Platform SDK Developer's Guide

Configuring Platform SDK Channel Encoding for String Values

12/13/2025

Configuring Platform SDK Channel Encoding for String Values

While sending string attributes/values to a server (or to the other side of any connection), Platform SDK packs strings to their binary representation. The binary representation depends on actual charset encoding, so it is important that this data will be unpacked with correct encoding when received on the other side of the connection.

Genesys protocols do not allow client and server sides to synchronize or exchange details about the encoding being used, so application developers may need to handle this configuration manually. (**Exception:** A Configuration Server 8.1.2+ deployment that is configured as UTF-8 multilingual can automatically synchronize UTF-8 encoding with Platform SDK 8.1.3 or later. For details, see Connecting Using UTF-8 Character Encoding.) The most common situation requiring this type of configuration occurs when an application using Platform SDK connects with a Genesys server that has different localization settings, causing the default encoding to be different on both sides.

In this scenario, you can use one of the following possible solutions to synchronize client side encoding with that of the server side:

1. **(Java only)** Change default jvm encoding with the jvm argument: `java -Dfile.encoding=...`
   This changes the charset encoding for the entire jvm, so will affect the main application and any Platform SDK connections to other servers. It may affect the client application relation with other components on the client host.

2. **(Java only)** Starting with Platform SDK 8.1.3, the new `com.genesyslab.platform.defaultcharset` system property can be used to set default charset encoding for Platform SDK connections without the need to change default encoding for whole jvm.
   Platform SDK checks this property once before opening the first connection, and if a value is specified then it will be used as the default encoding for all Platform SDK connections (instead of the value defined for the jvm).

3. Configure a particular Platform SDK connection to use the server side encoding with following connection configuration option (added in Platform SDK 8.0.1):

```
[Java]

protocol = ...;

PropertyConfiguration conf = new PropertyConfiguration();
conf.setOption(Connection.STR_ATTR_ENCODING_NAME_KEY, "windows-1252");

protocol.configure(conf);
protocol.open();

[C#]

protocol = ...;

PropertyConfiguration conf = new PropertyConfiguration();
conf.SetOption(CommonConnection.StringAttributeEncodingKey, "windows-1252");

protocol.Configure(conf);
protocol.Open();
```