



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

Building Regular Expression Files

9/1/2025

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>.