

# iWD and GRS

intelligent Workload Distribution 8.1.0

2/3/2022

# Table of Contents

# IWD and the Genesys Rules System

These topics describe how iWD interoperates with the Genesys Rules System.

> **Tip**
>
> The following documents will be helpful:
>
> - **The Genesys Rules System Overview**
> - **The Genesys Rules System deployment summary**
>
> Unless otherwise stated, all the procedures described in these topics require a fully installed iWD Manager and Genesys Rules System as a prerequisite.

## IWD and GRS Overview

Starting with release 8.1.0, the iWD solution no longer has its own embedded rules engine service, and rules development and authoring user interfaces are no longer integrated into iWD Manager. Instead, iWD now uses the Genesys Rules System to provide all of this functionality.

Genesys Rules System provides all the business rules functionality for the Genesys intelligent Workload Distribution (iWD) solution, a business application for dynamically prioritizing the distribution of work tasks to the people who are best suited to handle them.

For information about how to migrate your existing rule templates and rules, see the Genesys System Migration Guide.

Rule templates are created in the Genesys Rules Development Tool and the templates are published to the rules repository. Users then use the Genesys Rules Authoring Tool to create a rule package that incorporates one or more rule templates. The Rules Authoring Tool is also where users create new rule packages that incorporate rule templates, author rules inside the rule package based on the rule templates, validate the rules, and then deploy their rule package to the Genesys Rules Engine. iWD provides a Standard Rules Template for use with the Genesys Rules System, and the Genesys Rules Authoring Tool (GRAT) can be launched from iWD Manager without the need for separate user authentication.

Client applications such as the iWD business process (IWDBP) then make requests to the Genesys Rules Engine to have rules in the rule package evaluated at various decision points in a task's lifecycle.

The Genesys Rules System is not only used by Genesys iWD. It is also used by other Genesys solutions, including Genesys Conversation Manager. As such, there are some objects that must be properly configured for iWD, when working with the Genesys Rules System. These include:

- Configuring the proper rule template type when creating a new template

- Configuring the proper rule package type when creating a rule package
- Proper use of rule phases.

These objects are described in more detail in these topics.

# Rule Templates

Rule templates for iWD are created in the Genesys Rules Development Tool, which is an Eclipse plug-in that can either be installed into a standalone Eclipse application or can be installed into Genesys Composer.

Rule templates are used to define the building blocks that are used by rules authors to build rules for task classification and prioritization at the Global, Department, and Processes levels of the business structure of an iWD Solution.

Normally rule templates are created or modified by IT personnel. Defining new rule conditions or actions sometimes requires a basic knowledge of the Java programming language, although in many cases it is possible to review the syntax of existing rule conditions and actions as a guide.

> **Tip**
>
> For more detail about rule template components, click **here** (opens in a new window).

All iWD rule templates include a fact called `_GRS_Environment`, which must not be modified. Unlike rule templates for other Genesys solutions, it is not necessary to create any additional facts for iWD rules.

> **Important**
>
> Rule conditions and actions are made up of a Language Expression and a Rule Language Mapping. The Language Expression defines what the rule condition or action does, in plain language. It is what the rules author (who is often a business user) will see when constructing a rule. The Rule Language Mapping defines the same Language Expression in Java code. This is the real code that is executed. Rule Language Mapping follows Drools syntax (which is Java with some minor extensions). Regarding available methods, there is a single variable available to rules called `$data`, which represents task attributes as a key-value collection. It is an instance of the `KeyValueCollection` class which is part of Platform SDK. For more information, see the Platform SDK Developer's Guide, (release 7.6 or later).

# The iWD Standard Rules Template

## Overview

iWD comes with an out-of-the-box rule template called the iWD Standard Rules Template, that a user will import into the Genesys Rules Development Tool. This template is the foundation for the rules that are most frequently used in iWD deployments. The template can be used as an example when new rule conditions or actions are required for a particular iWD deployment.

Although the functionality defined in the iWD Standard Rules Template can be extended, Genesys recommends creating one or more custom rule templates for this purpose. One reason for this is that you can use Configuration Server object permissions to control which users can access specific rule templates. For example, you might have one custom rule template for all users of the Genesys Rules Authoring Tool, which includes common rule conditions and actions, to which all users should have access. Then you can have a second custom rule template that contains advanced rule conditions and actions, which you only make accessible to experienced rule authors.

> **Important**
>
> If Genesys provides an updated iWD Standard Rules Template, all custom changes to the previous version will be lost after importing of the updated template.

After you modify an existing rule template, or create a new one, the rule template is published to the Genesys Rules System rules repository. This makes the template available to users of GRAT, so the template can be added to a new rule package, or an existing rule package can be updated with the latest version of a template.

## Importing the Standard Rules Template

# Importing the iWD Standard Rules Template into the Genesys Rules Development tool and publishing it to the Rules repository

## Purpose

To import the iWD Standard Rules Template into the Genesys Rules Development Tool and then publish it to the Genesys Rules System rules repository. This will make the Standard Rules Template available to rules authors, from within the Genesys Rules Authoring Tool.

1. Launch Eclipse (or Composer).
2. Open the Template Development perspective (if it is not already open).
3. Navigate to `Window > Open Perspective > Other > Template Development.`
4. In the `Eclipse` menu, navigate to `File >Import.`
5. In the `Import` dialog box, select Existing Projects into Workspace from within the `General` folder.
6. Click `Next.`
7. Select the `Select root directory` radio button
8. Select `Browse` to browse to the iWD Standard Rules Template folder.

> **Tip**
> The iWD Standard Rules Template will be in the rule `Templates` subdirectory in the folder where the iWD Manager supporting files were installed on your file system. For example, `C:\Program Files\GCTI\iWD Manager\ruleTemplates.`

9. Click `Finish.`

   The iWD Standard Rules Template is now visible in the Eclipse (or Composer) `Project Explorer` view.

10. In the `Project Explorer` view, right-click `iWD_Standard_Rules project` and select `Publish…`
11. Using the `Publish Template Wizard`, select the `iWD_Standard_Rules` template project.
12. Select the `Edit properties` link.
13. In the `Properties` dialog, under `Template Properties`, ensure that `Type = iWD.`
14. For the `Tenant` property, select the Configuration Server tenant to which this rule template will be associated.
15. Click `OK.`
16. Using the `Publish Template Wizard`, select the `iWD_Standard_Rules` template project.
17. Click `Finish.`

The `Publishing` dialog, with a progress indicator, will display `Publish successful` after the template project has been successfully published to the Genesys Rules System repository.

> **Important**
>
> You cannot import the iWD Standard Rules Template project into GRDT if it already exists in the Eclipse workspace. Therefore, if you have a new version of the iWD Standard Rules Template to import, you must first delete the existing one. For this reason, Genesys strongly recommends that you use custom rule templates for new rule conditions or actions, rather than extending the iWD Standard Rules Template.

## Next Steps

If you do not plan to create any custom rule templates at this time, go to the Genesys Rules Authoring Tool to create a rule package. See **Rule Authoring for IWD**. To review information about the parameters, conditions, actions, and functions provided in the iWD Standard Rules Template, see the iWD Standard Rules Template Contents tab on this page.

## iWD Standard Rules Template Contents

## Conditions, Actions, Parameters, and Functions

> **Important**
>
> The ACME sample application (supplied on the software CD) contains sample rules that show how some of the Standard Rules Template conditions and actions can be used in practice. This includes assigning a task to an iWD process, based on specific attributes of the task, such as Product Type, as well as setting priority and business values based, on the iWD process. The sample also includes examples of how to set the reprioritization interval in a rule.

The iWD Standard Rules Template defines the most commonly used rule conditions, actions, parameters, and functions.

# Conditions

| Condition | Explanation |
|---|---|
| Business value is "businessValue_From" to "businessValue_To" | If the business value of the task is between businessValue_From and businessValue_To, then... This rule condition is designed to test inclusive conditions. That is, the "businessValue_From" and "businessValue_To" conditions that are being checked are ">=" and "<=", respectively. This must be understood in order to use this rule condition effectively. |
| Capture point is "capturePoint" | If the capture point is capturePoint, then... This rule condition is designed to be used at the rule package (Global) level, as a way to classify tasks, based on the capture point from which they originated. This is in lieu of the Capture Point-level rules that were available in iWD 8.0 but are not available in the Genesys Rules System. This rule condition requires that the capturePoint rule parameter be properly configured so that GRAT can access the list of Capture Points from the iWD configuration database. |
| Channel is "taskChannels" | If the specified channel of the task is taskChannels, then... This rule condition requires that the taskChannels rule parameter be properly configured so that GRAT can access the list of task channels from the iWD confutation database. |
| Department is "department" | If the specified department of the task is department, then... |
| Due Time is in "periodFrom" to "periodTo" "periodType" | If the task due date/time is between periodFrom and periodTo specified time periodType, then... This rule condition is designed to test conditions that are not inclusive. That is, the "periodFrom" and "periodTo" conditions that are being checked are ">" and "<", respectively. This must be understood in order to use this rule condition effectively. |
| Integer "attribute" "operator" "integerValue" | When the value of a specified custom attribute of type integer is greater than/less than/equal to the specified integerValue, then... |
| Is first prioritization | When the rules being applied are part of the initial prioritization step (the value of the task attribute reprioritizeDateTime is empty), then... |
| Is reprioritization | When the rules being applied are part of the reprioritization step (the value of the task attribute reprioritizeDateTime is not empty), then... |
| Is working day | When today is a working day, then... This condition is evaluated, based on the default business calendar assigned to the rule package or a business calendar that has been explicitly assigned to the task in a previous rule. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |

| Condition | Explanation |
|---|---|
| Is working time | When the current time is working time, then... This condition is evaluated based on the default business calendar assigned to the rule package or a business calendar that has been explicitly assigned to the task in a previous rule. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Media type is `"mediaType"` | When the specified media type of the task is `mediaType`, then... |
| No process selected | If process is not assigned to the task, then... |
| Priority is `"operator" "priority"` | If iWD priority of the task is equal to/not equal to/less than/less than or equal to/greater than/greater than or equal to p `riority`, then... |
| Process is `"process"` | If the task is assigned to `process`, then... |
| String `"attribute" equals "stringValue"` | When the value of a specified custom attribute of type `string` is equal to the specified `stringValue`, then... |
| Task is overdue | When the task is overdue, then... |
| Task overdue in `"period" "periodType"` | Checks if task will be overdue after a given time interval. |

## Actions

| Action | Explanation |
|---|---|
| Activate task in `"period" "periodType"` | Activate the task in the specified period of time, using time unit `periodType` . The activation date/time is used in the IWDBP business process to determine when the task should be moved from the `iWD_Captured` queue into the Prioritization routing strategy. |
| Activate task in `"period" working "periodType"` | Activate the task in the specified period of time, using time unit `periodType`, according to the task's business calendar (working days, working hours and so on). The activation date/time is used in the IWDBP business process to determine when the task should be moved from the `iWD_Captured` queue into the Prioritization routing strategy. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Activate task beginning next working day | Activates the task at the beginning of the next business day. A |

| Action | Explanation |
|---|---|
| | business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Activate task end working day | Activates the task at the end of the working day when the task was received. A business calendar must be assigned to the rule or in a separate rule action before this action can be used. |
| Archive destination `"archive"` | Set the archive destination for the task. Note: This action has been maintained from iWD 8.0 and 8.1.0 but it is no longer necessary to use it, due to changes in the way task archiving works in the out-of-box iWD business process (IWDBP) in iWD 8.1.1. See **Task Archiving** for more information. |
| Assign business calendar `"businessCalendar"` | Assign the specified `businessCalendar` to the task. |
| Assign distribution point `"distributionPoint"` | Assign the specified `distributionPoint` to the task. This rule condition requires that the `distributionPoint` rule parameter be properly configured so that GRAT can access the list of distribution points from the iWD configuration database. See the Deployment Guide for more information. |
| Assign iWD department `"department"` | Assign the task to the specified department. |
| Assign iWD process `"process"` | Assign the task to the specified process. |
| Assign WFM Activity `"wfmActivity"` | Assign the specified Genesys Workforce Management Activity called `wfmActivity` to the task. To ensure GRS can read objects from Genesys Workforce Management, you must configure the following object in the Configuration Manager under specific tenant's `Scripts` folder: `Name` — `WFM Connection` (or other descriptive name) `Type` — `Data Collection`<br><br>Under `Annex` tab, configure the following section: `Section` — `wfm`<br><br>Under this section, configure the following properties: `wfmCfgServerApplName` —CME application name of the WFM Server `wfmCfgServerUserName` —User name that is used to connect to WFM Server `wfmCfgServerPassword` —Password that is used to connect to WFM Server `wfmServerUrl` —URL that is used to connect to WFM Server. The URL must have the following format: `http://<wfm_server_host_name>:<wfm_server_port_number>`<br><br>For example, if the WFM Server is running on a host named `WFMHost001` on port `7023`, the URL would be: `http://wfmhost001:7023` |
| Increase iWD priority `"priority"` | Increase the iWD priority of the task by the specified amount defined in `priority`. |
| Multiply business value `"multiplier"` | Multiply the business value of the task by the specified |

| Action | Explanation |
| --- | --- |
| | multiplier. |
| Reprioritize after "period" "periodType" | Reapply prioritization rules to the task after the specified period of time, using time unit periodType from the current time. |
| Reprioritize after "period" working "periodType" | Reapply prioritization rules to the task after the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours, and so on). A business calendar must be assigned to the rule package, or directly to the task, before this action can be used. |
| Request agent "agent" | Request a specific agent for the task. |
| Request skill "skill" | Request a specific skill for the task. |
| Request agent group "agentGroup" | Specify which agentGroup is required to process the task. |
| Request place group "placeGroup" | Specify which placeGroup is required to process the task. |
| Set activation date from "customAttribute" | Set the activation date/time of the task from the specified customAttribute of the task. The activation date/time is used in the IWDBP business process to determine when the task should be moved from the iWD_Captured queue into the Prioritization routing strategy. |
| Set activation time "time" | Set the time when the task will be activated. The activation date/time is used in the IWDBP business process to determine when the task should be moved from the iWD_Captured queue into the Prioritization routing strategy. The value for "time" must be entered by the rules author in UTC, because it will be attached to the task in the IWD_activationDateTime attribute. The value of this attribute will be displayed to users in the Global Task List based on their operating system's time zone settings. Entering time values in UTC is not intuitive for a business user, therefore, if there is a business reason to attach a specific activation time to a task, you might want to create a custom rule condition that displays the local time (considering the time zone of the business, for example) in the Language Expression, while assigning the corresponding value in UTC, in the Rule Language Mapping. Take care with date changes when specifying the activation time in a local time zone. For example, 23:00 in GMT-5 is 04:00 in UTC on the next day, so date should also be incremented by one day. |
| Set business value "businessValue" | Set business value of the task to the specified businessValue. |
| Set due date from "customAttribute" | Set the due date/time of the task from the specified customAttribute of the task. |

| Action | Explanation |
| --- | --- |
| Set due time "time" | Set the time when the task is due. The value for "time" must be entered by the rules author in UTC, because it will be attached to the task in the IWD_dueDateTime attribute. The value of this attribute will be displayed to users in the Global Task List, based on their operating system's time zone settings. Entering time values in UTC is not intuitive for a business user, therefore, if there is a business reason to attach a specific due time to a task, you might want to create a custom rule condition that displays the local time (considering the time zone of the business, for example) in the Language Expression, while assigning the corresponding value in UTC, in the Rule Language Mapping. Take care with date changes when specifying the activation time in a local time zone. For example, 23:00 in GMT-5 is 04:00 in UTC on the next day, so date should also be incremented by one day. |
| Set integer "attribute" value "integerValue" | Set the value of a specified custom attribute of type integer to the specified integerValue. |
| Set priority "priority" | Set the iWD priority of the task to the specified value priority. |
| Set string "attribute" value "stringValue" | Set the value of a specified custom attribute of type string to the specified stringValue. |
| Task Due in "period" "periodType" | The task is due after the specified period of time, using time unit periodType. |
| Task Due in "period" working "periodType" | The task is due after the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours, and so on). A business calendar must be assigned to the rule package, or directly to the task, before this action can be used. |
| Task expires in "period" "periodType" | Sets the task's expiration date/time after the specified period of time, using time unit periodType.<br><br>A task's expiration date/time is what is used in the IWDBP business process to determine when a task should be deleted from the Interaction Server database. |
| Task expires in "period" working "periodType" | Sets the task's expiration date/time after the specified period of time, using time unit periodType, according to the task's business calendar (working days, working hours, and so on). A task's expiration date/time is what is used in the IWDBP business process to determine when a task should be task should be deleted from the Interaction Server database.<br><br>**Note:** A business calendar must be assigned to the rule package or directly to the task, before this action can be used. |

## Parameters

| Parameter | Description |
|---|---|
| agent | Presents you with a list of agents' user names that are read from the Genesys Configuration Server database. This list is dynamic; it changes as agents are added or removed. The agents displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| agentGroup | Presents you with a list of Agent Groups that are read from the Genesys Configuration Server database. This list is dynamic; it changes as agents are added or removed. The agent groups displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| archive | Presents you with an enumerated list of possible archive destinations for expired tasks. This parameter uses an Enum, which is included in the Standard Rules Template, called archive.<br><br>Note: This action has been maintained from iWD 8.0 and 8.1.0 but it is no longer necessary to use it, due to changes in the way task archiving works in the out-of-box iWD business process (IWDBP) in iWD 8.1.1. See the IWD Overview for more information. |
| attribute | Enables you to enter text that represents the name of a task attribute. |
| businessCalendar | Presents you with a list of business calendars that are defined in the rule package. This list is dynamic; it changes as business calendars are added or removed. |
| businessValue | Enables you to enter the numeric value that represents business value. |
| businessValue_From | Enables you to enter a numeric value that represents the lower boundary of business value. |
| businessValue_To | Enables you to enter a numeric value that represents the upper boundary of business value. |
| capturePoint | Presents you with a list of Capture Points that is generated dynamically by reading the iWD configuration database. |
| customAttribute | Enables you to enter text that represents the name of a task custom attribute. |
| department | Presents you with a list of departments that are defined in the iWD Solution. This list is dynamic; it changes as departments |

| Parameter | Description |
|---|---|
| | are added or removed. |
| distributionPoint | Presents you with a list of distribution points that are defined in the iWD Solution. This list is dynamic; it changes based on entries in the distributionPoints lookup table. See the Deployment Guide for more information. |
| diffByPeriod | Returns the difference between the two dates in the specified time units (minutes, hours, days). |
| difWorkingDays | Returns the difference between two dates in working days. |
| difWorkingHours | Returns the difference between two dates in working hours. |
| difWorkingMinutes | Returns the difference between two dates in working minutes. |
| integerValue | Enables you to enter an integer value for use in rule conditions or actions that evaluate the value of task custom attributes. |
| mediaType | Presents you with a list of media types that are defined in the Genesys Configuration Server under the tenant. This list is dynamic; it changes as media type attributes get added or removed. |
| multiplier | Enables you enter to enter a numeric value by which some other parameter (such as priority ) will be multiplied in a rule. |
| operator | Provides an enumerated list of operators equal to, not equal to, less than, less than or equal to, greater than, and greater than or equal to, that can be used to compare values of custom attributes of type integer, to values specified in a rule condition. This parameter uses an Enum, which is included in the Standard Rules Template, called operator. |
| period | Enables you to enter a numeric value. Combined with the period type, it gives the actual value of the time period. |
| periodFrom | Enables you to enter a numeric value that represents the start point of a period, in time units, according to period type. |
| periodTo | Enables you to enter the numeric value that represents the end point of a period, in time units, according to period type. |
| periodType | Presents an enumerated list of types of time periods, such as days, hours or minutes. This parameter uses an Enum, which is included in the Standard Rules Template, called periodType. |

| Parameter | Description |
|---|---|
| placeGroup | Presents you with a list of place group names that are read from the Genesys Configuration Server database. This list is dynamic; it changes as place groups are added or removed. The place groups displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| priority | Enables you to enter the numeric value that represents iWD priority. |
| process | Presents you with a list of processes that are defined in the iWD Solution. This list is dynamic; it changes as processes are added or removed. |
| skill | Presents you with a list of skills that are read from the Genesys Configuration Server database. This list is dynamic; it changes as skills are added or removed. The skills displayed will be based on the access control of the Genesys Rules Authoring Tool user. |
| stringValue | Enables you to enter a string value for use in rule conditions or actions that evaluate the value of task custom attributes. |
| taskChannels | Presents the user with a list of task channels, read from an iWD Lookup Table (see the Deployment Guide for more information). |
| time | Enables you to enter a time value that is used in various rule conditions and actions. |
| wfmActivity | Presents a list of WFM activities, retrieved dynamically from the WFM Server. |

## Functions

| Function | Description |
|---|---|
| adjustWorkingDate | Adds or subtracts a given time interval from the given date according to the current business calendar. |
| compareDate | Compares the value of two dates. For a list of possible returned values, see comments within the function. |
| compareInteger | Compares the value of two integers, taking as inputs the two integers and a comparison operator. Returns `true` or `false`. |

| Function | Description |
|---|---|
| `getCurrentCalendar` | Returns the current business calendar ID. |
| `getCurrentDT` | Returns the current date and time, in UTC. |
| `getDTValue` | Gets the value of a task attribute as date/time. |
| `getIntValue` | Gets the value of a task attribute as an integer. |
| getPeriodDTFrom | Adds or subtracts a given time interval from the given date, regardless of the business calendar. |
| `getStringValue` | Gets the value of a task attribute as a string. |
| `getWFMActivity` | Retrieves WFM Activity. |
| `getWFMActivityContext` | Retrieves WFM Activity context (Business Unit or Site). |
| `increaseIntegerValue` | Increases the value of an integer by adding to it. |
| `isNull` | Is `true` if a given task attribute value is `Null`, otherwise, the value is `false`. |
| `isProcess` | Is `true` if a given task is assigned to a given process, otherwise, the value is `false`. |
| `isWorkingDay` | Returns `true` if the current day is the working day. |
| `isWorkingTime` | Returns `true` if the current time is the working time. |
| `multiplyIntegerValue` | Increases the value of an integer by multiplying it by some factor. |
| `notNull` | Is `true` if a task attribute value is not `Null`, otherwise, the value is `false`. |
| `setDepartment` | Sets the department for a task. |
| `setDepartmentAndProcess` | Sets the department and process of the given task from a given string, in `department\|process` format. |

| Function | Description |
| --- | --- |
| `setDTValue` | Sets the value of a task attribute as date/time. |
| `setIntValue` | Sets the value of a task attribute as an integer. |
| `setProcess` | Sets the process for a task. |
| `setStringValue` | Sets the value of a task attribute as a string. |
| `setTime` | Sets the time portion of a given date/time value from a string value, in `hh:mm` format. |

# Custom Templates

## Creating a Custom Template

## Creating a Custom Template

### Procedure

1. Start the Genesys Rules Development Tool.
2. To open the Template Development perspective (if not already opened); go to Window > Open Perspective > Template Development.

> **Tip**
> If the Template Development perspective does not appear in the list, select Other and then, select the Template Development perspective.

3. Select File > New > Rule Template Project.

> **Tip**
> If the Rule Template Project type is not visible, select Other, and then, within the Genesys Rules System folder in the New dialog, select Rule Template Project.

4. Click Next.
5. On the Rule Template Project screen of the New Rule Template Project dialog, enter a name for the template project and select the Eclipse workspace where the project files will be stored.
6. Click Next.
7. On the Template Properties screen, for the Type property, select iWD.
8. For the Tenant property, select the Configuration Server tenant to which this rule template will be associated.

> **Tip**
> Optionally, you can provide a description for the rule template project.

9. Click `Finish`.

The custom rule template is displayed in the `Eclipse Project Explorer` view.

## Next Steps

After parameters, conditions, actions, and functions are created as necessary for the custom rule template, you must publish it to the Genesys Rules System rules repository. See steps 8-15 of **Importing the iWD Standard Rules Template into the Genesys Rules Development tool and publishing it to the Rules repository**.

> **Tip**
> For more detail about rule template components click here or go to the Genesys Rules System 8.1 Development Tool Help.

## Conditions and Actions

## Custom Rule Template Conditions and Actions

## Language Expression Syntax

If a rule action Language Expression contains one or more parameters, the expression must terminate with one of those parameters. For example, the following Language Expression for a rule action generates an error:

```
Set Priority "{priority}" and update
```

whereas, the following Language Expression is valid:

```
Set and Update Priority "{priority}"
```

## Using iWD Lookup Tables in Rules

The Genesys Rules System offers several ways to create rule parameters that will present an enumerated list of items to the rules author as a drop-down list. However, there are certain situations, in which you might want to use a Lookup Table that has been configured in iWD Manager, as the basis for a dynamic rule parameter. This might be valuable because Lookup Tables provide user-friendly name/label pairs that can be changed easily by a business user, while some other types of dynamic parameters, such as List Objects in Configuration Server, cannot be edited easily by a business user. If you want to use a Lookup Table in a rule parameter, it will be a parameter of type `database`, where the database being read is the iWD configuration database. The best way to create a new rule parameter that will look up values from an iWD Manager Lookup Table is to use an existing parameter as an example. For example, in the iWD Standard Rules Template, you can look at the `taskChannels` rule parameter and, in the ACME Rules Template, you can look at the `ACMEproduct` parameter.

Using the `ACMEproduct` parameter as an example, we can see that the SQL query that is executed will read data from several tables in the iWD configuration database:

```
SELECT cmc_lookup_attr_entry.valueKey,cmc_lookup_attr_entry.valueLabel
FROM ((cmc_lookup_attr_entry INNER JOIN cmc_lookup_attr_type
ON cmc_lookup_attr_entry.attributeTypeId=cmc_lookup_attr_type.id)
INNER JOIN cmc_tenant
ON cmc_lookup_attr_type.tenantId=cmc_tenant.id)
WHERE cmc_lookup_attr_type.name='Products'
AND cmc_tenant.name='ACME';
```

In this example, the only parts of the query that would need to change are the name of the Lookup Table, which is referenced in the query as `cmc_lookup_attr_type.name`, and the name of the tenant, which is referenced in the query as `cmc_tenant.name`. If the Lookup Table is configured under the System tenant, then the `cmc_tenant.name` would be set to `System` in the query.

## Operational Parameters

Operational parameters provide another user-friendly way to make rules dynamic, enabling a business user to change a dynamic value that will be referenced in a rule, without having to change the rule itself. Operational parameters are a special type of parameter that is created by users through Genesys Administrator Extension and, when deployed, are stored as options of `Transaction` objects of the type `List` in the Genesys Configuration Server database. At rule execution time, when the Genesys Rules Engine evaluates a rule that contains an operational parameter, it obtains the current value of the associated `Transaction` object option from Configuration Server. The rule developer determines from which `Transaction` object, and which option of that object, the value of the operational parameter should be fetched, and the rule author uses this parameter as part of a condition or action.

For example, an operational parameter might be called `waitTimeThreshold`. If a caller is waiting longer than this threshold for an available agent, perhaps some other action will be performed. Instead of specifying a value for the threshold in the rule like the following:

`When`

```
Caller's wait time is greater than 30 seconds
```

Then

```
Offer a callback
```

the rule author could specify: When

```
Caller's wait time is greater than {waitTimeThreshold}
```

Then

```
Offer a callback
```

The value of `{waitTimeThreshold}` can be changed at any time by a user that is using Genesys Administrator Extension and this action will have an immediate effect without the user having to modify and redeploy a rule package.

# 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.

## Overview

## Overview

## Changes in Archiving in iWD Release 8.1.1

A new archiving solution was implemented in iWD release 8.1.1. Pre-8.1.1 , archive rules (the rules that are applied during the archiving phase) are no longer applicable. The iWD Standard Rules Template still contains a dedicated phase called `Archiving`. This is done for compatibility reasons to allow migration from earlier versions of the iWD, where archiving rules may have been used. After migration, users should review the migrated rules and adjust them according to the new archiving strategy.

## 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

## 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

## 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 in either of the following scenarios:

- During the migration process from an earlier version of iWD to iWD 8.1.
- If the iWD business structure has been pushed to the Genesys Rules System from iWD Manager.

See the Pushing iWD Business Structure to the Genesys Rules System tab on this page.

## 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.

# Pushing the iWD Business Structure to GRS

Pushing the iWD Business Structure to GRS

## Purpose

To synchronize the iWD business structure that has been created in iWD Manager with Genesys Configuration Server. After you complete this procedure, the business structure will become available to the Genesys Rules System, specifically to the Genesys Rules Authoring Tool so it can display the appropriate business structure for use with iWD.

## Prerequisites

- Some business structure (iWD Solution, Departments, and/or Processes) is created in iWD Manager.

## Start

1. Log in to iWD Manager.
2. Click `Departments and Processes`.
3. Expand the iWD Solution you want to push to the Genesys Rules System, and select `Push to Rules System`.

   Alternatively, if you have not already performed this action, you will see the following notification on the top of the iWD Manager screen informing you that “There are changes to be pushed to Rules System: [Solution Name]”, with a hyperlink that will take you to the appropriate screen where you can execute the push action.
4. At the bottom of the `Push to Rules System` screen, click `Execute`.

   In the `Messages` pane, you will see one or more messages indicating the success or failure of the push action.

> **Important**
>
> If the action fails, it might be because the user does not have the appropriate permissions to the *parent* folder. For example, if you have just added a new Process under an existing Department, and the iWD Manager user does not have the appropriate permission (`Full Control`) to that Department business structure

folder in Configuration Server, you will not be able to create the new Process.

## Next Steps

Create a rule package in Genesys Rules Authoring Tool that you can use to create business rules. If you have successfully created your business structure in iWD Manager and pushed it to the Genesys Rules System, you are ready to create a rule package in Genesys Rules Authoring Tool, to start creating business rules.

### Important

Prior to release 8.1.1, if you delete a Process or Department in iWD Manager, you can also push these changes to the Genesys Rules System. However, if you delete an entire Solution in iWD Manager, you cannot push these changes to the Genesys Rules System through iWD Manager. This is because the `Push to Rules System` action is only visible in the navigation tree under the Solution. In this case, you must manually delete the

Solution folder under `Business Structure`, through Genesys Administrator or Genesys Configuration Manager.

Until you manually delete the Solution folder by using Genesys Administrator or Genesys Configuration Manager, you will continue to see this unwanted Solution in the Genesys Rules Authoring Tool.

Also, to delete a parent folder in Configuration Manager or Genesys Administrator, such as an iWD Solution, you must first delete the child folders, such as the Departments. If there are Processes under those Departments, those must be deleted first.

After you have a working environment, from time to time you might need to modify the business structure in iWD Manager. After making those modifications, you must the following steps in this order:

1. Push the business structure changes to the Genesys Rules System, as described in this section.
2. Modify your business rules, if necessary, in GRAT.
3. Deploy your rule package in GRAT.
4. Deploy your iWD Solution in iWD Manager.

From release 8.1.1 onwards, if you delete a solution, you will be prompted to select whether you want to delete it from the rules system as well. If you confirm the deletion, the solution with all business objects will be removed. If there was a rules package deployed from that solution, you need to first delete the rules package manually.

## 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 8.1 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.
- Templatesection—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.

- 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

## 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

In iWD 8.1. you can use the rule condition `Capture Point is…` from the iWD Standard Rules Template, at the Global Rules level. Classification rules are no longer created at the Capture Point level (as they were in iWD 7.6.1 and 8.0).

## 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 <Scriptobjects that represent each template. In Configuration Server, under each tenant, there is a Scriptfolder 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.

For an example of a decision table, see xxx

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 Scriptfolder 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 8.1 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. |

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

In addition to the standard rule actions that use business calendars, you can build other rule expressions that use business calendar functions. For more information about how to build these other rule expressions, see https://sites.google.com/a/iwdlab.com/iwd8/rules/bc.

## 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 | `businessCalendarService.add(getStringValu $data), getDTValue("IWD_createdDateTime", $data), "hours", 4)); // get timestamp of 4 working hours after the task was created` |

| businessCalendarService Method | Description | Example(s) |
|---|---|---|
| | adding 1 day it will result in the end of the next working day. | |
| 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(g<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// is today a working day<br><br>businessCalendarService.dayIsWorkingDay(getStringValue("I<br>getDTValue("IWD_dueDateTime",$data));<br>// does IWD_dueDateTime fall on a<br>working day? |
| boolean timeIsWorkingTime(String businessCalendarName, Date time) | Returns true if the given timestamp is a working time. | businessCalendarService.timeIsWorkingTime<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// is it working time right<br>now<br><br>businessCalendarService.timeIsWorkingTime(getStringValue<br>getDTValue("IWD_dueDateTime",$data));<br>// does IWD_dueDateTime fall within<br>working hours? |
| int diffWorkingDays(String businessCalendarName, Date date1, Date date2) | Returns the number of working days between two given timestamps. | businessCalendarService.diffWorkingDays(g<br>$data),<br>getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// how many working days<br>have passed since creation<br>of the task? |
| int diffWorkingHours(String businessCalendarName, Date date1, Date date2) | Returns the number of working hours between two given timestamps. | businessCalendarService.diffWorkingHours<br>$data),<br>getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// how many working hours<br>have passed since creation<br>of the task? |
| int diffWorkingMinutes(String businessCalendarName, Date date1, Date date2) | Returns the number of working minutes between two given timestamps | businessCalendarService.diffWorkingMinute<br>$data),<br>getDTValue("IWD_createdDateTime",<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// how many working minutes<br>have passed since creation<br>of the task? |
| Date beginningOfWorkingDay(String businessCalendarName, Date utcDate) | 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 | businessCalendarService.beginningOfWorkin<br>$data),<br>evo.common.TimeHelper.getUTCTime());<br>// get today's opening hours |

| businessCalendarService Method | Description | Example(s) |
|---|---|---|
| | working hours for the given date. If date is not a working day, a null value will be returned. | |
| `Date endOfWorkingDay(String businessCalendarName, Date utcDate)` | 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 ond of working hours for the given date. If date is not a working day, a null value will be returned. | `businessCalendarService.endOfWorkingDay($data), evo.common.TimeHelper.getUTCTime()); // get today's closing hours` |

**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?**

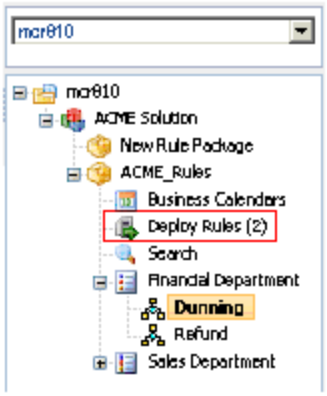
```
businessCalendarService.diffWorkingHours(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime(),
 businessCalendarService.endOfWorkingDay(getStringValue("IWD_businessCalendarId", $data), evo.common.TimeHelper.getUTCTime()));
```

# Rule Deployment

## Rule Deployment

Once you have created all the necessary rules in a rule package, it is time to deploy the rule package to an instance of the Genesys Rules Engine. 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, (as was the case in iWD 7.6.1 and 8.0). 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 changes made in GRS release 8.1.3, including scheduled deployment and deployment history, and snapshots, see the **GRAT 8.1 Help.**

# 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.**

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 Business Context Management Service (BCMS) application, and to correlate the `IWD_SolutionId` to the name of the Genesys Rules Engine. The BCMS is the External Service Protocol (ESP) server that the IWDBP business process uses to communicate with 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, iWD_Canceled`, or `iWD_Rejected`.

## 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.

> **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, Rule-265 will be evaluated prior to Rule-294. If the rules author preferred Rule-294 to be evaluated first, s/he 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.

## 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 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`.