



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.

Platform SDK Developer's Guide

[Hide or Tag Sensitive Data in Logs](#)

12/14/2025

Hide or Tag Sensitive Data in Logs

The [Genesys Security Deployment Guide](#) describes common options to filter out or tag sensitive data in logs (in KeyValueCollection entries).

- The default-filter-type option in the [log-filter] section defines the treatment for all Key-Value pairs.
- The <key-name> options in the [log-filter-data] section define the treatment for specific keys in the log, overriding the default treatment specified by default-filter-type.

Corresponding configurations can also be applied for the Platform SDK KeyValuePrinter:

```
KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(globalPrinterOpts,
individualKeyMapping));
```

where globalPrinterOpts and individualKeyMapping are KeyValueCollection objects with filter names and filter options.

Using Default Filters

Most KeyValueCollection objects (CfgApplication configuration options) can be read from Configuration Server and applied to the KeyValuePrinter directly:

```
CfgApplication application = ...;

KeyValueCollection options = application.getOptions();
KeyValueCollection globalPrinterOpts= options.getList("log-filter");
KeyValueCollection individualKeyMapping = options.getList("log-filter-data");
KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(globalPrinterOpts,
individualKeyMapping));
```

Prior to 8.5.102.00, standard tag filters configuration could not be applied as-is and required additional parsing.

The table below demonstrates filter samples from the [Genesys Security Deployment Guide](#) and corresponding KeyValuePrinter settings.

Masking Partial Values	
Configuration options in CME	Corresponding KeyValueCollection content
[log-filter] default-filter-type=hide-first,3	<pre>[Java] KeyValueCollection globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.addString("default-filter- type", "hide- first,3"); KeyValuePrinter.setDefaultPrinter(new</pre>

Masking Partial Values	
	<pre> KeyValuePrinter(globalPrinterOpts , null)); [.NET] var globalPrinterOpts = new KeyValuePairCollection(); globalPrinterOpts.Add("default-filter- type", "hide- first,3"); KeyValuePair.DefaultPrinter = new KeyValuePair(globalPrinterOpts , null); </pre>
KVList: <pre> 'DNIS' [str] = "****0" 'PASSWORD' [str] = "****111111" 'RECORD_ID' [str] = "****3427" </pre>	
Using Default Tags	
Configuration options in CME	Corresponding KeyValuePairCollection content
[log-filter] default-filter-type=tag()	<pre> [Java] KeyValuePairCollection globalPrinterOpts = new KeyValuePairCollection(); globalPrinterOpts.addString("default-filter- type", "tag()"); KeyValuePair.setDefaultPrinter(new KeyValuePair(globalPrinterOpts , null)); [Java, Prior to 8.5.102.00] KeyValuePairCollection globalPrinterOpts = new KeyValuePairCollection(); globalPrinterOpts.addString("default-filter- type", "custom-filter"); globalPrinterOpts.addString("custom-filter- type", "PrefixPostfixFilter"); KeyValuePairCollection filterOpts = new KeyValuePairCollection(); filterOpts.addString("key-prefix-string", ""); filterOpts.addString("key-postfix-string", ""); filterOpts.addString("value-prefix-string", "<#"); filterOpts.addString("value-postfix- string", "#>"); globalPrinterOpts.addList("custom-filter- options", filterOpts); KeyValuePair.setDefaultPrinter(new KeyValuePair(globalPrinterOpts, null)); [.NET] var globalPrinterOpts = new KeyValuePairCollection(); globalPrinterOpts.Add("default-filter- </pre>

Masking Partial Values	
	<pre>type", "tag()"); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(globalPrinterOpts , null); [.NET Prior to 8.5.102.00] var globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.Add("default-filter- type", "custom-filter"); globalPrinterOpts.Add("custom-filter-type", "Genesyslab.Platform.Commons.Collections.Filters.PrefixPostfixFilter"); var filterOpts = new KeyValueCollection(); filterOpts.Add("key-prefix-string", ""); filterOpts.Add("key-postfix-string", ""); filterOpts.Add("value-prefix-string", "<#"); filterOpts.Add("value-postfix-string", "#>"); globalPrinterOpts.Add("custom-filter- options", filterOpts); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(globalPrinterOpts, null);</pre>
KVList: 'DNIS' [str] = <#"8410"#> 'PASSWORD' [str] = <#"11111111"#> 'RECORD_ID' [str] = <#"8313427"#>	
Using User-defined Tags for All Attributes	
Configuration options in CME	Corresponding KeyValueCollection content
	<pre>[Java] KeyValueCollection globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.addString("default-filter- type", "tag(<**, **>)"); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(globalPrinterOpts , null));</pre>
[log-filter] default-filter-type=tag(<**, **>)	<pre>[Java, Prior to 8.5.102.00] KeyValueCollection globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.addString("default-filter- type", "custom-filter"); globalPrinterOpts.addString("custom-filter- type", "PrefixPostfixFilter"); KeyValueCollection filterOpts = new KeyValueCollection(); filterOpts.addString("key-prefix-string", ""); filterOpts.addString("key-postfix-string", "");</pre>

Masking Partial Values	
	<pre> filterOpts.addString("value-prefix-string", "<*""); filterOpts.addString("value-postfix- string", "**>"); globalPrinterOpts.addList("custom-filter- options", filterOpts); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(globalPrinterOpts, null)); [.NET] var globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.Add("default-filter- type", "tag(<*,**>)"); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(globalPrinterOpts, null); [.NET Prior to 8.5.102.00] var globalPrinterOpts = new KeyValueCollection(); globalPrinterOpts.Add("default-filter- type", "custom-filter"); globalPrinterOpts.Add("custom-filter-type", "Genesyslab.Platform.Commons.Collections.Filters.PrefixPostfix"); var filterOpts = new KeyValueCollection(); filterOpts.Add("key-prefix-string", ""); filterOpts.Add("key-postfix-string", ""); filterOpts.Add("value-prefix-string", "<*""); filterOpts.Add("value-postfix-string", "**>"); globalPrinterOpts.Add("custom-filter- options", filterOpts); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(globalPrinterOpts, null); </pre>
KVList: <pre> 'DNIS' [str] = <*"8410"*> 'PASSWORD' [str] = <*"1111111111"*> 'RECORD_ID' [str] = <*"8313427"*> </pre>	
Masking Individual Values in Selected KV Pairs	
Configuration options in CME	Corresponding KeyValueCollection content
[log-filter-data] PASSWORD=hide	<pre> [Java] KeyValueCollection individualKeyMapping = new KeyValueCollection(); individualKeyMapping.addString("PASSWORD", "hide"); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(null, individualKeyMapping)); </pre>

Masking Partial Values	
	<pre>[.Net] var individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.Add("PASSWORD", "hide"); KeyValuePairPrinter.DefaultPrinter = new KeyValuePairPrinter(null, individualKeyMapping);</pre>
KVList: <pre>'DNIS' [str] = "8410" 'PASSWORD' [output suppressed] 'RECORD_ID' [str] = "8313427"</pre>	
Masking Partial Values in Selected KV Pairs	
Configuration options in CME	Corresponding KeyValuePairCollection content
[log-filter-data] PASSWORD=unhide-last,5	<pre>[Java] KeyValuePairCollection individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.addString("PASSWORD", "unhide-last,5"); KeyValuePairPrinter.setDefaultPrinter(new KeyValuePairPrinter(null, individualKeyMapping)); [.NET] var individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.Add("PASSWORD", "unhide-last,5"); KeyValuePairPrinter.DefaultPrinter = new KeyValuePairPrinter(null, individualKeyMapping);</pre>
KVList: <pre>'DNIS' [str] = "8410" 'PASSWORD' [str] = "*****11111" 'RECORD_ID' [str] = "8313427"</pre>	
Tagging Specific KV Pairs with Default Tags	
Configuration options in CME	Corresponding KeyValuePairCollection content
[log-filter-data] PASSWORD=tag()	<pre>[Java] KeyValuePairCollection individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.addString("PASSWORD", "tag()"); KeyValuePairPrinter.setDefaultPrinter(new KeyValuePairPrinter(null, individualKeyMapping));</pre>

Masking Partial Values	
	<pre>[Java, Prior to 8.5.102.00] KeyValueCollection customFilter = new KeyValueCollection(); customFilter.addString("custom-filter- type", "PrefixPostfixFilter"); KeyValueCollection individualKeyMapping = new KeyValueCollection(); individualKeyMapping.addList("PASSWORD", customFilter); KeyValueCollection filterOpts = new KeyValueCollection(); filterOpts.addString("key-prefix-string", ""); filterOpts.addString("key-postfix-string", ""); filterOpts.addString("value-prefix-string", "<#"); filterOpts.addString("value-postfix- string", "#>"); customFilter.addList("custom-filter- options", filterOpts); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(null, individualKeyMapping)); [.NET] var individualKeyMapping = new KeyValueCollection(); individualKeyMapping.Add("PASSWORD", "tag()"); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(null, individualKeyMapping); [.NET, Prior to 8.5.102.00] var customFilter = new KeyValueCollection(); customFilter.Add("custom-filter-type", "Genesyslab.Platform.Commons.Collections.Filters.PrefixPostfixFilter"); var individualKeyMapping = new KeyValueCollection(); individualKeyMapping.Add("PASSWORD", customFilter); var filterOpts = new KeyValueCollection(); filterOpts.Add("key-prefix-string", ""); filterOpts.Add("key-postfix-string", ""); filterOpts.Add("value-prefix-string", "<#"); filterOpts.Add("value-postfix-string", "#>"); customFilter.Add("custom-filter-options", filterOpts); KeyValuePrinter.DefaultPrinter = new KeyValuePrinter(null, individualKeyMapping);</pre>
KVList: <pre>'DNIS' [str] = "8410" 'PASSWORD' [str] = <#"11111111"#></pre>	

Masking Partial Values	
'RECORD_ID' [str] = "8313427"	
Tagging Individual KV Pairs with Different Tags	
Configuration options in CME	Corresponding KeyValueCollection content
<pre>[log-filter-data] PASSWORD=tag() RECORD_ID=tag(<**, **>)</pre>	<pre>[Java] KeyValueCollection individualKeyMapping = new KeyValueCollection(); individualKeyMapping.addString("PASSWORD", "tag(<!--, -->)"); individualKeyMapping.addString("RECORD_ID", "tag(<**, **>)"); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(null, individualKeyMapping)); [Java prior to 8.5.102.00] KeyValueCollection opts1 = new KeyValueCollection(); opts1.addString("key-prefix-string", ""); opts1.addString("key-postfix-string", ""); opts1.addString("value-prefix-string", "<!-- "); opts1.addString("value-postfix-string", "-- >"); KeyValueCollection opts2 = new KeyValueCollection(); opts2.addString("key-prefix-string", ""); opts2.addString("key-postfix-string", ""); opts2.addString("value-prefix-string", "<**"); opts2.addString("value-postfix-string", "**>"); KeyValueCollection filter1 = new KeyValueCollection(); filter1.addString("custom-filter-type", "PrefixPostfixFilter"); filter1.addList("custom-filter-options", opts1); KeyValueCollection filter2 = new KeyValueCollection(); filter2.addString("custom-filter-type", "PrefixPostfixFilter"); filter2.addList("custom-filter-options", opts2); KeyValueCollection individualKeyMapping = new KeyValueCollection(); individualKeyMapping.addList("PASSWORD", filter1); individualKeyMapping.addList("RECORD_ID", filter2); KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(null,</pre>

Masking Partial Values	
	<pre> individualKeyMapping)); [.NET] var individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.Add("PASSWORD", "tag(<!--,-->"); individualKeyMapping.Add("RECORD_ID", "tag(<**,**)"); KeyValuePairPrinter.DefaultPrinter = new KeyValuePairPrinter(null, individualKeyMapping); [.NET prior to 8.5.102.00] var opts1 = new KeyValuePairCollection(); opts1.Add("key-prefix-string", ""); opts1.Add("key-postfix-string", ""); opts1.Add("value-prefix-string", "<!--"); opts1.Add("value-postfix-string", "-->"); var opts2 = new KeyValuePairCollection(); opts2.Add("key-prefix-string", ""); opts2.Add("key-postfix-string", ""); opts2.Add("value-prefix-string", "<***"); opts2.Add("value-postfix-string", "***>"); var filter1 = new KeyValuePairCollection(); filter1.Add("custom-filter-type", "Genesyslab.Platform.Commons.Collections.Filters.PrefixPostfix"); filter1.Add("custom-filter-options", opts1); var filter2 = new KeyValuePairCollection(); filter2.Add("custom-filter-type", "Genesyslab.Platform.Commons.Collections.Filters.PrefixPostfix"); filter2.Add("custom-filter-options", opts2); var individualKeyMapping = new KeyValuePairCollection(); individualKeyMapping.Add("PASSWORD", filter1); individualKeyMapping.Add("RECORD_ID", filter2); KeyValuePairPrinter.DefaultPrinter = new KeyValuePairPrinter(null, individualKeyMapping); </pre>
KVList: <pre> 'DNIS' [str] = "8410" 'PASSWORD' [str] = <!--"11111111"--> 'RECORD_ID' [str] = <***"8313427"***> </pre>	

Note that the `KeyValuePairPrinter` class has predefined String constants. For example, `KeyValuePairPrinter.DEF_FILTER_OPTION` is equivalent to `default-filter-type`. See the `KeyValuePairPrinter` documentation in the [Platform SDK API Reference](#) guide for details.

Implement Custom Filter

It is possible to write your own filter implementation. To do that, extend the `KeyValueAbstractOutputFilter` class and register it in `KeyValuePrinter` using the `custom-filter-type` option.

A sample filter implementation is provided below:

```
public class SimpleHideFilter extends KeyValueAbstractOutputFilter {
    private KeyValueCollection opts;
    public void configure(final KeyValueCollection options) {
        this.opts = options;
    }
    @Override
    protected String doAppendPairValue(final StringBuffer buf,
        final String key, final Object value,
        final KeyValuePrinterContext context) {
        if (opts != null && "true".equals(opts.getString("enabled"))) {
            buf.append("*** Hidden by simple filter ***");
        } else {
            super.doAppendPairValue(buf, key, value, context);
        }
        return null;
    }
}
```

And here is some code showing how to register your filter:

```
KeyValueCollection filterOpts = new KeyValueCollection();
filterOpts.addString("enabled", "true");

KeyValueCollection customFilterDef = new KeyValueCollection();
customFilterDef.addString(KeyValuePrinter.CUSTOM_FILTER_TYPE,
    SimpleHideFilter.class.getName());
customFilterDef.addList(KeyValuePrinter.CUSTOM_FILTER_OPTIONS, filterOpts);

KeyValueCollection individualKeyMapping = new KeyValueCollection();
individualKeyMapping.addList("PASSWORD", customFilterDef);
KeyValuePrinter.setDefaultPrinter(new KeyValuePrinter(null, individualKeyMapping));
```

The resulting log might look like this:

```
'EventPartyInfo' (109) attributes:
  AttributeCallType [int] = 4 [Consult]
  AttributeConnID [long] = 008b012ece62c8be
  AttributeUserData [bstr] = KVList:
    'DNIS' [int] = 8410
    'PASSWORD' [str] = *** Hidden by simple filter ***
    'RECORD_ID' [int] = 8313427
  AttributeThisDN [str] = "8899"
```