Red Hat Process Automation Manager 7.2

Testing a decision service using test scenarios

Last Updated: 2020-05-04

Red Hat Process Automation Manager 7.2 Testing a decision service using test scenarios

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Abstract

This document describes how to test a decision service using test scenarios in Red Hat Process Automation Manager 7.2.

Table of Contents

PREFACE

As a business analyst or business rules developer, you can use test scenarios in Business Central to test a decision service before a project is deployed. Testing a decision service ensures that rule assets in the project function properly and as expected. You can test a decision service at any time during project development.

Prerequisites

- The team and project for the decision service have been created in Business Central. For details, see *Getting started with decision services* .
- Business rules and their associated data objects have been defined for the decision service. For example, see *Designing a decision service using guided decision tables* .

NOTE

Having defined business rules is not a technical prerequisite for test scenarios, because the scenarios can test the defined data that constitutes the business rules. However, creating the rules first is helpful so that you can also test entire rules in test scenarios and so that the scenarios more closely match the intended decision service.

CHAPTER 1. TEST SCENARIOS

Test scenarios in Red Hat Process Automation Manager enable you to validate the functionality of business rules and business rule data before deploying them into a production environment. With a test scenario, you use data from your project to set given conditions and expected results based on one or more defined business rules. When you run the scenario, the expected results and actual results of the rule instance are compared. If the expected results match the actual results, the test is successful. If the expected results do not match the actual results, then the test fails.

Using the **Test Scenario** editor, test scenarios can be executed one at a time or as a group whereas when using the **Test Scenario (Preview)** editor, all scenarios from the ***.scesim** file are executed at once. The group execution contains all the scenarios from one package. Test scenarios are independent, so one scenario cannot affect or modify the other. You can run test scenarios at any time during project development in Business Central. You do not have to compile or deploy your decision service to run test scenarios.

You can import data objects from different packages to the project package as the test scenario. Assets in the same package are imported by default. After you create the necessary data objects and the test scenario, you can use the **Data Objects** tab of the test scenarios designer to verify that all required data objects are listed or to import other existing data objects by adding a **New item**.

CHAPTER 2. DATA OBJECTS

Data objects are the building blocks for the rule assets that you create. Data objects are custom data types implemented as Java objects in specified packages of your project. For example, you might create a **Person** object with data fields **Name**, **Address**, and **DateOfBirth** to specify personal details for loan application rules. These custom data types determine what data your assets and your decision services are based on.

CHAPTER 3. USING THE TEST SCENARIO EDITOR IN BUSINESS CENTRAL

In Red Hat Process Automation Manager, there are two test scenarios editors. One is the **Test Scenario** editor and the other is **Test Scenario (Preview)** editor. The **Test Scenario (Preview)** editor is a technology preview only and is not supported for production use. However, you can use either of them to test your business rules & data. This chapters covers how to use the **Test Scenario** editor to create and run test scenarios.

3.1. CREATING DATA OBJECTS

The following procedure is a generic overview of creating data objects. It is not specific to a particular business process.

Procedure

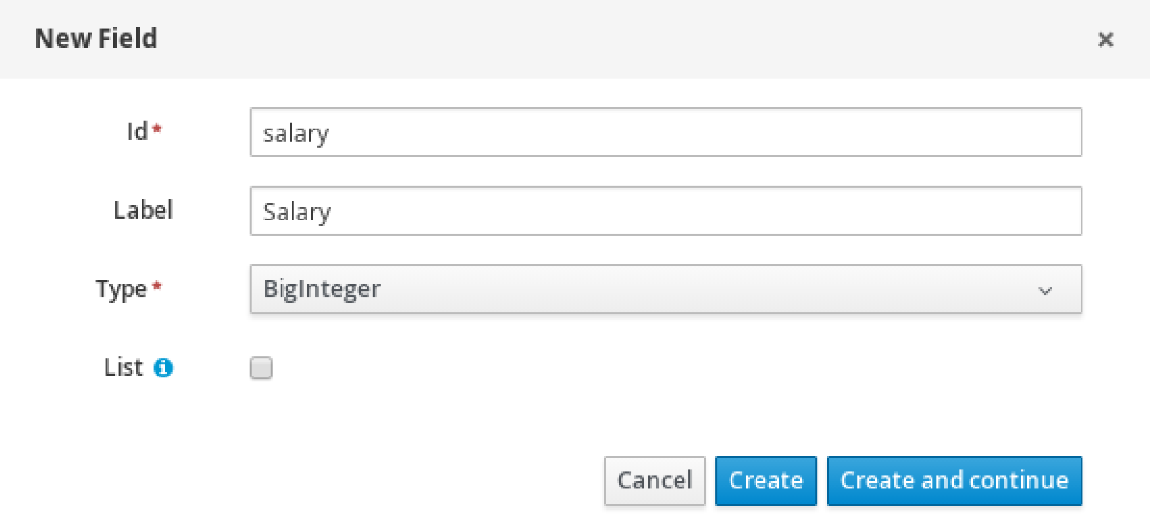
1. In Business Central, go to **Menu → Design → Projects** and click the project name.
2. Click **Add Asset → Data Object**.
3. Enter a unique **Data Object** name and select the **Package** where you want the data object to be available for other rule assets. Data objects with the same name cannot exist in the same package. In the specified DRL file, you can import a data object from any package.

   IMPORTING DATA OBJECTS FROM OTHER PACKAGES

   You can import an existing data object from another package directly into the asset designer. Select the relevant rule asset within the project and in the asset designer, go to **Data Objects → New item**to select the object to be imported.

4. To make your data object persistable, select the **Persistable** checkbox. Persistable data objects are able to be stored in a database according to the JPA specification. The default JPA is Hibernate.
5. Click **Ok**.
6. In the data object designer, click **add field** to add a field to the object with the attributes **Id**, **Label**, and **Type**. Required attributes are marked with an asterisk (*).
   - **Id:** Enter the unique ID of the field.
   - **Label:** (Optional) Enter a label for the field.
   - **Type:** Enter the data type of the field.
   - **List:** Select this check box to enable the field to hold multiple items for the specified type.

Figure 3.1. Add data fields to a data object


New Field

Id * salary

Label Salary

Type * BigInteger

List □

7. Click **Create** to add the new field, or click **Create and continue** to add the new field and continue adding other fields.

NOTE

To edit a field, select the field row and use the **general properties** on the right side of the screen.

3.2. CREATING AND RUNNING A TEST SCENARIO

You can create test scenarios in Business Central to test the functionality of business rule data before deployment. A basic test scenario must have at least the following data:

- Related data objects
- **GIVEN** facts
- **EXPECT** results

With this data, the test scenario can validate the expected and actual results for that rule instance based on the defined facts. You can also add a **CALL METHOD** and any available **globals** to a test scenario, but these scenario settings are optional.

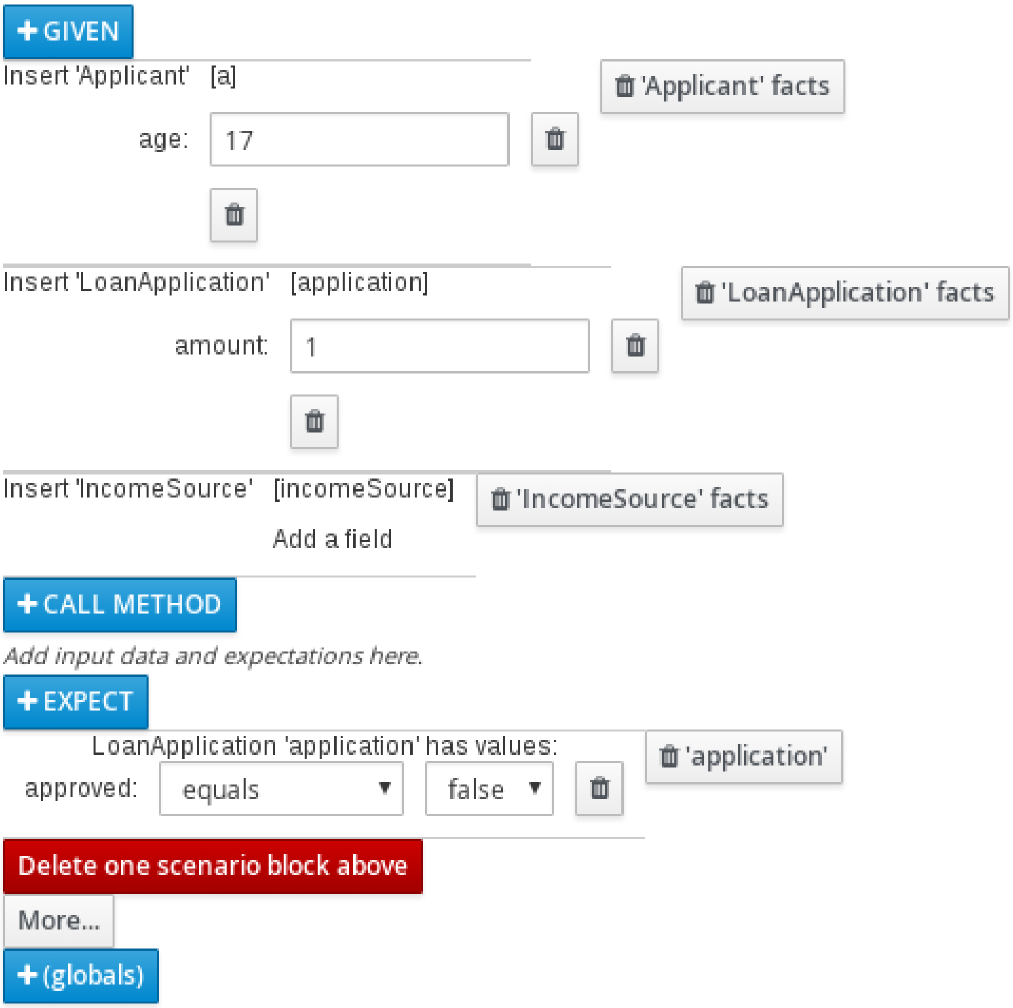
Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name.
2. Click **Add Asset** → **Test Scenario**.
3. Enter an informative **Test Scenario** name and select the appropriate **Package**. The package that you specify must be the same package where the required rule assets have been assigned or will be assigned. You can import data objects from any package into the asset's editor.
4. Click **Ok** to create the test scenario.
   The new test scenario is now listed in the **Test Scenarios** panel of the **Project Explorer**,

5. Click the **Data Objects** tab to verify that all data objects required for the rules that you want to test are listed. If not, click **New item** to import the needed data objects from other packages, or create data objects within your package.

6. After all data objects are in place, return to the **Model** tab of the test scenarios designer and define the **GIVEN** and **EXPECT** data for the scenario, based on the available data objects.

Figure 3.2. The test scenarios designer



Insert 'Applicant' [a]

age: 17

Insert 'LoanApplication' [application]

amount: 1

Insert 'IncomeSource' [incomeSource]

Add input data and expectations here.

LoanApplication 'application' has values:

approved: equals false

The **GIVEN** section defines the input facts for the test. For example, if an **Underage** rule in the project declines loan applications for applicants under the age of 21, then the **GIVEN** facts in the test scenario could be **Applicant** with **age** set to some integer less than 21.

The **EXPECT** section defines the expected results based on the **GIVEN** input facts. That is, **GIVEN** the input facts, **EXPECT** these other facts to be valid or entire rules to be activated. For example, with the given facts of an applicant under the age of 21 in the scenario, the **EXPECT** results could be **LoanApplication** with **approved** set to **false** (as a result of the underage applicant), or could be the activation of the **Underage** rule as a whole.

7. Optionally, add a **CALL METHOD** and any **globals** to the test scenario:

   - **CALL METHOD:** Use this to invoke a method from another fact when the rule execution is initiated. Click **CALL METHOD**, select a fact, and click ▶ to select the method to invoke. You can invoke any Java class methods (such as methods from an ArrayList) from the Java library or from a JAR that was imported for the project (if applicable).

   - **globals:** Use this to add any global variables in the project that you want to validate in the test scenario. Click **globals** to select the variable to be validated, and then in the test scenarios designer, click the global name and define field values to be applied to the global variable. If no global variables are available, then they must be created as new assets in Business Central. Global variables are named objects that are visible to the process engine but are different from the objects for facts. Changes in the object of a global do not trigger the re-evaluation of rules.

8. Click **More** at the bottom of the test scenarios designer to add other data blocks to the same scenario file as needed.

9. After you have defined all **GIVEN**, **EXPECT**, and other data for the scenario, click **Save** in the test scenarios designer to save your work.

10. Click **Run scenario** in the upper-right corner to run this **.scenario** file, or click **Run all scenarios** to run all saved **.scenario** files in the project package (if there are multiple). Although the **Run scenario** option does not require the individual **.scenario** file to be saved, the **Run all scenarios** option does require all **.scenario** files to be saved.
    If the test fails, address any problems described in the **Alerts** message at the bottom of the window, review all components in the scenario, and try again to validate the scenario until the scenario passes.

11. Click **Save** in the test scenarios designer to save your work after all changes are complete.

For more details about adding GIVEN facts to test scenarios, see Section 3.2.1, “Adding GIVEN facts in test scenarios”.

For more details about adding EXPECT results to test scenarios, see Section 3.2.2, “Adding EXPECT results in test scenarios”.

3.2.1. Adding GIVEN facts in test scenarios

The **GIVEN** section defines input facts for the test. For example, if an **Underage** rule in the project declines loan applications for applicants under the age of 21, then the **GIVEN** facts in the test scenario could be **Applicant** with **age** set to some integer less than 21.

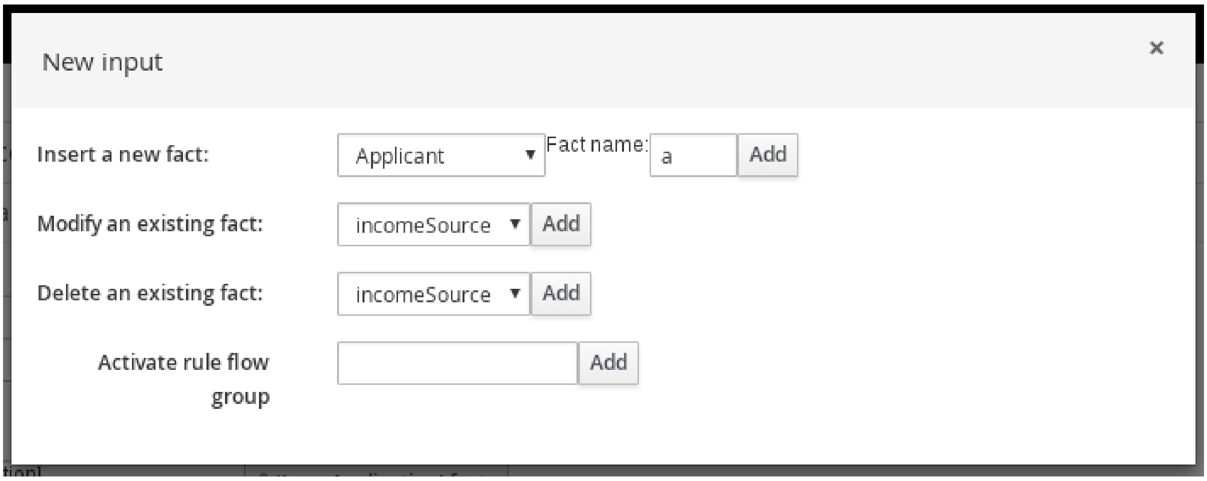
Prerequisite

All data objects required for your test scenario have been created or imported and are listed in the **Data Objects** tab of the test scenarios designer.

Procedure

1. In the test scenarios designer, click **GIVEN** to open the **New input** window with the available facts.

Figure 3.3. Add GIVEN input to the test scenario



New input

Insert a new fact: Applicant ▾ Fact name: a Add

Modify an existing fact: incomeSource ▾ Add

Delete an existing fact: incomeSource ▾ Add

Activate rule flow group Add

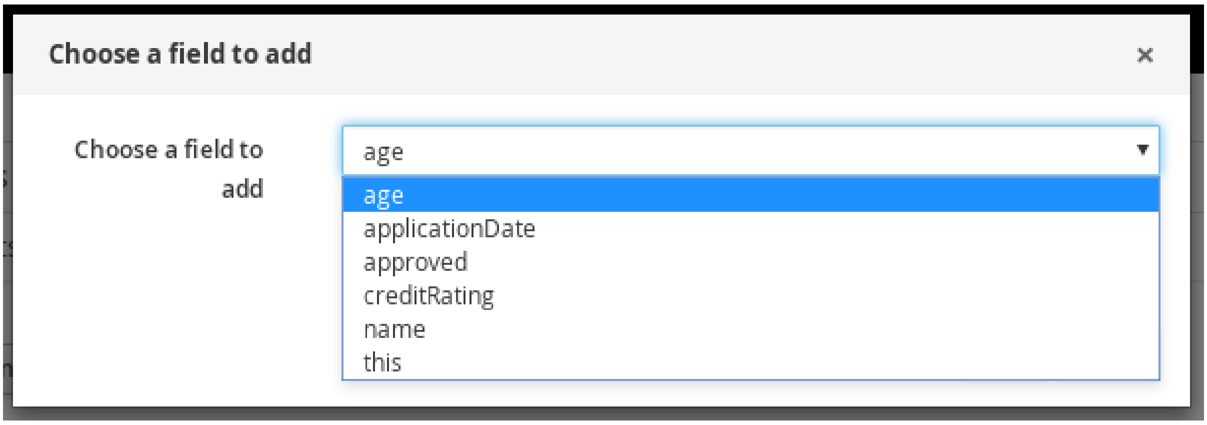
The list includes the following options, depending on the data objects available in the **Data Objects** tab of the test scenarios designer:

- **Insert a new fact:** Use this to add a fact and modify its field values. Enter a variable for the fact as the **Fact name**.
- **Modify an existing fact:** (Appears only after another fact has been added.) Use this to specify a previously inserted fact to be modified in the process engine between executions of the scenario.
- **Delete an existing fact:** (Appears only after another fact has been added.) Use this to specify a previously inserted fact to be deleted from the process engine between executions of the scenario.
- **Activate rule flow group:** Use this to specify a rule flow group to be activated so that all rules within that group can be tested.

2. Choose a fact for the desired input option and click **Add**. For example, set **Insert a new fact:** to **Applicant** and enter **a** or **app** or any other variable for the **Fact name**.

3. Click the fact in the test scenarios designer and select the field to be modified.

Figure 3.4. Modify a fact field



Choose a field to add

Choose a field to add: age ▾

age
applicationDate
approved
creditRating
name
this

4. Click the edit icon ( ) and select from the following field values:

- **Literal value:** Creates an open field in which you enter a specific literal value.

- **Bound variable:** Sets the value of the field to the fact bound to a selected variable. The field type must match the bound variable type.

- **Create new fact:** Enables you to create a new fact and assign it as a field value of the parent fact. Then you can click the child fact in the test scenarios designer and likewise assign field values or nest other facts similarly.

5. Continue adding any other **GIVEN** input data for the scenario and click **Save** in the test scenarios designer to save your work.

3.2.2. Adding EXPECT results in test scenarios

The **EXPECT** section defines the expected results based on the **GIVEN** input facts. That is, **GIVEN** the input facts, **EXPECT** other specified facts to be valid or entire rules to be activated. For example, with the given facts of an applicant under the age of 21 in the scenario, the **EXPECT** results could be **LoanApplication** with **approved** set to **false** (as a result of the underage applicant), or could be the activation of the **Underage** rule as a whole.

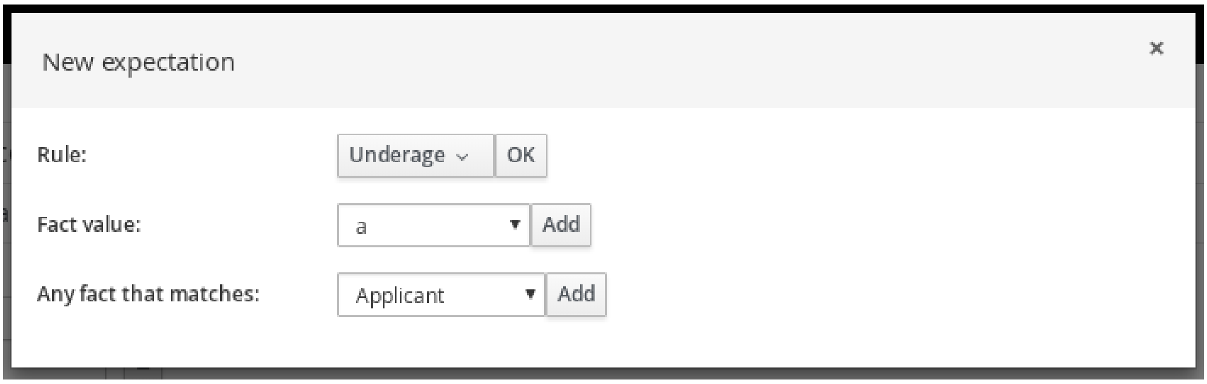
Prerequisite

All data objects required for your test scenario have been created or imported and are listed in the **Data Objects** tab of the test scenarios designer.

Procedure

1. In the test scenarios designer, click **EXPECT** to open the **New expectation** window with the available facts.

Figure 3.5. Add EXPECT results to the test scenario



New expectation

Rule: Underage

Fact value: a

Any fact that matches: Applicant

The list includes the following options, depending on the data in the **GIVEN** section and the data objects available in the **Data Objects** tab of the test scenarios designer:

- **Rule:** Use this to specify a particular rule in the project that is expected to be activated as a result of the **GIVEN** input. Type the name of a rule that is expected to be activated or select it from the list of rules, and then in the test scenarios designer, specify the number of times the rule should be activated.

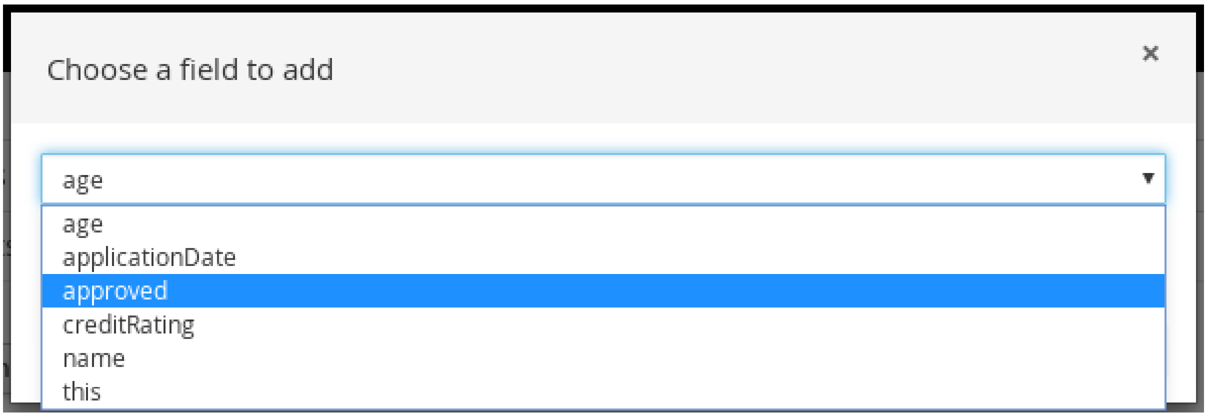
- **Fact value:** Use this to select a fact and define values for it that are expected to be valid as a result of the facts defined in the **GIVEN** section. The facts are listed by the **Fact name** previously defined for the **GIVEN** input.

- **Any fact that matches:** Use this to validate that at least one fact with the specified values exists as a result of the **GIVEN** input.

2. Choose a fact for the desired expectation (such as **Fact value: application**) and click **Add** or **OK**.

3. Click the fact in the test scenarios designer and select the field to be added and modified.

   Figure 3.6. Modify a fact field

4. Set the field values to what is expected to be valid as a result of the **GIVEN** input (such as **approved** | **equals** | **false**).

5. Continue adding any other **EXPECT** input data for the scenario and click **Save** in the test scenarios designer to save your work.

6. After you have defined and saved all **GIVEN**, **EXPECT**, and other data for the scenario, click **Run scenario** in the upper-right corner to run this **.scenario** file, or click **Run all scenarios** to run all saved **.scenario** files in the project package (if there are multiple). Although the **Run scenario** option does not require the individual **.scenario** file to be saved, the **Run all scenarios** option does require all **.scenario** files to be saved.
   If the test fails, address any problems described in the **Alerts** message at the bottom of the window, review all components in the scenario, and try again to validate the scenario until the scenario passes.

7. Click **Save** in the test scenarios designer to save your work after all changes are complete.

CHAPTER 4. USING THE TEST SCENARIO (PREVIEW) EDITOR IN BUSINESS CENTRAL

Red Hat Process Automation Manager ships with two editors for you to create and run test scenarios to validate your business rules and data. The difference between them is that the **Test Scenario** editor is supported by Red Hat for production use whereas **Test Scenario (Preview)** editor is a technology preview only and is not supported for production use. The sections covered in this chapter will help you understand how to use the preview editor to create and run test scenarios.

4.1. TEST SCENARIO (PREVIEW) EDITOR

Red Hat Process Automation Manager now comes with a new **Test Scenario (Preview)** editor. The preview editor provides a new and enhanced user experience when you test or validate your business rules and data.

IMPORTANT

This feature is for Technology Preview only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and Red Hat does not recommend to use them for production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information on Red Hat Technology Preview features, see Technology Preview Features Scope.

4.1.1. Enabling Test Scenario (Preview) editor

The **Test Scenario (Preview)** editor is available as a *Technology Preview* feature in Red Hat Process Automation Manager and is disabled by default.

Procedure

1. Log in to Business Central.
2. Click **Admin** from the top-right main panel and then click **Roles**.
3. From **All roles**, select a role.

   NOTE

   Any changes you make to roles or groups affect all of the users associated with that role or group.

4. In the **Permissions** section, expand **Editors**.
5. Click **Read** in front of the **Test Scenario (Preview)** editor to enable it.
6. Click **Save**.

NOTE

Currently, the preview editor works only using the default KIE session.

4.1.2. Test Scenario (Preview) editor layout

The preview editor introduces a tabular layout which helps you in defining a scenario template and all the associated test cases. The editor layout consists of a table which has a header and the individual rows. The header consists of three parts, the **GIVEN** and **EXPECT** row, a row with instances, and a row with corresponding fields. The header is also known as test scenario template and the individual rows are called test scenarios definitions.

The test scenario template or header has the following two parts:

- **Given** data objects and their fields - represents the input information
- **Expected** data objects and their fields - represents the objects and their fields whose exact values are checked based on the given information and which also constitutes the expected result.

The test scenarios definitions represent the separate test cases of a template.

You can access the **Project Explorer** from the left panel of the editor whereas from the right panel you can access the **Test Tools** and the **Test Report**. You can use the **Test Tools** to configure the data object mappings (**Test Editor**) or for accessing the cheat sheet (**Scenario Cheatsheet**) that contain notes which you can use as reference.

4.1.3. Importing data objects

Test Scenario (Preview) editor loads all data objects that are located in the same package as the test scenario. You can view all the data objects from the **Data Objects** tab in the editor. The loaded data objects are also displayed in the **Test Editor** tab in the **Test Tools** panel.

You need to close and reopen the preview editor in case the data objects change (for example, when a new data object is created or when an existing one is deleted). Select a data object from the list to display its fields and the field types.

In case you want to use a data object located in a different package than the test scenario, you need to import the data object first. Follow the procedure below to import a data object.

Procedure

1. Go to **Project Explorer** panel in the **Test Scenario (Preview)** editor.
2. From **Test Scenario (Preview)**, select a test scenario.
3. Select **Data Objects** tab and click **New Item**.
4. In the **Add import** window, choose the data object from the drop-down list.
5. Click **Ok** and then **Save**.
6. Close and reopen the **Test Scenario (Preview)** editor to view the new data object from the data objects list.

4.1.4. Importing a test scenario

You can import an existing test scenario using the **Import Asset** button in the **Asset** tab from the project view.

Procedure

1. In Business Central, go to **Menu → Design → Projects** and click the project name.
2. From the project's **Asset** tab, click **Import Asset**.
3. In the **Create new Import Asset** window,
   - Enter the name of the import asset.
   - Select the package from the **Package** drop-down list.
   - From **Please select a file to upload**, click **Choose File...** to browse to test scenario file.
4. Select the file and click **Open**.
5. Click **Ok** and the test scenario opens in the review editor.

4.2. BASIC OPERATIONS USING THE TEST SCENARIO (PREVIEW) EDITOR

You can perform the following operations using the **Test Scenario (Preview)** editor:

- Save a test scenario
- Delete a test scenario
- Rename a test scenario
- Copy a test scenario
- Run a test scenario
- Download a test scenario
- Switch between different versions of a test scenario
- View or Hide Alerts

4.2.1. Saving a test scenario

You can save a test scenario at any time while creating a test scenario template or defining the test scenarios.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Save**.
2. On the **Confirm Save** window,
   a. If you wish to add a comment regarding the test scenario, click **add a comment**.
   b. Click **Save** again.

A message stating that the test scenario was saved successfully appears on the screen.

4.2.2. Deleting a test scenario

You can delete existing test scenarios which were created using the **Test Scenario (Preview)** editor.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Delete**.
2. In the **Confirm Delete** window,
   - To add a comment regarding the deletion of the test scenario, click **add a comment**
   - Click **Delete**.

A message stating that the test scenario was deleted successfully appears on the screen.

4.2.3. Renaming a test scenario

You can rename existing test scenarios by using the **Rename** button from the upper-right toolbar in the editor.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Rename**.
2. In the **Rename Asset** window,
   a. Enter a name in the **Asset Name** field.
   b. If you wish to add a comment, click **add a comment**
   c. Click **Rename**.

      NOTE

      Clicking on **Rename** simply renames the test scenario file.

   d. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Save**.
   e. In the **Confirm Save** window,
      i. If you wish to add a comment, click **add a comment**
      ii. Click **Save** again.
   f. Alternately, you could click **Save and Rename** to save and rename the scenario at the same time.

A message stating that the test scenario was renamed successfully appears on the screen.

4.2.4. Copying a test scenario

You can copy an existing test scenario to the same package or to some other package by using the **Copy** button from the upper-right toolbar.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Copy**.
2. In the **Make a Copy** window,
   a. Enter a name in the **New Name** field.
   b. Select the package you want to copy the test scenario to.
   c. Optionally, to add a comment, click **add a comment**.
   d. Click **Make a Copy**.

A message stating that the test scenario was copied successfully appears on the screen.

4.2.5. Downloading a test scenario

You can download a copy of the test scenario to your local machine for future reference or as backup.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Download**.

The .scesim file is downloaded to your local machine.

4.2.6. Switching between versions of a test scenario

Business Central provides you the ability to switch between the various versions of a test scenario. Every time you save the scenario, a new version of the scenario is listed under **Latest Versions**. To use this feature, you must save the test scenario file at least once.

Procedure

1. From the **Test Scenario (Preview)** editor toolbar on the upper-right, click **Latest Version**. All the versions of the file are listed under **Latest Version**, if they exist.
2. Click the version you want to work on.
   The selected version of the test scenario opens in the **Test Scenario (Preview)** editor.
3. From the editor toolbar, click **Restore**.
4. In the **Confirm Restore**,
   a. To add a comment, click **add a comment**.
   b. Click **Restore** to confirm.

A message stating that the selected version has been reloaded successfully in the editor appears on the screen.

4.2.7. View/hide the alerts panel

In the **Test Scenario (Preview)** editor, the message panel appears at the bottom of the editor with the test results and the **Alert** messages. From the preview editor toolbar on the upper-right, click **Hide Alerts**/**View Alerts** to enable or disable the reporting panel.

4.3. CONTEXTUAL MENU OPTIONS

The **Test Scenario (Preview)** editor provides contextual menu options, which enables you to perform basic operations on the table such as adding, deleting, and, duplicating rows and columns. To use the contextual menus, you need to right-click a table element. Menu options differ based on the table element you select.

Table 4.1. Contextual menu options

Table element	Cell label	Available context menu options
Header	# & Scenario description	Insert top row, Insert bottom row
	GIVEN & EXPECTED	Insert leftmost column, Insert rightmost column, Insert top row, Insert bottom row
	INSTANCE 1, INSTANCE 2 & PROPERTY 1, PROPERTY 2	Insert column left, Insert column right, Delete column, Insert top row, Insert bottom row
Rows		Insert column left, Insert column right, Delete column, Insert top row, Insert bottom row, Insert row above, Insert row below, Duplicate row, Delete row

Table 4.2. Description of table interactions

Table interaction	Description
Insert top row	Inserts a new row at the top of the table.
Insert bottom row	Inserts a new row at the bottom of the table.
Insert leftmost column	Inserts a new leftmost column (in either the GIVEN or EXPECTED section of the table based on user selection).
Insert rightmost column	Inserts a new rightmost column (in either the GIVEN or EXPECTED section of the table based on user selection).
Insert column left	Inserts a new column to the left of the selected column. The new column is of the same type as the selected column (in either the GIVEN or EXPECTED section of the table based on user selection).
Insert column right	Inserts a new column to the right of the selected column. The new column is of the same type as the selected column (in either the GIVEN or EXPECTED section of the table based on user selection).
Delete column	Deletes the selected column.
Insert row above	Inserts a new row above the selected row.

Table interaction	Description
Insert row below	Inserts a new row below the selected row.
Duplicate row	Duplicates the selected row.
Delete row	Deletes the selected row.

The **Insert column right** or **Insert column left** context menu options behave differently,

- if the selected column does not have a type defined, a new column without a type is added.
- if the selected column has a type defined, either a new empty column or a column with the parent instance type is created.
- if the action is performed from an instance header, a new column without a type is created.
- if the action is performed from a property header, a new column with the parent instance type is created.

4.4. TEST SCENARIO TEMPLATE

Before specifying test scenario definitions, you need to create a test scenario template. The header of the test scenario table defines the template for each scenario. You need to set the types of the instance and property headers for both the GIVEN and EXPECTED sections.

4.4.1. Concept of aliases

In the Test Scenario (Preview) editor, once you map a header cell with a data object, the data object is removed from the **Test Editor** tab in the **Test Tools** panel. You can re-map a data object to another header cell by using the concept of **Alias**. Aliases enable you to specify multiple instances of the same data object. Ensure that the aliases are uniquely named. By doing so, the data object again becomes available in the **Test Editor** tab for remapping.

4.4.2. Creating a test scenario template

Create a test scenario template by following the procedure below to validate your rules and data.

Procedure

1. In Business Central, go to **Menu → Design → Projects** and click the project for which you want to create the test scenario.
2. Click **Add Asset → Test Scenario (Preview)**.
3. Enter a **Test Scenario** name and select the appropriate **Package**. The package you select must contain all the required data objects and rule assets have been assigned or will be assigned.
4. Click **Ok** to create and open the test scenario in the **Test Scenario (Preview)** editor.
5. To map the **GIVEN** column header to a data object,
    a. Click an instance header in the **GIVEN** section.

   b. Select the data object from the **Test Editor** tab.

   c. Click **Add**.

6. To insert more properties of the data object, right-click the property header and select **Insert column right** or **Insert column left** as required.

7. To map a data object field to a property cell,

   a. Click a property cell.

   b. Select the data object field from the **Test Editor** tab.

   c. Click **Add**.

8. To map the **EXPECTED** column header to a data object,

   a. Click an instance header in the **EXPECTED** section.

   b. Select the data object from the **Test Editor** tab.

   c. Click **Add**.

9. To insert more properties of the data object, right-click the property header and select **Insert column right** or **Insert column left** as required.

10. To map a data object field to a property cell,

    a. Click a property cell.

    b. Select the data object field from the **Test Editor** tab.

    c. Click **Add**.

Use the contextual menu to add or remove columns as needed.

After you have created and mapped both the **GIVEN** & **EXPECTED** columns to a data object & its properties, you have to define the test scenario which comes next.

4.5. DEFINING A TEST SCENARIO

After creating a test scenario template you have to define the test scenario next. The rows of the test scenario table define the individual test scenarios. A test scenario has a unique index number, description, set of input values (the given values), and a set of output values (the expected values).

Prerequisite

Ensure that the test scenario template exists for the selected test scenario.

Procedure

1. Open the test scenario in the **Test Scenario (Preview)** editor.

2. Enter a description of the test scenario and fill in required values in each cell of the row. Use the contextual menu to add or remove rows as required.

After defining the test scenario, you can run the test next.

4.5.1. Expressions syntax

The following test scenario definition expressions are supported by **Test Scenario (Preview)** editor:

Table 4.3. Description of expressions syntax

Operator	Description
=	Specifies equal to a value. This is default for all columns and is the only operator supported by the GIVEN column.
=, =!, <>	Specifies inequality of a value. This operator can be combined with other operators.
<, >, < =, >=	Specifies a comparison: less than, greater than, less or equals than, and greater or equals than.
[value1, value2, value3]	Specifies a list of values. If one or more values are valid, the scenario definition is evaluated as true.
expression1; expression2; expression3	Specifies a list of expressions. If all expressions are valid, the scenario definition is evaluated as true.

NOTE

An empty cell is equal to null. To define an empty string, use **=**.

Table 4.4. Example expressions

Expression	Description
–1	The expected value is equal to –1.
< 0	The expected value is less than 0.
! > 0	The expected value is not greater than 0.
[–1, 0, 1]	The expected value is equal to either –1 or 0 or 1.
<> [1, –1]	The expected value is neither equal to 1 nor –1.
! 100; 0	The expected value is not equal to 100 but is equal to 0.
!= < 0; <> > 1	The expected value is neither less than 0 nor greater than 1.
<> < = 0; >= 1	The expected value is neither less than 0 nor equal to 0 but is greater than or equal to 1.

NOTE

You can refer to the supported commands and syntax in the **Scenario Cheatsheet** tab on the **Test Tools** panel.

4.6. RUNNING A TEST SCENARIO

After creating a test scenario template and defining the test scenarios, you can run the tests in the preview editor to validate your business rules and data.

Procedure

1. On the **Test Scenario (Preview)** editor, click **Run Test**.
2. The **Test Report** panel pops out from the right of the editor and displays the test result overview and the scenario status.
3. After the tests are executed, click **View Alerts** to open the **Alerts** section for error messages. Alternatively, you can also view the test result messages from the **Alerts** panel at the bottom of the editor.
4. In case the tests fails, review the error messages. Messages with a red icon indicate test failures where as a green icon indicates successful test.
   In the preview editor, erroneous rows or cells are highlighted in red.
5. Make the necessary changes and run the test again till the scenario passes.

4.7. CREATING TEST SCENARIO USING THE SAMPLE MORTGAGES PROJECT

This chapter illustrates creating and executing a test scenario from the sample **Mortgages** project shipped with Business Central using the **Test Scenario (Preview)** editor. The test scenario example in this chapter is based on the **Pricing loans** guided decision table from the **Mortgages** project.

Prerequisite

Ensure that the Test Scenario (Preview) editor is enabled.

Procedure

1. In Business Central, go to **Menu → Design → Projects** and click **Mortgages**.
2. If the project is not listed under **Projects**, from **MySpace**, click the three dots (⋮) in the upper-right corner of the page.
3. Click **Try Samples → Mortgages → OK**.
   The Assets window appears.
4. Click **Add Asset → Test Scenario (Preview)**.
5. Enter **scenario_pricing_loans** as the **Test Scenario** name and select the default **mortgages.mortgages** package from the **Package** drop-down list.
   The package you select must contain all the required rule assets.

6. Click **Ok** to create and open the test scenario in the **Test Scenario (Preview)** editor.

7. Expand **Project Explorer** and verify the following:

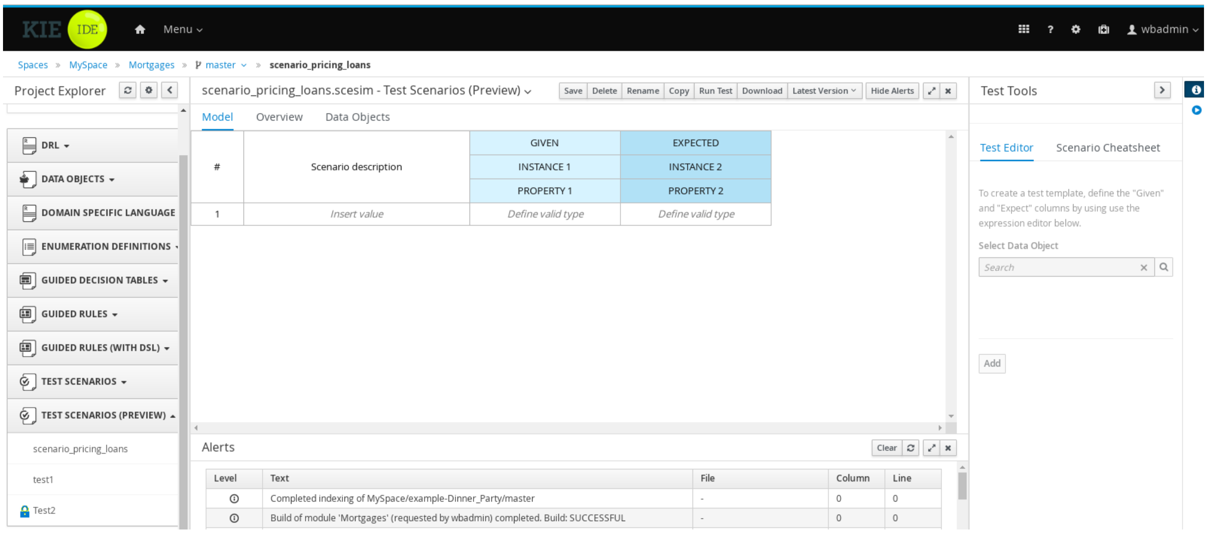
   - **Applicant**, **Bankruptcy**, **IncomeSource**, and **LoanApplication** data objects exist.

   - **Pricing loans** guide decision table exists.

   - Verify that the new test scenario is listed under **Test Scenario (Preview)**

8. After verifying that everything is in place, return to the **Model** tab of the test scenarios designer and define the **GIVEN** and **EXPECT** data for the scenario, based on the available data objects.

   Figure 4.1. A blank Test Scenario (Preview) editor

#	Scenario description	GIVEN	EXPECTED
		INSTANCE 1	INSTANCE 2
		PROPERTY 1	PROPERTY 2
1	*Insert value*	*Define valid type*	*Define valid type*

   To create a test template, define the "Given" and "Expect" columns by using use the expression editor below.

Level	Text	File	Column	Line
ⓘ	Completed indexing of MySpace/example-Dinner_Party/master	-	0	0
ⓘ	Build of module 'Mortgages' (requested by wbadmin) completed. Build: SUCCESSFUL	-	0	0

9. Define the **GIVEN** column details,

   a. Click the cell named **INSTANCE 1** under the **GIVEN** column header.

   b. From **Test Tools → Test Editor**, select **LoanApplication** data object.

   c. Click **Add**.

10. To create properties of the data object, right-click the property cell and select **Insert column right** or **Insert column left** as required. For this example, you need to create two more property cells under the **GIVEN** column.

11. Click the first property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

    b. Click **amount** and then **Add** to map the data object field to the property cell.

12. Click the second property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

    b. Click **deposit** and then **Add**.

13. Click the third property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

b. Click **lengthYears** and then **Add**.

14. Right-click **LoanApplication** header cell and select **Insert column right**. A new **GIVEN** column to the right is created.

15. Click the new header cell,

    a. From **Test Tools → Test Editor**, select the **IncomeSource** data object.

    b. Click **Add** to map the data object to the header cell.

16. Click the property cell below **IncomeSource**,

    a. From **Test Tools → Test Editor**, select and expand the **IncomeSource** data object.

    b. Click **type** and then **Add** to map the data object field to the property cell.
    You have now defined all the **GIVEN** column cells.

17. Next, define the **EXPECTED** column details,

    a. Click the cell named **INSTANCE 2** under the **EXPECTED** column header.

    b. From **Test Tools → Test Editor**, select **LoanApplication** data object.

    c. Click **Add**.

18. To create properties of the data object, right-click the property cell and select **Insert column right** or **Insert column left** as required. Create two more property cells under the **EXPECTED** column.

19. Click the first property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

    b. Click **approved** and then **Add** to map the data object field to the property cell.

20. Click the second property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

    b. Click **insuranceCost** and then **Add**.

21. Click the third property cell,

    a. From **Test Tools → Test Editor**, select and expand the **LoanApplication** data object.

    b. Click **approvedRate** and then **Add**.

22. Now for defining the test scenario, enter the following data in the first row:

    - Enter **Row 1 test scenario** as the **Scenario Description**, **150000** as the **amount**, **19000** as the **deposit**, **30** as the **lenghtYears**, and **Asset** as the **type** for the **GIVEN** column values.

    - Enter **true** as **approved** and **0** as the **insuranceCost** for the **EXPECTED** column values.

23. Next enter the following data in the second row:

    - Enter **Row 2 test scenario** as the **Scenario Description**, **100002** as the **amount**, **2999** as the **deposit**, **20** as the **lenghtYears**, and **Job** as the **type** for the **GIVEN** column values.

- Enter **true** as **approved** and **10** as the **insuranceCost** for the **EXPECTED** column values.

24. After you have defined all **GIVEN**, **EXPECT**, and other data for the scenario, click **Save** in the test scenarios designer to save your work.

25. Click **Run Test** in the upper-right corner to run the **.scesim** file.
    The test result is displayed in the **Test Report** panel. Click **View Alerts** to display messages from the **Alerts** section. If a test fails, refer to the messages in the **Alerts** section at the bottom of the window, review and correct all components in the scenario, and try again to validate the scenario until the scenario passes.

26. Click **Save** in the test scenarios designer to save your work after you have made all necessary changes.

CHAPTER 5. NEXT STEPS

Packaging and deploying a Red Hat Process Automation Manager project

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Tuesday, May 28, 2019.