



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.

Genesys Rules System Deployment Guide

Rule Template

12/17/2025

Rule Template

The rule template that is created for this example consists of two facts:

- `_GRS_Environment`
- `Customer`

The `_GRS_Environment` fact is a mandatory fact for all Genesys Rules Systems rule templates. It is used to establish two important fact properties:

- `businessContext_Level*`—Used in the request to the Rules Engine, to determine the node(s) of the business structure at which to evaluate rules
- `phase`—Used within the request to the Rules Engine, to determine which rules to evaluate.

Each rule that you create in Genesys Rules System must have a rule phase defined. The list of rule phases can be modified by changing the values of the enumeration that is called `Phases`, in the rule template. In this example, the phase that is selected is called `segmentation`, so we can assume that the values for the `Phases` enumeration contains at least one value called `segmentation`, and possibly others.

The `Customer` fact contains three properties that we will use in our business rule:

- `segment`
- `order`
- `offer`

Our rule template contains two conditions and one action, as well as the necessary parameters that are used within these conditions and actions. See the two following tables for details of these parameters.

Rule Language Mapping

Rule Language Mapping Parameters

Name	Language Expression	Rule Language Mapping
Segment	Customer segment is {customerSegment}	Customer(segment=='{customerSegment}')
OrderValue	Order value is greater than {orderValue}	Customer(order>{orderValue})
SpecialOffer	Offer special promotion {specialOffer}	\$Customer.offer='{specialOffer}'

Language Expression Details

Name	Type	Comments
customerSegment	Enumeration	An enumeration must be created in the rule template that contains the values for Customer Segment from which the rules author will be able to select (for example, Bronze, Silver, and so on). Note that there are two properties that you must provide for each value of the enumeration: Name and Label. The Label is what will appear to the business rules author when the business rules author is using a rule condition or action that includes a parameter that references this enumeration. The Name is what is used in the request/response to/from the Rules Engine; therefore, case is important. For example, you may want to use uppercase for the labels of these enumeration values, and lowercase for the names. shows an example of how that might appear in the Genesys Rules Development Tool:
orderValue	Input Value (Numeric)	Optionally, you can supply upper and lower bounds for this parameter. If these are supplied in the template, the rules author will be constrained as to the values the rules author can provide in the rule condition that uses this parameter.
specialOffer	Input Value (Boolean)	Because the parameter type is Boolean, this will present a checkbox to the rules author when this parameter is used in the rule action.

The figure below shows how the enumeration is configured.

The screenshot shows two sections of a rule template editor. The top section, titled 'Enumeration Details', has a 'Name' field containing 'CustomerSegment' and an empty 'Description' field. The bottom section, titled 'Values', contains a table with two columns: 'Name' and 'Label'. The table lists four values: bronze, gold, platinum, and silver, each with a corresponding label.

Name	Label
bronze	Bronze
gold	Gold
platinum	Platinum
silver	Silver

Enumeration Details

Note that for this template, because the `orderValue` parameter is numeric, when it is used in a rule condition, there are no single quotation marks (") surrounding it in the rule language mapping, whereas there are single quotation marks surrounding the `customerSegment` string parameter.

With the Drools language you cannot set the value of a Fact property by referring to the Fact's name. In the condition section you must first declare a variable and associate this variable with a Fact object. Once this association has been made within a condition then the variable can be used in actions (and other conditions) to reference fields contained within the fact. A period (".") is used to access the fields on a fact. Use a colon (":") when you want to create a variable in a condition. So, in the preceding example, the "." is used in the rule language mapping for the action (`$Customer:Customer()` in the condition, `$Customer.offer` in the action).

Since the 8.1.2 release, conditions are automatically added to declare variables which are referenced in actions. For a variable to be automatically declared, the variable name must be the name of the fact preceded by a '\$' sign. So in this example, `$Customer` is referenced in an action so the condition `$Customer:Customer()` will automatically be added to the rule.

Before 8.1.2, variables had to be declared by a condition within the template and added to the rule by the rule developer. With this example the following generic rule condition needed to be defined within the template and the rules developer would have added this condition to any rule that referenced the `$Customer` variable.

Important

A variable cannot be declared twice.

Language Expression: `Customer exists`
Rule Language Mapping: `$Customer:Customer()`