GENESYS™

Genesys Rules Authoring Tool Help

Parameters Editor

4/25/2025

Contents

Parameters Editor

The Parameters Editor allows you to create rule parameters, which are optionally used in rule conditions and actions.

> ## Important
> In release 9.0.0, mapping of parameters to fact models is not supported.

Each parameter contains the same fields in the `Details` section:

Name—A name for the parameter. **Description**—A brief description of the parameter. **Type**—The Parameter type. Full information on types is in the following topics.

Parameter Names

The underscore (_) character in parameter names has a special meaning when building rule templates. It is used to specify an index of the parameter in circumstances where the rule expression requires additional instances of the parameter. The most common example is a range definition.

For example, suppose you need to create a condition which has to check if the task's due date is in either the date1 to date2 range, or in the date3 to date4 range. You could create a condition such as:

```
Due is in "{dueDT1}" to "{dueDT2}" or in "{dueDT3}"  to "{dueDT4}")
```

But this requires the definition of 4 parameters with type `InputDate` in the `Parameters` section. This approach can become inefficient, especially if there is more than one occurrence of the condition/action.

A better solution is to use the underscore and index approach:

```
Due is in "{dueDT_1}" to "{dueDT_2}" or in "{dueDT_3}" to "{dueDT_4}"
```

Using this approach, you need to specify only one parameter, with name dueDT and type InputDate.

Categories of Parameter

The **Configuration** section contains information that is dependent on the parameter type. When a type is selected from the drop-down list, different fields are displayed that relate to that type.

There are eight main categories of parameters:

- String

- Integer
- Numeric
- Date
- Time
- Boolean
- Calendar
- Configuration Server

Input values

Boolean, Integer, Numeric, String, Date and Time are simply parameters for which the rules author can provide a value based on the defined parameter type. These parameters can also be constrained. For instance, an integer value can be constrained to be within a defined range.

Matching Patterns

For Input Value parameters of type `String`, you can enter a matching pattern that must be followed. Enter a Javascript regular expression to define the matching pattern. For example, a Zip Code parameter might have the matching pattern:

```
>^\d{5}$|^\d{5}-\d{4}$
```

which represents a 5-digit zip code. A Phone Number parameter might have the matching pattern:

```
^\(?(\d{3})\)?[- ]?(\d{3})[- ]?(\d{4})$
```

which represents a 10-digit phone number in the format (xxx)-xxx-xxxx.

Custom Tooltips

Use Custom Tooltip allows you to enter useful tooltip text when defining all Input Value parameters (except for the Boolean parameter type, which does not need a tooltip). If you check **Use Custom Tooltip**, the text that you enter in the tooltip field is displayed in GRAT when this parameter is used in a rule condition or action. If you do not check **Use Custom Tooltip**, GRAT displays an automatically generated tooltip; for example, `{age} is an integer between 1 and 99`.

The regex pattern supported should conform to the browser's Javascript engine and may vary slightly depending on the browser version.

Calendars

Calendar parameters indicate to GRAT that it should display a drop-down list of business calendars associated with the rule package being edited. The rule author can then choose one of the calendars.

Example: Calendar parameters can be used in a rule to dynamically assign a calendar as follows:

```
Assign business calendar "{businessCalendar}"
```

When defining a Calendar parameter, the template designer only needs to provide the parameter name and select a type of Calendar. There is no other configuration required.

Configuration Server

Configuration Server parameters give the rule author the ability to choose a single value from a drop-down list of values. For example, a Configuration Server parameter may be configured to pull a list of Agent Groups from the Configuration Server database. The list is populated from Configuration Server. Configuration Server parameters require you to select the Object type:

- Agent
- Agent Groups
- Agent Skills
- Business attribute
- Business context
- Extension
- External Route Point
- Interaction Queue
- Media Type
- Place
- Place Groups
- Route Point
- Switch
- T-Server
- Virtual Route Points

Selecting **Business Attribute** prompts you to select the name of the Business Attribute from a list defined in Configuration Manager.

Selecting **Business Context** prompts you to enter the level of the Business Context that is of interest for this parameter. Here, Business Context refers to the level of the hierarchy under the **Business Structure** folder in Configuration Server.

Database

Database parameters give the rule author the ability to choose a single value from a drop-down list of values. For example, a Database parameter may be configured to pull a list of Order Types from a database. The list is populated by a database query. Database parameters require the Profile Name (the name of the Configuration Server Script object that contains your database connection

information), Query Type (Single Value or List, depending on what you want to show up in GRAT), and the SQL Query to be executed.

Note: The list of values is fetched at the time the rules author logs into GRAT. If any values are updated from the external system after the user has logged into GRAT, the user must click the Logoff button and then log back in again to see any changes.

Example

To use a Database parameter, a parameter profile must previously have been configured for the Tenant in Configuration Server. This is a Script object that specifies the JDBC driver, as well as the database url, username, and password required to perform the query. Refer to the Genesys Rules System Deployment Guide for information on configuring these profiles. The name of this Script object is used as the profile name for the database parameter.

To obtain values from the database, a valid SQL Select statement must be specified. For instance, to get all the values of a column use a select statement of the following form:

```
SELECT column_name FROM table_name
```

For dynamic Database parameters, you can configure the parameter so that both a name (an internal value that is included with a rule evaluation request) and a label (the information that is displayed to a rules author when authoring a rule that uses this parameter) can be retrieved from two different database columns.

Database Profile Parameter Properties

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

In template development, you can only configure database parameters with an SQL SELECT

statement. Any other type of statement will fail when configured.

Operational

Operational parameters are 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 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 is fetched, and the rule author uses this parameter as part of a condition or action.

Example An Operational parameter called **waitTimeThreshold** may be defined. If a caller is waiting longer than this threshold for an available agent, some other action may 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 using Genesys Administrator Extension and will have an immediate effect without having to modify and redeploy a rule package.

For example, use the following condition when you define the mapping:

```
Queue(waitTime >  {waitTimeThreshold} )
```

To configure an Operational parameter you need two IDs:

- The **List ID**, which corresponds to the name of the Transaction object in which the Operational parameter is stored
- The **Parameter ID**, which corresponds to the name of an option of that Transaction object.

The option value contains the actual value of the Operational parameter that is retrieved by the Rules Engine when the rule is evaluated. Operational parameters are always stored as Transaction objects of the type `List`, but the precise configuration of the options within that List object varies depending on how the Operational parameter was configured.

> ## Warning
>
> When configuring an "operational parameter", it is important that you *do not* enclose the variable in either double or single quotes in either the language expression or the rule language mapping. For example, a condition that uses an operational parameter {opParmEwtThreshold} should be configured like this:
>
> - **Language Expression**—`Estimated wait time is greater than {opParmEwtThreshold}`
>
> - **Rule Language Mapping**—`CallInfo ( ewt > Integer.parseInt({opParmEwtThreshold}))`
>
> Note that there are no single or double quotes around `{opParmEwtThreshold}`
>
> For additional examples, see the "Operational Parms" sample template and rule package shipped with GRAT (in the **examples** folder).
>
> Refer to the Genesys Administrator Extension Help for general information about Operational parameters.

Select Enumeration

Select Enumeration parameters are linked with an Enumeration. This provides the rules author with a specific list from which to select.

Web Service

Web Service parameters give the rule author the ability to choose a single value from a drop-down list of values. For example, a Web Service parameter may be configured to pull a list of stock symbols from an external web service. The list is populated by a Web Service query. Web Service parameters require the Profile Name (the name of the Configuration Server Script object that contains your web service connection information), Query Type (Single Value or List), and the XPath Query to be executed. In addition, Web Service parameters require that some Protocol Settings be entered, specifically the HTTP Method, Path, and the Message Body.

Note: The list of values is fetched at the time the rules author logs into GRAT. If any values are updated from the external system after the user has logged into GRAT, the user must click the Logoff button and then log back in again to see any changes.

Example

Similar to a Database parameter, a parameter profile must also have been previously created. This profile will contain information such as the server's address (host and port), the path to the service, and any other necessary HTTP settings. Refer to the Genesys Rules System Deployment Guide for information about configuring these profiles.

To obtain values from the service, a valid message for the service must be specified. This message must be constant. In other words, no variable substitution will occur.

Note: No message can be sent for HTTP GET requests. All the information in the request is provided through the query string and/or headers.

For instance, to obtain the weather forecast for San Francisco, the following request can be made to the Weather Underground REST service:

http://api.wunderground.com/auto/wui/geo/ForecastXML/index.xml?query=94129

However, this is the complete request. The host (api.wunderground.com) and the base path (/auto/wui/geo/ForecastXML/), must be specified in the profile.

To define a parameter to make this request, the profile name must reference the correct information described above. In addition, the Protocol Settings must specify GET as the method, along with `index.xml?query=94129` as the path. No message is needed for this request.

To obtain values from the result, a valid XPath expression must be specified. The Web Service must return results in XML or JSON. Please see the XPath specification for more information on XPath expressions.

For example, to get the forecast highs from the previously described request, the following XPath expression may be used:

`//high/fahrenheit/text()`

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 Profile 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]`

You cannot specify headers or parameters that contain "," in the value. 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`.

> ## Important
>
> In Template Development 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.

> ## Important
>
> 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.

In Template Development, 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 the GRDT 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

Workforce Management (WFM) parameters enable the rules author to select a value from a drop-down list of activities (a WFM database object that represents contact center tasks in which agents can be engaged) and multi-site activities (a collection of activities performed at multiple physical sites) that is dynamically retrieved from the Genesys Workforce Management Server. Workforce Management parameters require the WFM Profile (the Configuration Server Script object of type Data Collection).

> ### Important
>
> The list of values is fetched at the time the rules author logs into GRAT. If any values are updated from the external system after the user has logged into GRAT, the user must click the Logoff button and then log back in again to see any changes.

Example

An activity is the main planning object that is used when building forecasts and schedules. An activity can be associated with an individual WFM Site object, or multi-site activities can be created at the WFM Business Unit level, which aggregates information from multiple "child" activities across multiple WFM Sites. So, when providing the rules author a list of WFM activities that are dynamically fetched from the WFM Server, the name of the WFM activity or multi-site activity is prefixed with the name of the WFM Site or the WFM Business Unit, respectively.

For example, if the WFM configuration has the following structure:

Business Unit with the name 'ACME'

- Site with the name 'San Francisco'
- Activity with the name 'Disputes'
- Activity with the name 'Billing Inquiries'
- Site with the name 'Chicago'
- Activity with the name 'Disputes'
- Activity with the name 'Address Changes'
- Multi-Site Activity with the name 'Billing' (comprised of 'Billing Inquiries' from the San Francisco site and 'Address Changes'' from the Chicago site)
- Multi-Site Activity with the name 'Disputes' (comprised of 'Disputes' from the San Francisco site and 'Disputes' from the Chicago site)

The rules author will see the following items in the drop-down list when using the rule action Assign WFM Activity in a rule:

```
B.U. ACME: Billing B.U. ACME: Disputes Site Chicago: Address Changes Site Chicago:
Disputes Site San Francisco: Billing Inquiries Site San Francisco: Disputes
```

> ### Important
>
> The names of the Business Units and Sites are pre-fixed with 'B.U.' and 'Site' respectively, to help the rules author understand the context.

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

Workforce Management Profile 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" in Template Development, 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, GRAT will fetch the current list of WFM Activities from the WFM Server and display them to the rule author.