



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.

iWD Deployment Guide

Rule Evaluation by the IWD Business Process

12/16/2025

Rule Evaluation by the IWD Business Process

Rule Evaluation

After a rule package has been deployed to a Genesys Rules Engine, it can be invoked by a client. The standard client that invokes the Genesys Rules Engine is the iWD business process (IWDBP). For information about other ways to invoke rule packages, see the [Genesys Rules System Deployment Guide](#).

Important

There are two separate business processes, one for [Genesys IRD](#) and one for [Genesys Composer/ORS](#).

It is important to understand how rules are evaluated, when an evaluation is requested from the IWDBP business process. IWDBP has the following two routing strategies that invoke rule evaluation from the Genesys Rules Engine.

- Classification—Calls the Genesys Rules Engine and requests an evaluation of all rules for the iWD Solution's associated rule package that belong to the Classification phase.
- Prioritization—Calls the Genesys Rules Engine and requests an evaluation of all rules that belong to the Prioritization phase.

Associating the iWD Solution with a Rule Package and Rules Engine

To ensure the IWDBP business process invokes the correct Genesys Rules Engine and the correct rule package, two List Objects must be properly configured:

- Iwd_Esp_List
- Iwd_Package_List

Important

Make sure that both of these List Objects are correctly configured, otherwise IWDBP will not work.

Iwd_Esp_List

The IWDBP uses the data from the Iwd_Esp_List List Object to correlate the IWD_SolutionId to the name of the Genesys Rules Engine (or cluster).

Important

Before 8.5.1, the IWDBP used the Business Context Management Service (BCMS)—an External Service Protocol (ESP)— to communicate with the Genesys Rules Engine. Since 8.5.1, BCMS functionality has moved into the Genesys Rules Engine.

Iwd_Esp_List is also used, optionally, to correlate the IWD_SolutionId to the name of a Genesys Universal Contact Server (UCS) that is connected to Interaction Server. If this association is configured in Iwd_Esp_List, the business logic in IWDBP will update the interaction history in the UCS database to mark the interaction as done (the value of the **Status** column in the **Interaction** table in the UCS database will be set to 3) when the interaction enters one of these queues:

- iWD_Completed (or iwd_bp_comp.Main.iWD_Completed for Composer/ORS)
- iWD_Canceled (or iwd_bp_comp.Main.iWD_Canceled for Composer/ORS)
- iWD_Rejected (or iwd_bp_comp.Main.iWD_Rejected for Composer/ORS)

Iwd_Package_List

The Iwd_Package_List List Object is used to correlate the IWD Solution ID (IWD_SolutionId) to the name of the rule package that will be evaluated when requests are made to the Genesys Rules Engine from the IWDBP business process.

Iwd_Package_List contains a single list called, RulePackageList. Create a new key/value pair for each iWD Solution that you have configured under your Configuration Server tenant, where the key or option is the IWD Solution ID and the value is the Package Name of the rules package. For multiple rule packages, specify each Package Name separated by a comma—for example; rule.package1, rule.package2, rule.package3

Important

The user configures two names for a rule package in GRAT. There is the **Package Name** and the **Business Name**. The value that the user enters in the RulePackageList list, must reference the Package Name (not the Business Name) of the rule package.

Rule Evaluation Order

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

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

To ensure the Genesys Rules Engine performs its evaluation in this order, the **sequential-mode** option (in the **[settings]** section in the Genesys Rules Engine Application) must be set to `false`. This is the default setting.

To ensure the Genesys Rules Engine 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 Genesys Rules Authoring Tool 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), the Genesys Rules Engine 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 Genesys Rules Authoring Tool. In the example in Controlling Rule Evaluation Order (below), Rule-265 will be evaluated prior to Rule-294. If the rules author preferred Rule-294 to be evaluated first, they could move that rule up in the evaluation order by 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.

{[NoteFormat]Rules authors can change the sequence of rows in a decision table to permit more sophisticated outcomes. So, you can specify whether the the rows in a decision table are processed top-down or bottom-up by configuring an option (from GRAT 8.5.0) called `evaluate-decision-table-rows-top-down`.

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 the Genesys Rules Engine might get into a loop during rule evaluation.

Also in the first 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.