Genesys Rules System Deployment Guide

Creating the GRE Application Object in Configuration Manager

5/7/2025

Creating the GRE Application Object in Configuration Manager

Procedure

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

1. 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..

- Click OK to save the template.

2. 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. Logging options are as follows:

log

Description	Valid values	Default value	Takes effect
all			
Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile	• stdout—Log events are sent to the Standard output (stdout). • stderr—Log events are sent to the Standard error output (stderr). • network—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the all log level option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database. • memory—Log events are sent to the memory output on the local disk. This is the safest output in terms of the	stdout	After restart

Description	Valid values	Default value	Takes effect
	application performance. • `[filename]`—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.		
expire			
Determines how many log files will be kept on disk. If set, expire specifies the maximum number of log files kept on disk.	Any number	(blank)	After restart
segment			
Determines whether a log output written to file is split in multiple segments. If it is, segment specifies the maximum size of each segment file.	Any number that represents the log size in megabyte	(blank)	After restart
standard			

Description	Valid values	Default value	Takes effect
Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example: `standard = stderr, network`	• `stdout`—Log events are sent to the Standard output (`stdout`). • `stderr`—Log events are sent to the Standard error output (`stderr`). • `network`— Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. • `memory`—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. • `[filename]`—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.	stdout	After restart
trace (not in application template by default)			
Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and	• `stdout`—Log events are sent to the Standard output (`stdout`). • `stderr`—Log events are sent to the Standard error output (`stderr`). • `network`—Log events are sent to Message Server, which can reside anywhere on the	stdout	After restart

Description	Valid values	Default value	Takes effect
Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: `trace = stderr, network`	network. Message Server stores the log events in the Log Database. • `memory`—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. • `[filename]`—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.		
verbose			
Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.	• `all`—All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated. • `debug`—The same as all. • `trace`—Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated. • `interaction`—Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels)	standard	After restart

Description	Valid values	Default value	Takes effect
	are generated, but log events of the Trace and Debug levels are not generated. • standard Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated. • none—No output is produced.		

10. Configure the options on the Settings tab as follows:

Settings in GRE

Description	Valid values	Default value	Takes effect
deployed-rules-directory			
Specifies the directory in which to keep the working copy of deployed rule packages. When a package is deployed, a copy of the deployed package is placed here. When the rules engine is restarted, all packages defined in this directory are loaded and made		/GCTI/logs/GRS_Engine	After restart

Description	Valid values	Default value	Takes effect
available for execution. Specifying a deployed-rules-directory is recommended. If a value is not assigned to the deployed-rules-directory, the rule packages are placed in the `WEB-INF\config` sub-directory within the `genesys-rules-engine` web application directory. At this location the deployed rule packages may be deleted when an updated `.war` file is deployed. If you choose to change the default value, ensure that the path exists and that the application server can write to the specified directory.			
max-number-rule-executions			

Description	Valid values	Default value	Takes effect
The maximum number of rules to be executed during a request. This is used to detect unwanted recursion when `sequential-mode` is false. If this maximum is reached an error is reported. May be set to `-1` to denote no maximum.	Any positive integer or -1	10,000	Next rules execution
sequential-mode			
Indicates whether to run the rules engine in sequential mode. In sequential mode, after the initial data set, no more data can be inserted or modified. This allows for the rules engine to operate in a simplified way.	true/false	false	On rules deployment
verify-deployer-address			
Indicates whether to verify the TCP address of the	true/false	true	Immediately

Description	Valid values	Default value	Takes effect
application deploying rules to be that of an associated Genesys Rules Authoring Tool.			
esp-worker-threads			
Specifies the maximum number of worker threads available when using the ESP interface to execute rules.	Any positive integer	5	Immediately
load-packages-on-start			
Indicates whether to load deployed rule packages at application start up. If packages are not loaded at startup (value= `false`), then a package is loaded on its first execution request.	true/false	true	Immediately
json-hierarchical-driver			
With value `true`, the `JsonHierarchic`	true/false	false	Immediately

Description	Valid values	Default value	Takes effect
`alStreamDriver` class is used to serialize JSON responses. With value `false`, the `JettisonMapped XmlDriver` class is used. The Jettison driver is unaware of the original data type and will try to detect numerical values and omit the quotes, whereas the `JsonHierarchic alStreamDriver` will maintain the data type.			
cache-operational-parameters (new in 8.5.0)			
Operational parameters are rule parameters whose value is obtained at rule execution time. They are configured in GAX as Parameter Groups, and stored in the Configuration Server database. Prior to 8.5,	true/false	true	Immediately

Description	Valid values	Default value	Takes effect
whenever an operational parameter was referenced during the execution of a rule, GRE would fetch the current value from Configuration Server. In high-volume environments, this could put unnecessary stress on Configuration Server. In GRS 8.5, the value of the operational parameters can be cached inside GRE, to make fetching faster. Instead of fetching the value with each reference, GRE will set up a listener to Configuration server and maintain the value in a local cache. When the administrator changes the value of the			

Description	Valid values	Default value	Takes effect
parameter using GAX, GRE will receive an event and update its local cache. If `cache-operational-parameters` is set to `true` (default), this new caching mechanism will be enabled. If `cache-operational-parameters` is set to `false`, no caching will be used and each reference will fetch the current value from Configuration Server (as was done prior to 8.5).			
parameter-cache-timeout (new in 8.5.0)			
When `cache-operational-parameters` is set to `true`, `parameter-cache-timeout` defines how long (in hours) an	Integer	168	Immediately

Description	Valid values	Default value	Takes effect
operational “parameter group” will remain in the cache. After the timeout expires, the transaction will be removed from the cache until the next time the value is requested. This is used to clean up old subscriptions to parameter groups which are no longer being referenced. The default value for this will be 168 (168 hours = 1 week).			
clear-cache-on-disconnect (new in 8.5.0)			
When `cache-operational-parameter` is set to `true`, the `clear-cache-on-disconnect` parameter defines what the behavior should be if GRE loses connection with the Configuration Server. If `clear-`	true/false	false	Immediately

Description	Valid values	Default value	Takes effect
`cache-on-disconnect` is set to `false`, GRE will continue to use the cached value for any rule evaluations, until such time as the Configuration Server is restored. With this option, there is a risk that GRE could use “stale” values for rule evaluation during the time the connection to Configuration Server is down. If `clear-cache-on-disconnect` is set to `true`, the cache will be cleared and a null (“”) value will be used in the rules. With this option, there is potential that rules will fail evaluation during the period that the Configuration Server connection is down.			
include-rule-evaluation-detail-in-response (new in 8.5.001)			

Description	Valid values	Default value	Takes effect
Returns rules that did not fire, conditions that evaluated `false` and rule evaluation time back to the REST client invoking the rule evaluation request. Prior to 8.5.001, only the results of rules that fired were returned. Note: Currently, the `rulesDisqualified` and `executionTime` is not returned via ESP to iWD.	true/false	false	Immediately
unload-inactive-package-timeout (new in 8.5.1)			
Specifies the interval (in minutes) after which, if a rule package remains unused by GRE, it is unloaded from memory. If the option is not specified, then packages are loaded in GRE with no timeout.	Integer	No default	At GRE start/restart

Description	Valid values	Default value	Takes effect
If a request for a rule package is received after the package has been unloaded, it is automatically loaded into memory again and the timer is restarted.			
iwd-set-department-from-process (new in 8.5.100.21)			
Enables (value = `true`), GRE to determine the Department from the properties of its Process, for ESP server requests. The setting of the Department from the Process properties will only occur if the Department is not specified and the business context level 1 is not specified.	true/false	false	At GRE start/restart
enable-memory-monitor (implemented in HF 8.5.100.15)			
Enables or disables the Memory Monitor feature.	true/false: Absence of this property or invalid value results in `false`	false	At GRE start/restart

Description	Valid values	Default value	Takes effect
memory-monitor-interval (implemented in HF 8.5.100.15)			
The interval in seconds between periodic memory usage checks.	integer: min 1	60	At GRE start/restart
memory-monitor-threshold (implemented in HF 8.5.100.15)			
The memory usage threshold expressed as a percentage. If memory usage goes above the threshold, GRE's `status.jsp` returns HTTP 503 status with a message `SYSTEM_STATUS_MEMORY_USAGE_ABOVE_THRESHOLD`. Genesys Management layer is also notified about GRE's unavailability via status set in LCA Connection. When memory usage is back below the threshold, GRE's `status.jsp` returns HTTP 200 status and Genesys Management	integer: min 40, max 80	70	Immediately

Description	Valid values	Default value	Takes effect
Layer is notified that GRE is available.			
memory-monitor-threshold-strategy (implemented in HF 8.5.100.15)			
Allows you to change the out of memory error handling behavior of memory monitor. • `adaptive`—At out-of-memory error, a new threshold is calculated and it is obtained by reducing the configured `memory-monitor-adaptive-threshold-safety-margin` amount from the percentage memory usage at the time Memory Monitor receives the out-of-memory notification. The threshold is reset only if the new calculated value is less than the configured threshold (or less than current override)—for example, if the configured threshold is 80 %, the safety margin is 10 % and if an out-of-memory error notification is retrieved when memory usage is 70 %, the new	adaptive/forced	adaptive	Immediately

Description	Valid values	Default value	Takes effect
override threshold will be 70 - 10 = 60 %. In this scenario, Memory Monitor learned that out-of-memory error can happen at 70 % memory usage, so it adjusts the threshold to be 60 %. The override threshold that the "adaptive" strategy sets can be removed by temporarily setting the strategy to "forced". It must be kept as "forced" for at least the `memory-monitor-interval` time. The override can also be removed by reducing the configured threshold value so that the new configured value is equal to, or lower than, the override threshold. The override is removed if GRE is restarted, so it is recommended to change the configured threshold to match the override threshold before restarting the GRE. • `forced`—At out-of-memory error, it does nothing except logging the current memory			

Description	Valid values	Default value	Takes effect
usage. It forces Memory Monitor to raise an alarm only when memory usage is above the threshold. If using this approach, the threshold must be set low enough so that no out-of-memory errors occur. Temporarily setting this strategy allows the removal of the override threshold set by the "adaptive" strategy.			
memory-monitor-adaptive-threshold-safety-margin (implemented in HF 8.5.100.15)			
The safety margin percentage used by the "adaptive" strategy, when set. The new threshold, set when application memory is exhausted, is obtained by reducing this percentage amount from the percentage memory usage at the time of memory exhaustion.	integer: min 10, max 30	10	Immediately

11. Save your changes.