exe /?**

Figure 11: Help



```
Administrator: cmd.exe
C:\Program Files\GetData\Forensic Explorer CLI v5>FEX_CommandLine.exe /?
Forensic Explorer CommandLine v5.1.0(8918) (64bit) - GetData Forensics
Application:      C:\Program Files\GetData\Forensic Explorer CLI v5\FEX_CommandLine.exe
Processor Cores:  8
Memory:           7.30 GB / 15.89 GB

Usage : FEX_CommandLine [options]

Options:
/c:"Case Name"          Case name.
/e:"schema://evidence?param1=value&param2=value"  Add evidence.
                                                                image://filename.E01
                                                                file://filename.ext
                                                                folder://folder path
                                                                device://device name
                                                                remote://0.0.0.0:443/device name
                                                                params <Optional>
                                                                timezone!tz=local
                                                                credential!pw=password!filename
                                                                group!grp=group name
/i:"{GUID}"            Forensic Explorer investigator GUID. <Optional>
/n:tz=local            Create new case. <Optional - TimeZone also Optional>
/v                    Verbose logging.
/w:"Cases Folder\"    Cases folder. <Optional>
/x:"Configuration XML File Name (*.xml)"  TXML processing file.
/s:<T|F>              Save case on exit. <Optional, default is true>

C:\Program Files\GetData\Forensic Explorer CLI v5>
```

5.1.2 REQUIRED CLI SWITCHES

When processing a case from the FEX CLI, there are required CLI switches, namely:

- **Case folder:** This is the name of the Forensic Explorer case. A folder will be created in the Cases path, for example:

```
/c:"My_CLI_Case_1"
```

- **Evidence:** Specifies the evidence added to the case, for example:

```
image://filename.E01
file://filename.ext
folder://folder path
device://device name
remote://0.0.0.0:443/device name
```

- **TXML Processing file:** The .TXML file is used to specify the processing options (e.g., verification, signature analysis, file carving, file export, etc., see Chapter 7), for example:

```
/x:"Processing_File.TXML"
```

5.1.3 OPTIONAL CLI SWITCHES

Optional Forensic Explorer CLI switches include:

- **Working path:** Specifies the path to the case, for example:

`/w:"C:\GetData\fex_cli\cases"`

If the working path is not specified, the path will default to the Forensic Explorer GUI case path (if present).

- **Create new case:** Specifies if a new case is to be created. If the `/n` is not present the CLI will **open** the existing case, as specified by the working path and case name.
- **Investigator GUID:** The investigator GUID is used to assign a specific investigator to the CLI processing, for example:

`/i:{Investigator GUID}`

If the `/I` switch is not used, or the specified investigator GUID is not found in the existing investigator database (...\\Documents\\Forensic Explorer CLI v5\\DataBases\\LocalInvestigator.xml) then the CLI process is given the default investigator GUID {D7DEB64C-45C5-49FA-8802-A719CA134A7B}. The default investigator GUID will appear as **Investigator (CLI)** in the Forensic Explorer GUI.

To locate an investigator GUID:

1. Run the Forensic Explorer GUI;
2. Click the **New** button and create a new case;
3. Click the Forensic Explorer orange button and select Investigators from the drop-down menu.

5.1.4 CLI EXECUTION

A typical execution of **FEX_CommandLine.exe** from the Forensic Explorer CLI installation folder may therefore look like this:

Figure 12: CLI Statement

```
C:\fex_cli\bin64\FEX_CommandLine /w:"C:\GetData\fex_cli\cases" /n /c:"My_CLI_Case_1"  
/x:"C:\GetData\fex_cli\txml\examples\read_file_system.xml" /e:"image://D:\Forensic_Image_1.E01?grp="
```

In this execution shown in Figure 12 above the following takes place:

- A new case (`/n`) is created in the working folder (`/w`) `C:\GetData\fex_cli\cases` with case name: `My_CLI_Case_1`;
- The forensic image file: `D:\Forensic_Image_1.E01` is added to the case;
- The image is processed using the .TXML file: `C:\GetData\fex_cli\txml\examples\read_file_system.xml`.

The Forensic Explorer CLI output window for this execution is shown in Figure 13 below:

Figure 13: Output from the execution of the CLI

```

C:\Users\Owner\Desktop>"C:\GetData\fex_cli\bin64\FEX_CommandLine.exe" /w:"C:\GetData\fex_cli\cases" /n
/c:"My_CLI_Case_1" /x:"C:\GetData\fex_cli\txml\examples\read_file_system.xml"
/e:"image://D:\Forensic_Image_1.E01?grp="

Forensic Explorer CommandLine 4.4.6.7744 - GetData Forensics
Application: C:\GetData\fex_cli\bin64\FEX_CommandLine.exe
Registered To: GetData Forensics
Maintenance Valid To: 01 June, 2019

Investigator: Investigator (CLI)
Investigator GUID: {D7DEB64C-45C5-49FA-8802-A719CA134A7B}
Working Directory: C:\GetData\fex_cli\cases\
Creating New Case: My_CLI_Case_1
Process XML: C:\GetData\fex_cli\txml\examples\read_file_system.xml

-----+-----+-----+-----+-----+
Task          |Description          |%  |Time  |State
-----+-----+-----+-----+-----+
Search for Known ISO Tracks  Devices 1, ISO/DVD tracks 0      100 00:00:00 Complete
Search for Known MBRs       Devices 2, MBRs 1, Partitions 3  100 00:00:00 Complete
Search for FileSystems      Files and folders 1006          100 00:00:00 Complete
Saving Case: My_CLI_Case_1  Saved case "My_CLI_Case_1"      100 00:00:00 Complete
-----+-----+-----+-----+-----+
Total Time: 00h:00m:02s

```


Chapter 6 – CLI Tools

In This Chapter

CHAPTER 6 – CLI TOOLS

- 6.1 CLI Tools.....35
 - 6.1.1 Batch Files35
 - 6.1.2 Python.....35

6.1 CLI TOOLS

The Forensic Explorer CLI can be launched by any tool capable of issuing CLI statements (e.g., batch files, Python, HTML5 etc.). Use of such tools can increase:

- Readability: It is easier to identify individual CLI arguments;
- Repeatability: Tasks are saved and re-executed as needed;
- Flexibility: Files can be easily modified and saved to perform variations of the task.
- Expandability: In an enterprise environment, custom tools can be used to maximize processing power and performance.

6.1.1 BATCH FILES

A batch file is a simple method of saving and executing a CLI statement.

To create a batch (.bat) file:

1. Right click on the desktop and select New > Text Document.
2. Rename the newly created text document to: **My_FEX_CLI_Batch_File_1.bat**
3. Copy and paste the statement in **Figure 12** above into the file.
4. Save the file.

Open **My_FEX_CLI_Batch_File_1.bat** to launch the command.

Examples of more complex batch files are provided at **Appendix 1 – Sample Batch Files**.

6.1.2 PYTHON

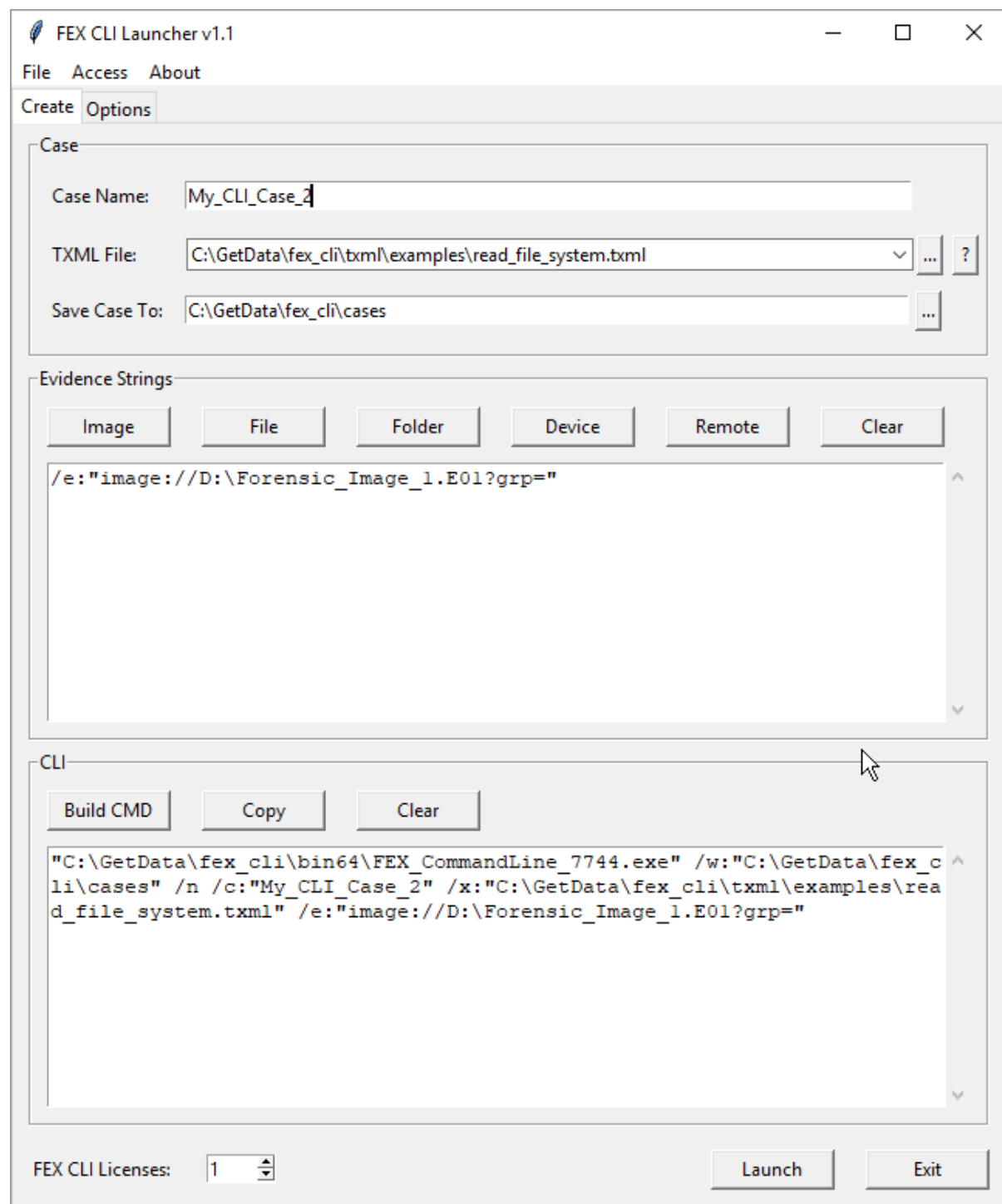
Provided in the root of the FEX CLI portable installation folder are two Python programs provided as sample front ends to execute CLI commands:

- **fex_cli_launcher.exe:** Launches individual instances of the FEX CLI (shown in Figure 14 above).
- **fex_cli_tree_processor.exe:** Used to process multiple folders containing forensic image files as separate cases.

Source code for these programs is provided in the **cli_launcher_python_source** folder of the FEX CLI portable version.

The **fex_cli_launcher.exe** is used to build and execute CLI commands, as shown in Figure 14 below:

Figure 14: fex_cli_launcher.exe



Chapter 7 – TXML Processing File

In This Chapter

CHAPTER 7 – TXML PROCESSING FILE	
7.1	TXML Processing File39
7.2	XML Terminology39
7.3	Forensic Explorer TXML Elements39
7.3.1	XML Declaration39
7.3.2	TXML Version39
7.3.3	Command Task (TCommandTask, File Carve shown).....40
7.4	Natural Order of Command Tasks40
7.5	TCommandTask_Parallel41

7.1 TXML PROCESSING FILE

TXML files in this folder can be examined with XML reader software such as Notepad ++ (<https://notepad-plus-plus.org/download/>). At the most basic level the principal .TXML Command Tasks are:

- Read evidence;
- Locate and determine the file system; and
- Populate the File System module.

7.2 XML TERMINOLOGY

XML tag A tag is used to **mark** the start or **end** of an element. A tag can be in the format:

```
<task><-- Stand-alone task -->/>
```

Or;

```
<task>
  <-- Use this format when there are sub tasks -->
</task>
```

XML Element An element is considered to include the start and end tags, and everything in between.

7.3 FORENSIC EXPLORER TXML ELEMENTS

The following elements are found in a Forensic Explorer TXML file:

7.3.1 XML DECLARATION

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

Description: The XML declaration typically appears as the first line in an XML document. The XML declaration is not required, however, if used it must be the first line in the document and no other content or white space can precede it. [https://msdn.microsoft.com/en-us/library/ms256048\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms256048(v=vs.110).aspx)

Use: The XML Declaration is required. It is the first line of the TXML.

7.3.2 TXML VERSION

Example: `<TXML version="2">`
`<-- All other elements fall within -->`
`</TXML>`

Description: The TXML version element is the Forensic Explorer version number.

Use: The TXML version element is **required**. This element will start at the second line of the TXML and end in the last line of the TXML and encase all other elements.

7.3.3 COMMAND TASK (TCOMMANDTASK, FILE CARVE SHOWN)

A list of available Forensic Explorer command tasks is available at Appendix 2 – TXML Command Tasks.

CommandTasks are built from **standard values** and **custom values** specific to that task. The command task **TCommandTask_SearchforLostFiles** is provided as an example below:

Table 1: CommandTask SearchforLostFiles (File Carve)

```
<task classname="TcommandTask_SearchforLostFiles"
  caption="My File Carve"
  description="JPG and Zip"
  enabled="false"
  searchmode="1"
  freespaceonly="true"
  byteoffset="0">
  <drivers>
    <driver classname="TJPGDriver"/>
    <driver classname="TZIPDriver"/>
  </drivers>
</task>
```

STANDARD VALUES

caption:	The task caption.
description:	A command task specifies the task to be run by Forensic Explorer. The element is encased in the <task> tag.
priority:	Processing priority: 0 = Minimum, 1 = Normal, 2 = High, 3 = Maximum
logging:	Log file: 0 = None, 1 = Normal, 2 = Verbose
enabled:	Whether the task will be executed (in the GUI this is represented by the checkbox).

CUSTOM VALUES – FILE CARVE

searchmode:	0 = Cluster, 1 = Sector (default), 2 = Byte (Warning: Will slow processing)
Freespaceonly:	True/False (carve freespace only)
Byteoffset:	Byte Offset to begin carving. Default as 0
driver:	Individual file types for which to carve.

7.4 NATURAL ORDER OF COMMAND TASKS

There is a natural order for command tasks and consideration should be given to the order in which CLI tasks are executed. For example:

Verify Image Hash:	Can be performed prior to the File System command tasks because it relies only on accessing the device not the File System.
--------------------	---

Expand Compound Files: A Signature Analysis should be run prior to Expand Compound Files to identify any compound files with the wrong extension. After the expansion it is also prudent to run a second Signature Analysis to determine the file types in the expanded compound files.

7.5 TCOMMANDTASK_PARALLEL

The purpose of the TCommandTask_Parallel is to ensure that dependent processes are executed most efficiently by running simultaneously in their own thread (parallel processing). The TCommandTask_Parallel tags surround the dependent tasks:

Table 2: TCommandTask_Parallel

```
<task classname="TCommandTask_Parallel"
  caption="Process in Parallel"
  description=""
  enabled="true">
  <task><!-- Hash files --></task>
  <task><!-- Cache thumbnails --></task>
  <task><!-- Index --></task>
</task>
```


APPENDIX 1 – SAMPLE BATCH FILES

The following sample Forensic Explorer CLI batch files are formatted as follows:

- **ECHO**: prints the line in the console window.
- **rem**: is a comment and is ignored by the batch file.
- **SET**: defines a variable.
- **%XXXXX%**: Substitutes the defined variable.
- The example batch files process evidence using the **ReadFS.xml** which reads the File System with no additional processing (see Chapter 7 for more detail).

7.5.1 FEX_CLI_1.BAT (ADD SPECIFIC EVIDENCE TO A CASE)

The batch file in Figure 15: FEX_CLI_1.bat below:

1. Creates a **new case** (/n) called **My_CLI_Case_1**;
2. Two forensic image files are added as **variables**:
 - I. **D:\Forensic_Image_1.E01** and
 - II. **D:\Forensic_Image_2.E01**;
 and **concatenated** into an **EVIDENCE_STR** for use in the command line argument.
3. Validation of **variables** (i.e., paths) is conducted prior to the command line execution.
4. The case is processed using **ReadFS.xml**.

Figure 15: FEX_CLI_1.bat

```
@ECHO OFF

rem =====
rem Set Evidence Variables
rem =====
SET ADD_EVIDENCE_IMAGE_1=D:\Forensic_Image_1.E01
SET ADD_EVIDENCE_IMAGE_2=D:\Forensic_Image_2.E01
rem SET ADD_EVIDENCE_IMAGE_3=D:\Forensic_Image_3.E01
rem SET ADD_EVIDENCE_FOLDER_1=D:\afolder\
rem SET ADD_EVIDENCE_FILE_1=D:\afile.jpg

rem =====
rem Validate Evidence Variables
rem =====
IF DEFINED ADD_EVIDENCE_IMAGE_1 IF NOT EXIST "%ADD_EVIDENCE_IMAGE_1%" ECHO Not
Found: Image File:      "%ADD_EVIDENCE_IMAGE_1%"  && SET ERROR=TRUE
IF DEFINED ADD_EVIDENCE_IMAGE_2 IF NOT EXIST "%ADD_EVIDENCE_IMAGE_2%" ECHO Not
Found: Image File:      "%ADD_EVIDENCE_IMAGE_2%"  && SET ERROR=TRUE
IF DEFINED ADD_EVIDENCE_IMAGE_3 IF NOT EXIST "%ADD_EVIDENCE_IMAGE_3%" ECHO Not
Found: Image File:      "%ADD_EVIDENCE_IMAGE_3%"  && SET ERROR=TRUE
```

```

IF DEFINED ADD_EVIDENCE_FOLDER_1 IF NOT EXIST "%ADD_EVIDENCE_FOLDER_1%" ECHO Not
Found: Evidence Folder: "%ADD_EVIDENCE_FOLDER_1%" && SET ERROR=TRUE
IF DEFINED ADD_EVIDENCE_FILE_1 IF NOT EXIST "%ADD_EVIDENCE_FILE_1%" ECHO Not
Found: Evidence File: "%ADD_EVIDENCE_FILE_1%" && SET ERROR=TRUE
if "%ERROR%" == "TRUE" ECHO The batch will terminate. && PAUSE && EXIT /B

rem =====
rem Set Investigator GUID: (default CLI_Investigator GUID used)
rem =====
SET INVESTIGATORID={D7DEB64C-45C5-49FA-8802-A719CA134A7B}

rem =====
rem Set Working Variables
rem =====
SET FEXCLIEXE=C:\Program Files\GetData\Forensic Explorer CLI
v5\FEX_CommandLine.exe
SET CASESPATH=%USERPROFILE%\Documents\Forensic Explorer CLI v5\Cases\
SET CASENAME=My CLI_Case_1
SET TXMLFOLDER=%USERPROFILE%\documents\Forensic Explorer CLI v5\Startup\
SET TXMLFILE=ReadFS.xml

rem =====
rem Validate Working Variables
rem =====
IF NOT EXIST "%FEXCLIEXE%" ECHO Not Found: FEX_CommandLine.exe: "%FEXCLIEXE%" &&
SET ERROR=TRUE
IF NOT EXIST "%CASESPATH%" ECHO Not Found: FEX Cases Folder: "%CASESPATH%" && SET
ERROR=TRUE
IF NOT DEFINED CASENAME ECHO Not Defined: Case Name && SET ERROR=TRUE
IF NOT EXIST "%TXMLFOLDER%%TXMLFILE%" ECHO Not Found: FEX TXML File:
"%TXMLFOLDER%%TXMLFILE%" && SET ERROR=TRUE
if "%ERROR%" == "TRUE" ECHO The batch will terminate. && PAUSE && EXIT /B

rem =====
rem Build the Add Evidence String
rem =====
SET EVIDENCE_STR=
IF DEFINED ADD_EVIDENCE_IMAGE_1 SET EVIDENCE_STR=%EVIDENCE_STR%
/e:"%ADD_EVIDENCE_IMAGE_1%"
IF DEFINED ADD_EVIDENCE_IMAGE_2 SET EVIDENCE_STR=%EVIDENCE_STR%
/e:"%ADD_EVIDENCE_IMAGE_2%"
IF DEFINED ADD_EVIDENCE_IMAGE_3 SET EVIDENCE_STR=%EVIDENCE_STR%
/e:"%ADD_EVIDENCE_IMAGE_3%"
IF DEFINED ADD_EVIDENCE_FOLDER_1 SET EVIDENCE_STR=%EVIDENCE_STR%
/d:"%ADD_EVIDENCE_FOLDER_1%"
IF DEFINED ADD_EVIDENCE_FILE_1 SET EVIDENCE_STR=%EVIDENCE_STR%
/f:"%ADD_EVIDENCE_FILE_1%"
rem Trim leading spaces ----
for /f "tokens=* delims=" %s in ("%EVIDENCE_STR%") do set EVIDENCE_STR=%s

:about
ECHO This FEX batch file will:
ECHO[
ECHO 1. Create new case: %CASENAME%
ECHO 2. Add evidence: %EVIDENCE_STR%
ECHO 3. Process using: %TXMLFILE%
ECHO[

:choice
set /P Q=Are you sure you want to continue[Y/N]?
if /I "%Q%" == "Y" goto :continue
if /I "%Q%" == "N" goto :exit

```

```
goto :choice

:continue
ECHO[
ECHO =====
ECHO FEX Command Executed at: %date% %time%
ECHO =====
"%FEXCLIEXE%" /i:%INVESTIGATORID% /w:"%CASESPATH%" /c:"%CASENAME%" /n
%EVIDENCE_STR% /x:"%TXMLFOLDER%%TXMLFILE%"
ECHO FEX Command Finished at: %date% %time%
ECHO[

PAUSE
EXIT

:exit
ECHO Batch canceled by user.
PAUSE
EXIT
```

7.5.2 FEX_CLI_2.BAT (ADD EVIDENCE IN A FOLDER/SUB-FOLDER TO A CASE)

The batch file in Figure 16 below:

1. An **EVIDENCEPATH** variable specifies the location of evidence files;
2. Validation of the **variables** is conducted prior to the command line execution.
3. A test is conducted to determine if a case called **My_CLI_Case_1** already exists. If true, a loop will create a unique case folder, e.g. **My_CLI_Case_2**.
4. A **new case** is created (/n) with the unique case name **My_CLI_Case_[x]**;
5. All .AD1, E01, Ex01 and L01 files in the EVIDENCEPATH (sub-folders can be included by setting the **INCLUDE_SUBFOLDERS** to **TRUE**) are **concatenated** into an **EVIDENCE_STR** for use in the command line argument.
6. The case is processed using **ReadFS.xml**.

Figure 16: FEX_CLI_2.bat

```
@ECHO OFF

rem =====
rem Set Evidence Folder/Sub-Folders
rem =====
SET EVIDENCEPATH=E:\GetData-4GSD-Mixed-Formats
SET INCLUDE_SUBFOLDERS=FALSE
SET SHOW_ADDED_FILES=TRUE

rem =====
rem Set Investigator GUID: (default CLI_Investigator GUID used)
rem =====
SET INVESTIGATORID={D7DEB64C-45C5-49FA-8802-A719CA134A7B}

rem =====
rem Set Working Variables
rem =====
SET FEXCLIEXE=C:\Program Files\GetData Forensic Explorer CLI
v5\FEX_CommandLine.exe
SET CASESPATH=%USERPROFILE%\Documents\GetData Forensic Explorer CLI v5\Cases\
SET TXMLFOLDER=%USERPROFILE%\documents\GetData Forensic Explorer CLI v5\Startup\
SET TXMLFILE=ReadFS.xml

rem =====
rem Validate Working Variables
rem =====
IF NOT EXIST "%FEXCLIEXE%" ECHO Not Found: FEX_CommandLine.exe: "%FEXCLIEXE%" &&
SET ERROR=TRUE
IF NOT EXIST "%CASESPATH%" ECHO Not Found: FEX Cases Folder: "%CASESPATH%" && SET
ERROR=TRUE
IF NOT EXIST "%TXMLFOLDER%%TXMLFILE%" ECHO Not Found: FEX TXML File:
"%TXMLFOLDER%%TXMLFILE%" && SET ERROR=TRUE
IF NOT EXIST "%EVIDENCEPATH%" ECHO Not Found: FEX Evidence Path: "%EVIDENCEPATH%"
&& SET ERROR=TRUE
if "%ERROR%" == "TRUE" ECHO The batch will terminate. && PAUSE && EXIT /B

rem =====
rem Create a Unique Case Name
```

```

rem =====
SET /A "COUNTER=1"
:while
SET CASENAME=My_CLI_Case_%COUNTER%
IF EXIST "%CASESPATH%%CASENAME%" (
    SET /A "COUNTER=COUNTER + 1"
    goto :while
)

:about
ECHO This FEX batch file will:
ECHO[
ECHO 1. Create new case:          %CASENAME%
ECHO 2. Add evidence from:       %EVIDENCEPATH%
ECHO 3. Include sub-folders:     %INCLUDE_SUBFOLDERS%
ECHO 3. Process using:          %TXMLFILE%
ECHO[

rem =====
rem Collect file names in folder and build EVIDENCE_STR
rem =====
IF "%SHOW_ADDED_FILES%" == "TRUE" (
    ECHO The following evidence will be added to the case:
    ECHO[
)
SET DIR=%EVIDENCEPATH%
setlocal ENABLEDELAYEDEXPANSION
SET EVIDENCE_STR=

rem Include Subfolders (see https://ss64.org/viewtopic.php?id=910)
IF "%INCLUDE_SUBFOLDERS%" == "TRUE" (
    for /f "delims=" %%A in ('dir /b /s "%EVIDENCEPATH%\*.AD1" "%EVIDENCEPATH%\*.E01" "%EVIDENCEPATH%\*.Ex01" "%EVIDENCEPATH%\*.L01" 2^>nul') do (
        SET EVIDENCE_STR=!EVIDENCE_STR! /e:"%%A"
        IF "%SHOW_ADDED_FILES%" == "TRUE" ECHO %%A
    ) ELSE (
        for /f "delims=" %%A in ('dir /b "%EVIDENCEPATH%\*.AD1" "%EVIDENCEPATH%\*.E01" "%EVIDENCEPATH%\*.Ex01" "%EVIDENCEPATH%\*.L01" 2^>nul') do (
            SET EVIDENCE_STR=!EVIDENCE_STR! /e:"%EVIDENCEPATH%\%%A"
            IF "%SHOW_ADDED_FILES%" == "TRUE" ECHO %%A
        )
    )

:choice
ECHO[
set /P Q=Are you sure you want to continue[Y/N]?
if /I "%Q%" == "Y" goto :continue
if /I "%Q%" == "N" goto :exit
goto :choice

:continue
ECHO[
ECHO =====
ECHO FEX Command Executed at: %date% %time%
ECHO =====
"%FEXCLIEXE%" /i:%INVESTIGATORID% /w:"%CASESPATH%" /c:"%CASENAME%" /n
%EVIDENCE_STR% /x:"%TXMLFOLDER%%TXMLFILE%"
ECHO FEX Command Finished at: %date% %time%
ECHO[

PAUSE
EXIT

:exit

```

```

ECHO Batch canceled by user.
PAUSE
EXIT

rem Dir Command: https://ss64.com/nt/dir.html
rem /b = Bare format (no heading, file sizes or summary).
rem /a:d-h-s = Folder, not hidden, not system.
rem /s = include all sub-folders.

```

7.5.3 FEX_CLI_3.BAT (CREATE SEPARATE CASES FROM FOLDERS)

In the batch file in Figure 17 below:

1. A **separate case** is created in the name of each folder found in the **EVIDENCEPATH**.
2. Evidence within each folder is added to the case (sub-folders can be included by setting **INCLUDE_SUBFOLDERS** to **TRUE**).
3. The case is processed using **ReadFS.xml**.

Figure 17: FEX_CLI_3.bat

```

@ECHO OFF

rem =====
rem Set Evidence Folder/Sub-Folders
rem =====
SET EVIDENCEPATH=E:
SET INCLUDE_SUBFOLDERS=FALSE

rem =====
rem Set Investigator GUID: (default CLI_Investigator GUID used)
rem =====
SET INVESTIGATORID={D7DEB64C-45C5-49FA-8802-A719CA134A7B}

rem =====
rem Set Working Variables
rem =====
SET FEXCLIEXE=C:\Program Files\GetData\GetData Forensic Explorer CLI
v5\FEX_CommandLine.exe
SET CASESPATH=%USERPROFILE%\Documents\GetData Forensic Explorer CLI v5\Cases\
SET TXMLFOLDER=%USERPROFILE%\documents\GetData Forensic Explorer CLI v5\Startup\
SET TXMLFILE=ReadFS.xml

rem =====
rem Validate Working Variables
rem =====
IF NOT EXIST "%FEXCLIEXE%" ECHO Not Found: FEX_CommandLine.exe: "%FEXCLIEXE%" &&
SET ERROR=TRUE
IF NOT EXIST "%CASESPATH%" ECHO Not Found: FEX Cases Folder: "%CASESPATH%" && SET
ERROR=TRUE
IF NOT EXIST "%TXMLFOLDER%%TXMLFILE%" ECHO Not Found: FEX TXML File:
"%TXMLFOLDER%%TXMLFILE%" && SET ERROR=TRUE
IF NOT EXIST "%EVIDENCEPATH%" ECHO Not Found: FEX Evidence Path: "%EVIDENCEPATH%"
&& SET ERROR=TRUE
if "%ERROR%" == "TRUE" ECHO The batch will terminate. && PAUSE && EXIT /B

:about
ECHO This FEX batch file will:

```



```

ECHO[
ECHO 1. Create cases from folders in:      %EVIDENCEPATH%
ECHO 2. Include evidence in sub-folders:  %INCLUDE_SUBFOLDERS%
ECHO 3. Process using:                    %TXMLFILE%
ECHO[

:choice
set /P Q=Are you sure you want to continue[Y/N]?
if /I "%Q%" == "Y" goto :continue
if /I "%Q%" == "N" goto :exit
goto :choice

:continue
for /f "usebackq tokens=*" %%A in (`dir %EVIDENCEPATH% /b /a:d-h-s`) do (

    rem Change current folder, store previous folder for popd.
    pushd %%A
    ECHO %%A

    rem =====
    rem Collect file names in folder and build EVIDENCE_STR
    rem =====
    SET DIR=%EVIDENCEPATH%
    setlocal ENABLEDELAYEDEXPANSION
    SET EVIDENCE_STR=

    rem Include Subfolders (see https://ss64.org/viewtopic.php?id=910)
    IF "%INCLUDE_SUBFOLDERS%" == "TRUE" (
        for /f "delims=" %%B in ('dir /b /s "%EVIDENCEPATH%\%%A\*.AD1"
"%EVIDENCEPATH%\%%A\*.E01" "%EVIDENCEPATH%\%%A\*.Ex01" "%EVIDENCEPATH%\%%A\*.L01"
2^>nul') do (
            SET EVIDENCE_STR=!EVIDENCE_STR! /e:"%%B"
        ) ELSE (
            for /f "delims=" %%B in ('dir /b "%EVIDENCEPATH%\%%A\*.AD1"
"%EVIDENCEPATH%\%%A\*.E01" "%EVIDENCEPATH%\%%A\*.Ex01" "%EVIDENCEPATH%\%%A\*.L01"
2^>nul') do (
                SET EVIDENCE_STR=!EVIDENCE_STR! /e:"%EVIDENCEPATH%\%%B"
            )
        )

    ECHO[
    ECHO =====
    ECHO FEX Command Executed at: %date% %time%
    ECHO =====

    SET CASENAME=%%A
    ECHO Processing folder: !CASENAME!

    IF DEFINED EVIDENCE_STR (
        rem =====
        rem Construct CLI Using Variables
        rem =====
        "%FEXCLIEX%" /i:%INVESTIGATORID% /w:"%CASESPATH%" /c:"!CASENAME!" /n
!EVIDENCE_STR! /x:"%TXMLFOLDER%TXMLFILE%"
        ECHO FEX Command Finished at: %date% %time%

    ) ELSE (
        ECHO No evidence files located in: %%A
    )
    endlocal

    rem Reset the EVIDENCE_STR and CASENAME for the next loop
    SET EVIDENCE_STR=
    SET CASENAME=

```

```
rem - Change back to folder stored by the PUSHHD command.
popd
)

ECHO[
PAUSE
EXIT

:exit
ECHO Batch canceled by user.
PAUSE
EXIT

rem Dir Command: https://ss64.com/nt/dir.html
rem Dir output is evaluated once, so the list of directories is fixed.
rem /b = Bare format (no heading, file sizes or summary).
rem /a:d-h-s = Folder, not hidden, not system.
rem /s = include all sub-folders.
```

APPENDIX 2 – TXML COMMAND TASKS

All Tasks have the following properties:

Caption:	string, title of the task
Description:	string, task description
Priority :	integer, Low=0, Normal=1 (default), High=2, Critical=3
Logging :	integer, None=0, Normal=1, Verbose=2
Enabled:	boolean, True (default)

The following TXML Command Tasks are available in the Forensic Explorer CLI.

TcommandTask_AcquireMemory
TCommandTask_AppleBackup
TCommandTask_Bookmark
TCommandTask_CacheThumbnails
TCommandTask_CacheVideoThumbnails
TCommandTask_ClamAV
TCommandTask_CreateHash
TCommandTask_DataStore
TCommandTask_DtSearch
TcommandTask_Entropy
TCommandTask_ExpandCompoundFiles
TCommandTask_ExportEmailToPST
TCommandTask_ExportEntryList
TCommandTask_ExportFiles
TCommandTask_ExportFilesL01
TCommandTask_ExportVIC
TCommandTask_ExportZIP
TCommandTask_FileTypeAnalysis
TCommandTask_Filter
TCommandTask_FolderCarve
TCommandTask_ImageAnalyzer
TCommandTask_KeywordSearchExternal
TCommandTask_LiveBoot
TCommandTask_MatchHash
TCommandTask_Parallel
TCommandTask_ReportGenerator
TCommandTask_Root
TCommandTask_Script

TCommandTask_SearchforEmails
TCommandTask_SearchforKnownFS
TCommandTask_SearchforKnownISOTracks
TCommandTask_SearchforKnownMBR
TCommandTask_SearchforLostFiles
TCommandTask_SearchRegistryHive
TCommandTask_SendTo
TCommandTask_VerifyDevice
TcommandTask_Yara

Command Tasks are described in more detail below:

1.1 ACQUIRE MEMORY

```
<!-- Acquire Memory (RAM) -->
<task classname="TCommandTask_AcquireMemory"
outputfile="%CASE_EXPORTED%memory.raw"
memorymode="2"
caption="Acquire Memory"
enabled="true"/>
```

If output is not specified it will be: %CASE_EXPORTED%memory_yyyy_mmm_dd_hhnss.raw

0 = use MnMapIoSpace method.

1 = use \Device\PhysicalMemory method (default for 32bit OS)

2 = use PTE remapping (AMD64 only - default for 64bit OS) – the default if not specified

1.2 APPLEBACKUP – IDENTIFY APPLE BACKUPS

```
<!--Identify and bookmark Apple Backups -->
<task classname="TCommandTask_AppleBackup"
caption="Apple Backup"
enabled="true"/>
```

1.3 BOOKMARK – BLANK BOOKMARK FOLDER CREATE

Apply a filter before TcommandTask_Bookmark to work with filtered files.

```
<!--BOOKMARK: Create and/or populate Bookmark folders -->
<task classname="TCommandTask_Bookmark"
caption="Add Bookmarks"
enabled="true"

<!-- Create BM folders -->
<task classname="TCommandTask_Bookmark" caption="Add BM Folder"
enabled="True" fullpathname="My Bookmarks\Special 1"
foldercomment="Folder comment for Special #1" folderonly="true"/>

<task classname="TCommandTask_Bookmark" caption="Add BM Folder"
enabled="True" fullpathname="My Bookmarks\Special 2"
foldercomment="Folder comment for Special #2" folderonly="true"/>

<task classname="TCommandTask_Bookmark" caption="Add BM Folder"
enabled="True" fullpathname="My Bookmarks\Special 2\Sub-Folder"
foldercomment="Folder comment for sub-folder" folderonly="true"/>

<!-- Create BM folder and add files -->
<task classname="TCommandTask_Bookmark" caption="Add Bookmarks"
enabled="True" fullpathname="My Bookmarks\Pictures" entrycomment="Comment
for bookmarked file"/>

<task classname="TCommandTask_Bookmark" caption="Add Bookmarks"
enabled="True" fullpathname="My Bookmarks\Wombat" entrycomment="Comment
for bookmarked file"/>
```

1.4 CACHE THUMBNAILS

```
<!-- Cache pictures to thumbnails -->
<task classname="TCommandTask_CacheThumbNails"
caption="Cache Thumbnails"
description=""
enabled="true"
minfilesize="0"
maxfilesize="100"/>
```

1.5 CACHEVIDEOTHUMBNAILS

```
<!-- Cache videos in Video View -->
<task classname="TCommandTask_CacheVideoThumbNails"
caption="Cache Thumbnails"
description="Cache videos"
enabled="true"
minfilesize="0"
maxfilesize="100"/>
```

1.6 ENTROPY

```
<!-- Calculate Entropy -->
<task classname="TCommandTask_Entropy"
caption="Entropy"
description="Calculate Entropy"
enabled="true"
forcecalc="false"
minfilesize="0"
maxfilesize="1024"
newcolumnindex="3"/>
```

1.7 EXPAND COMPOUND FILES

```
<!-- Expand Compound Files
NOTE: This runs Expand Compound files from the TXML. Unless you filter above, it
will expand ALL compound files (e.g. Movies, DOCX etc.) -->
<task classname="TCommandTask_ExpandCompoundFiles" caption="Expand Compound Files"
enabled="true"/>
```

```
<!-- Extract X Video Keyframes -->
<task classname="TCommandTask_ExpandCompoundFiles" caption="Extract X
Video Keyframes" description="" enabled="true" video="1" VideoValue="3"
/>
```

```
<!-- Extract Time Sliced Video Keyframes -->
<task classname="TCommandTask_ExpandCompoundFiles" caption="Extract Time
Sliced Keyframes" description="" enabled="true" video="2" VideoValue="3"
/>
```

1.8 EXPORT EMAIL TO PST

```
<!-- Export Email To PST -->
<task classname="TCommandTask_ExportEmailToPST"
caption="Export Email To PST"
enabled="true"
filename="%CASE_PATH%Exported\Email_Messages.pst">
description=""
enabled="true"
examiner="Investigator"
caseno="CLI PST Export"
desc=" CLI PST Export"
evidno=""
notes="PST Created by Forensic Explorer CLI"/>
```

1.9 EXPORT FILE LIST (STANDARD COLUMN)

Only the first column is shown, "TColumnCHECK"

```
<!-- Export File List -->
<task classname="TCommandTask_ExportEntryList"
caption="Export File List - All Files"
enabled="true"
exportemptylist="true"
filename="%CASE_PATH%Exported\Export File List - All Files.csv">

  <colhandler name="ExportEntryList" ver="44">
    <col class="TColumnCHECK">
      <xmldata
        color="-16777201"
        color_enabled="false"
        color_functionname=""
        exportemptylist="false"
        font_enabled="false"
        font_functionname=""
        font_color="536870911"
        font_style="0"
        image_functionname=""
        image_enabled="false"
        formatstring=""
        issearchable="true"
        style="0"
        header=""
        header_alignment="0"
        alignment="1" sort="0"
        sortmode="0"
        width="74"
        minwidth="0"
        maxwidth="2048"
        visible="true"
        index="0"
        maxdate="1899-12-30T00:00:00Z"
        mindate="1899-12-30T00:00:00Z"/>
      </col>
    </colhandler>
  </task>
```


1.10 EXPORT FILE LIST (METADATA COLUMN)

```
<col class="TColumnDataStoreField"
meta_name="Hash (MD5)"
meta_alias="Exif 33432: Copyright"
field_type="2">

  <xmldata
    color="-16777201"
    color_enabled="false"
    color_functionname=""
    font_enabled="false"
    font_functionname=""
    font_color="536870911"
    font_style="0"
    image_functionname=""
    image_enabled="false"
    formatstring=""
    issearchable="true"
    style="0"
    header=""
    header_alignment="0"
    alignment="1" sort="0"
    sortmode="0"
    width="74"
    minwidth="0"
    maxwidth="2048"
    visible="true"
    index="0"
    maxdate="1899-12-30T00:00:00Z"
    mindate="1899-12-30T00:00:00Z"/>

</col>
```

File Type:

ftUnknown = 0

ftBoolean = 1

ftBytes = 2

ftByte = 3

ftVarBytes = 4

ftWord = 5

ftLongWord = 6

ftSmallint = 7

ftShortint = 8

ftInteger = 9

ftLargeInt = 10

1.11 EXPORT FILES TO FOLDER

```
<!-- Export files to folder --> (apply a Filter to export specific files)
<task classname="TCommandTask_ExportFiles"
caption="Export Files"
description=""
enabled="true"
destinationfolder="%CASE_PATH%Exported\"
folderstructure="true"
emptyfolders="true"
saveassinglefile="false"
savelogical="true"
keepdatetimes="true"
batesid="false"
batessuffix="false"
splitsizebytes="-1"/>
```

1.12 EXPORT L01

```
<!-- Export L01 --> (apply a Filter to export specific files)
<task classname="TCommandTask_ExportFilesL01"
caption="Export L01"
description=""
enabled="true"
filename="%CASE_PATH%Exported\CLI_Export.L01"
segmentsize="2000"
md5hash="true"
shalhash="false"
sha256hash="false"
compression="1"
dirdata="false"
examiner="Investigator"
caseno="CLI Export"
desc=" CLI Export"
evidno=""
notes="L01 Created by Forensic Explorer CLI"/>
```

1.13 EXPORT PROJECT VIC

```
<!-- Export to Project VIC. -->
<task classname="TCommandTask_ExportVIC"
caption="Export VIC"
enabled="true"
version="2"
filename="%CASE_PATH%Exported\Project_VIC_Export.json"
exportfilelink="true"/>
```

Version: Is not required. When not present the latest supported version of Project Vic is used.

1.14 EXPORT ZIP

```
<!-- Export to Project ZIP. -->
<task classname="TCommandTask_ExportZIP"
caption="Export ZIP"
compression="1"
dirdata="false"
filename="%CASE_PATH%Exported\New_File.zip"
savefilesack="false"/>
```

compression: (0=clNone, 1=clFastest (default), 2=clNormal, 3=clMax)
dirdata: True or False (default)
savefilesack: True or False (default)

1.15 FILE CARVE

```
<!-- File Carve -->
<task classname="TCommandTask_SearchforLostFiles"
caption="My File Carve"
description="JPG and Zip"
enabled="true"
searchmode="1"
freespaceonly="true"
byteoffset="0">
<drivers>
<driver classname="TJPGDriver"/>
<driver classname="TZIPDriver"/>
</drivers>
</task>
```

SearchMode:
0 = cluster/block
1 = sector
2 = byte

1.16 FILTER SCRIPT

```
<!-- Filter (JPEG.pas filter shown) -->
<task classname="TCommandTask_Filter"
caption="Filter"
description=""
enabled="true"
script="JPEG.pas">
<!-- To work with filtered content (e.g., export) embed further tasks here -->
</task>
```

1.17 HASH FILES

```
<!-- Hash files -->
<task classname="TCommandTask_CreateHash"
caption="Hash Files (MD5)"
description=""
enabled="true"
md5="true"
sha1="false"
sha256="false"
crc32="false"
fuzzy="false"
differential="false"
photodna="false"
forcecalc="false"
findduplicates="false"
minfilesize="0"
maxfilesize="1024"
newcolumnindex="3"/>
```

1.18 HASH MATCH

```
<!-- Hash Match the filtered files using hash sets-->
<task classname="TCommandTask_MatchHash"
caption="Hash Match"
description=""
enabled="true"
hashmethod="0"
clearpreviousmatches="true"
usemultiplesets="true"
newcolumnindex="3">
  <hashsets>
    <hashset filename="C:\HashSets\Cat5.hash" enabled="true"/>
    <hashset filename="C:\HashSets\Fish.db3" enabled="true"/>
    <hashset filename="C:\HashSets\Fish2.db3" enabled="true"/>
  </hashsets>
</task>
```

1.19 INDEX FILES

```
<!-- Index files -->
<task classname="TCommandTask_DTSearch"
caption="Index Name"
description="Index Description"
enabled="true"
fileslack="true"
unallocatedspace="False"/>
```

1.20 IMAGE ANALYZER (AI GRAPHICS)

```
<!-- Image Analyzer -->
<task classname="TCommandTask_ImageAnalyzer"
caption="AI Graphics"
description="Categorize graphics."
CatIDs="2754,7175,5250,4123"
enabled="true"
minfilesize="20"
maxfilesize="512000"
cutoff="60"/>
```

1.21 KEYWORD SEARCH

```
<!-- Keyword Search -->
<task classname="TCommandTask_KeywordSearchExternal"
caption="Keyword Search"
description="Keyword Search"
enabled="true"
filelimit="1000000"
searchlimit="0"
searchfileslack="True">
  <keywordfiles>
    <keywords filename="%APP_PATH%..\words.txt" enabled="true" />
  </keywordfiles>
</task>
```

IMPORTANT: It is necessary to run the Keyword Search inside a filter so that unwanted files are not included:

```
<!-- FILTER -->
<task classname="TCommandTask_Filter" caption="Filter Keyword Search"
enabled="true"
filename="%APP_PATH%..\filters\for_keyword_search\cli_filter_keyword_seac
h_no_freespace.pas" >
  <!-- Then Keyword Search -->
</task>
```

1.22 RECOVER FOLDERS

```
<!-- Recover Folders -->
<task classname="TCommandTask_FolderCarve"
caption="Recovered Folders"
description=""
enabled="true"
FindFAT="true"
FindMFT="true"
FindEXFAT="true"
FindHFS="true"
searchmode="1"
freespaceonly="true"
byteoffset="0"/>
```

1.23 SENDTO - MODULE

```
<!-- SendToModule -->
<task classname="TCommandTask_SendTo"
caption="Send To Module"
enabled="true"
module="Registry"/>
```

First filter the required file types.

Works for modules:

- Registry
- Email

IMPORTANT: After sending files to the module, it is necessary to populate the DataSotre:

For Email:

```
<task classname="TCommandTask_SearchforEmails"
caption="Search for Email"
enabled="true" />
```

For Registry:

```
<task classname="TCommandTask_SearchRegistryHive"
caption="Search for Registry"
enabled="true" />
```

1.24 SIGNATURE ANALYSIS - ALL

```
<!-- File Signature Analysis All Files -->
<task classname="TCommandTask_FileTypeAnalysis"
caption="Signature Analysis"
enabled="true"/>
```

1.25 SIGNATURE ANALYSIS - INDIVIDUAL

```
<!-- Signature Analysis -->
<task classname="TCommandTask_FileTypeAnalysis"
  caption="Signature Analysis"
  description=""
  enabled="true"
  newcolumnindex="3">
  <drivers>
    <driver classname="TJPGDriver"/>
    <driver classname="TZIPDriver"/>
  </drivers>
</task>
```

1.26 VERIFY DEVICE HASHES

```
<!-- Verify Device Hashes -->
<task classname="TCommandTask_VerifyDevice"
  caption="Verify Device Hashes"
  description=""
  enabled="true"
  md5hash="true"
  shalhash="false"
  sha256hash="false"/>
```

1.27 YARA

```
<!-- Yara -->
<task classname="TCommandTask_Yara"
  caption="Yara"
  description="Yara rules."
  enabled="true"
  MaxScanSizeKB="50"
  FastScan="true"
  FileTimeout="60"
  BookmarkMatched="true"/>
```


APPENDIX 3 - FORENSIC EXPLORER FILE DRIVER

File drivers are used in tasks such as Signature Analysis and File Carve.

```

</drivers>
<driver classname="T3GPDriver"/>
<driver classname="T7ZIPDriver"/>
<driver classname="TABDriver"/>
<driver classname="TACCESSDriver"/>
<driver classname="TAFTDriver"/>
<driver classname="TAIFFDriver"/>
<driver classname="TAMRDriver"/>
<driver classname="TAppleNumberDriver"/>
<driver classname="TApplePagesDriver"/>
<driver classname="TASFDriver"/>
<driver classname="TAVIDriver"/>
<driver classname="TBKFDriver"/>
<driver classname="TBMPDriver"/>
<driver classname="TBplist_iOSSearchHistory"/>
<driver classname="TBplist_iTunesBackup"/>
<driver classname="TBplist_SafariSearches"/>
<driver classname="TBPLISTDriver"/>
<driver classname="TBZ2Driver"/>
<driver classname="TCABDriver"/>
<driver classname="TCAFFDriver"/>
<driver classname="TCAN1Driver"/>
<driver classname="TCAN2Driver"/>
<driver classname="TCDCDriver"/>
<driver classname="TCDIDriver"/>
<driver classname="TCINEDriver"/>
<driver classname="TCORELDriver"/>
<driver classname="TCPFDriver"/>
<driver classname="TCRWDriver"/>
<driver classname="TCTWDriver"/>
<driver classname="TCWPDriver"/>
<driver classname="TD2SDriver"/>
<driver classname="TDBFDriver"/>
<driver classname="TDBXDriver"/>
<driver classname="TDGNDriver"/>
<driver classname="TDICOMDriver"/>
<driver classname="TDocXDriver"/>
<driver classname="TDSNDriver"/>
<driver classname="TDSSDriver"/>
<driver classname="TDSStoreDriver"/>
<driver classname="TDVFDriver"/>
<driver classname="TDWGDriver"/>
<driver classname="TDXFDriver"/>
<driver classname="TDYNDriver"/>
<driver classname="TEFAXDriver"/>
<driver classname="TELFDriver"/>
<driver classname="TEmailDriver"/>
<driver classname="TEMFDriver"/>
<driver classname="TENLDriver"/>
<driver classname="TEPSDriver"/>
<driver classname="TEPUBDriver"/>
<driver classname="TESEDriver"/>
<driver classname="TESEPrivStoreDriver"/>
<driver classname="TESEWndSrchDriver"/>
<driver classname="TEEventLogDriver"/>
<driver classname="TEEventLogXDriver"/>
<driver classname="TEXEDriver"/>
<driver classname="TexFATRecord"/>

```

```
<driver classname="TFATRecord"/>
<driver classname="TFATRecord"/>
<driver classname="TFBXDriver"/>
<driver classname="TFDRDriver"/>
<driver classname="Tff7Driver"/>
<driver classname="TFH10Driver"/>
<driver classname="TFH8Driver"/>
<driver classname="TFLACDriver"/>
<driver classname="TFLGDriver"/>
<driver classname="TFLPDriver"/>
<driver classname="TFLVDriver"/>
<driver classname="TFMPDriver"/>
<driver classname="TFMZDriver"/>
<driver classname="TFOXDriver"/>
<driver classname="TFWADriver"/>
<driver classname="TFZDDriver"/>
<driver classname="TGBKDriver"/>
<driver classname="TGDBDriver"/>
<driver classname="TGIFDriver"/>
<driver classname="TGUEDriver"/>
<driver classname="TGZIPDriver"/>
<driver classname="THDPhotoDriver"/>
<driver classname="THDTVDriver"/>
<driver classname="THEIFDriver"/>
<driver classname="THELPDriver"/>
<driver classname="THFDDriver"/>
<driver classname="THFSCatalogRecord"/>
<driver classname="THPGDriver"/>
<driver classname="THTMLDriver"/>
<driver classname="TICODriver"/>
<driver classname="TIEFAVDriver"/>
<driver classname="TIEIdxDatDriver"/>
<driver classname="TINDDDriver"/>
<driver classname="TINTERBASEDriver"/>
<driver classname="TiPhoneWIFIDriver"/>
<driver classname="TISO_DirRecord"/>
<driver classname="TJB2Driver"/>
<driver classname="TJETAUDDriver"/>
<driver classname="TJISDriver"/>
<driver classname="TJPG2000Driver"/>
<driver classname="TJPGDriver"/>
<driver classname="TJSONDriver"/>
<driver classname="TLACERTEDriver"/>
<driver classname="TLNKDriver"/>
<driver classname="TLSODriver"/>
<driver classname="TLWODriver"/>
<driver classname="TLWPDDriver"/>
<driver classname="TLWSDriver"/>
<driver classname="TLZHDriver"/>
<driver classname="TM4ADriver"/>
<driver classname="TMAYADriver"/>
<driver classname="TMBDBDriver"/>
<driver classname="TMDXDriver"/>
<driver classname="TMFTRecord"/>
```

```
<driver classname="TMIDIIDriver"/>
<driver classname="TMimeEmail"/>
<driver classname="TMimeFile"/>
<driver classname="TMKVDriver"/>
<driver classname="TMoneyDriver"/>
<driver classname="TMOVDriver"/>
<driver classname="TMP3Driver"/>
<driver classname="TMPEGDriver"/>
<driver classname="TMPSDriver"/>
<driver classname="TMUSDriver"/>
<driver classname="TMYOBDriver"/>
<driver classname="TNK2Driver"/>
<driver classname="TNSFDriver"/>
<driver classname="TODS_OpenOffice_Driver"/>
<driver classname="TODT_OpenOffice_Driver"/>
<driver classname="TOggDriver"/>
<driver classname="TOLE3DStudio"/>
<driver classname="TOLECrystal"/>
<driver classname="TOLEDesignCAD"/>
<driver classname="TOLEDesignPro"/>
<driver classname="TOLEDGNCAD"/>
<driver classname="TOLEDriver"/>
<driver classname="TOLEFamilyTree"/>
<driver classname="TOLEFlash"/>
<driver classname="TOLEHANGUL"/>
<driver classname="TOLEI3F"/>
<driver classname="TOLEIchitaro"/>
<driver classname="TOLEInstallFile"/>
<driver classname="TOLEJumpList"/>
<driver classname="TOLELabelMighty"/>
<driver classname="TOLEMSAccess"/>
<driver classname="TOLEMSExcel"/>
<driver classname="TOLEMSOutlookMsg"/>
<driver classname="TOLEMSPowerPoint"/>
<driver classname="TOLEMSProject"/>
<driver classname="TOLEMSVisio"/>
<driver classname="TOLEMSWord"/>
<driver classname="TOLEMSWorks"/>
<driver classname="TOLEMSWorksDB"/>
<driver classname="TOLEPageMaker"/>
<driver classname="TOLEPhotoDraw"/>
<driver classname="TOLEPrintMaster"/>
<driver classname="TOLEPublisher"/>
<driver classname="TOLEQuattroPro"/>
<driver classname="TOLERss"/>
<driver classname="TOLEThumbNail"/>
<driver classname="TOLETurboCAD"/>
<driver classname="TOLEWordPerfect"/>
<driver classname="TOLEWPPresentation"/>
<driver classname="TOMNISDriver"/>
<driver classname="TONEDriver"/>
<driver classname="TORFDriver"/>
<driver classname="TOSTDriver"/>
<driver classname="TPABDriver"/>
<driver classname="TPAFDriver"/>
<driver classname="TPAPERDriver"/>
<driver classname="TPCXDriver"/>
<driver classname="TPDFDriver"/>
<driver classname="TPFDriver"/>
<driver classname="TPNGDriver"/>
<driver classname="TPPJDriver"/>
<driver classname="TPPMDriver"/>
<driver classname="TPPTXDriver"/>
```

```
<driver classname="TpsdDriver"/>
<driver classname="TPSPDriver"/>
<driver classname="TPSTDDriver"/>
<driver classname="TqbbDriver"/>
<driver classname="TQBWDriver"/>
<driver classname="TQCADDriver"/>
<driver classname="TQCFDriver"/>
<driver classname="TQDFDriver"/>
<driver classname="TQTAXDriver"/>
<driver classname="TQXDDriver"/>
<driver classname="TRAFDriver"/>
<driver classname="TRAMDriver"/>
<driver classname="TRARDriver"/>
<driver classname="TRegDriver"/>
<driver classname="TRETRODriver"/>
<driver classname="TRIFFDriver"/>
<driver classname="TRNSDriver"/>
<driver classname="TRPMDriver"/>
<driver classname="TRPTDriver"/>
<driver classname="TRTFDriver"/>
<driver classname="TRW2Driver"/>
<driver classname="TSAFARICOOKIEDriver"/>
<driver classname="TSafariHistoryDriver"/>
<driver classname="TSAS1Driver"/>
<driver classname="TSASDriver"/>
<driver classname="TSDDriver"/>
<driver classname="TSHDDriver"/>
<driver classname="TSHPDriver"/>
<driver classname="TSIBDriver"/>
<driver classname="TSIMDriver"/>
<driver classname="TSKFDriver"/>
<driver classname="TSPLDriver"/>
<driver classname="TSPSSDriver"/>
<driver classname="TSQLDriver"/>
<driver classname="TSQLite_AndroidContacts"/>
<driver classname="TSQLite_AndroidSMS"/>
<driver classname="TSQLite_AndroidWIFI"/>
<driver classname="TSQLite_CocoAndroid"/>
<driver classname="TSQLite_CocoiOS"/>
<driver classname="TSQLite_iMessenger"/>
<driver classname="TSQLite_iOSCalendar"/>
<driver classname="TSQLite_iOSContacts"/>
<driver classname="TSQLite_iOSNotes"/>
<driver classname="TSQLite_iOSv3SMS"/>
<driver classname="TSQLite_iOSv4SMS"/>
<driver classname="TSQLite_iTunesManifest"/>
<driver classname="TSQLite_KikAndroid"/>
<driver classname="TSQLite_KikiOS"/>
<driver classname="TSQLite_LineiOS"/>
<driver classname="TSQLite_MessageMeiOS"/>
<driver classname="TSQLite_TouchiOS"/>
<driver classname="TSQLite_WhatsAppiOS"/>
<driver classname="TSQLiteChromeAutofill"/>
<driver classname="TSQLiteChromeCookies"/>
```

```
<driver classname="TSQLiteChromeFavicons"/>
<driver classname="TSQLiteChromeHistory"/>
<driver classname="TSQLiteChromeLogins"/>
<driver classname="TSQLiteChromeShortcuts"/>
<driver classname="TSQLiteChromeTopSites"/>
<driver classname="TSQLiteFirefoxCookies"/>
<driver classname="TSQLiteFirefoxFormHistory"/>
<driver classname="TSQLiteFirefoxHistory"/>
<driver classname="TSQLiteSafariBookmarks"/>
<driver classname="TSQLiteSafariCache"/>
<driver classname="TSQLiteSKYPE"/>
<driver classname="TSQLiteVIBERAndroid"/>
<driver classname="TSQLiteVIBERiPhone"/>
<driver classname="TSQLiteDriver"/>
<driver classname="TSQLOGDriver"/>
<driver classname="TSummaryInformation"/>
<driver classname="TSWFDriver"/>
<driver classname="TSWIDriver"/>
<driver classname="TTARDriver"/>
<driver classname="TTAXACTDriver"/>
<driver classname="TTAXCUTDriver"/>
<driver classname="TTAXDriver"/>
<driver classname="TTEXTDriver"/>
<driver classname="TTIFFDriver"/>
<driver classname="TTorrentDriver"/>
<driver classname="TTTFDriver"/>
<driver classname="TUMVDriver"/>
<driver classname="TUNIDriver"/>
<driver classname="TUserNetMessage"/>
<driver classname="TUTF8Driver"/>
<driver classname="TUyapDriver"/>
<driver classname="TVIDDriver"/>
<driver classname="TVSSBlockHeaderDriver"/>
<driver classname="TVSSCatalogFileDriver"/>
<driver classname="TVSSStoreFileDriver"/>
<driver classname="TVSSStoreInfoFileDriver"/>
<driver classname="TWABDriver"/>
<driver classname="TWAVEDriver"/>
<driver classname="TWin7ThumbDriver"/>
<driver classname="TWK3Driver"/>
<driver classname="TWKSDriver"/>
<driver classname="TWMFDriver"/>
<driver classname="TWOFFDriver"/>
<driver classname="TWPDDriver"/>
<driver classname="TWRIDriver"/>
<driver classname="TX3FDriver"/>
<driver classname="TXARADriver"/>
<driver classname="TXLSXDriver"/>
<driver classname="TXML_iPhoneBugReport"/>
<driver classname="TXML_iTunesBackup"/>
<driver classname="TXMLDriver"/>
<driver classname="TXPSDriver"/>
<driver classname="TYAHOODriver"/>
<driver classname="TZIPDriver"/>
<drivers>
```

APPENDIX 4 – CLI ERROR CODES

The following error codes can be returned by the CLI:

- `ERRORCODE_OK = 0;`
- `ERRORCODE_FAIL = -1;`
- `ERRORCODE_DIRNONWRITE = -2;`
- `ERRORCODE_CASE_INVALID = -3;`
- `ERRORCODE_CASE_TIMEZONE_NOTFOUND = -4;`
- `ERRORCODE_EVIDENCE_ADD_FAILED = -5;`
- `ERRORCODE_EVIDENCE_TIMEZONE_NOTFOUND = -6;`
- `ERRORCODE_PARAMS_INVALID = -7;`
- `ERRORCODE_UNHANDLED_EXCEPTION = -8;`
- `ERRORCODE_ACTIVATION = -9;`
- `ERRORCODE_INVALID_TXML = -10;`
- `ERRORCODE_INVALIDSECTORSIZE = -11;`