Genesys Rules System Deployment Guide

Genesys Rules System 9.0.0

2/2/2023

Table of Contents

Genesys Rules System Deployment

Welcome to the Genesys Rules System 9.0.x deployment pages. This document describes how to install and configure Genesys Rules System 9.0.x.

Genesys Rules System provides the ability to develop, author, and evaluate business rules. A business rule is a piece of logic defined by a business analyst. These rules are evaluated in a Rules Engine based on requests received from client applications.

Overview

Overview

New Features by Release

Migration to 9.0.x

Preparing for Installation

Installing Genesys Rules System - Task Summary

Configuring the Rules Repository Database for Configuration Manager

Configuring the Rules Repository for Genesys Administrator

Values for DAP Connection Parameters

Installing Genesys Rules Engine

Deploying GRE in Genesys Administrator

Creating the GRE Application object in Configuration Manager

Installing the GRE Component

Creating GRE Application Clusters

Importing the GRE Smart Cluster Template

Creating an Application Cluster in Configuration Manager

Creating an Application Cluster in Genesys Administrator

Installing Genesys Rules Authoring Tool

Deploying GRAT in Genesys Administrator

Installing the GRAT Component

Creating the GRAT Application Object in Configuration Manager

Creating GRAT Application Clusters

Creating the GRAT Application Cluster Object in GA

Further deployment

Deploying WAR Files

Performance Tuning with JVM

Configuring WebSphere 8.5

Post Installation

Testing the Installation

Installing GRS on Unix Platforms

High Availability Support

Configuring GRS for Secure Sockets Layer (SSL)

Troubleshooting

Configuration Considerations

The log4j2.xml File

Localization

Localization

About Business Structure

Defining the Business Structure

Role-Based Access Control

Role-Based Access Control
Role Permissions
User Logins
Business Hierarchy
Role Task Permissions
Template Script Objects
Configuring a User

Overview

Genesys Rules System (GRS) provides the ability to develop, author, and evaluate business rules. A business rule is a piece of logic defined by a business analyst. These rules are evaluated in a Rules Engine based on requests received from client applications.

GRS consists of two components in release 9.0:

- **Genesys Rules Authoring Tool (GRAT)** is a browser-based application that:
  - Allows advanced users (business rules developers) to develop *templates* that define the discrete rule conditions and actions that will comprise the rules. Each rule condition and action includes the plain-language label that the business rules author will see, as well as the rule language mapping that defines how the underlying data will be retrieved or updated. For each rule condition and action, the template developer decides what kind of data the rules author will be providing. Some examples include whether the input should be an integer value, a non-integer numeric value, a string, a selection from a pre-defined list, or a selection from a dynamic list.
  - Allows business analysts to create, edit and deploy business *rule packages* based on the templates. Rule packages provide:
    - The ability to partition rules and facts so that they are small, well-defined, and apply only to a particular application or use. This makes them easier to debug and understand.
    - The ability to isolate rule packages from one another when executing rules. This also improves performance because the Rules Engine has fewer candidates to examine during the evaluation.
    - The ability to update individual rule packages without affecting other deployed packages.
    - The ability to import and export an entire rule package containing the rule definitions, business calendars, and also the templates that the rule package is dependent on.
      A rule package contains one or more rules plus the fact model that is needed to support the rules.
- **Genesys Rules Engine (GRE)** evaluates the rule packages (groups of rules). Rule packages are deployed to the Rules Engine by the Rules Authoring Tool. When a rule package has been deployed, Genesys applications will be able to request the Rules Engine to evaluate the logic that is defined in this rule package.

New Features by Release

9.0.000.17

- Support for Oracle 18g RAC database.
- Full support for creation and editing of templates that support test scenarios (not implemented in the initial 9.0.000.13 release.)

9.0.000.13

- The Genesys Rules Development Tool (GRDT—last release 8.1.4) has been merged with the Genesys Rules Authoring Tool (GRAT) to provide one platform for development of templates through to authoring of business rules and their publication to the Genesys Rules Engine or other specified engine.
- The user interface of the new GRAT has been upgraded to the current Genesys branding and style guidelines.
- A number of enhancements have been made to Genesys Rules Engine to better handle package expiration and serialization. These new features will:
  - Speed up GRE startup time
  - Reduce initial memory usage of GRE on startup
  - Reduce GRE memory usage over time
  - Improve performance when unloading expired packages
  - Improve performance when loading rule packages into memory

    A new configuration option—enable-package-serialization—allows you to enable or disable this functionality globally for rules packages. The feature can be disabled at the rule package level if required by checking the **Disable Package Serialization** checkbox on the **General** tab of a rule package.
- Support for OpenJDK 1.8.
- Undo/re-do buttons have been added to the user interface to assist in editing rules, calendars and test scenarios.
- GRAT has been upgraded to support latest browser releases for improved security and support.

Migration to 9.0

From 8.5.3 to 9.0

1. Undeploy the 8.5.3 .**war** file from your application server.

> **Important**
>
> Do not just copy the 9.0.war file over the 8.5.3 **.war** file in your application server directory. Genesys recommends undeploying the previous file first and letting the application server clean up its files, then deploying the new **.war** file.

2. Ensure you are running the latest supported Tomcat 8.5 application server (Note: Tomcat 9 is not yet supported).
3. Ensure your application server uses the latest Oracle Java 8 or Open JDK 8.
4. Adjust or add any of the new configuration options.
5. Deploy the **9.0.war** file to your application server.
6. Log into 9.0 Genesys Rules Authoring Server.

Summary of installation steps

The following table outlines the task flow for installation of GRS 9.0 The procedures in this table provide instructions about installing GRS components on Microsoft Windows. For information about how to install on UNIX-based operating systems, refer to Installing Genesys Rules System on UNIX Platforms.

Objective	Related Procedures and Actions
1. Prepare for installation and review prerequisites.	• Ensure that your environment meets the prerequisites that are outlined in the Prerequisites section of the GRS page of the Supported Environment Reference Guide. • Ensure that the required CD is available.
2. Create the database for the Rules Repository.	• Configuring the Rules Repository database using Configuration Manager • Configuring the Rules Repository database using Genesys Administrator • Configure DAP Connection Parameters/JDBC Drivers

3. Install the Genesys Rules Engine	• Genesys Administrator: Deploying the Genesys Rules Engine in Genesys Administrator
	  • Creating an Application Cluster in Genesys Administrator
	• Configuration Manager: Creating the Genesys Rules Engine Application object in Configuration Manager
	  • Creating an Application Cluster in Configuration Manager
	• Installing the Genesys Rules Engine

4. Install the Genesys Rules Authoring Tool	• Genesys Administrator:
	  • Deploying the Genesys Rules Authoring Tool in Genesys Administrator
	  • Creating a GRAT Application Cluster
	• Configuration Manager:

Objective	Related Procedures and Actions
	• Creating the Genesys Rules Authoring Tool Application objects in Configuration Manager • Installing the Genesys Rules Authoring Tool
5. Deploy the .war files.	Deploy the `genesys-rules-authoring.war` and `genesys-rules-engine.war` files to your application server.
6. Define your business structure	See Defining the Business Structure.
7. Test the installation	Testing the Installation
8. Review the Troubleshooting section for configuration tips and considerations	See Troubleshooting.
9. Redeploy any standard rule packages that have been previously deployed only to 8.1.2 (or earlier) Genesys Rules Engines. This step is not necessary for standard rule packages that have been deployed to 8.1.3 or later Genesys Rules Engines.	In release 8.1.3, the rules engine was updated from Drools 5.1 to 5.5. The rules engine (up to and including release 8.1.2) writes serialized objects to file. These serialized objects are no longer loadable due to the Drools upgrade. To avoid future upgrade issues, rules engines later than 8.1.3 will maintain the rules package in its DRL form.

Configuring the Rules Repository Database using Configuration Manager

This procedure creates and configures the database that will be used as the Rules Repository by using Configuration Manager.

Most database distributions include the JDBC connector that is needed; if this is not the case, you must download it from the vendor's site. Genesys does not provide the JDBC connector.

> **Important**
> Genesys Rules System 9.0 can use only Java 8. Java 7 is no longer supported.

Once the configuration below is complete, the same database configuration will be used whenever GRAT is installed or updated. There will be no need to specify it again. Also, if any of the database information changes (for example, DB Server location, DB name, DB user or DB password), users can simply update the DAP and restart GRAT.

Prerequisites

Please consult the GRS page of the *Supported Operating Environments Reference Guide*.

Procedure

1. Create a new database using the normal DBMS procedures for the database type you are working with. There is no need to create tables within this database—it will be populated by GRAT automatically on startup.
2. In Configuration Manager, right-click the **Environment > Applications** folder and select **New > Application**. This opens the **Browse** dialog box that lists the available application templates.
3. In the **Browse** dialog box, select the DAP template file, and click 0K. This opens the **Properties** dialog box for the new DAP `Application` object.
4. On the **General** tab:
    a. Enter a name for the DAP. A DAP can have the same name as the database itself. However, it is recommended that you make their names unique if you are using multiple access points for the same database.
    b. Do not enter anything in the **DB Server** field.
    c. Select the **JDBC Connection** check box. This will disable the **DBMS Name** field on the **DB Info**

tab.

d. Ensure that the **State Enabled** check box is checked.

5. On the **DB Info** tab:

a. Enter the DBMS type, database name, username, and password. The database username must have full permissions to this database in order to create the tables. GRAT uses these database credentials when it starts for the first time in order to create the necessary database schema.

b. Set `Case Conversion` to any, and leave the **DBMS Name** field clear.

3. On the **JDBC Info** tab, set the following values:

a. **Role** field—`Main`

b. **Debug** field—`Unknown`

c. **Query Timeout** field—`0` (zero)

4. On the **Server Info** tab, enter the host name and port number of the DBMS that handles the database. For Oracle RAC configuration, use the SCAN name host and the SCAN port.

- Add this newly created DAP to the **Connections** tab of the GRAT Application object.

Configuring the Rules Repository Database using Genesys Administrator

This procedure creates and configures the GRAT Rules Repository database using Genesys Administrator.

Most database distributions include the JDBC connector that is needed. If this is not the case, you must download it from the vendor's site. Genesys does not provide the JDBC connector.

Genesys Rules System 9.0 can use only Java 8.

Once the configuration below is complete, the same database configuration will be used whenever GRAT is installed or updated. There will be no need to specify it again. Also, if any of the database information changes (for example, DB Server location, DB name, DB user or DB password), users can simply update the DAP and restart GRAT.

Prerequisites

Either Oracle 11g or 12c or Microsoft SQL or DB2 or PostGRE SQL 9.4

> **Important**
> Oracle 10g is not supported.

Procedure

1. Create a new database using the normal DBMS procedures for the type you are working with. This is the database that will be populated by GRAT. Create a DB user/password that will have full access to this new database.
2. In Genesys Administrator, navigate to the **Environment > Applications > GRS** folder.
3. Create a new application to be the new Database Access Point.
4. Select the Application Template type **Database_Access_Point_900** (or later).
5. Ensure that the **State: Enabled** check box is checked.
6. In the **Server Info** panel, enter values for the **Host** and the **Listening Ports** of the DBMS server.
7. In the **DB Info** panel, enter JDBC as the **Connection Type**. This will disable the **DBMS Name** field.
8. Set the **Role** field to value `Main`.

9. Set the **Debug** field to value `0` (zero).
10. Set the **Query Timeout** field to value `0` (zero).
11. Select the value in the **DBMS Type** field (MSSQL, DB2, Oracle or PostGRE SQL).
12. Enter the name of the database created in Step 1. For Oracle, this is the "service name".
13. Enter the database username and password created in Step 1.
14. Ensure that the **Case Conversion** field has the value any.
15. Save your changes.
16. Add this newly created DAP as a Connection on the GRAT Application object. When GRAT initializes, it will use the information in this DAP to connect to the repository database.

Configure DAP Connection Parameters/ JDBC Drivers

Enter these parameters in the DAP as described in Creating the Rules Repository Database).

Default Values

> **Important**
>
> You should consult your database vendor's documentation for specific information. Choose the most current version of the JDBC driver that supports your operating system and Java version. The JDBC driver(s) must be copied to the **lib** directory of your application server. Typical examples are shown in the table.

Database Type	Connector Class	Default Database URL	Example JDBC Driver to be Copied (see note above)
MSSQL	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:jtds:sqlserver://{host}:{port};databaseName={database_name}	sqljdbc42.jar
Oracle	oracle.jdbc.driver.OracleDriver	jdbc:oracle:thin:@//{host}:{port}/{SID}	ojdbc6.jar
DB2	com.ibm.db2.jcc.DB2Driver	jdbc:db2://{host}:{port}/{database_name}	db2jcc.jar db2jcc_license_cu.jar
PostGRE SQL	org.postgresql.Driver	"jdbc:postgresql://{host}:{port}/databaseName"	postgresql-9.3-1102.jdbc41.jar

Overriding Default Values on the Options Tab

More advanced users can use the DAP's `Options` tab to override the default values mentioned above; for example, if a database vendor makes changes to the JDBC driver class, or if additional options need to be specified on the DB URL. Note that this is very rarely needed—entering the

parameter during DAP configuration is almost always sufficient.

If the **GRS** section is present, the value of any options specified here overrides the defaults generated by GRAT.

Procedure

1. On the **Options** tab, create a section called GRS.
2. Use the **URL** field to override the URL value generated by GRAT.
3. Use the **Driver** field to override the default driver value generated by GRAT.

Installing GRE

GRE can be configured by using either Genesys Administrator or Configuration Manager.

If you use Genesys Administrator, you can deploy the installation package from within Genesys Administrator.

If you use Configuration Manager, you will have to:

1. Create the application.
2. Run the installation package manually.

Deploying GRE in Genesys Administrator

Prerequisites

To install GRE on Configuration Servers 8.1.0 or later, Genesys Administrator 8.1.0 or later is required.

Procedure Summary

1. Import the installation package into Genesys Administrator.
2. Install the GRE IP.
3. Configure the Rules Engine application.

Procedure

Import the installation package into Genesys Administrator.

1. On the **Deployment** tab of Genesys Administrator, select **Import**.
2. Select **Installation CD-ROM**.
3. Click **Next**.
4. Browse to the **MediaInfo.xml** file on the CD or the CD image location on the network (the path must be in UNC format).
5. Click **Next**.
6. To import the installation package, select GRE for your operating system as well as the appropriate type in the list:
    - For Management Framework 8.1, the type is `Business Rules Execution Server`.
    - For Management Framework 8.0 and earlier, the type is `Genesys Generic Server`.
7. Select **Next** to start the import.
8. Click **Finish** when the import is complete.

Install the GRE IP

1. Select the **Deployment** tab in Genesys Administrator. The list of installation packages will now display GRE.

2. Right-click and select **Install Package** for the IP for your operating system and type.
3. Click **Next** to start the installation wizard. The following parameters must be defined/selected:
    a. **Application Name** for the GRE application
    b. **Target Host**—The host to which the **.war** file will be copied during the installation procedure.
    c. **Working Directory**—The directory in which the **.war** file will be created.
    d. **Client Side IP Address** (optional).
    e. **Client Side Port** (optional).
    f. Configuration Server hostname.
    g. Configuration Server port.

> **Important**
> For a secure connection, the Configuration Server port should be of type `Auto Detect (Upgrade)`.

    h. Connection delay time in seconds
    i. **Reconnect Attempts**.

Configure the Rules Engine application

1. In the **[Server Info]** section, verify the default listening port, as well as the connector port on which the Rules Engine Servlet receives requests:
    - The **ID** value is the name of the Rules Engine web application. The default name of this application is `genesys-rules-engine`.
    - The **Listening port** is the connector port of the servlet container. For example, on Tomcat the default listening port is 8080.
    - The **Connection Protocol** must be `http`.
2. On the **Tenants** tab, add the Tenants that will be available to the Rules Engine.
3. On the **Connections** tab, add a connection to Message Server if you want to use network logging.
4. On the **Options** tab, configure options. In addition to the standard logging options that you can configure, you can configure an option named **fileEncoding** in the **[log]** section. **fileEncoding** specifies the encoding that is to be used during creation of the log file, for example, `UTF-8`. This value is optional. If you do not specify this option, the server's locale information will determine the log file encoding. This option is available for both GRE and GRAT. Also, the **log4j2.xml** file that is included in both components supports a similar option, **log4j.appender.runtime.Encoding**. The **log4j2.xml** file is used for initial log configuration prior to the reading of the log configuration from the Configuration Server database.
5. There are several optional configuration options to work with in the Configuration Options Reference Guide.
6. Save your changes.

Next Steps

Deploy the **genesys-rules-engine.war** file to your application server. See Deploying the .WAR files.

Creating the GRE Application Object in Configuration Manager

Procedure Summary

1. Import the GRE application template into Configuration Manager.
2. Configure the Rules Engine application.

Procedure

To create the application object for GRE in Configuration Manager, do the following:

Import the GRE application template into Configuration Manager

1. In Configuration Manager, navigate to the **Application Templates** folder.
2. Right-click the **Application Templates** folder, and select **Import Application Template**.
3. Browse to the **templates** folder of the installation CD, and select the appropriate template for your version of Management Framework.
    - For Management Framework 8.1.1, select `Genesys_Rules_Engine.apd.`.
    - For Management Framework 8.1 and earlier, select `Genesys_Rules_Engine_Generic_Server.apd.`.
4. Click **OK** to save the template.

Configure the Rules Engine application

1. Right-click the **Applications** folder and select **New > Application**.
2. Select the template that you imported in the previous procedure.
3. On the **General** tab, enter a name for the application, such as `Rules_Engine`.
4. On the **Tenants** tab, add the Tenants that will be available to the Rules Engine.
5. On the **Server Info** tab, select the Host on which the application will be installed.
6. Add a default listening port.
7. Add an additional port. This port is the connector port on which the Rules Engine Servlet receives requests:
    - The **ID value** is the name of the Rules Engine web application. The default name of this

application is `genesys-rules-engine`.

- The **Listening Port** is the connector port of the Servlet Container. For example, on Tomcat the default listening port is `8080`.
- The **Connection Protocol** must be `http`.

8. On the **Start Info** tab, enter `x` for each field. These fields are not used, but you must enter some text there in order to save the configuration.
9. On the **Options** tab, configure options in the **[log]** section and the **[settings]** sections.
10. Save your changes.

Installing the GRE Component

Purpose

- To run the installation package for the GRE, after the application has been created in Configuration Manager.

Prerequisites

- Creating the GRE Application Object in Configuration Manager

Start

1. From the host on which the GRE is to be installed, locate and double-click `Setup.exe` in the **rulesengine** folder of the Genesys Rules System CD.
2. Click **Next** on the **Welcome** screen of the installation wizard.
3. Enter the connection parameters to connect to Configuration Server (**Host**, **Port**, **User name**, and **Password**).
4. On the **Client Side Port Configuration** screen, if you do not want to configure client-side port parameters, leave the checkbox empty and click **Next**. If you do want to configure these settings, select the checkbox to display to additional options: **Port** and **IP Address**. Enter values for these options and click **Next**.
5. Select the Rules Engine application that you created in Creating the GRE Application Object in Configuration Manager. Click **Next**.
6. Specify the destination directory for the installation, or accept the default location, and click **Next**.
7. Enter the host and port of the optional backup Configuration Server and click **Next**.
8. Enter the number of times that the Rules Engine application should attempt to reconnect to Configuration Server (**Attempts**) before switching to the backup Configuration Server, and the amount of time (**Delay**) between attempts. Click **Next**.

> **Important**
>
> After the specified number of attempts to connect to the primary Configuration Server all fail, then connection to the backup Configuration Server is attempted. If these attempts to the backup Configuration Server fail, then once again connection to the Primary Configuration Server is attempted. If no backup Configuration Server is configured, there is no limit on the number of connection attempts.

9. Enter Application Name and click **Install**.
10. Click **Finish**.

End

Next Steps

- Deploy the **genesys-rules-engine.war** file to your application server. See Deploying the .WAR files.

Importing the GRE "Smart Cluster" Template

From Genesys Administrator

1. Navigate to **Application Templates**.
2. Click **Upload Templates** (upper right corner).
3. Choose the **Genesys_Rules_Engine_Application_Cluster_900.apd** file.
4. Click **Save** and **Close**.
5. Go to **Applications**.
6. Create a **New** application.
7. For the template, choose the application template that you just created in steps 1-4.
8. Fill in the mandatory fields, including the host (which is not used, but GA requires you to complete this field).
9. In the **Connections** section, add each GRE in the cluster.
10. On the **Options** tab, configure the three auto-synchronization options:
    - auto-synch-rules
    - auto-synch-rules-interval
    - auto-synch-rules-at-startup

Creating an Application Cluster in Configuration Manager

> **Important**
>
> If you do not require the auto-synchronization feature you can continue to use the cluster configuration for 8.5.1—click **here** for details. Only Genesys Rules Engine (GRE) supports the auto-synchronization feature.

You can use a Configuration Server application template of type `Genesys_Rules_Engine_Application_Cluster` to define a group of Genesys Rules Engine (GRE) or Genesys Web Engagement engines. Engines in the group must be all of the same type—either all GRE engines or all Genesys Web Engagement engines.

When a user deploys a package in GRAT, the deployment target list may contain cluster application names. Partial deployments are possible when the allow-partial-cluster-deployment option in GRAT is set to value `true`. If this option is set to `false`, deployed packages are placed in service only after the deployment to all engines in the cluster is successful.

If deployment to any of the engines fails, details of the failure(s) are shown to the GRAT user and logged in the GRAT log. Partial deployments, where configured, are also displayed to the GRAT user and in the GRAT log.

Procedure

1. Import an application template of type `Genesys_Rules_Engine_Application_Cluster`, if one does not already exist in your environment.
2. Create a Configuration Server application of type `Genesys_Rules_Engine_Application_Cluster`.
3. Enter a host and other mandatory information.

> **Important**
>
> Host information is required by the configuration user interface, but it is not used.

4. Add as connections to this cluster application the engine applications you wish to treat as a cluster. For each connection be sure to select a Port ID (it does not matter which port) for the Rules Engine Web Application (either GRE or Genesys Web Engagement).
5. Add the cluster application as a connection to the GRAT application.
6. Save the changes.

This cluster application will now appear in the **Location** drop-down list in the **Deploy** window of GRAT and rules authors can select it as a deployment target.

Creating an Application Cluster in Genesys Administrator

> **Important**
>
> If you do not require the auto-synchronization feature, you can continue to use the cluster configuration for 8.5.1—click **here** for details. Only Genesys Rules Engine (GRE) supports the auto-synchronization feature.

Purpose

To create an application cluster in Genesys Administrator to which rules packages can be deployed.

Procedure

1. Import an application template of type `Genesys_Rules_Engine_Application_Cluster`, if one does not already exist in your environment.
2. Create a Genesys Administrator application of type `Genesys_Rules_Engine_Application_Cluster`.
3. Go to **Provisioning > Environment > Applications**. If required, navigate to the folder in which you want to store the new Application object.
4. Open the **Tasks** panel, if necessary, and click **Create Application** in the **Create** section.
5. Follow the steps in the **Create New Application** wizard. Make sure to select Genesys Rules Authoring Tool (GRAT) server as the Host in **Server Info** section.
6. Add as connections to this cluster application the engine applications you wish to treat as a cluster. For each connection be sure to select the default port ID for the Rules Engine Web Application (either GRE or Genesys Web Engagement).
7. Add the cluster application as a connection to the GRAT application.
8. Save the changes.

Installing GRAT

Genesys Administrator

Genesys recommends that you configure the GRAT by using Genesys Administrator. If you use Genesys Administrator, you can deploy the installation package from within Genesys Administrator.

Configuration Manager

You can configure the GRAT by using Configuration Manager if you are using an older version of Configuration Server, prior to 8.0.2, where Roles are not supported. If you use Configuration Manager, you will have to:

1. Create the applications.
2. Run the setup program manually.

Non-English Environments

When operating the GRAT in a non-English environment, you will need to configure the **URIEncoding** option to properly operate and integrate with the Genesys Framework environment. By default, Tomcat uses ISO-8859-1 character encoding when decoding URLs received from a browser. If you wish to use characters not included in this character set, you will need to do the following:

1. Set the **URIEncoding** option to `UTF-8` in the **server.xml** file on the Connector that is used for the Genesys Rules Authoring Tool. For example:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443"
URIEncoding="UTF-8" useBodyEncodingForURI="true"/>
```

2. Start Tomcat with the environment variable:

```
-Dcom.genesyslab.platform.defaultcharset=UTF-8
```

Deploying GRAT in Genesys Administrator

Purpose

To configure the GRAT applications and deploy the GRAT installation package using Genesys Administrator.

Prerequisites

To install GRAT on Configuration Servers 8.1.1 or later, Genesys Administrator 8.1.1 or later is required.

Procedure Summary

1. Import the GRAT IP into Genesys Administrator.
2. Install the GRAT IP.
3. Configure the GRAT application.

Import the GRAT IP into Genesys Administrator

1. Import the installation package into Genesys Administrator:
2. On the **Deployment** tab of GA select the **Import** button.
    a. Select the **Installation CD-ROM** radio button.
    b. Click **Next**.
    c. Browse to the **MediaInfo.xml** file on the CD or the CD image location on the network (the path must be in UNC format).
    d. Click **Next**.
5. Select GRAT for your operating system as well as the appropriate type in the list in order to import the installation package.
    - For Management Framework 8.1.1, the type is `Business Rules Application Server`.
    - For Management Framework 8.1 and earlier, the type is `Genesys Generic Server`.
6. Select **Next** to start the import.
7. Click **Finish** when the import is complete.

Install the GRAT IP

1. Select the **Deployment** tab in Genesys Administrator. The list of installation packages will now show the Genesys Rules Authoring Tool.
2. Right-click and select **Install Package** for the IP for your operating system and type.
3. Click **Next** to start the installation wizard. The following parameters must be defined/selected:
   a. **Application Name** for the Genesys Authoring Tool server application.
   b. **Target Host**—The host to which the **.war** file will be copied during the installation procedure.
   c. **Working Directory**—The directory in which the **.war** file will be created.
   d. **Client Side IP Address** (optional).
   e. **Client Side Port** (optional).
   f. **Backup Configuration Server** hostname.
   g. **Backup Configuration Server** port.
   h. Connection delay time in seconds.
   i. **Reconnect Attempts**.

   > **Important**
   > After the specified number of attempts to connect to the primary Configuration Server all fail, connection to the backup Configuration Server is attempted. If these attempts to the backup Configuration Server fail, then once again connection to the Primary Configuration Server is attempted. If no backup Configuration Server is configured, there is no limit on the number of connection attempts.

   j. Client application name—The name of the GRAT client application.

   > **Important**
   > Items *a* through *i* will be written to the **bootstrapconfing.xml** file in the **.war** file. Any subsequent updates to the parameters will have to be made in that file.

11. On the next screen, enter the **Connection ID** and **Connection Port** for the Genesys Rules Authoring Server. Specify the connections for the Rules Authoring Server on the next screen (select the GRE application). You can also add this connection later under the Configuration for the application. Verify the previously-defined installation parameters on the **Deployment Summary** screen.

Configure the GRAT application

Configuration options are described here.

Creating the GRAT Application Object in Configuration Manager

Purpose

To create the Application objects in Configuration Manager that will link the GRAT with Configuration Server. The GRAT requires two applications in Configuration Server: a server application and a client application.

Procedure Summary

1. Import the GRAT application template for the server.
2. Import the GRAT application template for the client.
3. Configure the server application.
4. Configure the client application.

Import the GRAT application template for the server

To import the application template that is to be used for the server application:

1. In Configuration Manager, navigate to the **Application Templates** folder.
2. Right-click the **Application Templates** folder, and select **Import Application Template**.
3. Browse to the **templates** folder of the installation CD, and select the appropriate template for your version of Management Framework.
    - For Management Framework 8.1.1, select **Genesys_Rules_Authoring_Server_900.apd**.
    - For Management Framework 8.1 and earlier, select **Genesys_Rules_Authoring_Generic_Server_900.apd**.
4. Click **OK** to save the template.

Import the GRAT application template for the client

To import the template that is to be used for the client application:

1. Right-click the **Application Templates** folder.
2. Select **Import Application Template**.
3. Browse to the **templates** folder of the installation CD.

4. Select **Genesys_Rules_Authoring_Generic_Client_900.apd**.
5. Click **OK** to save the template.

Configure the server application

Configuration options are described here.

Configure the client application

To configure the client application:

1. Right-click the **Applications** folder.
2. Select **New > Application**.
3. Select the **Genesys_Rules_Authoring_Generic_Client** template.
4. On the **General** tab, enter a name for the application, such as `Rules_Authoring_Client`.
5. Click **Save**.

Next Steps

Installing the GRAT Component

Installing the GRAT Component

Purpose

To run the installation package for the GRAT after the applications are configured in Configuration Manager.

Prerequisites

- Configuring the Rules Repository Database
- Creating the GRAT Application Objects in Configuration Manager

Procedure

1. From the host on which the GRAT is to be installed, locate and double-click **Setup.exe** in the **rulesauthoring** folder of the Genesys Rules System CD.
2. Click **Next** on the **Welcome** screen of the installation wizard.
3. Enter the connection parameters to connect to Configuration Server (**Host**, **Port**, **User name**, and **Password**).
4. On the **Client Side Port Configuration** screen, if you do not want to configure client-side port parameters, leave the checkbox empty and click **Next.** If you do want to configure these settings, select the checkbox to display to additional options: **Port** and **IP Address**. Enter values for these options and click **Next**.
5. Select the GRAT application that you created in Creating the GRAT Application Objects in Configuration Manager. Click **Next**.
6. Specify the destination directory for the installation, or accept the default location, and click **Next**.
7. Enter the host and port of the optional backup Configuration Server and click **Next**.
8. Enter the number of times that the GRAT Server application should attempt to reconnect to Configuration Server (**Attempts**) and the amount of time (**Delay**) between attempts. Click **Next**.
9. On the screen that is shown in Creating the GRAT Application Objects in Configuration Manager, specify the name of the rules authoring *client* application and click **Next**.
10. Select **Application Server Type**. Click **Next**.
11. Click **Install**.
12. Click **Finish**.

Next Steps Before using GRAT, you will need to set up users and roles. See **Role Task Permissions** and Configuring a User for more information.

Creating the GRAT Cluster Application Object

Using Genesys Administrator

1. Navigate to **Application Templates**.
2. Click **Upload Templates** (upper right corner).
3. Choose the `.apd` file `GRAT_Rules_Authoring_Tool_Application_Cluster_900.apd`.
4. Click **Save** and **Close**.
5. Go to **Applications**.
6. Create a **New** GRAT cluster Application based on the new template. Adjust the configuration options as needed.
7. Fill in the mandatory fields, including the host (which is not used, but GA requires you to complete this field).
8. For each GRAT node to be added to the cluster, add its configuration Application as a connection in the cluster Application.
9. Add the DAP (Database Access Point) configuration Application to be used by the GRAT cluster as a connection in the cluster Application.

> **Important**
>
> An existing standalone GRAT database can be used.

10. If you are re-using an existing standalone GRAT instance in the cluster, re-deploy the GRAT application to ensure that its local cache is re-initialized.

> **Important**
>
> Any change in the cluster configuration will only take effect upon re-start of the GRAT servers.

> **Important**
>
> For high availability, GRAT instances must have a high-speed connection to the

database. Slow connection may result in the type of issues commonly seen when the local repository cache is corrupted.

Deploying .WAR Files

> **Important**
> Make sure you have configured DAP connection parameters and JDBC drivers before deploying .WAR files.

The **genesys-rules-authoring.war** and **genesys-rules-engine.war** files must be copied or deployed to your web container. When the **.war** files have been deployed, you will be able to launch GRE and GRAT.

The **.war** files can be found in the destination folder that you specified when you installed the IPs.

- If you are using Tomcat, copy the files and paste them into the Tomcat **webapps** folder.
- If you are using WebSphere, deploy the **.war** files by using WebSphere Administrative Console.

> **Important**
> GRS 9.0 requires Java 8 and as a consequence of this requirement, you must deploy WebSphere Liberty—this is the only Websphere application server that supports Java 8.

- Refer to your Tomcat documentation for specific deployment instructions.
- WebSphere 8.5 configuration information can be found **here**.
- WebSphere Liberty configuration information can be found here.

Genesys recommends using **the information here** to tune performance for GRE and GRAT. This may need to be adjusted based on your configuration and depending upon any other applications deployed to your application server.

Configuring WebSphere 8.5

Procedure

1. Extract:
   - `httpclient-[VERSION].jar`
   - `httpcore-[VERSION].jar`
   - `jackson-core-asl-[VERSION].jar`
   - `jackson-mapper-asl-[VERSION].jar`

   from the **WEB-INF/lib** directory of `genesys-rules-engine.war` or `genesys-rules-engine-authoring.war` and store them in:

   ```
   ${WAS_INSTALL_ROOT}\optionalLibraries
   ```

   > **Important**
   > The `jackson-core-asl-[VERSION].jar` and `jackson-mapper-asl-[VERSION].jar` libraries are required for the auto-synchronization feature.

2. Configure these four JAR files as Isolated Shared Libraries.
   a. From the WS Admin console select `Environment->SharedLibraries->New`.
   b. Set the name to `sharedStuff`.
   c. Set the classpath to:

      ```
      ${WAS_INSTALL_ROOT}/optionalLibraries/httpclient-[VERSION].jar
      ```

      and

      ```
      ${WAS_INSTALL_ROOT}/optionalLibraries/httpcore-[VERSION].jar
      ```

      and

      ```
      ${WAS_INSTALL_ROOT}/optionalLibraries/jackson-core-asl-[VERSION].jar
      ```

      and

      ```
      ${WAS_INSTALL_ROOT}/optionalLibraries/jackson-mapper-asl-[VERSION].jar
      ```

   d. Check the **Use an isolated class loader for this shared library** check box. Click **Apply** and **Save**.

5. Navigate to **Enterprise Applications->genesys-rules-engine->Shared library references** and add the sharedStuff shared library reference to the Module.
6. Navigate to **Enterprise Applications->genesys-rules-authoring>Shared library references** and add the sharedStuff shared library reference to the Application and Module.

Configuring Clusters in Websphere

Background

Before release 8.5.2 of GRS, it was not possible to configure multiple cluster nodes running on the same machine and controlled by the same cluster manager because separate entries for the same host could not be created in **bootstrapconfig.xml** to represent different GRE nodes. The pre-8.5.2 format of the **bootstrapconfig.xml** allowed for a single node to be defined per host. The xml format was as follows:

```
<xs:complexType name="node">
      <xs:sequence>
       <xs:element name="cfgserver" type="cfgserver" minOccurs="1" maxOccurs="1"/>
       <xs:element name="lcaserver" type="lcaserver" minOccurs="0" maxOccurs="1"/>
       <xs:element name="application" type="application" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="host" type="xs:string"/>
      <xs:attribute name="ipaddress" type="xs:string"/>
</xs:complexType>
```

What's New?

In GRS 8.5.2, an additional attribute called **servername** has been added to the node definition. This makes it possible to define multiple nodes for the same host. The server name is defined via the WebSphere Application Server (WAS) Deployment Manager when the cluster node is created.

For example, you can replicate the “node” definition for each GRE that is running on the same host. Then, by adding **servername=**, you can make the entry unique. Each entry then points to the corresponding Configuration Server application for that GRE instance. In this way, a single bootstrapconfig.xml file can be used to define all nodes in the Websphere cluster, whether or not there are multiple GRE nodes defined on a given host.

To ensure backward compatibility, if no node is found within the **bootstrapconfig.xml** that matches both the hostname and **serverName** then the node that contains the **hostname** with no server name defined serves as the default.

Editing the bootstrapconfig.xml file

To edit this file, manually extract the `bootstrapconfig.xml` file from the `.war` file, edit and save the `bootstrapconfig.xml` file, then repackage the `bootstrapconfig.xml` file back into the `.war` file.

Sample bootstrapconfig.xml files

> **Important**
>
> Terminology—In the `bootstrapconfig.xml` files, the <node> element corresponds to an individual member of a WebSphere cluster.

For a cluster with one host and two server instances on that host

Below is a sample **`bootsrapconfig.xml`** definition for a GRE cluster running on one host, `GenSrv1000`, with server instances `server01` and `server02` on that host:


```
<bootstrap name="BRS_applications" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="bootstrapconfig.xsd">
  <node host="GenSrv1000" servername="server01"
   <cfgserver host="192.xxx.x.x" port="2020"
       <reconnectperiod></reconnectperiod>
       <reconnectattempts></reconnectattempts>
       <!-- <protocoltimeout>30</protocoltimeout> -->
       <clienttransport ipaddress="" port="" />
       <backup host="" port="" />
       <!-- <addp clienttimeout="30" servertimeout="30" /> -->
    </cfgserver>
    <application id="brs.rules.engine"> <cfgappname>GRE_85200-S01</cfgappname> </application>
 </node>
<node host="GenSrv1000" servername="server02"
   <cfgserver host="192.xxx.x.x port="2020"
       <reconnectperiod></reconnectperiod>
       <reconnectattempts></reconnectattempts>
       <!-- <protocoltimeout>30</protocoltimeout> -->
       <clienttransport ipaddress="" port="" />
       <backup host="" port="" />
       <!-- <addp clienttimeout="30" servertimeout="30" /> -->
    </cfgserver>
    <application id="brs.rules.engine"> <cfgappname>GRE_85200-S02</cfgappname> </application>
 </node>
</bootstrap>
```

For a cluster with two hosts and two server instances on each host

Below is a sample **`bootsrapconfig.xml`** definition for a GRE cluster running on two hosts, `GenSrv1000` and `GenSrv2000`, with server instances `server01` and `server02` on each host:


first host name and first server instance
first host name and second server instance
second host name and first server instance
second host name and second server instance
unique configuration application names

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**.
3. 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
   ```

6. 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):

   ```
   <!-- GRAT Application: IMPORTANT! Remove classLoader element in below Application if
   not using an external database otherwise replace "[database driver file]" with the
   appropriate driver file name -->
   <application context-root="genesys-rules-authoring" type="war" id="genesys-rules-
   authoring"
          location="genesys-rules-authoring.war" name="genesys-rules-authoring">
          <classloader>
               <privateLibrary>
                         <file name="ExternalLibs/[database driver file].jar"
   id="databasedriver"></file>
                    </privateLibrary>
           </classloader>
   ```

```
</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>
            <privateLibrary>
                       <file 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-rules-authoring.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**.

Testing the Installation

Test the installation by logging in a user to the GRAT.

See Configuring a User for the GRAT to verify that the user has the correct permissions.

1. Start your web application container (Tomcat or WebSphere) on the server(s) that are hosting the GRAT and the GRE.
2. Open a web browser and enter the URL for the GRAT—for example **http://<host>:<port>/genesys-rules-authoring/login.jsp** (or **http://<host>:<port>/genesys-rules-authoring/singlesignon.jsp** if this is configured) where <host> is the name of the server on which the web container is running that is hosting the GRAT server, and <port> is the listening port for your web container (such as 8080). The default name is **genesys-rules-authoring**, but you can override this name during deployment.
3. On the login screen, enter the credentials for a user to login to the GRAT. Users who log into the GRAT must have access to one or more tenants in a multi-tenant environment, with, at minimum, Read permission to the tenant(s). In addition, users or access groups must have, at a minimum, Read and Execute permissions to this GRAT client Application object in Configuration Server, in order to log in to the GRAT.

Installing GRS On Unix Platforms

For the supported UNIX versions, please consult the Supported Operating Environments Reference Guide

Procedure

To install the GRE or the GRAT on UNIX systems:

1. Create the Application objects in Configuration Manager or Genesys Administrator. Please refer to the following topics:
   - Creating the Genesys Rules Engine Application Object in Configuration Manager
   - Deploying GRE in Genesys Administrator
2. Locate and run the **install.sh** scripts for each component (found in their respective directories on the CD).

Example of the command terminal from an installation of the GRE on a Linux host

This example includes the script's prompts, as well as the user's input (in bold).

```
bash-3.2$ ./install.sh
-------------------------------------------------
 Welcome to the Genesys 9.0 Installation Script
-------------------------------------------------

Installing Genesys Rules Engine, version 9.0.xxx.xx

Please enter the hostname or press enter for “rh5x64-vm1” => <ENTER> was selected

Unable to find configuration information.
Either you have not used configuration wizards and the
GCTISetup.ini file was not created or the file is corrupted.

Please enter the following information about your Configuration Server:

Configuration Server Hostname =>host1
Network port =>2020
User name =>default
Password => the password was entered

Client Side Port Configuration
Select the option below to use a Client Side Port. If you select
this option, the application can use Client Side Port number for initial connection to Configuration Server.
Do you want to use Client Side Port option (y/n)?y
Client Side Port port =>8888
Client Side IP Address (optional), the following values can be used
135.xxx.xx.xxx
=><ENTER> was selected
Backup Configuration Server Hostname =>host2
Backup Network port =>2020

Please choose which application to install:
1 : GRE90xxxxx_rh5x64-vm1
=>1
```

```
Press ENTER to confirm “0” as
the Number of attempts to reconnect to primary Configuration Server or enter a new one =>6

Press ENTER to confirm “0” as
the Delay in seconds between reconnect attempts or enter a new one =>3

Please enter full path of the destination directory for installation =>/home/GRS/GRE/90xxxxx/linux

The target install directory /home/GRS/GRE/9.0.xxx.xx/linux
has files in it. Please select an action to perform:
1. Back up all files in the directory
2. Overwrite only the files contained in this package
3. Wipe the directory clean
1, 2, or 3 =>2

Extracting tarfile: data.tar.gz to directory: /home/user/GRS/GRE/9.0.xxx.xx/linux
….

Installation of Genesys Rules Engine, version 9.0.xxx.xx has completed successfully.
```

Example of the command terminal from an installation of the GRAT on a Linux host

This example includes the script’s prompts, as well as the user’s input (in bold).

```
bash-3.2$ ./install.sh
------------------------------------------------
 Welcome to the Genesys 9.0 Installation Script
------------------------------------------------

Installing Genesys Rules Authoring Tool, version 9.0.xxx.xx

Please enter the hostname or press enter for “rh5x64-vm1” =><ENTER> was selected

Unable to find configuration information.
Either you have not used configuration wizards and the
GCTISetup.ini file was not created or the file is corrupted.

Please enter the following information about your Configuration Server:
```

```
Configuration Server Hostname =>host1
Network port =>2020
User name =>default
Password => the password was entered

Client Side Port Configuration
Select the option below to use a Client Side Port. If you select this option,
the application can use Client Side Port number for initial connection to
Configuration Server.

Do you want to use Client Side Port option (y/n)?y
Client Side Port port =>9999
Client Side IP Address (optional), the following values can be used
135.xxx.xx.xxx
 =><ENTER> was selected
Backup Configuration Server Hostname =>host2
Backup Network port =>2020

Please choose which application to install:
1 : GRAT9000011_rh5x64-vm1
2 : GRE9000011_rh5x64-vm1
=>1

Press ENTER to confirm “0” as
the Number of attempts to reconnect to primary Configuration Server or enter a new one =>3

Press ENTER to confirm “0” as
the Delay in seconds between reconnect attempts or enter a new one =>6

Client connection application =>GRSRuleClient


Please enter full path of the destination directory for installation =>/home/GRS/GRAT/9.0.xxx.xx/linux

The target install directory /home/GRS/GRAT/9.0.xxx.xx/linux
has files in it. Please select an action to perform:
1. Back up all files in the directory
2. Overwrite only the files contained in this package
3. Wipe the directory clean
1, 2, or 3 =>2

Extracting tarfile: data.tar.gz to directory: /home/user/GRS/GRAT/9.0.xxx.xx/linux
```

….

```
Installation of Genesys Rules Authoring Tool, version 9.0.xxx.xx has completed successfully.
```

High Availability Support

GRE

The Genesys Rules Engine (GRE) can be set up in a cluster in order to provide a highly available configuration. GRE is considered a critical path application because the execution of rules depends upon at least one node in the system being available. Since GRE is stateless, each rule execution request can be dispatched to any node in the cluster, and should a node fail, another node could execute the request.

The load balancer can be set up to dispatch requests to each GRE node at random, or in a round-robin fashion. There is no need to configure "session stickiness" as there are no sessions to maintain between rule execution requests. The load balancer should only route rule evaluation requests to a node that returns an `HTTP 200/ SYSTEM_STATUS_OK`, as described in GRE Status below.

GRE Status

GRE has a `status.jsp` URL that can be used for a health check. The following statuses are available via `/genesys-rules-engine/status.jsp`.

Status	Response Text/Meaning
HTTP 503	• `SYSTEM_STATUS_CONFIG_SERVER_NOT_CONNECTED`—Configuration Server is not connected (same as pre-8.5.2 response) • `SYSTEM_STATUS_ENGINE_NOT_INITIALIZED`—Engine is not initialized • `SYSTEM_STATUS_CLUSTER_SYNCHING`—Engine synching with Cluster
HTTP 200	• `SYSTEM_STATUS_OK`—Ready to take rule execution requests (same as pre-8.5.2 response)

GRAT

Single GRAT instance

GRAT is not considered a critical path application because it only handles the creation, editing and deployment of rules. Where only one GRAT instance is connected to a particular rules repository database at a time, if GRAT should fail, rule execution continues uninterrupted. Only rule editing

becomes unavailable.

GRAT Status

GRAT has a `status.jsp` URL that can be used for a health check.

Status	Response Text/Meaning
HTTP 200	• `SYSTEM_STATUS_OK`—GRAT server is up and running
HTTP 503	• `SYSTEM_STATUS_CONFIG_SERVER_NOT_CONNECTED`—GRAT server is not connected to Configuration Server • `SYSTEM_STATUS_DB_INITIALIZING`—GRAT server is currently initializing local cache from repository database. This can take several minutes for a large repository. • `SYSTEM_STATUS_DB_NOT_CONNECTED`—GRAT Server cannot connect to the repository database. Check the database status and/or check the database credentials that are specified in the DAP on the GRAT application object. • `SYSTEM_STATUS_UNKNOWN`—GRAT server is down. Check logs for more details.

Configuring GRAT clusters

You can configure clusters of GRAT servers which deliver much greater resilience and availability, enabling instant switchovers between GRAT nodes that are members of the cluster. All cluster members connect to the same database repository. No single GRAT node is considered primary—they are all equal partners in the n-node cluster.

An n-node cluster configuration can be used to deliver High Availability and Load Balancing. For example, 2 or 3 GRATs can be configured in a cluster. It is not recommended to have more than 4 GRATs in a cluster, due to the network demands on the repository database. It is also not recommended to have GRATs which are geographically distant from the repository database server, due to the high network demands placed on the database. It is recommended to have all nodes of a GRAT cluster in the same region as the database server. Users can access the GRAT cluster from different regions via browser (the browser's GRAT bandwidth requirement is insignificant compared to the GRAT's databse bandwidth, so it is best to have users remote via browser and have GRAT close to the DB).

A load balancer can front-end the GRAT UI, and evenly distribute the load across the available healthy GRAT nodes in the cluster. The load balancer must provide session stickiness for the GRAT user interface requests by sending all the requests pertaining to a session to the same GRAT node that initiated the session after successful login. Similarly, in the case of the GRAT REST API, after successful login, the load balancer must send the subsequent requests to the same GRAT node that handled the login request. More information on the GRAT REST API Authentication mechanism is

available here.

The load balancer can poll each node's "health check" URL (**/genesys-rules-authoring/status.jsp**) to determine the health of that node in the cluster. In the event of a failure (or planned maintenance) of one of the GRAT nodes in the cluster, the load balancer will detect this (via the healthcheck URL) and no longer send traffic to the node that is down. Any users that were currently logged into that node would be reassigned to another node. They would be prompted to log in again to resume their work: however, any "unsaved" changes at the time of the node failure would be lost.

When a GRAT node is part of a cluster, you should in general set the value of its **clear-repository-cache** option to `false`.

Configuring GRS for Secure Sockets Layer (SSL)

Introduction

Purpose

This topic describes how set to up GRAT and/or GRE in SSL (Secure Sockets Layer) mode using the Java *Keytool* utility which should be available if Java is installed. In these steps, Keytool is used to generate the self-signed certificate and update Java's keystore to add the certificate. The browser is used to download the public certificate.

Supported Configurations

Supported configurations are:

- GRAT in SSL mode and GRE in non-SSL mode
- GRAT in non-SSL mode and GRE in SSL mode
- Both GRAT and GRE in SSL mode

Changes in GRE's Communication Port in Configuration

- Make sure the correct SSL communication port is provided. In Tomcat, by default, 8080 is a non-SSL port and 8443 is for SSL.
- The value for the Connection Protocol in the GRE Port must remain as `http` (**NOT** to be changed to `https`).
- The listening mode must be set to `Secured`.

Setting up GRAT in SSL Mode

> **Important**
>
> The steps here demonstrate how to create a self-signed certificate and use it for SSL. For security reasons, it is very important that you consult your organization's IT Security Administrator to check whether you must use a certificate signed by an external CA (Certificate Authority) like VeriSign or Thawte.

1. **[+] Create the Certificate if it is not already available.**

   On GRAT, use Keytool utility to create a self-signed certificate to be used for SSL.

   ```
   keytool -genkey -alias <alias name> -keyalg <security key algorithm> -validity 360
   ```

   > **Important**
   >
   > When prompted for input `What is your first and last name?`, enter the name of GRAT's Host object in Configuration Server. It must be either GRAT's hostname or the IP address. The value entered here is used in the commonName (CN) property of the certificate.

   For example:

   ```
   /usr/local/java/jdk1.7.0_79/jre/bin/keytool -genkey -alias linux-grat -keyalg RSA
   ```

   A self-signed certificate will be created by file name `.keystore` in the user's home directory. If the certificate must be signed by an external CA (Certificate Authority), a CSR needs to be created and submitted to the CA. You can use Keytool to create a CSR. Please see Java documentation for a complete list of Keytool options.

2. **[+] Enable SSL in the server configuration by using the Certificate and disable non-SSL mode.**

   For example, to enable SSL in the case of Tomcat, the SSL configuration in `.../[TOMCAT_HOME]/conf/server.xml` looks like this:

   ```
   <Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
   ```

```
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/home/certificates/.keystore" keystorePass="changeit"/>
```

Where:

- `keystoreFile` is the path to the certificate file generated in step 1.
- `keystorePass` is the password created for the certificate in step 1.

3. **[+] On the GRE machine, get the public certificate of GRAT.**
   a. Open GRAT's link `https://[ GRAT IP address]:[SSL port number]/genesys-rules-authoring/index.jsp` in your browser.
   b. When it shows the warning about certificates, accept the certificate to be added in the browser's Trusted Certificates list.
   c. Once the certificate has been downloaded by the browser, export it using the browser's export certificate feature.

<toggledisplay linkstyle font-size:larger showtext="[+] BROWSER DETAILS" hidetext="[-] BROWSER DETAILS">

Browser	Procedure
IE 11	1. When the GRAT application is open in Internet Explorer using HTTPS, click the **Lock** icon in the address bar. (It is located beside the refresh icon on the right side in the address bar.) 2. Navigate to the **View Certificates link > Details** tab. 3. Click the **Copy to File...** button. 4. Select the format `DER encoded binary X.509 (.CER)` and click **Next**. 5. Enter the file name into which you want to save the certificate. Click **Next**, then **Finish**. OR 1. Navigate to **Internet Options > Content > Certificates**. 2. Locate the certificate. 3. Select the certificate and export it selecting `DER encoded binary X.509 (.CER)` format from the three format choices.

Browser	Procedure
Firefox 40.02	1. When the GRAT application is open in Firefox using HTTPS, click the **Lock** icon in the address bar. (It is located just before "https") 2. Click **More information...**. 3. Navigate to the **Security Tab > View Certificate button > Details Tab**. 4. Click the **Export...** button. OR 1. Navigate to **Options > Advanced > Certificates**. 2. Locate the certificate. 3. Select the certificate and export it.
Chrome 44.0	1. When the GRAT application is open in Chrome using HTTPS, click the **Lock** icon in the address bar. (It is located just before "https"). 2. In the popup that opens, navigate to **Connection tab > Certificate information link > Details** tab. 3. Click **Copy to File...**. 4. In the Certificate Export Wizard, enter the file name to which you want to save the certificate. Click **Next**, then **Finish**. OR 1. Navigate to **Customize > Settings >**. 2. Enter `certificate` in **Search Settings** and press the enter key. 3. Click **Manage Certificates...**. 4. Locate the certificate.

Browser	Procedure
	5. Select the certificate and export it.

4. **[+] On the GRE machine, add the public certificate to Java Keystore using the Java Keytool.**

   ```
   keytool -import -alias <alias> -keystore <cacerts_file> -trustcacerts -file <certificate_filename>
   ```

   > **Important**
   > This will prompt for the Java Keystore password. The default password for Java Keystore is changeit

   For example:

   ```
   /usr/local/java/jdk1.7.0_79/jre/bin/keytool -import -alias linux-grat -keystore /usr/local/java/jdk1.7.0_79/jre/lib/security/cacerts
    -trustcacerts -file /home/certificates/linux-grat
   ```

   Where:

   - `alias` is the alias to be used for this certificate.
   - `keystore` is the path to Java's Keystore in which we want to add the certificate. Make sure to update the Keystore of Java that is used by the Server.
   - `file` is the path to the certificate file (exported in step 3) that we can to add into Java Keystore.

5. **As for GRE**, repeat step 3 and 4 for any other Java clients of GRAT which would need to use HTTPS to send requests to GRAT.

Setting up GRE in SSL mode

The procedure to set up GRE in SSL mode is similar to the procedure for GRAT. In step 3, use:

```
https://[ GRE IP address]:[SSL port number]/genesys-rules-engine/status.jsp
```

to get GRE's public certificate on the GRAT machine.

Similar to the steps above, where you added GRAT's public certificate to GRE's Java keystore, for GRE you need to add GRE's public certificate (exported from the browser) to GRAT's Java Keystore.

Performance Tuning with Java Virtual Machine

Use the following parameter settings in Java Virtual Machine (JVM) Garbage Collection (GC) to maximize the performance of GRE and GRAT.

> **Important**
> For performance reasons, GRAT and GRE should run under separate JVMs.

Required

```
-server -Xms2G -Xmx2G -XX:+UseG1GC -XX:MaxGCPauseMillis=200 -XX:ParallelGCThreads=5
-XX:ConcGCThreads=5 -XX:InitiatingHeapOccupancyPercent=10
```

Highly Recommended For GC log

These settings help in debugging JVM performance problems.

> **Important**
> Adjust the value for parameter **-Xlogg** parameter as needed—this is the log file path.

```
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5
-XX:GCLogFileSize=20M -Xloggc:[path to gc log file]
```

Highly Recommended for Out Of Memory Heap Dump

> **Important**
> Adjust the value for parameter **-XX:HeapDumpPath** parameter as needed.

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=[path to heap dump file]
```

Notes

- For most cases 2G for -Xms (heap space min) and -Xmx (heap space max), as in the above required settings, will suffice. In some marginal cases, you may need to increase or decrease the heap space.
- If GRE has a lot of rules deployed, then the Xms/Xmx values must be increased to leave enough heap space to execute the requests at the required rate.

Examples

- If rule packages alone consume about 3G of heap when loaded in memory, you should use approximately 6G as the value for Xms and Xmx.
- If there are only few rule packages (each with only few rules) then 1GB of heap space should be enough.

Important

GRE's Memory Monitor statistics log output can be used to determine the amount of heap space needed by GRE. Before enabling the Memory Monitor in production environment, please make sure to read about its purpose and adjust the settings according to the heap space.

Troubleshooting

This section contains the following topics:

Configuration Considerations

This section contains some considerations that you should keep in mind when you are configuring your Genesys Rules System environment.

Genesys Rules Authoring Tool (Server)

In a multi-tenant environment, the authorized tenant(s) must be added to the Tenants tab.

- This application must have a connection to at least one GRE application, Genesys Web Engagement Engine application, or application cluster.
- A default listening port must be specified in the configuration.
- On the Security tab, under Log On As, you must provide the username of a user who has Read, Change, and Create permissions to the Scripts folder.

The Security tab is available only in Genesys Administrator 8.1.0 or later. Otherwise, you must perform this part of the configuration through Configuration Manager.

Genesys Rules Authoring Tool (Client)

- Users or access groups must have, at a minimum, Read and Execute permissions to this application, in order to log in to the Genesys Rules Authoring Tool.
- Users or access groups must have, at a minimum, Read and Execute permissions to this application, in order to access the Repository through the Rules Development Tool. That is, on the Repository Server preferences screen in the Genesys Rules Development Tool, the user whose name and password is provided must have Read and Execute permission—or must belong to an access group that has those permissions—to the GRAT client application object.

Genesys Rules Engine

- Tenants that may use this Rules Engine must be specified.
  - When deploying a rule package from the Rules Authoring Tool, if there are no "target" Rules Engines to select from, check that the correct tenants have been specified for both the Rule Authoring Tool and Rules Engines. Only those Rules Engines whose tenants match will be displayed.
- A default listening port must be specified in the configuration.
- A second port must be specified in the configuration:
  - ID: genesys-rules-engine (the name of the Rules Engine web application; can be changed by the installer)

- Port: (port being used by Tomcat or WebSphere)
- Protocol: http
- Secured: Optionally, select to activate deployment over a secured connection.

Access Groups

No access groups are created out of the box for Genesys Rules System. Suggested access groups to create, at a minimum, are the following:

- Rule Authors
- Rule Developers

Roles

- Requires Configuration Server and Genesys Administrator 8.0.2 or later.
- No roles are created out of the box for Genesys Rules System.
- Suggested roles to create, at a minimum, are the following:
  - Rules Administrator (all privileges)
  - Rules Author (relevant privileges in the Rule Authoring and Business Calendar groups)
  - Rules Developer (all privileges in the Rule Templates group)
- Users may be assigned individually to these roles, and/or access groups to which the users belong may be assigned to these roles.
- Role changes take effect immediately. See Role-Based Access Control for more information about roles and role-based access control.

Users/Persons

- No users are created out of the box for Genesys Rules System.
- Genesys Rules System users can be agents or non-agents.
- Users who log in to the GRAT must have access to one or more tenants, in a multi-tenant environment, with at least Read permission to the tenant(s).
- In addition to the users for the GRAT, you must create one non-agent user (for example, GRAT_Application_Proxy) who has Read and Change permissions to the Scripts folder.

Business Structure

- No business structure is created out of the box for Genesys Rules System.
- If you are using the Genesys Rules System with intelligent Workload Distribution, the business structure is created in the iWD GAX Plug-in and is then synchronized with Configuration Server, after which it becomes available for use by the Genesys Rules System.
- A top-level folder must be created, of type Business Unit (called Configuration Unit in Configuration Manager) or Site, with the exact name of Business Structure.
- Within the Business Structure folder, at least one more Business Unit or Site must be created (it does not matter which one).

These first level nodes under Business Structure represent the Solution(s). Within each solution, additional levels of hierarchy may be created, as needed, using either Business Units or Sites. Those levels of hierarchy beneath the Solution level will represent the business context.

- Multiple solutions may be created by creating additional Business Units or Sites directly beneath the Business Structure folder.
- Business Structure is created under Resources for single tenant Configuration Server or under a Tenant for a multi-tenant Configuration Server.
- Read permission to the Business Structure folder must be provided to the users and/or access groups that you want to use the Rules Authoring Tool. Normally, if the user or access group has permission to the Tenant object, this will be propagated automatically. If you do not want a user or access group to have permission to see all nodes of the business structure, you can control this by not giving that user or access group(s) Read permission to those folders. See Defining the Business Structure for more information.

Scripts

- A user (such as `GRAT_Application_Proxy`) on whose behalf the GRAT server will update the `Scripts` folder must have Read, Create, and Change permissions to this folder.
- Individual GRAT Rules Development users, or one or more access group(s) to which they belong, must have Read permissions to the individual Script objects that represent the rule templates to which they should have access. Alternatively, you might decide to grant permission to the entire Template Access Control scripts folder to individual users or an access group such as Rule Developers, and allow that permission to propagate to all scripts that might be created in the future.
- Individual GRAT users, or one or more access group(s) to which they belong, must have read permissions to individual Script objects that represent the rule templates that rule authors should be able to add to a rule package when creating it.
- Users need Read access to parameter scripts. These scripts are maintained via Genesys Administrator Extension.

The log4j2.xml File

The `log4j2.xml` file is used to configure initial logging for the Rules Engine and for the Genesys Rules Authoring Tool. Once the Rules Engine and GRAT are initialized, logging is done through the configured Application options. The `log4j2.xml` file contains logging attributes that are used during the startup of the application, before the configured log settings are read by Configuration Server. In general, you should not have to modify this file and you can accept the default values. But should you need to change the defaults, perform the steps in the following procedure:

Procedure

1. Locate the `log4j2.xml` file. This file can be found in the `.war` file, which is located in the installation directory.
2. Extract the `.war` file by using WinZip or a similar tool for extraction. (For the Rules Engine and the Rules Authoring Tool, the `.war` files are named `genesys-rules-engine.war` and `genesys-rules-authoring.war`, respectively).
3. Open the file in a text editor, and update any logging parameters.
4. Save the file.
5. Add the modified `log4j2.xml` file back into the original .war file by using WinZip (or a similar tool). Be very careful to preserve the “path” of that file during this step.

Locating Log Files

Before connecting to Configuration Server, GRAT and GRE will log by default to the relative path:

- `"logs/GRATInit"`
- `"logs/GREInit"`

These logs are useful in debugging start-up issues; for example, if GRAT or GRE are unable to connect to Configuration Server.

- **In UNIX based systems,** these initial logs can be found under your application servers "logs" directory.
- **In Windows based servers,** these initial logs can be found in the `\Windows\SysWOW64\logs` directory.

You can change the initial location of these logs by editing the `log4j2.xml` file as described in this section.

After GRAT or GRE connect to Config Server, the log location is determined by the application's configuration options (`log` section).

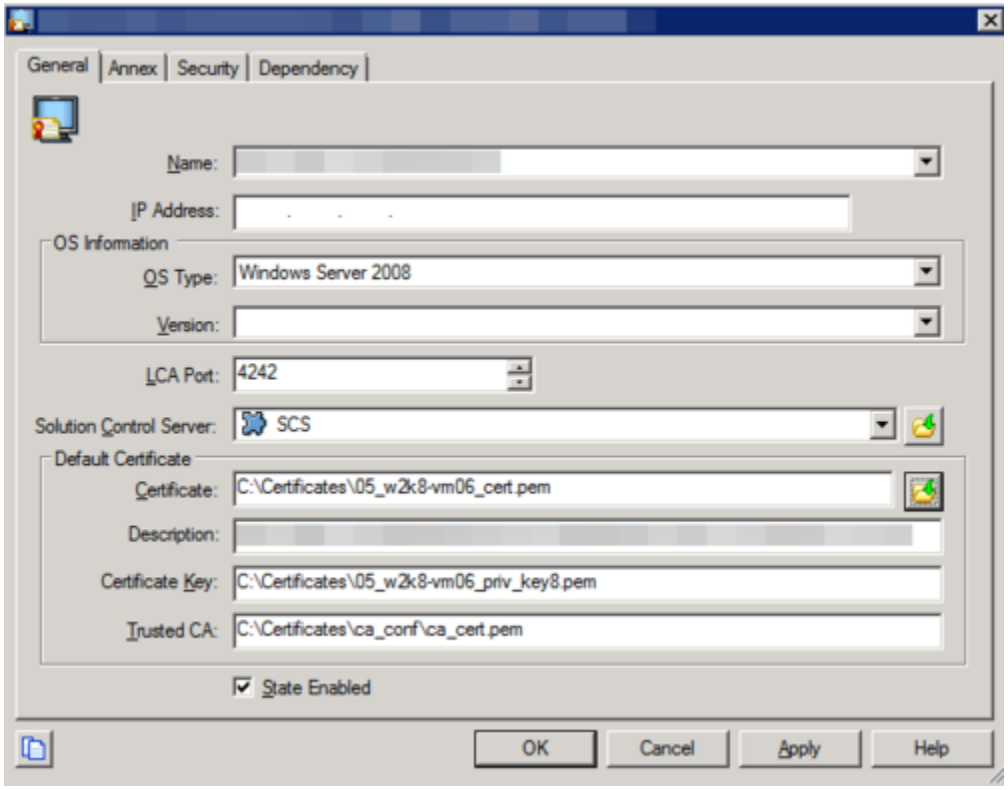
Security Certificates

GRE and GRAT applications are unable to understand thumbprint certificates from the Windows Microsoft Certificate Store while connecting to Configuration Server or Message Server. Consequently they cannot establish a TLS connection to either Configuration Server or Message Server, and a security error is generated, such as this:

```
15:14:31.445 Alarm 21363 [ServerConnectionMonitor][Thread-2]: connect(): caught exception
while
opening connection to server '<GRAT/GREServerVersion>'. Nested exception: Could not configure
TLS.
```

Workaround

Create certificates in .PEM format and private-keys in PKCS#8 format—see the example below.



Paths to the physical certificates can be configured either on the Application or the Connection level, but not on the Host level (despite this being a general Genesys recommendation). For Host level certificates, if GRAT and GRE are located on the same host as Configuration Server or Message Server or other C++ applications, the secured connection will not be established because C++ based applications do not accept PKCS#8 format.

You can convert private-key from PEM to PKCS#8 format using the following OpenSSL command:

```
openssl pkcs8 -topk8 –nocrypt -in tradfile.pem -out p8file.pem
```

Localization

> **Important**
> Localization is not available in the initial 9.0 release of GRS.

Defining the Business Structure

Overview

The business structure is a hierarchy of business units. No business structure is created out-of-box for Genesys Rules System; the business structure must be configured in Genesys Administrator or Configuration Manager. For customers who are using the Genesys Rules System with intelligent Workload Distribution, the business structure is created in iWD GAX Plug-in and then synchronized with Configuration Server, after which it becomes available for use by the Genesys Rules System.

The business structure that you configure will be visible in the Genesys Rules Authoring Tool. Each rule package will display the business structure for the Tenant. Each Tenant can contain one more Solutions as the first level of the hierarchy, and rules can be defined at each level (node) of the business structure from Solutions down.

Rules that are configured for the Solution, known as global rules, are executed first, followed by rules configured for the first node of the business structure, then rules configured for the second node, and so on. Global rules are only “global” within the defined rule package.

The business structure that you create can vary depending on a number of factors. Sample structures are provided in this chapter. The structure can be product- or business-specific.

Object permissions are used to determine which elements of a business structure are visible to various users. See Role-Based Access Control for more information.

Your Tenant’s business structure is created under Resources for single-tenant Configuration Server, or under a Tenant for a multi-tenant Configuration Server.

Procedure

1. Navigate to the `Resources` folder for a single-tenant Configuration Server, or to the specific Tenant for a multi-tenant Configuration Server.
2. Open the `Business Units/Sites` folder (in Genesys Administrator) or `Configuration Units` (in Configuration Manager) folder.
3. Create a new top-level folder named `Business Structure`. This folder **must** be named `Business Structure`.
4. Within the Business Structure folder, click either `New Unit` or `New Site` to create at least one more Business Unit or Site (it does not matter whether you create a site or a unit). This new site/unit will represent the Solution.
5. Within the new folder (the Solution), additional levels of hierarchy can be created as needed, using either Business Units or Sites. The levels of hierarchy beneath the Solution level will represent the business context.

Warning

You cannot have the same node name across different departments in the hierarchy. So, you must either:

- Ensure that all node names within the business structure are unique, or;
- Add a condition to your template (for example, location) and have have it passed in as a new Fact field.

Multiple Solutions can be created by creating additional Business Units or Sites directly beneath the Business Structure folder.

Read permission to the Business Structure folder must be provided to the users and/or access groups that you want to use the Rules Authoring Tool. Normally this will be propagated automatically, if the user or access group has permission to the `Tenant` object. If you do not want a user or access group to have permission to see all of the nodes of the business structure, you can control this by not giving that user or the access group(s) of that use read permission to those folders.

Role-Based Access Control

Genesys Rules System role-based access control utilizes Configuration Server-defined access groups and roles to control visibility and access to rule packages, rule templates, rules, and business calendars. Because these objects are not stored in the Configuration Server database they will not have security permissions associated with them, as Configuration Server objects do. The GRAT server will utilize the access permissions for the container object, and the Genesys Rules System objects will inherit these access permissions.

Role-based access control requires Configuration Server 8.0.2 or higher and Genesys Administrator 8.0.2 or higher.

Rule packages and business calendars inherit their access permissions from the Tenant object with which they are associated and the **Business Structure** folder access permissions. Business rules are associated with a specific node in the business structure. Their access permissions are inherited from the Configuration Server–defined node with which they are associated (the business structure nodes are created by using Configuration Manager or Genesys Administrator).

Rule templates have Script objects created in Configuration Server that are used to hold the individual access permissions of the rule template. Additionally, rule templates inherit the access permissions from the business structure node with which they are associated.

For a full discussion of Role-Based Access Control, please refer to the *Genesys 8.5 Security Deployment Guide*.

Role Permissions for Rules and Rule Packages

Genesys Rules System defines a set of role permissions for governing the tasks that can be performed in the Genesys Rules Authoring Tool.

Rules

The combination of the access permissions and the role permissions will determine whether a task can be performed. For example:

- To view a rule a user must have Read permission for the node with which the rule is associated as well as the Business Rule - View role permission.
- To delete a rule, the user must have Read permissions for the node and the Business Rule - Delete role permission. In this example, Read access permission is also needed for the delete task, because the user will not have visibility to any object that is associated with the node without Read access permissions.

Rule package-level overrides

You can make package-level overrides to these global roles—a user's role privileges can be restricted to specific rule packages by applying Role-Based Access Control at the rule package level.

Rule Package Level Roles (roles created specifically for use with rule packages only) can be mapped to rule packages to override the global-level roles. These Rule Package Level Roles will have no effect if not mapped to a rule package.

Role Permission—View Rule Package

View access for specific rule packages can be controlled by using the role permission **View Rule Package**. This permission is applicable to only the rule package level. All of the existing role permissions except **Create Rule Package** and template-related permissions are applicable at the rule package level too.

Example

You can now assign role permissions at both global/node level and at rule-package level to achieve the following outcome:

- Department A
  - Rule package 1

    - Rule package 2
    - Sales
        - Rule package 3
- Department B
    - Rule package 4

- User A—Can see Department A but not Department B
- User B—Can see Department B but not Department A
- User C—Can see rule package 1, but rule package 2 is hidden

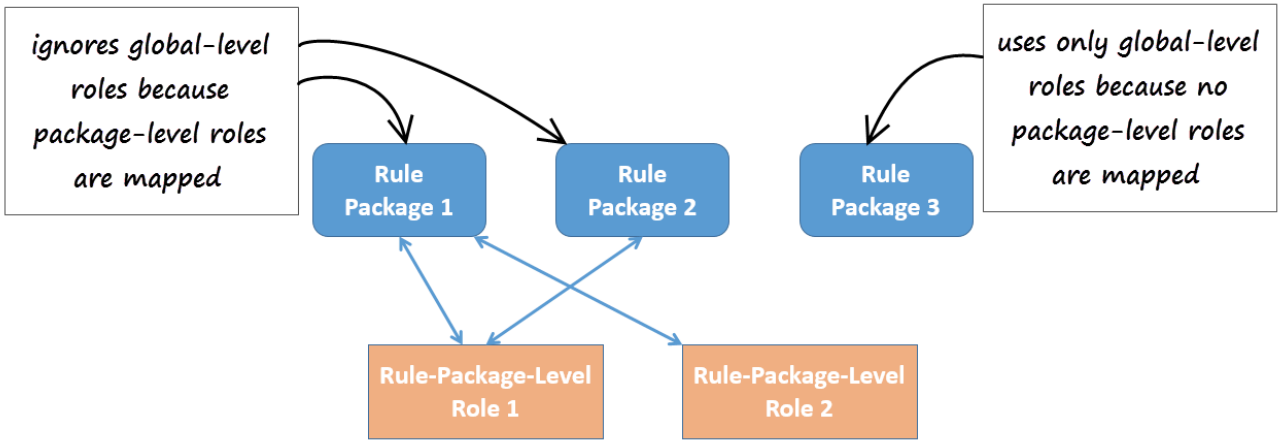
Location

To distinguish these roles from global-level roles, they are placed in this folder:

```
[Tenant] > Roles > GRS Rule Package Level Roles
```

Mapping

Where package-level roles are mapped to a rule package, they override global-level roles.



Managing the Mapping of Roles

The mapping of the rule packages to Rule Package Level Roles is managed in Genesys Administrator or Genesys Administrator Extensions, in the options under section **Rule Package Level** of the **\Scripts\GRS Access Control\GRS Role Mappings** script.

> **Important**
> Because the delimiter in the list of roles is a comma, you can't use commas in the names of any role.

Viewing GRAT User Permissions

To enable GRAT users to view their current list of permissions, a **Check My Permissions** button is now also available at the rule-package level and shows the permissions at selected package level.

List of Permissions

- Business Calendar - Create
- Business Calendar - Delete
- Business Calendar - Modify
- Business Calendar - View
- Business Rule - Create
- Business Rule - Delete
- Business Rule - Modify
- Business Rule - View
- Business Rule - Edit Only - allows a user to edit and save only the parameter values of a rule. No other permissions are granted.
- Rule Template - Create
- Rule Template - Delete
- Rule Template - Modify
- Rule Package - Create
- Rule Package - Delete
- Rule Package - Modify
- Rule Package - Deploy
- Rule Package - Undeploy (also requires Rule Package - Deploy permission)
- Rule Package History - Admin View—Allows viewing of complete package history for a rule package without checking access to the business hierarchy subnodes used inside the rule package. Even with this role privilege enabled, the package history will only be shown for packages that the user can view.
- Rule Package History - View Changed By—Allows users to view Changed By information in Package History.

- Locks - Override
- Test Scenario - Create
- Test Scenario - Modify
- Test Scenario - Delete
- Test Scenario - View
- Test Scenario - Execute
- Snapshot - Create
- Snapshot - Delete
- Snapshot - View: User can view and export snapshots. If this is not enabled, users will only see LATEST in the list of snapshots, which represents 8.1.2 functionality where users can only deploy the latest version.

> **Important**
>
> Snapshot permissions are active on the `Deployment` tab of GRAT, so all snapshot permissions also require `Rule Package - Deploy` permission.

User Logins

GRAT has multiple connections to Configuration Server:

- The server connection that is used by the Rules Authoring server to read application information and perform various server tasks
- The individual client connection of each user who logs on to the GRAT. This is limited based on the configuration of the user's login.

Business Hierarchy

Each Tenant should contain a folder called **Business Structure** (for single-tenant Configuration Servers, the **Business Structure** folder must be created under **Resources**). Under that folder there can be multiple levels (nodes) of sites/business units that represent the business hierarchy for this Tenant.

Each user login should be configured in Configuration Server with:

- Read permissions for only the Tenants that will be visible to this user (if there is more than one Tenant) and;
- Read permissions for only the nodes of the business hierarchy that this user can view.

Users who have Rule View permissions can see all of the rules that are associated with a node that is visible to them. See Defining the Business Structure for more information about business structures.

Role Task Permissions

When GRAT has been deployed by using Genesys Administrator, role task permissions can be configured in Genesys Administrator.

A new Role object can be created under **Provisioning > Accounts > Roles**. On the **Role Privileges** tab there is a check box to add the privileges that are associated with the Genesys Rule Authoring Generic Server.

You can grant users a specific set of permissions by adding them as members of a role—either individually or as part of an access group. There are six groups of privileges:

- **Rule Authoring**
  - Create
  - Delete
  - Modify (overrides Edit Only)
  - Edit Only (edit-only access to existing rule parameters, when in conjunction with View. Cannot create new rules, delete rules, or even add/remove conditions.)
  - View
- **Rule Packages**
  - Create Package (Global Roles Only)
  - Modify Package
  - Delete Package
  - Deploy Package
  - Undeploy Package
  - View Package (Rule Package Roles only)
  - Create Package Snapshot
  - Delete Package Snapshot
  - View Package Snapshot
  - View Package History Changed By
  - Package History Admin View
- **Rule Templates**
  - Create
  - Modify
  - Delete
- **Test Scenarios**

    - Create
    - Modify
    - View
    - Delete
    - Execute
- **Business Calendars**
    - Create
    - Delete
    - Modify
    - View
- **Locks**
    - Override Locks

Templates and their Script Objects

Templates

Role permissions for importing and exporting templates and rule packages must be set to the following values:

- To import a template, a user must have Create permission for the Rule Template.
- To export a template, a user must have read access to the Template Script Object representing the template.
- To import or export rule packages, a user must have full permissions granted. For example, if a user does not have the ability to view business calendars or test scenarios, they won't be exported in the rule package XML. Conversely, if a user doesn't have permission to create calendars or test scenarios on import, they will not be able to create these resources from the imported rule package.

Script Objects

Script objects are used to control visibility to templates. Whenever a template is created, a Script object is created automatically in the **Template Access Control** folder under the **Scripts** folder to represent that template. A user must have read access to that Script object to be able to view that template.

Genesys recommends that you give template developers View permissions to the **Template Access Control** folder and have that permission propagate to all sub-objects. This way, template developers can immediately view any template that they may create. All other users will not be able to see the newly created templates until View permissions are explicitly granted for that template.

Configuring a User

The following procedure provides the basic steps for setting up users for GRAT.

Procedure

1. Give the user Read access to all of the Tenants that they can access.
2. Add the user as a member of a role with the desired permissions, or add the user as a member of an access group which can be part of a role.
3. Give the user Read access to the **Business Structure** folder and all of the desired nodes for that user.
4. Give the user Read access to all of the desired templates through the Script objects.