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.

Integrated Capture Points Guide

Java Configuration

5/9/2025

Java Configuration

For JMS Capture Points and Groovy transformation scripts, as well as for File Capture Points if Groovy transformation scripts are used (for example, for iWD compatibility mode), Java 8 is the minimum required version. Java 11 is recommended. Here is a general description of the configuration requirements for Java:

- Configure the `jvm-path` option in Interaction Server. In the `java-config` section, the `jvm-path` option must specify the path to the `jvm.dll` file (for Windows) or `libjvm.so` file (for UNIX platforms). Interaction Server requires this to start JVM by means of JNI. This option is required for JMS Capture Points and Groovy transformation scripts.
- Configure the `jvm-options` section in Interaction Server. This section lists JVM option pairs, for example `["-Xmx256m",""]` or `["-Djava.class.path",".;C:\myjars\my-jar.jar;C:\myotherjars\my-other-jar.jar"]`. If JMS Capture Points or Groovy transformations are present, the option `-Djava.class.path` must contain a path to the Genesys-provided JAR files, as well as the Message Queue provider-specific JAR files, which are required in order for JMS and Groovy scripts to run.

These options are explained in more detail below. For OpenMQ, the provider-specific jar files are: `jms.jar`, `imq.jar`, and `fscontext.jar`. For TIBCO, the provider-specific jar files are `jms.jar` and `tibjms.jar`.

For more information about these and other Capture Point-related Interaction Server options, refer to the eServices Reference Manual.

Configuring Interaction Server to Load the Java Virtual Machine (JVM)

To enable JMS capture point functionality or Groovy transformation functionality, Interaction Server must be configured to load the Java Virtual Machine. Java 8 is the minimum required version. Java 11 is recommended.

Take care to specify the correct virtual machine with regard to the architecture; that is, for 64-bit Interaction Server, 64-bit JVM must be used and for 32-bit Interaction Server, 32-bit JVM must be used.

Interaction Server java-config Section

The section should contain only one option: `jvm-path`. This option specifies the full path to the `jvm.dll` (on the Windows platform) or to `libjvm.so` (on UNIX platforms). If this option is not present, Interaction Server does not attempt to load JVM. The following is an example of this option for the Windows platform:

```
jvm-path=C:\Program Files\ojdkbuild\java-11-openjdk-11.0.5-1\bin\server\jvm.dll
```

The following is an example of this option for Linux:

```
jvm-path=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-7.b13.el7.x86_64/jre/lib/amd64/
server/libjvm.so
```

Interaction Server jvm-options Section

This section specifies options that are used to run the JVM. Interaction Server composes the startup string for the JVM containing all of the options specified in this section.

-Xss1m

This option, with empty value, is required for all platforms. It specifies that the Java stack size should be 1 megabyte.

-Xoss1m

This option, with empty value, is required for all platforms. It specifies that the Native code stack size should be 1 megabyte.

It is important to note that Interaction Server creates many working threads to perform its tasks. If the stack size is set to be large, the multiplicity of threads will consume an unnecessarily large amount of memory. Many UNIX systems have unreasonably large default setting for stack size; the recommended stack size for Interaction Server is 1 megabyte.

-Djava.class.path

This option specifies all of the required JAR files. There are a few JAR files provided with the Interaction Server IP that implement Java wrappers to access JMS or Groovy transformation functionality. Additional JAR files are required to use different JMS providers or to use some specific features in customized transformation scripts. The following is the minimal class path that contains all the standard JAR files provided with Interaction Server:

```
-Djava.class.path=lib/ixn-java-aux.jar:lib/groovy-all-2.4.21.jar:lib/XmlTransformer/
xercesImpl.jar:lib/XmlTransformer/xsltc.jar
```

> **Important**
>
> Check your IXN Server installation for exact versions of third-party libraries provided because they may differ from ones mentioned in this documentation.

To make it work, for example, with Open MQ JMS provider on Windows platform, the following class path is required (considering the default installation path for Open MQ):

```
-Djava.class.path=lib/ixn-java-aux.jar;lib/groovy-all-2.4.21.jar;lib/XmlTransformer/
xercesImpl.jar;lib/XmlTransformer/xsltc.jar;
 C:\Program Files\Sun\MessageQueue\mq\lib\fscontext.jar; C:\Program Files\Sun\MessageQueue\mq\
lib\jms.jar;
 C:\Program Files\Sun\MessageQueue\mq\lib\imq.jar
```

For different platforms or different installation paths of the JMS provider, it must be adjusted accordingly.

> **Important**
>
> Staring from version 9.0.009.03, **slf4j-api-1.7.36.jar** must be explicitly specified in the `-Djava.class.path` option within the `java-options` section if Groovy Event Logger, JMS Event Logger/Capture Point, or Kafka Event Logger/Capture Point is used.

-Djava.library.path

This option specifies the path to native libraries that might be required by JVM or specific JMS providers. On the Windows platform it is usually not necessary to specify this option. On UNIX platforms this option must specify the path to the JRE libraries and in certain cases the path to `libjvm.so` itself.

Take extreme care to specify the library path to the same JRE directory from which `libjvm.so` is loaded (the `jvm-path` option). If these do not match, it is often hard to find the reason why the solution is not working.

Operating System Environment

Interaction Server itself does not make use of any environment variables and should not require Java to be in the path or `JAVA_HOME` environment variable to be set. But if these are set, they **must** refer to the same JRE that is configured in the Interaction Server configuration options.

Different operating systems have different default settings for maximum number of threads a process can create. Interaction Server can and will create a few dozen threads. It is important that limits set for the operating system allow creating a few hundred threads. The default value of `1024` should be sufficient for almost all purposes. Consult with your system administrator to check the operating system limits and ensure that these are adequate for Interaction Server.

Another important operating system parameter is the stack size for the thread. As previously mentioned, Interaction Server creates many threads and requires reasonable stack size for the threads. Some systems might have a default in the vicinity of 256 MB or more, which will definitely lead to problems when a process tries to create a few dozen threads. The stack size should be set to 2 MB for Interaction Server. The following command changes the thread stack size for most UNIX operating systems:

```
ulimit -s 2048
```

Again, consult your system administrator to check and ensure the correct operating system limits are in place before running Interaction Server.