

# **GENESYS**<sup>®</sup>

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.

## Genesys Rules System Deployment Guide

Configuring WebSphere Liberty in GRS 9.0

5/4/2025

## Configuring WebSphere Liberty in GRS 9.0

Support for Java 8 in GRS release 9.0 means that previous WebSphere versions supported in Java 7 are not available. The only WebSphere available for Java 8 is WebSphere Liberty. Verification has been performed using version 8.5.5.7.

### Configuring WebSphere Liberty to Run GRAT and GRE

- 1. Set the JAVA\_HOME and PATH environment variables. These are used by the ./server command.
- 2. Create the Liberty server by navigating to [WAS Liberty Home]/bin and executing the command ./server create grs.
- Navigate to [WAS Liberty Home]/usr/servers/grs and create a directory called ExternalLibs. Copy the JDT core .jar file to ExternalLibs from [WAS Liberty Home]/lib. The .jar file should be similar to com.ibm.ws.org.eclipse.jdt.core.[version].jar.
- 4. If GRAT is configured to use a database via a configuration Data Access Point, add the appropriate drive .jar in the ExternalLibs directory created above. Otherwise skip this step. For example, if using the PostgreSQL database, add a driver file like postgresql-[version].jdbc4.jar to the ExternalLibs directory.
- 5. Navigate to the [WAS Liberty Home]/usr/servers/grs directory and edit the server.env file to add JAVA\_HOME. For example, if JAVA\_HOME is /usr/java8\_64, then add this line at the end of the file:

JAVA\_HOME=/usr/java8\_64

- Create file jvm.options in directory [WAS Liberty Home]/usr/servers/grs to set the JVM memory parameters—Genesys recommends using the information here to tune performance for GRE and GRAT.
- 7. Update the **server.xml** file (the server configuration file at **[WAS Liberty Home]/usr/servers/grs**) thus:
  - a. Add <webContainer deferServletLoad="false"/> under the <server> element.
  - b. Add <applicationMonitor updateTrigger="disabled" /> under the <server> element.
  - c. Add host="\*" attribute to the httpEndpoint element if you want any hosts to be able to access this application.
  - d. Add the following application elements under the root element (that is, the **<server>** element):

```
</application>
<!-- GRE Application: IMPORTANT! Replace "[JDT core jar file name]" with the file
name -->
<application context-root="genesys-rules-engine" type="war" id="genesys-rules-engine"
location="genesys-rules-engine.war" name="genesys-rules-engine">
classloader>
classloader>
crivateLibrary>
cfile name="ExternalLibs/[JDT core jar file name].jar"
id="jdt"></file>
</privateLibrary>
</classloader>
</application>
```

#### Important

Make sure <featureManager> has only <feature>jsp-2.2</feature> and nothing else. Adding any other feature might interfere with the already existing libraries in the application.

#### 5. For GRAT:

- a. Create directory genesys-rules-authoring.war under [WAS Liberty Home]/usr/servers/grs/ apps directory.
- b. Extract the contents of the genesys-rules-authoring.war file into genesys-rulesauthoring.war directory. Navigate to genesys-rules-authoring.war directory and execute command jar -xvf [Path to GRAT .war].
- 3. For GRE:
  - a. Create directory genesys-rules-engine.war under [WAS Liberty Home]/usr/servers/grs/apps directory.
  - b. Extract the contents of the genesys-rules-engine.war file into genesys-rules-engine.war directory. Navigate to genesys-rules-engine.war directory and execute command jar -xvf [Path to GRE .war].
- 3. Start the server by navigating to [WAS Liberty Home]/bin and executing EITHER command ./server run grs OR command ./server start grs.