



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 and GRS

Custom Templates

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.