



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.

## GRDT Help

Developing Templates to Enable Test Scenarios

12/17/2025

# Developing Templates to Enable Test Scenarios

## Contents

- **1 Developing Templates to Enable Test Scenarios**
  - 1.1 Mapping Parameters to a Fact Model
  - 1.2 Using ESP-type Templates
  - 1.3 Fact Name
  - 1.4 Field Name

## Mapping Parameters to a Fact Model

Rules authors build rule-test scenarios using parameters, in the same way as they define rules. However, when the tests execute, GRAT maps the parameter values to the underlying fact model developed by the template developer, who understands the relationship between the parameters and the fact model. So, for rule testing to operate correctly, template developers must map parameters back to the underlying fact model.

For example, the {age} parameter may be related to the ageinyears field of the Prospect fact. So, if age is set to 25, then, when executing the test, GRAT needs to allocate a Prospect fact and set the ageinyears field to 25. The same is true for the expected results.

The **Associate Property** dialog enables this mapping.



In general, there should be a one-to-one mapping between a parameter and a fact. However, this may be too restrictive for all implementations. GRDT lets you map a single parameter to multiple fact values. For example, the {age} parameter could be defined once, but reused to represent both a customer's age and the age of an order.

So, {age} could map to both:

- Customer.ageinyears
- Order.ageoforder

Where this occurs, GRAT displays the parameter in the **Add Given** or **Add Expect** drop-down list in parentheses, so that the GRAT user can select the correct mapping

### Example

In the following example, only {age} has this special designation because of the ambiguity in the definition.

Add Given...

- {age} (Customer.ageinyears)
- {age} (Order.ageoforder)
- {gender}
- {education}

To hide this ambiguity from the rule author, you should declare a different parameter for each usage: for example, {customerAge} and {orderAge}.

## Using ESP-type Templates

There are some special considerations in developing templates for ESP-type templates, for products such as iWD.

With ESP templates, instead of building a set of facts and passing them to be executed, you create a KeyValueCollection (KVC) and populate it with key/value pairs of test data. In order to enable this mapping between a parameters and the correct key to use in the KVC collection, you need to create a dummy fact model in GRDT to represent the keys that will be inserted into the KVC.

For example, with iWD, this means modifying the iWD Standard Rules Table to insert a fact and a field for each key that is used in the template.

### Fact Name

You will need to define a reserved fact name (for example, \_GRS\_ESP\_Fact) that is processed differently. For this fact, the fields are mapped to a KVC instead of the traditional Fact model. The fact name must be used because there is no type associated with a fact. Types are only associated with individual fields.

### Field Name

You must develop a convention (such as prefixing all field names with grs\_) . This is because fact fields must start with a lower-case letter (a DROOLS restriction) and GRDT enforces this convention, but iWD key names all begin with IWD. Since the existing iWD key names are like “IWD\_businessValue”, you will need to adopt some naming convention. If the grs\_ prefix is present, GRAT will remove it and insert the remaining value into the KVC as the key (for example, grs\_IWD\_channel is inserted into KVC as IWD\_channel)

The rule template developer must then map each parameter name (used in conditions/actions) to the

appropriate field within `_GRS_ESP_Fact`.

The rule author can then use GRAT to create a rule and a test scenario for that rule.