



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 Authoring for iWD

12/22/2025

Contents

- [1 Rule Authoring for iWD](#)
 - [1.1 Login Permissions](#)
 - [1.2 Configuring iWD Tenant and User for GRAT Access](#)
 - [1.3 Configuring GRAT Access Control to IWDBP](#)
 - [1.4 Creating a New Rules Package](#)
 - [1.5 Rule Levels and Types](#)
 - [1.6 Business Calendars](#)
 - [1.7 List of Solution's Business Calendar](#)
 - [1.8 Rule Deployment](#)

Rule Authoring for iWD

Rule authoring for iWD is done through the Genesys Rules Authoring Tool (GRAT). This section describes general information about using GRAT for iWD, and how to use it for creating decision tables, linear rules, and business calendars.

Login Permissions

The User permissions to the various capabilities of the GRAT are controlled by Role-Based Access Control, which is configured through Genesys Administrator (if you are not using Genesys Administrator, you will have full access to the GRAT).

See **Role-Based Access Control** in the Genesys Rules System Deployment Guide.

Configuring iWD Tenant and User for GRAT Access

To complete the necessary configuration of an iWD managed tenant and an iWD user's security role, so that the Rules Authoring link appears on the iWD Manager navigation bar, properly launch the Genesys Rules Authoring Tool web application, and log in the user, complete the following procedure.

Procedure

1. Log in to iWD Manager as a user with full administrative privileges.
2. From the **Tenant** drop-down list, select your iWD-managed tenant.
3. Select the **Profile** link.
4. Under Genesys Rules Authoring Tool URL, enter a valid URL for the Genesys Rules Authoring Tool.
5. Test this URL in a separate browser window to ensure that it brings up the login screen.

The default syntax for the URL should be:

`http://<host>:<port>/genesys-rules-authoring`

where;

- <host>—Is your application server host name.
- <port>—Is the listening port of your application server (usually 8080 by default).
- genesys-rules-authoring—Is the name of the Genesys Rules Authoring Tool web application that is deployed on your application server. (This will be the default name of that web application unless you have modified it.)

6. Click **Save**.
7. Select the **Security Policy** link.
8. Select the name of an existing Security Role, to which you want to give access to the Genesys Rules Authoring Tool, or create a new one.
9. Scroll down to the **Application Permissions** section.
10. Under the **Run** column, check the checkbox for the Rules Authoring permission.
11. Click **Save**.

Configuring GRAT Access Control to IWDBP

When you launch GRAT from iWD Manager, the active tenant will be the Configuration Server tenant that is linked to the iWD managed tenant you were working with in iWD Manager. The GRAT navigation tree will display the iWD Solutions that are configured under the tenant. If you don't see any Solutions in the navigation tree, but you have one or more Solutions configured in iWD Manager, you must ensure that the user has permission to the appropriate folders under the Business Structure folder in Configuration Server.

Important

The Business Structure folder is created during the migration process from an earlier version of iWD to iWD 9.0.

Procedure

To configure access control for the Genesys Rules Authoring Tool user, to allow him/her to see the appropriate iWD Solutions when logged into GRAT, do the following:

1. Log in to Genesys Administrator or Genesys Configuration Manager.
2. Navigate to the tenant that maps to the iWD managed tenant with which you are working.
3. Locate the Business Structure folder (in Genesys Administrator, go to **Environment > Business Units/Sites**).
4. In the Business Structure folder, locate the iWD Solution folder to which you want the Genesys Rules Authoring Tool user to have access.
5. If you are using Genesys Configuration Server:
 - a. On the **Security** tab of that iWD Solution, configure at least Read access for the Genesys Rules Authoring Tool user, or for at least one Configuration Server Access Group to which that user belongs.

If you are using Genesys Administrator:

- a. Highlight the iWD Solution folder and select **Edit**.
- b. Select the **Permissions** tab.

- c. Use either **Add Access Group** or **Add User** to configure at least Read access to this iWD Solution, for the necessary Genesys Rules Authoring Tool user or his Access Group(s).

Important

When you add the user or Access Group, you can decide whether or not to propagate the permissions. If you do not propagate the permissions, you will need to explicitly add permissions to the Department(s) and Process(es) under the iWD Solution, to which you want the user or access group to have access.

4. To grant access to any additional iWD Solutions, repeat Steps 4 to 6.

Creating a New Rules Package

Summary

The rule package is the parent object for all the business rules for an iWD Solution.

1. Log in to iWD Manager and launch the Genesys Rules Authoring Tool.
2. In the navigation tree, expand the Solution with which you want to work.
3. Select **New Rule Package**.
4. On the right side of the screen, fill in the properties for the rule package, such as.
 - **Package Name**—Used internally and primarily in the Configuration Server List Object that will be read in the iWD business process, so the business process knows which rule package should be evaluated by the Genesys Rules Engine. There are some reserved keywords that must not be used when you name the rule package. See [Creating Rule Packages](#) topic in the GRAT Help.
 - **Business Name**—Enter any user-friendly name you wish to provide to identify the rule package.
 - **Package Type**—Select iWD. Selecting iWD will ensure that the list of available rule templates of type iWD will be displayed.
 - **Rule Package** (Optional)—Enter a description for the rule package.
 - **Template** section—Select one or more rule templates. This will determine which collection of rule actions and conditions will be made available to the business rules authors who are responsible for creating rules in this package.
5. Save the rule package.

The new rule package will appear in the navigation tree on the left side of the pane, displaying all the Departments and Processes underneath the package, based on the access control of the user.

Next Steps

After a rule package has been created, you can start creating business rules. For the descriptions of

the various types of rules and rule objects, see the Rules Overview tab on this page.

Important

The list of rule templates that are available to the person creating the rule package will depend on that user's access to the Script objects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type Data Collection, for each rule template published to the rules repository. The access control defined on the **Security** tab of that Script object will determine which users and access groups can use that template in a rule package.

Rule Levels and Types

In the Genesys Rules Authoring Tool, there are three levels at which business rules can be created:

- Rule Package (referred to as Global Rules)
- Department
- Process

When the appropriate node is selected on the rule package tree, you can then select the **Rules** tab to view or edit the rules for that level of the business structure. Rules are presented in a list, with an associated phase. The order of the rules is relevant, because they will be evaluated, within a particular phase, in the same order as they appear. You can change the order of rules by clicking the up and down buttons. The logic of a particular rule can be expressed as either a linear rule or a decision table.

Any iWD extended or custom attribute can be read or updated by business rule conditions or actions, respectively.

Global Rules

Rules that are created at the rule-package level are also known as *Global Rules*. Global rules enable you to specify rules that will apply to the entire iWD Solution. For example, they enable you to configure rules that classify or prioritize all tasks globally, instead of at a lower level of the business structure. Global rules are applied before any other rules.

This means that each rule phase (classification and prioritization) is triggered from within the IWDBP business process in the following sequence:

- Global rules
- Department rules
- Process rules

Important

A classification rule must be configured that assigns the task to a Process, or the Process can be assigned directly in the `createTask` message when the task is captured by a capture point. If a Process is not assigned to the task in either of these two methods, the task status will be changed to `Error Held`. For example, a department can be assigned in a global rule, followed by assigning a process at the department level. Alternatively, a process might be directly assigned in a global rule. If a process is assigned in both global rules and department rules, based on the same conditions, the department rules will override.

Capture Point Rules

You can use the rule condition `Capture Point is...` from the iWD Standard Rules Template, at the Global Rules level. Classification rules are not created at the Capture Point level.

Linear Rules

A linear rule is a business rule that has a set of conditions (when) and actions (then), and is used for a simple (linear) business case. For example, when a task is due in 1 to 8 hours, set the task's priority to 20.

To specify a linear rule for a simple business case:

1. In Genesys Rules Authoring Tool, expand the rule package and select a node at which you want to create the rule.
2. On the right side of Genesys Rules Authoring Tool panel, on the **Rules** tab, click **New Linear Rule**.
3. Enter a **Name** for the rule that identifies it.
4. Optionally, enter a **Description** for the rule.
5. Select the **Phase** in which to apply the rule.

You can choose `classification` or `prioritization`.

6. Optionally, assign a **Calendar** (business calendar) to the rule.
7. If required, set the **Start Date** and **End Date**.

Important

If **Start Date** and **End Date** are left empty the rule activation period is unconstrained.

8. From the **Add Condition** combo box, select one or several conditions for the rule.

By default, conditions are concatenated by using the logical `AND` operator.

However, you can select other functions (for example, “or” or “and not”) from the **Add Condition** list, and you can also select multiple conditions and use the **Group** (or **Ungroup**) functions.

Important

Available actions and conditions are defined in rules templates. If a new action or condition type is required, add it to either an existing rules template or a new template. Genesys strongly recommends using a custom rule package for all new rule actions or conditions.

9. Select one or several actions for the rule from the **Add Action** combo box.
10. To save the specified rule, click **Save**.

You can optionally enter a check-in comment, which will be available later on the **Audit Trail** tab.

After you have created a linear rule, you can create additional linear rules or decision tables, or deploy your rule package.

Important

The available conditions and actions that are presented to the rules author is driven by the set of rule templates to which the user has access. This depends on the user's permissions to the Script objects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type **Data Collection**, for each rule template published to the rules repository. The access control that is defined on the **Security** tab of that Script object will determine which users can access which rule templates, which will control the final list of rule actions and conditions presented.

Decision Tables

Decision tables have a set of the same conditions (when) and actions (then), but have different parameters and are used for a complex (structured) business case. Use decision tables to avoid dozens of linear rules in the system. Defining a decision table is similar to defining a linear rule.

To specify a decision table rule for a complex business case:

1. On the **Rules** tab, click **New Decision Table**.
2. Specify the rule **Name**, **Description**, **Phase**, **Calendar**, **Start Date**, and **End Date**, if required.
3. From the **Add Condition** combo box, select zero or more conditions for the rule.
4. From the **Add Action** combo box, select one or more actions for the rule.
5. To add a new row to the decision table, select the green plus (+) icon to the right of the right-most rule

action.

6. Enter the required parameters for each rule condition and action.
7. Optionally, enter a **Name** for the row.
8. To add a new row to the decision table, select the green plus (+) icon to the right of the right-most rule action. until you have set all of the required cases.

The result is a table in which the columns represent rule conditions and actions and the rows contain real conditions and action parameter values.

Important

To delete a condition, action, or rule, click the delete icon (-), which is located on the right side of the column (for conditions and actions), or on the right side of the row (for rules).

9. To save the specified rule, click **Save**.
10. Optionally, enter a check-in comment, which will be available later on the **Audit Trail** tab.

After you have created a decision table, you can create additional decision tables or linear rules, or deploy your rule package.

Important

The available conditions and actions that are presented to the rules author is driven by the set of rule templates to which the user has access. This depends on the user's permissions to the Scriptobjects that represent each template. In Configuration Server, under each tenant, there is a **Script** folder that contains a subfolder called **Template Access Control**. In that folder, there is a script of type Data Collection, for each rule template published to the rules repository. The access control that is defined on the **Security** tab of that Script object will determine which users can access which rule templates, which will control the final list of rule actions and conditions presented.

Business Calendars

A business calendar is a set of rules that define working days and hours, and holidays that are applicable for the business. Business calendars can be used in iWD rules to perform date and time calculations, taking into account the working schedule of the business. Business calendars can be assigned once (for example, at the Global Rule level), or can be assigned dynamically in a rule when needed.

To create new (or to manage existing) business calendars, launch the Genesys Rules Authoring Tool and expand a rule package. Select the **Business Calendars** entry in the tree.

List of Solution's Business Calendar

The right side of the Genesys Rules Authoring Tool is split horizontally into two panes. The upper displays a list of a rule package's business calendars. The **New Calendar** button that is below this list is used to create a new business calendar. You can delete business calendars by clicking the delete button that is on the right side of the business calendar in the business calendar list. When a calendar list is selected, the lower pane displays the attributes of the selected business calendar.

Business Calendar Attributes

Business calendars consist of a set of standard mandatory attributes and optional business calendar rules. See **Business Calendars** in the Genesys Rules Authoring Tool Help.

Using Business Calendars in iWD Rules

After business calendars are defined, you can use them in rules. Business calendars must be assigned to a task before any business calendar-related calculations can be performed on task values. A Calendar can be assigned at the parent rule level, or to an individual rule. Only one calendar can be assigned to a task at a time, so a calendar can be assigned by one rule and then, overwritten by a later rule.

iWD Rules Conditions and Actions Using Business Calendars

Condition/ Action	Parameters	Description
Assign business calendar	{businessCalendar}	Assigns a business calendar to a task. A business calendar must be assigned to a task, before any business calendar-related calculations can be performed on task values. A drop-down list displays a list of business calendars that are defined for the rule package.
Is Working Day	N/A	Calculates whether the current date/time is a working day, according to the assigned business calendar.
Is Working Time	N/A	Calculates whether the current date/time is working time, according to the assigned business calendar.
Reprioritize after	{period} working {periodType}	Sets a task's re-prioritization date/time to value that is calculated, based on current date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies working minutes, hours, or days.
Task Due in	{period} working {periodType}	Sets a task's due date/time to a value that is calculated, based on the task's creation date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies working minutes, hours, or days.
Activate task in	{period} working {periodType}	Sets a task's activation date/time to a value that is calculated, based on task's creation date/time, the task's business calendar, and specified parameters. {period} is a numeric value, and {periodType} specifies working <i>minutes, hours, or days</i> .
Task expires in	{period} working {periodType}	Sets a task's expiration date/time to a value that is calculated, based on task's creation date/time, the task's business calendar, and specified parameters. {period} is a numeric

Condition/ Action	Parameters	Description
		value, and {periodType} specifies working minutes, hours, or days.

Working with the businessCalendarService object

Business calendar logic is exposed to rule expressions via the businessCalendarService object. Each method of this object accepts business calendar ID as the first parameter. Business calendar ID is usually stored in businessCalendarId attribute of the task and can be accessed from rules expressions like this:

```
getStringValue("IWD_businessCalendarId", $data)
```

businessCalendarService Method	Description	Example(s)
Date add(String businessCalendarName, Date addToThis, String type, int amount)	Adds working time to the timestamp according to the business calendar. Type parameter should be either "minutes", "hours" or "days". Amount parameter represents number of time units of the given type. It can also be negative. The method returns the resulting timestamp. When adding days, timestamp will be set to end of working day, for example when adding 1 day it will result in the end of the next working day.	businessCalendarService.add(getStringValue("IWD_businessCalendarId", \$data), getDTValue("IWD_createdDateTime", \$data), "hours", 4)); // get timestamp of 4 working hours after the task was created
boolean dayIsWorkingDay(String businessCalendarName, Date day)	Returns true if the day of the given timestamp is a working day (has any working hours).	businessCalendarService.dayIsWorkingDay(getStringValue("IWD_businessCalendarId", \$data), evo.common.TimeHelper.getUTCtime()); // is today a working day businessCalendarService.dayIsWorkingDay(getStringValue("IWD_businessCalendarId", \$data), getDTValue("IWD_dueDateTime", \$data)); // does IWD_dueDateTime fall on a working day?
boolean timeIsWorkingTime(String businessCalendarName, Date time)	Returns true if the given timestamp is a working time.	businessCalendarService.timeIsWorkingTime(getStringValue("IWD_businessCalendarId", \$data), evo.common.TimeHelper.getUTCtime()); // is

businessCalendarService Method	Description	Example(s)
		it working time right now <pre>businessCalendarService.timesWorkingTime(getStringValue("IWD_businessCalendarName", \$data), getDTValue("IWD_dueDateTime", \$data)); // does IWD_dueDateTime fall within working hours?</pre>
<pre>int diffWorkingDays(String businessCalendarName, Date date1, Date date2)</pre>	Returns the number of working days between two given timestamps.	<pre>businessCalendarService.diffWorkingDays(getStringValue("IWD_businessCalendarName", \$data), getDTValue("IWD_createdDateTime", \$data), evo.common.TimeHelper.getUTCtime()); // how many working days have passed since creation of the task?</pre>
<pre>int diffWorkingHours(String businessCalendarName, Date date1, Date date2)</pre>	Returns the number of working hours between two given timestamps.	<pre>businessCalendarService.diffWorkingHours(getStringValue("IWD_businessCalendarName", \$data), getDTValue("IWD_createdDateTime", \$data), evo.common.TimeHelper.getUTCtime()); // how many working hours have passed since creation of the task?</pre>
<pre>int diffWorkingMinutes(String businessCalendarName, Date date1, Date date2)</pre>	Returns the number of working minutes between two given timestamps	<pre>businessCalendarService.diffWorkingMinutes(getStringValue("IWD_businessCalendarName", \$data), getDTValue("IWD_createdDateTime", \$data), evo.common.TimeHelper.getUTCtime()); // how many working minutes have passed since creation of the task?</pre>
<pre>Date beginningOfWorkingDay(String businessCalendarName, Date utcDate)</pre>	Returns timestamp that contains opening hours for the given date. The result will be a composite timestamp, where the date part is taken from the given parameter, but the time part contains the beginning of working hours for the given date. If date is not a working day, a null value will be returned.	<pre>businessCalendarService.beginningOfWorkingDay(getStringValue("IWD_businessCalendarName", \$data), evo.common.TimeHelper.getUTCtime()); // get today's opening hours</pre>
<pre>Date endOfWorkingDay(String businessCalendarName, Date utcDate)</pre>	Returns timestamp that contains closing hours for the given date. The result will be a composite timestamp, where the date part is taken from the given parameter, but the time part contains the end of working hours for the given date. If date is	<pre>businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarName", \$data), evo.common.TimeHelper.getUTCtime()); // get today's closing hours</pre>

businessCalendarService Method	Description	Example(s)
	not a working day, a null value will be returned.	

How to retrieve today's opening hours?

```
businessCalendarService.beginningOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime());
```

How to retrieve today's closing hours?

```
businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime());
```

How many hours left till closing hours?

```
businessCalendarService.diffWorkingHours(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime(),  
    businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime()));
```

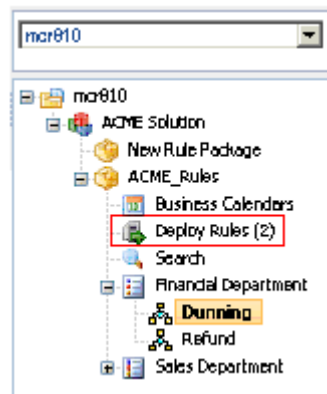
Rule Deployment

Once you have created all the necessary rules in a rule package, it is time to deploy the rule package to a single instance of the Genesys Rules Engine or a cluster of GREs. Once the package is deployed, it can be invoked by a client such as the iWD business process (IWDBP).

Rule package deployment is done through the Genesys Rules Authoring Tool, and is independent of the iWD Solution deployment in iWD Manager. Thus, you can deploy a new version of a rule package for an iWD Solution without having to redeploy the entire Solution.

When a rule is created or edited and it has not been deployed, there is a checkmark in the **Pending Deployment** column of the rule.

In addition, when a rule package requires deployment or redeployment, there will be a visual indication next to the **Deploy Rules** node in the **Genesys Rules Authoring Tool** navigation tree, under the rule package itself.



For more information about rule package deployment, including scheduled deployment and deployment history, and snapshots, see [Deploying/Undeploying Rules Packages](#) (new document).