Genesys Rules Authoring Tool Help

Rule Template Development Overview

4/24/2025

Rule Template Development Overview

Rules templates allow the business rules developer to create rule templates that define the conditions and actions that will be used by the business rule author. The developer creates the plain-language statements that the business author will see and maps them to the rule language statements that the rules engine will execute. For each rule condition and action, the developer decides what kind of data the rules author will be providing. Some examples include whether the input should be an integer value, a non-integer numeric value, a string, a selection from a pre-defined list, or a selection from a dynamic list. Rule templates 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 Genesys solution.

Quick links—Working with Templates

Importing and exporting templates
Creating and editing templates
Publishing templates
Examples of template development
Rule language mapping
Using Drools 5

Quick Links—Working with Editors

Actions Editor
Conditions Editor
Enumerations Editor
Fact Model Editor
Functions Editor
Parameters Editor

Template contents

Rule templates are made up of several elements:

- Actions
- Conditions
- Enumerations
- Fact models
- Events
- Functions
- Parameters

Actions and Conditions

Actions and conditions define WHEN/THEN scenarios, such as WHEN a customer is a Gold customer,

THEN target the GoldAgentGroup. The WHEN statement is the condition, and the THEN statement is the action. A rule may have zero or more conditions, and one or more actions. This example also includes parameters: the status of the customer (Gold) and the name of the Agent Group (GoldAgentGroup).

Whenever a condition contains a rule language mapping that begins with `eval(...)`, you must enclose the entire expression in parenthesis, as follows:

```
( eval(.... ) )
```

This will ensure it will compile properly when used with the NOT operator.

See Actions editor and Conditions editor.

Enumerations

Enumerations are used to define lists of possible choices that will be displayed to the business rule author, when the author is creating rules that are based on the rule template. In some cases, the list of possible choices will be selected dynamically from Genesys Configuration Server objects or from external data sources. For WFM Activities and Multi-Site Activities, the list of possible choices is retrieved dynamically from the Genesys WFM Server. Thus, enumerations are used during definition of a discrete list of choices that will not change dynamically.

See Enumerations editor

Fact models

All rule templates include a fact model with one or more facts. A fact model structures basic knowledge about business operations from a business perspective. A fact model focuses on logical connections (called facts) between core concepts of the business. It indicates what you need to know about the business operations in order to support (or actually do) those operations.

A good fact model tells you how to structure your basic thinking (or knowledge) about the business process based on a standard vocabulary. By using standard, business-focused vocabulary, it ensures that the business rules themselves can be well-understood by key stakeholders such as business analysts. For example, in your business you may have a Fact that represents a Customer, and another Fact that represents an Order.

The Customer could have fields such as name, age, location, credit rating, and preferred language. The Order may have fields such as order amount and order date. A rule could be constructed using these values such as:

When Customer is at least 21 years old and his order is > 100.00 then invite customer to participate in survey.

See Fact model editor

Events

In order to support Complex Event Processing, template developers need to be able to designate certain facts as events, and rules authors need to change the way that the DRL is generated when a fact is designated as an event.

So the fact model includes include events, and the fact model dialog now includes a Create Event button. An event has the following fields:

- Name
- Description
- An optional list of Properties.
- User-defined expiration metadata for the event

In GRAT, the `@role` meta-data tag determines whether we are dealing with a fact or an event. The `@role meta-data` tag can accept two possible values:

- `fact`—Assigning the fact role declares the type is to be handled as a regular fact. Fact is the default role.
- `event`—Assigning the event role declares the type is to be handled as an event.

Functions

Functions are used to define elements other than Conditions and Actions, for example, when parsing of timestamps is required. The Functions editor enables you to write specific Java functions for different purposes for use in rule templates. The specified functions may then be used in the rule language mappings (see Rule Language Mapping).

When the rule templates are created, the rule developer publishes them to the Rule Repository, making them available in the GRAT for business users to create rules.

Actions and conditions can contain parameters. Various types of parameters are supported.

Certain dynamic parameter types that refer to external data sources require a Profile to be selected. The Profile is defined as a Script object of Data Collection type, and it provides connection information that enables the GRAT to retrieve this dynamic data from the external data source. The next sections describe how to configure Profiles for database, Web Service, and Workforce Management parameters.

See Functions editor.

Parameters

Database, Web Service and Workforce Management parameters are used in the actions and conditions.

Database Parameters

Database Parameter Properties

Property	Mandatory/optional	Description
driver	Mandatory	The name of the jdbc driver to be used. For example,

Property	Mandatory/optional	Description
		`com.mysql.jdbc.Driver`
url	Mandatory	The url for the database in the correct format for the jdbc driver to be used.
username	Mandatory	A valid username to connect to the database.
password	Mandatory	The password needed for the user to connect to the database.
initSize	Optional	The initial size of the connection pool. The default is `5`.
maxSize	Optional	The maximum size of the connection pool. The default is `30`.
waitTime	Optional	The maximum time (in milliseconds) to wait for obtaining a connection. The default is `5000`.

In general, the optional values do not need to be set or changed.

You can only configure database parameters with an SQL SELECT statement. Any other type of statement will fail when configured.

Web Service Parameters

In Configuration Server, Web Service Scripts must have a section called webservice. The table below lists the properties that you can specify for web service parameters.

Web Service Parameter Properties

Property	Mandatory/optional	Description
host	Mandatory	The host for the service.
base-path	Mandatory	The base path to access the service.
protocol	Optional	The default is `http`.
port	Optional	The default is `80`.
headers	Optional	Any custom HTTP headers that are needed for the service.
parameters	Optional	Any custom HTTP settings that are needed to tune the connection.

In general, the parameters values do not need to be set or changed. Headers and parameters are lists in the following format:

```
key:value[,key:value]
```

Warning:	You cannot specify headers or parameters that contain "," in the value. Warning: If you are sending a message to the service, it is expected that `Content-Type` is specified in the header since it defines the overall message interaction with the server. An optional charset can be included. For example, `Content-Type:applicaton/json;charset=UTF-8`.

You have to completely define the message to be sent and it must be constant. No variable substitution is done. The XPath Query is used to pull values out of the response from the server. The response must be in XML or JSON, otherwise this will not work. A valid XPath query for the response must be specified. This depends entirely on the service you interface with.

Note:	The message is sent to the server only once per session. This is done both for performance reasons and because the values in the response are expected to be relatively constant.

The path for the parameter is added to the `base_path` in the script.

For example:

If the Script contains:

```
host = api.wunderground.com
base_path = /auto/wui/geo/ForecastXML/
```

and Template Development specifies:

```
query type = List
XPath Query = //high/fahrenheit/text()
HTTP Method = GET
path = index.xml?query=66062
message (not set)
```

then the message that is sent is:

```
GET /auto/wui/geo/ForecastXML/index.xml?query=66062 HTTP/1.1
```

This will return the week's highs in Fahrenheit:

```
81
77
81
81
83
85
```

Workforce Management Parameters

In Configuration Server, Workforce Management Scripts must have a section called wfm. Table 4 lists the properties that you can specify for Workforce Management parameters.

Workforce Management Parameter Properties

Property	Mandatory/optional	Description
wfmCfgServerApplName	Mandatory	Configuration Server application name for the WFM server.
wfmCfgServerUserName	Mandatory	Configuration Server user name.
wfmCfgServerPassword	Mandatory	Configuration Server password.
wfmServerUrl	Mandatory	URL of WFM Server.

When configuring a new parameter of type “Workforce Management”, simply name the parameter and choose the WFM profile (script object just created) from the drop-down list. When the author is using this parameter, the GRAT will fetch the current list of WFM Activities from the WFM Server and display them to the rule author.

Support for User-Defined Template Types

GRAT automatically displays the list of template types published to it, and template designers can select these user-defined template types or define new ones, according to their own needs.

Template versions

Each time a rule template is published, a new version is created in the repository. The rule author will be able to select any version of the template from the Template Selection dialog when creating a rule package. This dialog shows the last N versions of a template, where N is a value configured by using configuration option display-n-template-versions in Genesys Administrator.

When you are publishing newer versions of the rule template, be aware that certain changes could affect rules that already have been created using the earlier version of the template. Be careful not to make changes that could void existing rules, unless these changes are communicated to the rule author.

For example, if Rule Template version 1 contains a condition that is removed later in version 2, then if a rule were already built using that condition, it will no longer compile if the rule author upgrades to Rule Template version 2.

For example, if the configuration were set to show the last 3 versions of a template, the currently selected template is `GRS Template` version 2, and there are 5 versions in the repository, we would show `GRS Template` versions 5, 4 and 3, as well as `GRS Template` version 2. Users could choose between versions 3, 4, or 5.

Version Comment

In order to provide details about the differences between template versions, rules template developers can publish a version comment that describes specific changes made to individual template versions. This version comment appears in GRAT in the Template Selection table, and can

be edited by the rule author in GRAT.