Genesys Rules System Deployment Guide

High Availability Support

12/19/2025

High Availability Support

Contents

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

Pre-8.5.301

Before release 8.5.301, only one Genesys Rules Authoring Tool (GRAT) instance can be connected to a particular rules repository database at a time. GRAT is not considered a critical path application because it only handles the creation, editing and deployment of rules. If GRAT should fail, rule execution continues uninterrupted. Only rule editing becomes unavailable.

GRAT can be set up in a cold standby configuration. A standby GRAT can be installed as a mirror image on a separate machine and be configured to use the same configuration management

application, same HTTP ports, and so on. Should the primary GRAT fail (hardware failure, network), the standby GRAT could be brought online quickly to restore service. Both the primary and standby GRATs can be connected to the same repository database; however, they should not be connected simultaneously. The rule author would have to log in again and resume their activity.

When configuring a standby GRAT, use option `clear-repository-cache=true` for both the primary and backup GRAT instances. Setting this option to `true` can delay the startup of GRAT, since the cache must be rebuilt each time, but it ensures that it is properly synchronized with the rules repository database.

GRAT Status

GRAT has a `status.jsp` URL that can be used for a health check. GRAT's `status.jsp` always returns `HTTP 200`. The response text must be queried to determine if the GRAT server is up or down.

Status	Response Text/Meaning
HTTP 200	• `SYSTEM_STATUS_OK`—GRAT server is up and running • `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.

8.5.301+

In release 8.5.301, you can now 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.

Each node maintains a journal of changes, which resides on a separate repository database table. Nodes can periodically poll the database to ensure that they mirror updates made on the other nodes. This design also allows new nodes to be added to the cluster and synchronized with their peers.

In addition, when a GRAT is part of a cluster, you should in general set the value of its `clear-repository-cache` option to `true`. For GRATs in a cluster, setting this value to `true` can help avoid the repository corruption errors that can occur if you forget to clear the cache when a node is re-

added to the cluster or moved from a different cluster. However if there is a specific reason to avoid the possible slower startup of GRAT server that might ensue, then set the option to `false`.

If the GRAT startup delay is irrelevant or if GRAT startup is quick enough with the `clear-repository-cache` option set to `true`, then it should be set to `true`.

ADDITIONAL DETAILS.