This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Rules System Deployment Guide

New Features by Release

5/12/2025

Contents

New Features by Release

New in Hot Fix 8.5.100.15

GRE Memory Monitor

The Memory Monitor is a new feature built in to GRE itself. It is designed to periodically check GRE's memory usage, and set its operational state to "over threshold" if memory usage exceeds the configured threshold. The state is reset to back to "normal" if memory usage comes back below the threshold. It also provides an option ("adaptive" strategy) to automatically adjust the configured threshold if an out-of-memory error occurs before threshold is reached (for example, if the configured threshold was set too high).

The memory monitor sets this state in two places:

- `status.jsp`—This .jsp provides a "health check" URL for load balancers to use. If the memory usage is above the configured threshold, `status.jsp` returns `SYSTEM_STATUS_MEMORY_USAGE_ABOVE_THRESHOLD` (HTTP 503 status). Load balancers should be configured to route requests only to GRE nodes whose `status.jsp` returns `SYSTEM_STATUS_OK` (HTTP 200 status).
- Genesys Management Layer—The Memory Monitor will also notify Genesys Management Layer if the memory is in an overloaded state by setting the status to `SERVICE_UNAVAILABLE`.

CONFIGURATION OPTIONS

New in Release 8.5.100.21

Automatic Protocol Detection Override

You can use the `deploy-method` configuration option to override the automatic detection of the protocol that is used to construct the "callback" URL used by GRE to fetch the DRL. Previously, when deploying from GRAT to GRE, GRAT automatically detected whether it was running http or https, and used that protocol to construct the "callback" URL used by GRE to fetch the DRL. However, when running through an ELB, GRAT could sometimes wrongly think it was running over https, when the link between the ELB and GRAT server is http. This might cause issues in deployment. The default value of the new configuration option is `auto`. GRE will use the selected method to connect with the GRAT server during deployment.

CONFIGURATION OPTION

Enable/Disable Nested Solutions

In GRS 8.5.1, a new feature was added called **'Nested Solutions Business Hierarchy'**, which enables an authorized user to create a new rule package anywhere in the business hierarchy. Because some customers might want to restrict their users to creating rule packages under the Solution node only, in release 8.5.100.21 a new configuration option—`enable-nested-solutions`—has been implemented that allows users to enable or disable this feature. Disabling this feature is recommended for iWD users.

CONFIGURATION OPTION

Support for the French Canadian (FRC) language

Support for the French Canadian (FRC) language is implemented in this release.

Setting Department from Process Properties

Based on properties of the Process GRE can determine the relevant Department. Configuration option `iwd-set-department-from-process` with value `true` enables GRE make this determination and so to run rules created for Department and Process, if the Department is not specified elsewhere. If the option is set to `true` then the Engine will set the Department from the Process for ESP server requests. The setting of the Department from the Process will only occur if the Department is not specified and the business context level 1 is not specified.

CONFIGURATION OPTION

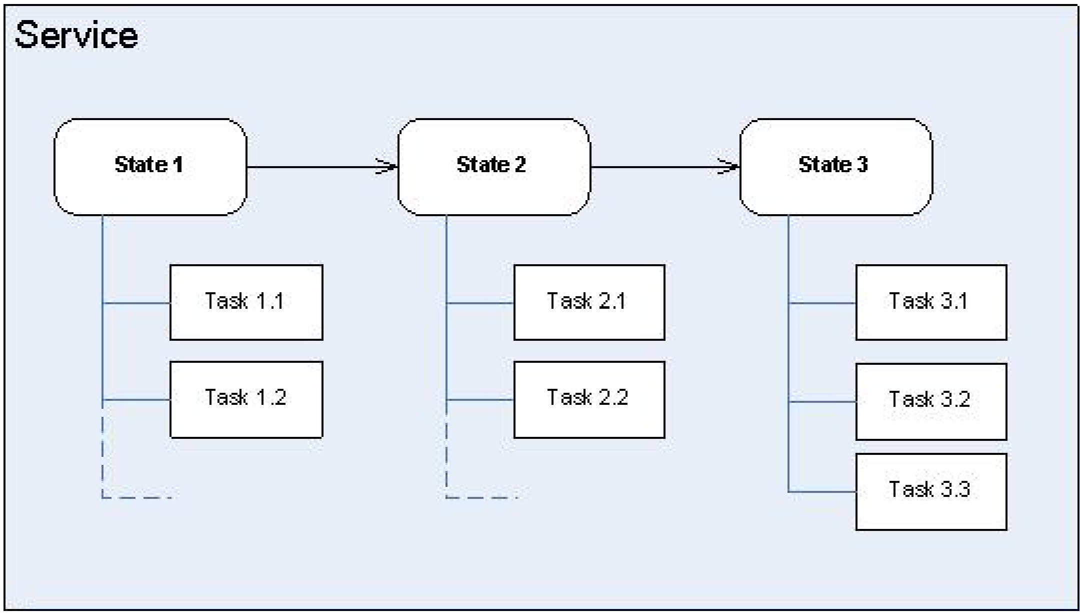
New in Release 8.5.1

Support for Test Scenarios in Conversation Manager

In the initial 8.5.001 release of GRS, the Test Scenario feature did not support rules that were created using the Conversation Manager (CM) template. This is because the Test Scenario feature in release 8.5.001 works by taking the input data (a set of one or more facts with different fields) that is configured by the user and building the appropriate Fact model, then running the rules under GRAT using that set of data. In release 8.5.1, the Test Scenario feature now supports rules based on the CM template.

Data Structure in CM

With Conversation Manager, the data is in a hierarchical JSON format of **Customer -> Service -> State -> Task**. Any given **Customer** may have one or more **Services**. Each **Service** may be in at most one **State** at a time. Each **State** may have one or more **Tasks**. **Tasks** may also be associated directly with **Services**.

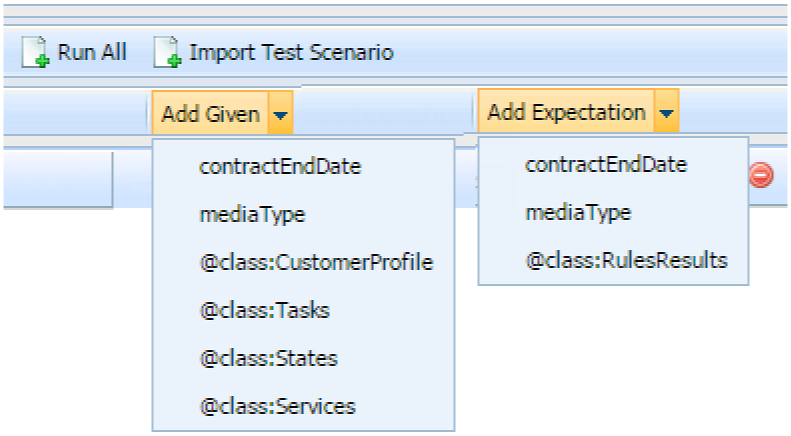


So the `Customer`, `Services`, `States` and `Tasks` Facts have now been added the lists of Facts that can be defined as **Given** fields, and the `RulesResults` Fact has been added to the list of Facts that can be defined as an **Expectation**.

> Important
>
> The current CM Template is only interested in the Type, Start Time, and Completion Time (if any) of Services, States, and Tasks.

Each of the new values is represented by a JSON string which will be the value for that field.

Now, when the type of rule for which you want to create a test scenario is a Conversation Manager rule (based on the Conversation Manager template), a series of different values for the **Given** and **Expectation** elements that reflect these more complex data structures are available. In the example below you can see the **Customer > Service > State > Task** structure is reflected by the four **@class** entries in the drop-down list of Givens and the **@class:RulesResults** entry in the drop-down list of Expectations.



When you select an **@class** entry, a new column is added. Click on a grid cell under the new column to bring up the edit dialog for that entry. The additional data listed below can be selected as either a **Given** or an **Expectation**.

Additional CM Template Objects

Givens

The list below shows the additional provided data.

- Available by selecting one of the **@class** entries:
  - `Add Customer Attribute`
  - `Add Service`
  - `Add Service Type`
  - `Add Service Start Time`
  - `Add Service Completion Time`
  - `Add State`
  - `Add State Type`
  - `Add State Start Time`
  - `Add State Completion Time`
  - `Add Task`
  - `Add Task Type`
  - `Add Task Start Time`
  - `Add Task Completion Time`
- Available for direct selection from **Givens**:

- `Add Interaction Media Type`
- `Add Contract End Date`

Expectations

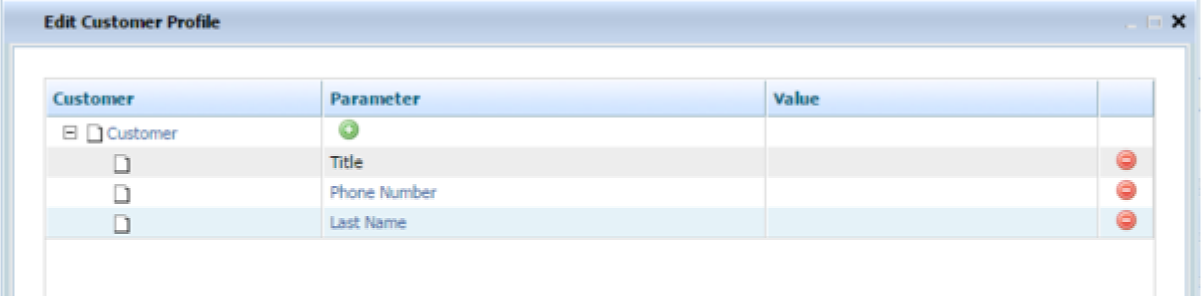
The list below shows the additional expected results:

- `Update Customer Attribute`
- `Request Specific Agent`
- `Request Agent Group`
- `Request Place Group`
- `Request Skill`
- `Send Communication to Customer`
- `Block Communication to Customer`
- `Offer Service Resumption`
- `Offer Survey to Customer`

Edit Dialogs

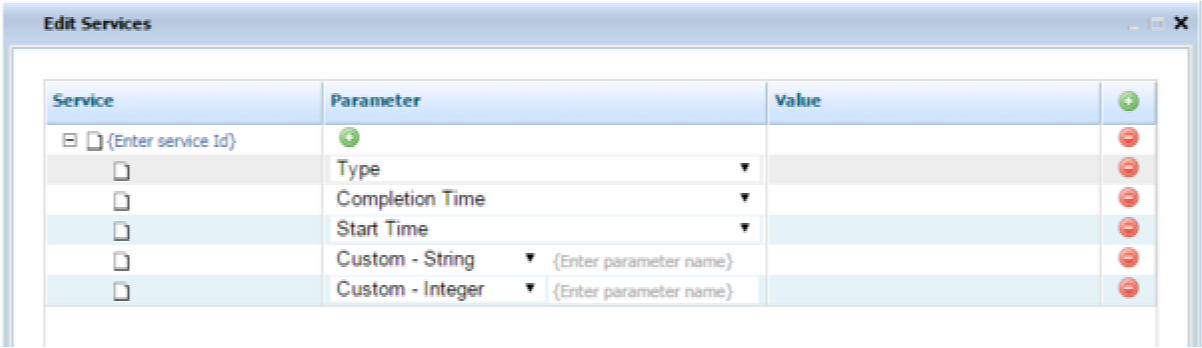
To create entries for the **Givens** and Expectations of your Conversation Manager test scenario, select the relevant **@class** item and use the sample additional edit dialogs shown below.

Givens



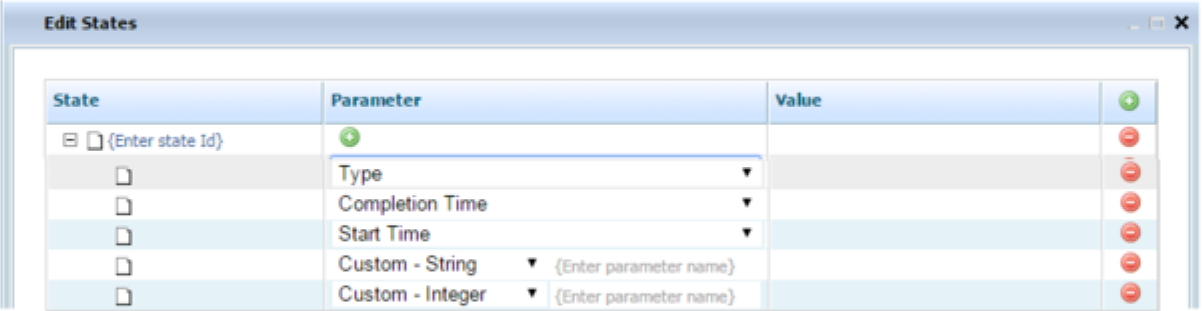
Edit Customer Profile

Customer	Parameter	Value	
⊟ Customer	⊕		
	Title		⊖
	Phone Number		⊖
	Last Name		⊖



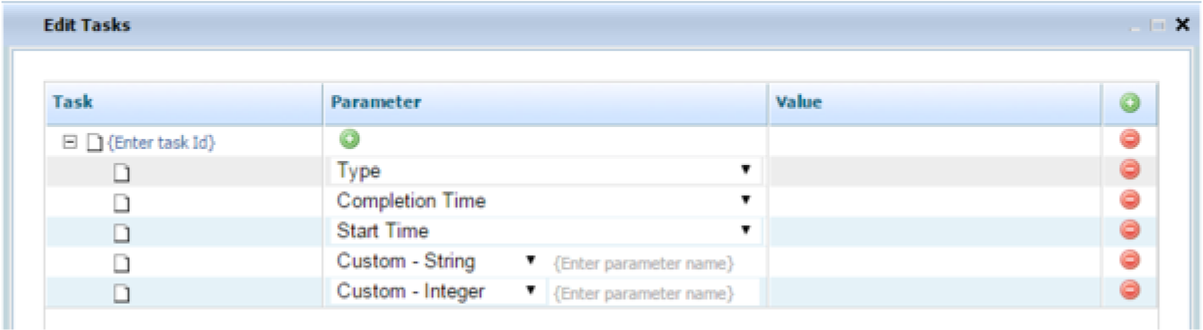
Edit Services

Service	Parameter	Value	⊕
⊟ □ {Enter service Id}	⊕		⊖
□	Type ▾		⊖
□	Completion Time ▾		⊖
□	Start Time ▾		⊖
□	Custom - String ▾ {Enter parameter name}		⊖
□	Custom - Integer ▾ {Enter parameter name}		⊖



Edit States

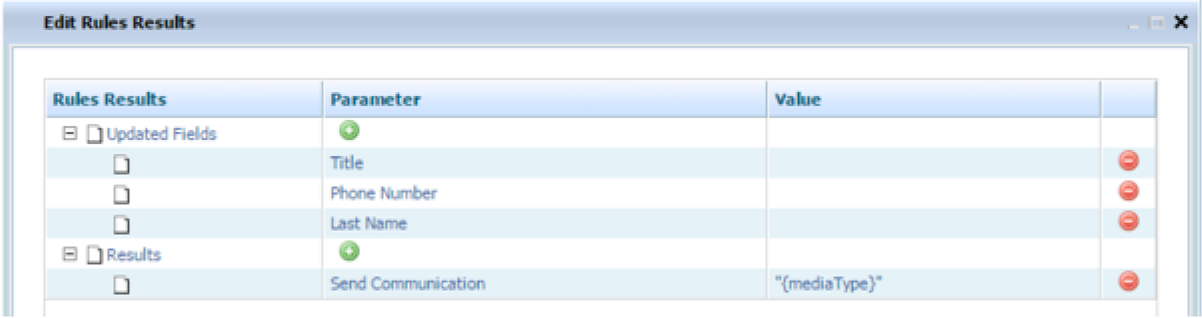
State	Parameter	Value	⊕
⊟ □ {Enter state Id}	⊕		⊖
□	Type ▾		⊖
□	Completion Time ▾		⊖
□	Start Time ▾		⊖
□	Custom - String ▾ {Enter parameter name}		⊖
□	Custom - Integer ▾ {Enter parameter name}		⊖



Edit Tasks

Task	Parameter	Value	⊕
⊟ □ {Enter task Id}	⊕		⊖
□	Type ▾		⊖
□	Completion Time ▾		⊖
□	Start Time ▾		⊖
□	Custom - String ▾ {Enter parameter name}		⊖
□	Custom - Integer ▾ {Enter parameter name}		⊖

Expectations



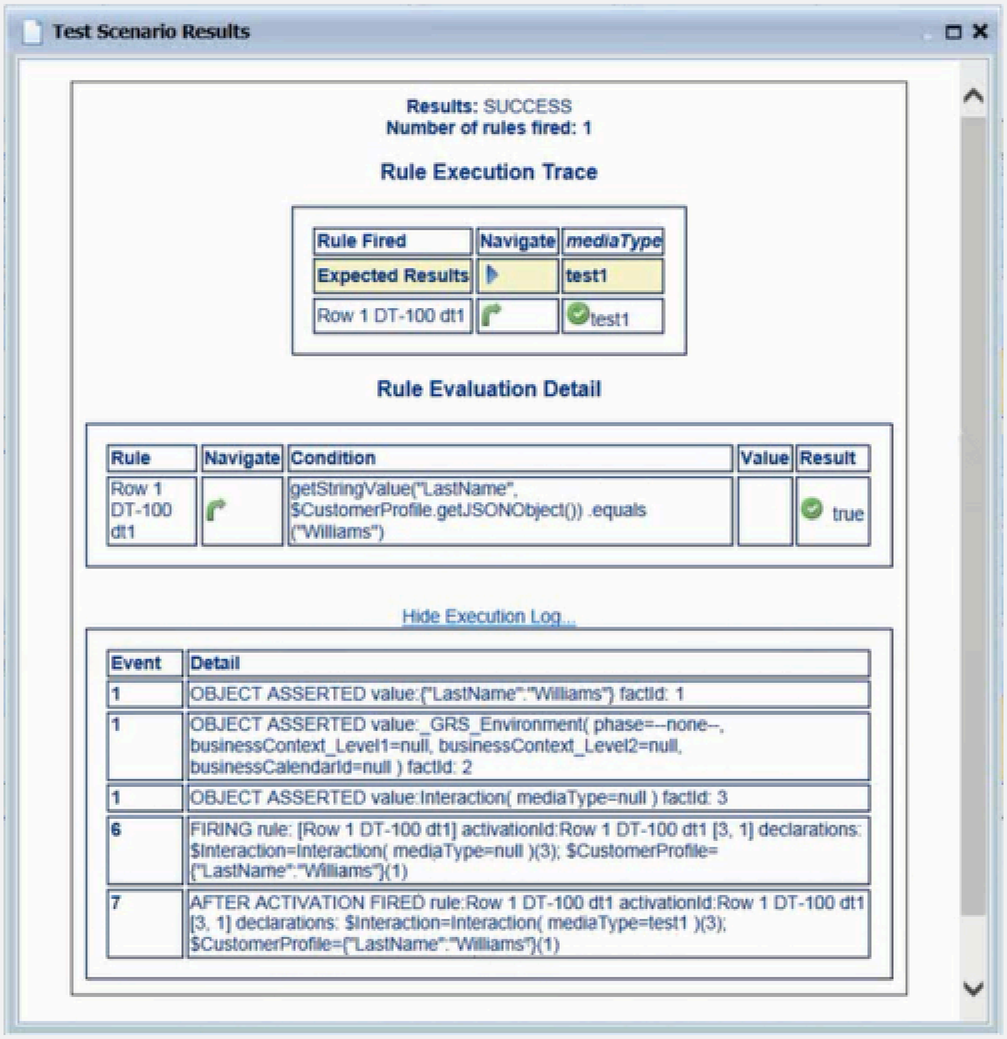
Edit Rules Results

Rules Results	Parameter	Value	
⊟ □ Updated Fields	⊕		
□	Title		⊖
□	Phone Number		⊖
□	Last Name		⊖
⊟ □ Results	⊕		
□	Send Communication	"{mediaType}"	⊖

Test Scenario Enhancement

Test scenario results can now show disqualified rules and details of how individual rule conditions were evaluated, enabling much more detailed debugging during rule development. Before the 8.5.1 release, only the result of rules that fired were returned to the REST client that made the rule evaluation request. Now, details of the rule execution log can be displayed when the GRE `include-rule-evaluation-detail-in-response` option (introduced in 8.5.001) is set to value `true`. Details include rules that were disqualified and why, as well as rules that fired, and other log-level data.

[+] SHOW RULE EXECUTION LOG



Test Scenario Results

Results: SUCCESS
Number of rules fired: 1

Rule Execution Trace

Rule Fired	Navigate	*mediaType*
Expected Results	▶	**test1**
Row 1 DT-100 dt1	↱	✓ test1

Rule Evaluation Detail

Rule	Navigate	Condition	Value	Result
Row 1 DT-100 dt1	↱	getStringValue("LastName", $CustomerProfile.getJSONObject()) .equals ("Williams")		✓ true

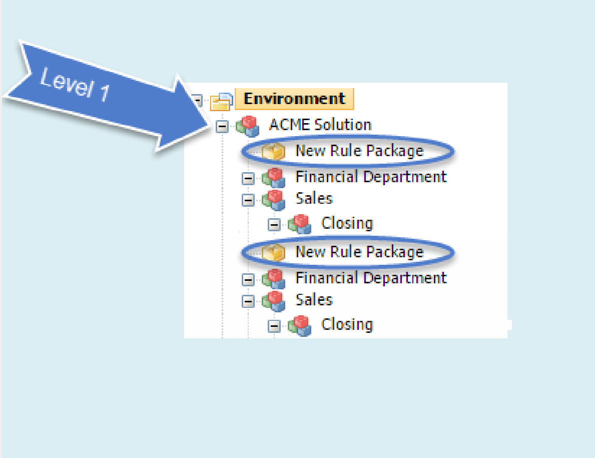
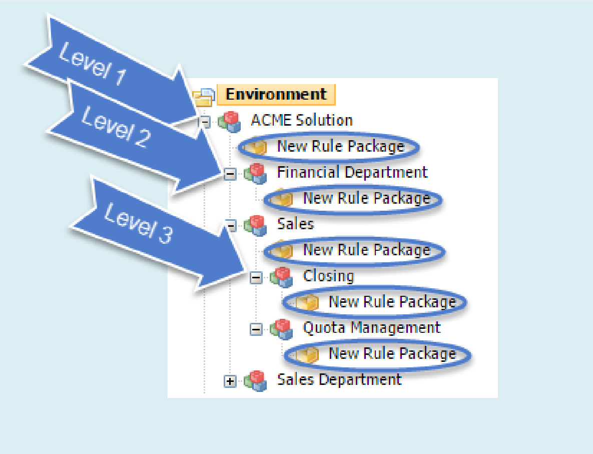
Hide Execution Log...

Event	Detail
1	OBJECT ASSERTED value:{"LastName":"Williams"} factId: 1
1	OBJECT ASSERTED value:_GRS_Environment(phase=--none--, businessContext_Level1=null, businessContext_Level2=null, businessCalendarId=null) factId: 2
1	OBJECT ASSERTED value:Interaction(mediaType=null) factId: 3
6	FIRING rule: [Row 1 DT-100 dt1] activationId:Row 1 DT-100 dt1 [3, 1] declarations: $Interaction=Interaction(mediaType=null)(3); $CustomerProfile={"LastName":"Williams"}(1)
7	AFTER ACTIVATION FIRED rule:Row 1 DT-100 dt1 activationId:Row 1 DT-100 dt1 [3, 1] declarations: $Interaction=Interaction(mediaType=test1)(3); $CustomerProfile={"LastName":"Williams"}(1)

Nested Solution Business Hierarchy

In release 8.5.1 of Genesys Rules Authoring tool, if you have permission to create a new rule (`Rule Package - Create`) you can now add a new Rule Package at any node in the business hierarchy (a *nested* solution), rather than just at the first level.

[+] FULL DESCRIPTION

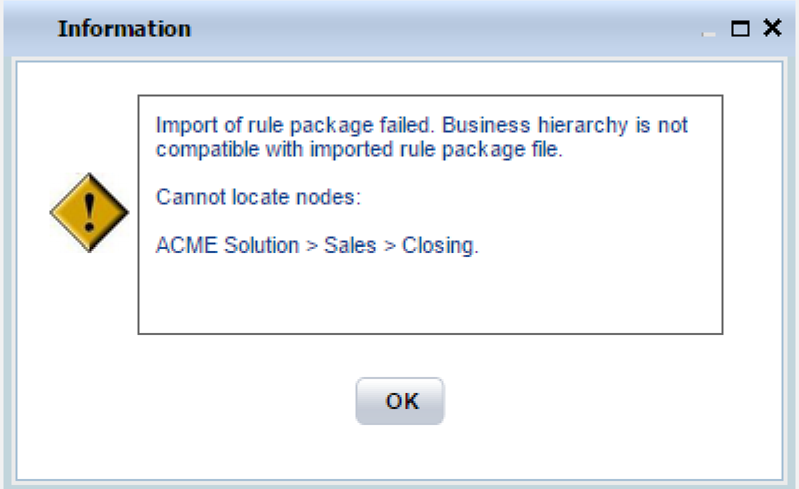
Non-Nested Solution	Nested Solution
 Level 1 Environment ACME Solution New Rule Package Financial Department Sales Closing New Rule Package Financial Department Sales Closing	 Level 1 Level 2 Level 3 Environment ACME Solution New Rule Package Financial Department New Rule Package Sales New Rule Package Closing New Rule Package Quota Management New Rule Package Sales Department

This means that:

- Your business hierarchy can be more easily organized.
- The need for lots of duplication and repetition at the Solution level in more complex business hierarchies is now removed.
- Individual users can be restricted using Role-Based Access Control to specific sub-nodes (for example, Departments and Processes).

Importing Rule Packages

Because rules can be associated with sub-nodes in a nested hierarchy, when a rule package is imported, GRAT ensures that the business structure is compatible, and prevents an import if it is not. If GRAT finds an incompatibility, an error such as the following is displayed:



Information

Import of rule package failed. Business hierarchy is not compatible with imported rule package file.

Cannot locate nodes:

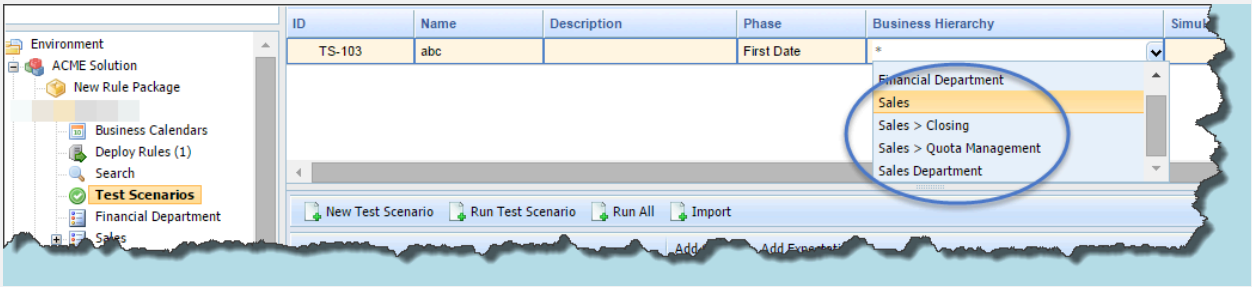
ACME Solution > Sales > Closing.

Important

Even if the **Auto-create business hierarchy during import** button is selected, GRAT prevents the same node name from being created anywhere in the hierarchy—uniqueness of business node names across the entire hierarchy is still enforced.

Test Scenarios

For Test Scenarios, the Business Hierarchy drop-down displays the relative path underneath the selected rule package:



ID	Name	Description	Phase	Business Hierarchy	Simul
TS-103	abc		First Date	*	

Deleting Business Nodes

Be careful never to delete any business structure nodes that contain active rule packages or rules, without first backing up the rule packages as XML files. While it is OK to add new nodes, or to "rename" nodes, proper permissions should be set up to prevent a GA/GAX user from accidentally deleting nodes which could cause rule packages / rules to become unreachable.

Enabling and Disabling the Feature

Because some customers might want to restrict their users to creating rule packages only under the **Solution** node, in release 8.5.100.21 a new configuration option—`enable-nested-solutions`—has been implemented to allow users to enable or disable this feature. Disabling this feature is recommended for iWD users.

- Option name—`enable-nested-solutions`
- Valid values—`true/false`
- Default value—`true`
- Description—Controls whether users can create new rule packages under any node in the hierarchy. For iWD, it is recommended to set this option to `false`.

Unloading from Memory of Unused Rule Packages

If a rule package is not used for a defined period of time, it can now be automatically unloaded from memory. A new configuration option controls this behavior:

- `unload-inactive-package-timeout`
- Value—The time (in minutes) for an inactive package to remain loaded before it is automatically unloaded.
- Default—No timeout.

If the option is not specified, then packages are loaded in GRE with no timeout. 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.

Enhanced cross-browser security features

Browser security has been improved to eliminate cross-site request forgery.

Platform/Database Support Changes

- **Additional Platform Support**
  - Java 7
  - Windows 2012
  - Red Hat Enterprise Linux 7 64-bit native
- **Discontinued Platform Support**
  - Java 6
  - Windows 2003
- **Additional database support**
  - Oracle 12c
  - MS SQL Server 2012
  - PostGRESQL 9.4

New in 8.5.0

New Features in 8.5.0 (new document)