



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 Overview

Rule Evaluation

Rule Evaluation

After a rule package has been deployed to GRE or to a cluster of GRE engines, it can be invoked by a client.

Rule Evaluation Order

Within all the rules that are configured for a particular phase, GRE will evaluate rules in this order:

- Global rules (package-level)
- Department rules
- Process rules

To ensure that GRE performs its evaluation in this order, the `sequential-mode` option (in the `settings` section in the GRE Application) must be set to `false`. This is the default setting.

To ensure that the GRE evaluates all rules within a particular phase, within a particular node of the business structure hierarchy, the `group-by-level` option (in the `settings` section in the GRAT server Application) must be set to `true`. This is the default setting.

Within the rules that are defined at a particular node of the business structure hierarchy (for example, Department), GRE will follow the salience (order) that is defined in the rule package. This is controlled by the rules author, who can move rules up and down in the GRAT using the arrow button on the right side of the rule.

Controlling Rule Evaluation Order

Within a set of rules for a particular phase, at a particular node in the business structure hierarchy, an action of one rule can influence a condition in a subsequent rule. To do this, it is necessary to include the `update($data)` parameter; in the Rule Language Mapping of the rule action or as a separate rule action. See the following example.

Example: Setting Rules at a Specific Phase

If you had one classification rule at the Department level that set the priority of a task to 100 and a second classification rule at the Department level that will increase the priority by 90, but only if the priority has already been set to 100:

The first classification rule would look like this:

WHEN

Priority is not equal to 100

THEN

Set Priority to 100

Update

The second classification rule would look like this:

WHEN

Priority is equal to 100

THEN

Increase Priority 90

In the first classification rule, it is important to specify that the priority be set to 100 only when it is not already equal to 100. This is to avoid a problem where GRE might get into a loop during rule evaluation. Also in the second classification rule, there is a rule action called Update. The Rule Language Mapping for this rule action is: `update($data);`

You could also modify the Rule Language Mapping of the Set Priority rule action, to add `update($data);` to the end of the Rule Language Mapping. For example:

```
setIntValue("Priority", {priority}, $data); update ($data);
```

In the latter case, however, you would want to create a new version of the Set Priority rule action and give it a different Language Expression to help differentiate it, such as `Set extended Priority`.

For more detailed information, see the working example in the **Deployment Guide** (new document).