



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 Message Attributes in Logs](#)

Contents

- 1 Hide Message Attributes in Logs
 - 1.1 How to Configure Hidden Attributes
 - 1.2 Using the Application Template Helper to Read Configuration from Config Manager

Hide Message Attributes in Logs

This article describes how to hide message attributes on logs generated by your Platform SDK applications.

The new `ToStringHelper` class described here was introduced in release 8.5.300.02 of Platform SDK.

How to Configure Hidden Attributes

There are two ways to configure hidden attributes within log files generated by Platform SDK, depending on what type of protocols you are working with.

Working With ESP Protocols

The UCS and EspEmail protocols use `Request3rdServer` and `Event3rdServerResponse` messages from the underlying ESP protocol to transport their messages. Thus, `Request3rdServer` and `Event3rdServerResponse` are printed to logs and `KeyValuePrinter` must be configured to hide any sensitive data in logs.

For example:

```
KeyValueCollection defaultCfg = new KeyValueCollection();
KeyValueCollection specificCfg = new KeyValueCollection();
specificCfg.addString("StructuredText", "hide");
KeyValuePrinter defaultKVPrinter = new KeyValuePrinter(defaultCfg, specificCfg);
KeyValuePrinter.setDefaultPrinter(defaultKVPrinter);
```

This is the standard Genesys [log-filter-data] implementation described by the [Hide or Tag Sensitive Data in Logs](#) article.

In cases where UCS protocol messages are printed out directly (for example, by calling `interactionContent.toString()` in your application code) it is possible to hide sensitive data by using the new `ToStringHelper` class:

```
/// Declare hidden attributes for Protocol message or structure
ToStringHelper.hideAttribute("ContactServer", "InteractionContent", "StructuredText");
```

Working With Non-ESP Protocols

To hide specific keys from message attributes of `KVLists` type in the logs, configure `KeyValuePrinter`.

To hide protocol messages attributes in logs, use the new `ToStringHelper` class. For example:

```
ToStringHelper.hideAttribute("FlexChat", "EventInfo", "Text");
```

Using the Application Template Helper to Read Configuration

from Config Manager

The Application Template will process the CfgApplication configuration, received from Config Server, and pass that information to ToStringHelper.setHiddenAttributes();.

Important

There is no standard format for defining log hidden attributes in Config Manger; only a format for KVLists structure exists).

The example below shows how to specify hidden attributes for log files that can be parsed by the Platform SDK AppTemplate mini-helper. However, you can specify settings in any other suitable format and parse them on your own.

```
[log-hidden-attributes]
<ProtocolName>.<Message Name | Complex Attribute Name> = <Attributes List>
```

For example:

```
[log-hidden-attributes]
ContactServer.InteractionContent = "Text, StructuredText"
ContactServer.SomeMessage = "Attr1, Attr2, Attr-n"
TServer.SomeEvent = "Attr-x"
```

Usage Sample

```
//Read application from ConfServer and set new log hidden attributes configuration:
String sectionName = "log-hidden-attributes"; //name of config section in the app options
CfgApplication app = confService.retrieveObject(CfgApplication.class, new
CfgApplicationQuery("AppName"));
MessagePrinterHelper.setHiddenAttributes(app.getOptions().getList(sectionName));
...

//Handle updates from Config Server:
if (cfgEvent.getCfgObject() instanceof CfgDeltaApplication) {
    CfgDeltaApplication delta = (CfgDeltaApplication) cfgEvent.getCfgObject();

    //update application configuration state
    app.update(delta);
    ...

    //set new configuration, if "log-hidden-attributes" section has been added\changed\removed
    if(MessagePrinterHelper.isConfigurationChanged(delta, sectionName)) {
        MessagePrinterHelper.setHiddenAttributes(app.getOptions().getList(sectionName));
    }
}
...
}
```