**CACIE Tool #00** – ***Handprinter Tool***

**Version** **1.0**

**QA**: **TEST** or **NA** or **QA**

1. **Description and Purpose**

The Handprinter tool is a batch processor that processes a collection of files and folders by invoking the Fingerprinter tool on each one. An input file lists the collection of files and folders to fingerprint. The Handprinter tool creates one fingerprint file for each item in the input file and stores the results in the target directory.

1. **Functional Requirements**

The following are the functional requirements of the Handprinter tool.

FR-1: Parse the files and/or folder listed in the input file.

FR-2: Invoke the Fingerprinter tool for each file and/or folder listed in the input file.

1. **Software Requirements Specifications**

The following documents the software requirements for the Handprinter tool.

Python 3.5

Python Standard Libraries:  
os  
sys  
argparse

Modules:  
.\fingerprint.py

1. **Software Design Description**

The following is a brief description of the required arguments and the output generated by the Handprinter tool:

Positional Arguments:

* inputfile: Path to a file containing the items to process through the Fingerprinter tool. Each line should contain at least a path, and optionally a filename. Blank lines or those starting with a '#' are ignored. The path specifies the path to a file or folder to fingerprint. The optional filename specifies the name of the fingerprint file that is generated for the specified path.

Optional Arguments:

* -h, --help
* - -outdir OUTDIR: Path to the folder where you want to store the Fingerprinter tool output. Defaults to the current directory.
* - -sep, SEP: the type of character that separates columns in the input file. Defaults to a comma (,); specify a tab or a space delimiter as ‘s’

Shell file configuration:

python [directory path]/pylib/handprint/handprint.py [optional arguments—see above] inputfile

1. **Requirements Traceability Matrix**

The requirements traceability matrix for the Handprinter tool is presented in Table 1.

| **Table 1. Handprinter Tool Requirements Traceability Matrix** | | |
| --- | --- | --- |
| **Functional Requirement ID** | **Acceptance Test ID** | **Test Case** |
| QA Level | IT-1 | Installation Test |
| FR-1 FR-2 | ATC-1 | FIngerprints files and/or directories listed in an input file |

1. **Test Plan and Cases**

The test plan for the Handprinter tool is presented in Table 2.

| **Table 2. Handprint Test Plan** | | |
| --- | --- | --- |
| **TEST ID** | **Test Case** | **Test Result  (Pass/Fail)** |
| *Note [Testing\_Directory] in acceptance test report* | | |
| IT-1 | *Navigate to [Testing\_Directory]\handprint\_test* | |
| *Invoke Tool Runner and Handprint tool using runner\_ITC-1.sh by entering the following at the command line:./runner\_ITC-1.sh* | |
| Verify Tool Runner is invoked and executes |  |
| Verify Handprint tool executes |  |
| *If testing on Windows, open git bash window in [Testing\_Directory].*  *If testing on Linux, navigate to [Testing\_Directory].* | | |
| ATC-1 | *Navigate to the [Testing\_Directory]\handprint\_test* | |
| *Enter the following command:*  *python handprint.py ATC-1\_input.txt - - outdir [Testing\_Directory]\ATC-1* | |
| Verify that the fingerprints outputted to *[Testing\_Directory]\ATC-1 correspond to the collection of files and directories listed in the input file ATC-1\_input.txt* |  |

1. **Acceptance Test Report**

The test report will state whether the tool is qualified for use, summarize test case results, and report all resolved incidents and resolution of unresolved incidents.

1. **User Guide**

The Handprinter tool can be invoked from the command line using the arguments as specified in Section 4 (Software Design) and the arguments for the invoked tool. The Handprinter tool can also be invoked using the Tool Runner tool.