This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Log File Masking Utility User Guide

Genesys Care/Support current

7/18/2022

Table of Contents

Log File Masking Utility User’s Guide

Overview

The Genesys Care Log File Masking Utility is provided for the purpose of masking sensitive data in log files.

The intent of the Genesys Care Log File Masking Utility is to provide Genesys customers and partners with a utility for masking sensitive data in any/all log files provided to Genesys Customer Care.

Sensitive data can include many different types of information, usually defined by the business (or its customers), examples could include credit card numbers, personal addresses, or server hostnames/IP addresses etc etc.

The protection of sensitive data is a major concern for any business data and must be handled properly.

Genesys Customer Care handles thousands of support cases a year, the resolution of these support cases usually requires the common practice of reviewing application log files.

Although Genesys applications can be configured to limit the presence of sensitive data in log files, it is possible for sensitive data to exist. This is attributable to:

1. Customer or partner created applications that integrate with Genesys applications or SDKs
2. No standard definitions of sensitive data.

Data Privacy

> **Important**
>
> Log File Masking Utility (LFMU) does not store data, it redacts log files based on customer regular expressions; please ensure any/all log files that are uploaded to Genesys Customer Care do NOT contain any Personally Identifiable Information (PII).

New in this Release

This section describes the new functionality for each release.

Release 9.0.000.20

- LFMU now supports multiline regex
- LFMU now uses log4j 2.17.1

Release 9.0.000.00

- LFMU now supports command line arguments enabling usage flexibility and automation.
- Matched values in the outputted log files can be **Hashed** or **Masked**
  - Hashed => "8F579D7F05607436981DBC2C88B08857"
  - Masked => "***"
- The advantage of using "Hash" redaction is post correlation analysis that isn't possible when values are simply replaced with "*" characters

Release 8.5.000.01

This is the first public release of the Genesys Care Log File Masking Utility.

Known Issues and Limitations

There are currently no known issues for this release of the Genesys Care Log File Masking Utility.

Prerequisites

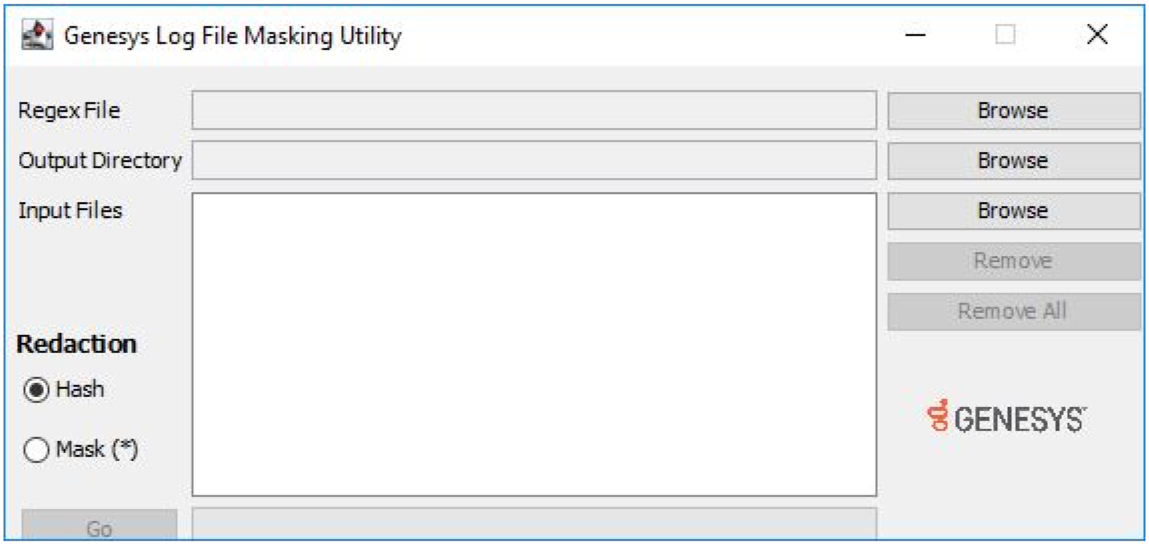
The following software must be installed before using the Genesys Care Log File Masking Utility:

- Java Runtime Environment (JRE) 1.8 or newer
- OpenJDK 11

Using the Utility

UI Mode

The Genesys Care Log File Masking Utility (LFMU) has been designed as an executable .jar file, opening the file, with no command line arguments, on a GUI OS, using a supported Java Runtime Environment, will display the main LFMU interface.



Redaction

● Hash

○ Mask (*)

Using this interface, the user must provide three inputs:

1. **Regex File** – The file containing regular expressions to determine which characters will be masked.
2. **Output Directory** – The location where the Log File Masking Utility will place the processed log files; **this must be a directory**.
3. **Input Files** – One or more files that will be processed by the Log File Masking Utility using the regular expressions found in the Regex File. This can be any plain text file.

- Each input location provides a **Browse** button that allows the user to select a file or folder from the local file system.
- Removing individual files from the **Input Files** list can be completed by selecting (or multi-selecting) files in the pane and clicking the **Remove** button.
- All files in the **Input Files** pane can be removed by clicking the **Remove All** button.
- When all three inputs are provided to the LFMU, the **Go** button will be enabled.
- Clicking the **Go** button will start the process of masking data.
- A status bar will show the tool’s progress.

- When the masking is completed, the processed files will be available in the **Output Directory** location with the suffix and extension *_mskd.log*.

New to the LFMU 9 UI are the "Redaction" options:

1. **Hash** (default) to hash the matched values
2. **Mask** to "*" the matched values

Command Line Mode

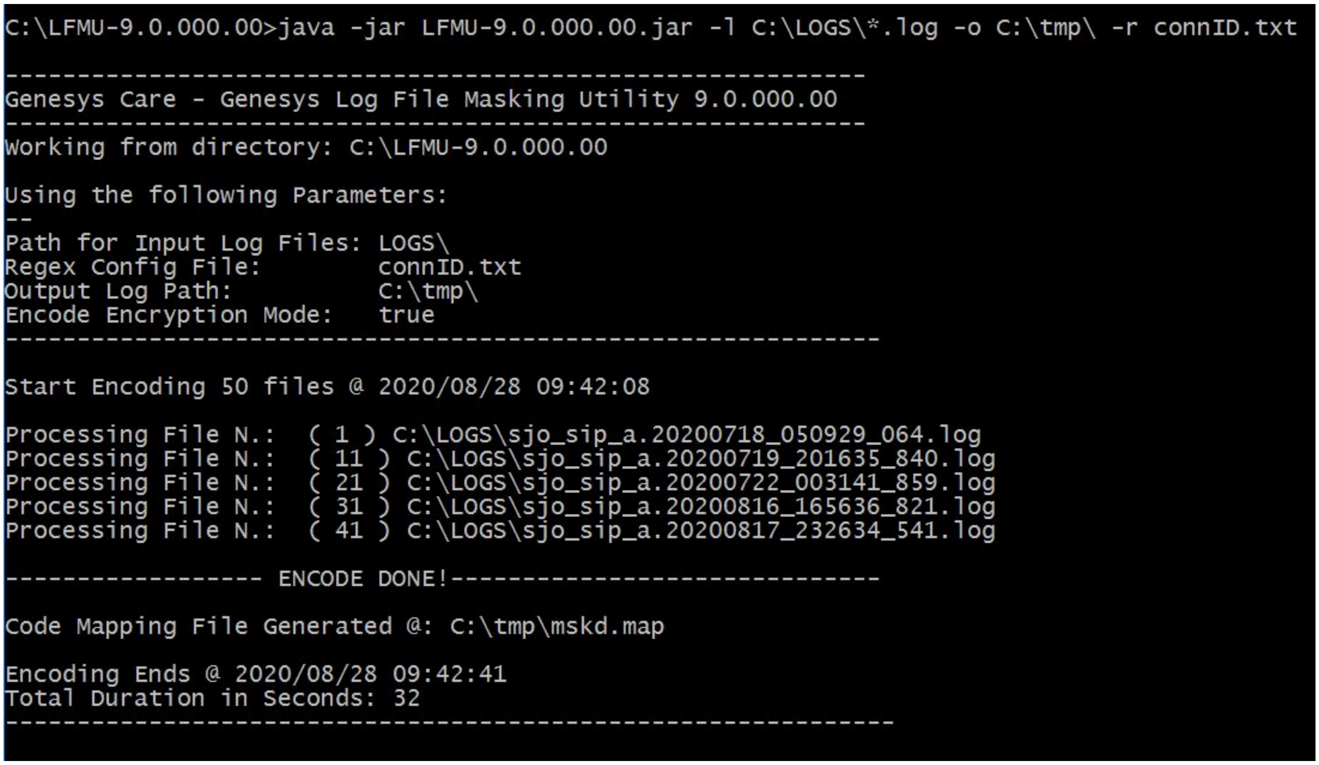
The LFMU can now be used using command line arguments, this enables flexibility and automation.
To view the LFMU command line argument options, use **"java -jar LFMU-9.x.xxx.xx.jar -h"**
The LFMU command line options being:

```
-h,--help             This help
-l,--logPath <arg>    Log File Name Path (Mandatory)
-m,--mask             Enable encoding Mask ("*") - For sensitive data use the default
encryption
-o,--outpath <arg>    Log File Destination Directory (Mandatory)
-r,--rgxCFile <arg>   Regex Configuration File PATH/Name (Mandatory)
-t,--threads <arg>    Number of Threads (default 10 - Max 50) - Adjust as needed
```

Command Line Mode - Example 1 - Hash/Encrypt

The example usage below:

- Processes ALL the **.log** files in the **C:\LOGS** directory - this is the **-l** argument
- Outputs the **Hashed** files to the **C:\tmp** directory - this is the **-o** argument
- The input files are matched/hashed based on the **connID.txt** Regex file - this is the **-r** argument
- Note the outputted **mskd.map** file, which is also in **C:\tmp**; this contains the *hashed to original* values for correlation/analysis



```
C:\LFMU-9.0.000.00>java -jar LFMU-9.0.000.00.jar -l C:\LOGS\*.log -o C:\tmp\ -r connID.txt

------------------------------------------------------------
Genesys Care - Genesys Log File Masking Utility 9.0.000.00
------------------------------------------------------------
Working from directory: C:\LFMU-9.0.000.00

Using the following Parameters:
--
Path for Input Log Files: LOGS\
Regex Config File:        connID.txt
Output Log Path:          C:\tmp\
Encode Encryption Mode:   true
-------------------------------------------------------------

Start Encoding 50 files @ 2020/08/28 09:42:08

Processing File N.:  ( 1 ) C:\LOGS\sjo_sip_a.20200718_050929_064.log
Processing File N.:  ( 11 ) C:\LOGS\sjo_sip_a.20200719_201635_840.log
Processing File N.:  ( 21 ) C:\LOGS\sjo_sip_a.20200722_003141_859.log
Processing File N.:  ( 31 ) C:\LOGS\sjo_sip_a.20200816_165636_821.log
Processing File N.:  ( 41 ) C:\LOGS\sjo_sip_a.20200817_232634_541.log

------------------ ENCODE DONE!------------------------------

Code Mapping File Generated @: C:\tmp\mskd.map

Encoding Ends @ 2020/08/28 09:42:41
Total Duration in Seconds: 32
--------------------------------------------------------------
```

An example of the **Hashed** content being:

- AttributeConnID ACC313D648E9907221466EE00BDE5B8B
- AttributeConnID D8C97E4C482AFD1C0CD4D3201579FC8F
- AttributeConnID 023CE218A53605C0A04231A79E0ED0BD

An example of the content of **mskd.map** being:

- 8F579D7F05607436981DBC2C88B08857 <=::=> 007602fd7d42a494
- 004B736BBE797DD101557B29C10FDFF9 <=::=> 0069030117e580de
- 02DC7565E3F5A308DD50A135E00350AA <=::=> 0068030117e581fd

The content of the **connID.txt** Regex file being:

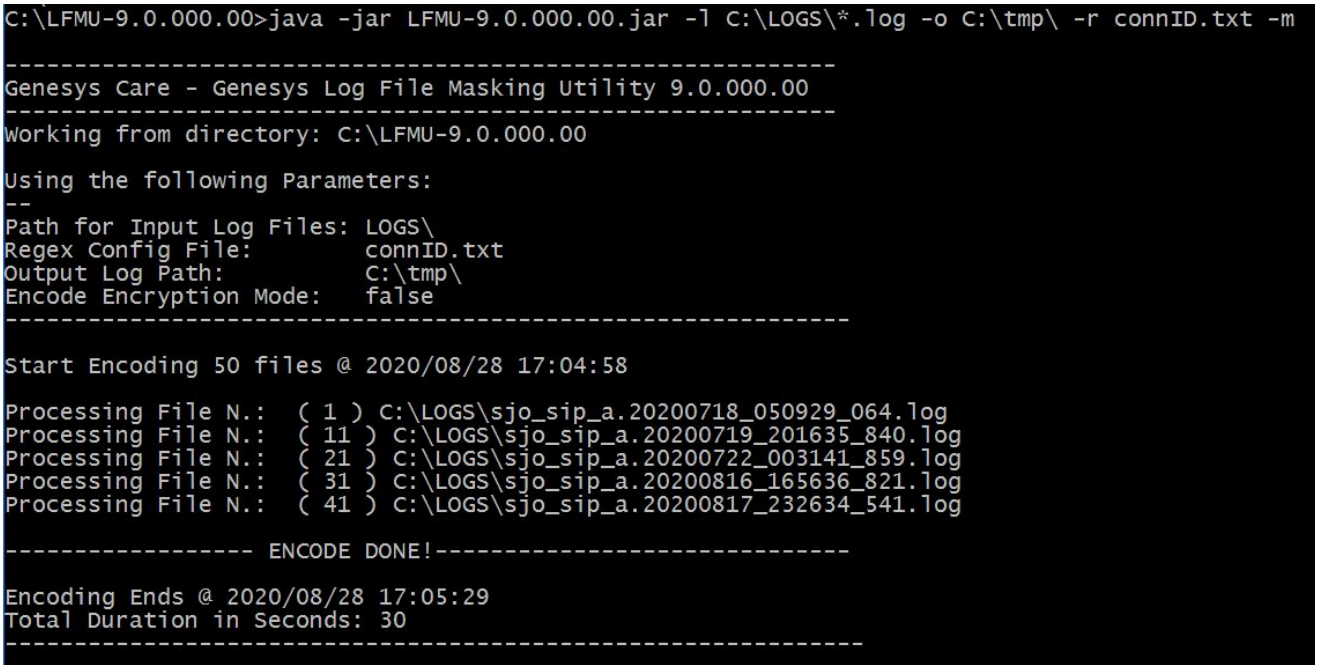
- \tAttributeConnID\t(\S\S\S\S\S\S\S\S\S\S\S\S\S\S\S\S)

Command Line Mode - Example 2 - Mask

The example usage below:

- Processes ALL the **.log** files in the **C:\LOGS** directory - this is the **-l** argument
- Outputs the **Masked** files to the **C:\tmp** directory - this is the **-o** argument
- The input files are matched/masked based on the **connID.txt** Regex file - this is the **-r** argument

- Given the "**-m**" argument, the outputted files in **C:\tmp** are **Masked** with **"*"** characters as opposed to hashed



```
C:\LFMU-9.0.000.00>java -jar LFMU-9.0.000.00.jar -l C:\LOGS\*.log -o C:\tmp\ -r connID.txt -m

------------------------------------------------------------
Genesys Care - Genesys Log File Masking Utility 9.0.000.00
------------------------------------------------------------
Working from directory: C:\LFMU-9.0.000.00

Using the following Parameters:
--
Path for Input Log Files: LOGS\
Regex Config File:        connID.txt
Output Log Path:          C:\tmp\
Encode Encryption Mode:   false
-------------------------------------------------------------

Start Encoding 50 files @ 2020/08/28 17:04:58

Processing File N.:  ( 1 ) C:\LOGS\sjo_sip_a.20200718_050929_064.log
Processing File N.:  ( 11 ) C:\LOGS\sjo_sip_a.20200719_201635_840.log
Processing File N.:  ( 21 ) C:\LOGS\sjo_sip_a.20200722_003141_859.log
Processing File N.:  ( 31 ) C:\LOGS\sjo_sip_a.20200816_165636_821.log
Processing File N.:  ( 41 ) C:\LOGS\sjo_sip_a.20200817_232634_541.log

------------------ ENCODE DONE!------------------------------

Encoding Ends @ 2020/08/28 17:05:29
Total Duration in Seconds: 30
--------------------------------------------------------------
```

An example of the **Masked** content being:

- Line 356633: AttributeConnID ****************
- Line 356680: AttributeConnID ****************
- Line 356712: AttributeConnID ****************

The content of the **connID.txt** Regex file being:

- \tAttributeConnID\t(\S\S\S\S\S\S\S\S\S\S\S\S\S\S\S\S)

Building Regular Expression Files

The regular expression file used by the Genesys Care Log File Masking Utility dictates what information will be masked by the tool, this regex file is in plain text format and contains one regular expression per line.

Regular Expressions are created using the Java Regular Expression protocol, Regular expression groups identify the data to be masked.

Genesys strongly recommends that you test regular expressions before adding them to your regular expression file.

The example below shows two sample regular expressions.

- The first line is to remove credit card numbers as they appear in a fictitious log file.
- The second line attempts to remove the first and last name of a user from the same log file.

Regex.txt

```
\tAttributeCreditCardNum\t’(\d+)’
\tAttributeFullName\t(\S+)\t\S+\t(\S+)
```

The regular expression on line one of **Regex.txt** will attempt to match:

1. **\t**– A tab character.
2. **AttributeCreditCardNum** - The text AttributeCreditCardNum.
3. **\t** – A tab character.
4. ' - A single quote.
5. **\d+** - A number of digits.
6. ' - A single quote.

- In the above expression, the **()** surround the **\d+** to define it as a *group*.
- The text matched inside the *group* will be that which is masked in the file.

The regular expression on line two of **Regex.txt** will attempt to match:

1. **\t**– A tab character.
2. **AttributeFullName** - The text AttributeFullName.
3. **\t** – A tab character.
4. **\S+** - A number of non-whitespace characters.
5. **\t** – A tab character.
6. **\S+** - A number of non-whitespace characters.

7. **\t** – A tab character.
8. **\S+** - A number of non-whitespace characters.

- In the above expression, *two groups* are created by surrounding the first and third instance of **\S+** with **()**.
- Applying the **Regex.txt** above to the following application log file **test.log** below gives the following results:

test.log

```
This is a test file to showcase the Genesys Care Log File Masking Utility.
     AttributeCreditCardNum         '4862458796590204'
     AttributeFullName         John Albert     Smith
The two lines above this one are indented with a tab character.
There is also a tab character between John Smith’s first, middle and last name.
```

test.log_mskd.log

```
This is a test file to showcase the Genesys Care Log File Masking Utility.
     AttributeCreditCardNum        '****************'
     AttributeFullName        **** Albert        *****
The two lines above this one are indented with a tab character.
There is also a tab character between John Smith’s first, middle and last name.
```

Notice that the output of the Log File Masking Utility masks characters with asterisks. Furthermore, take note that because the middle name in the second regular expression was not surrounded with parentheses, the middle name was not masked.

> ### Tip
>
> 1. Tools like Regex Coach (http://www.weitz.de/regex-coach/) can be used to test the syntax of the regular expressions.
> 2. Help on regular expression syntax can be found at https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html.