



Red Hat Process Automation Manager 7.0

Getting started with decision services

Red Hat Process Automation Manager 7.0 Getting started with decision services

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

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

In this tutorial, you will create and test a mortgage application decision service using Red Hat Process Automation Manager 7.0. The procedures in this document are based on the included Mortgage Process sample project.

Table of Contents

PREFACE	3
CHAPTER 1. OPENING THE MORTGAGE PROCESS SAMPLE PROJECT	4
CHAPTER 2. DATA OBJECTS	5
2.1. VIEWING THE MORTGAGE PROCESS DATA OBJECTS	5
2.1.1. Viewing the Applicant data object	5
2.1.2. Viewing the Application data object	6
2.1.3. Viewing the Property data object	6
2.1.4. Viewing the ValidationErrorDO data object	7
CHAPTER 3. GUIDED RULES	8
3.1. DEFINING BUSINESS RULES	8
3.1.1. Creating the Validate Down Payment guided rule	8
3.1.2. Defining the Validate Down Payment guided rule conditions	8
3.1.3. Defining the error conditions	9
3.1.3.1. Defining the ValidationErrorDO	10
3.1.3.2. Defining the retractValidationErr	10
CHAPTER 4. GUIDED DECISION TABLES	12
4.1. CREATING DECISION TABLES	12
4.1.1. Viewing the Mortgage Decision Table	12
CHAPTER 5. TEST SCENARIOS	13
5.1. TESTING THE LOAN APPLICATION SCENARIO	13
5.1.1. Create the Qualify Test Scenario asset	13
5.1.2. Input the Applicant facts and values	13
5.1.3. Input the Property facts and values	14
5.1.4. Input the Application facts and values	14
5.1.5. Input the expected values	15
CHAPTER 6. ADDITIONAL RESOURCES	17
APPENDIX A. VERSIONING INFORMATION	18

PREFACE

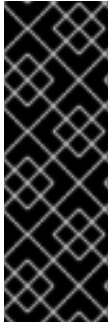
As a business analyst or rules developer, you can use Business Central in Red Hat Process Automation Manager to design a variety of decision services. In this tutorial, you will create and test a mortgage application decision service.

Prerequisites

- Red Hat JBoss Enterprise Application Platform 7.1.0 is installed. See [Red Hat JBoss EAP 7.1.0 Installation Guide](#).
- Red Hat Process Automation Manager is installed. For more information, see [Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss EAP 7.1](#).
- Red Hat Process Automation Manager is running and you can log in to Business Central with the **admin** role.

CHAPTER 1. OPENING THE MORTGAGE PROCESS SAMPLE PROJECT

The **Mortgage Process** sample project consists of predefined data objects, guided decision tables, guided rules, forms, and a business process. Using the sample project provides a quick way to get acclimated with Red Hat Process Automation Manager. In a real business scenario, you would create all of the assets by providing data that is specific to your business requirements.



IMPORTANT

The business process application example includes features that are Technology Preview only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs), might not be functionally complete, and are not recommended 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 about Red Hat Technology Preview support, see [Technology Preview Features Support Scope](#).

Procedure

Navigate to the Mortgage Process sample project to view the predefined assets.

1. Log in to Business Central and click **Menu** → **Design** → **Projects**.
2. Click the three vertical dots in the upper-right corner of the screen and select **Try Samples**.
3. Select **Mortgage Process** and click **Ok**.

The **Assets** view of the project opens.

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 service are based on.

For more information about creating data objects, see ["Creating data objects"](#) in *Designing a decision service using guided decision tables*.

2.1. VIEWING THE MORTGAGE PROCESS DATA OBJECTS

This tutorial utilizes the predefined data objects in the **Mortgage Process** sample project.

The **Mortgage Process** data model is composed of four data objects:

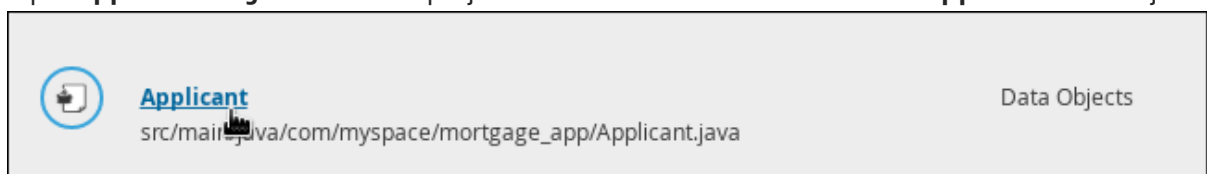
- **Applicant**
- **Application**
- **Property**
- **ValidationErrorDo**

2.1.1. Viewing the Applicant data object

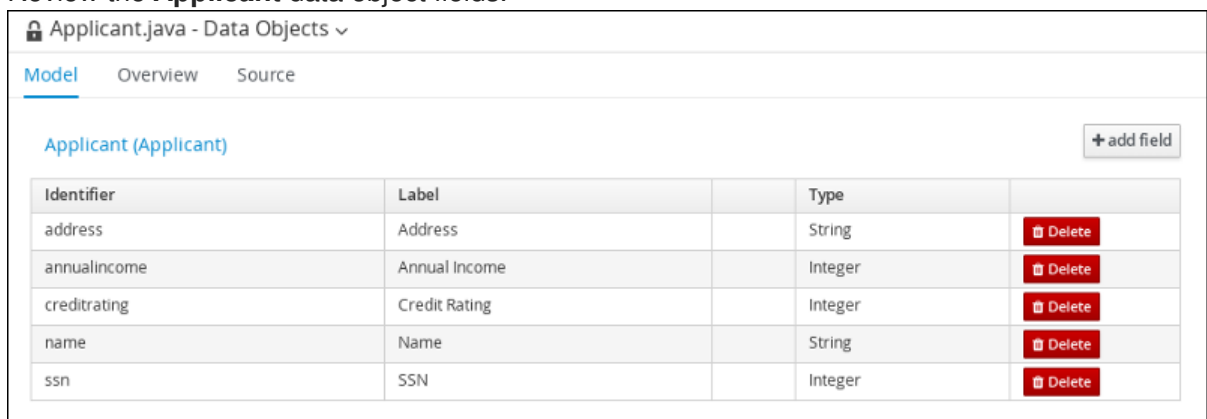
Follow these steps to familiarize yourself with the predefined **Applicant** data object.

Procedure

1. Click **Menu** → **Design** → **Projects**, then click **Mortgage Process**.
2. Input **Applicant.java** in to the project's asset search box and click the **Applicant** data object.



3. Review the **Applicant** data object fields.

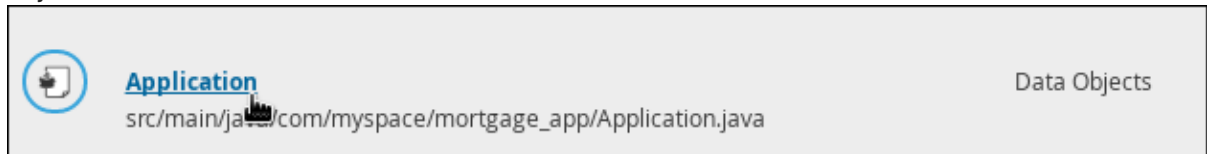


2.1.2. Viewing the Application data object

Follow these steps to familiarize yourself with the predefined **Application** data object.

Procedure

1. Click **Menu** → **Design** → **Projects**, then click **Mortgage Process**.
2. Input **Application.java** in to the project's asset search box and click the **Application** data object.



3. Review the **Application** data object fields.

Application.java - Data Objects ▾

[Model](#) [Overview](#) [Source](#)

Application + add field

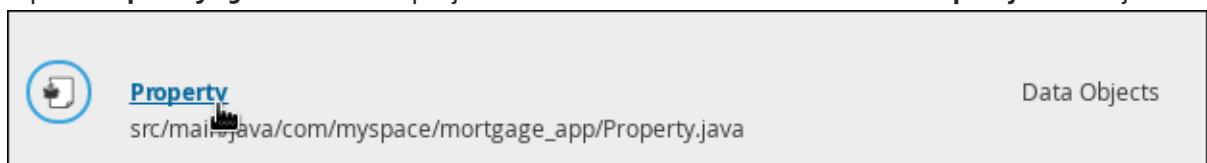
Identifier	Label	Type	
amortization	Years of amortization	Integer	Delete
applicant	Applicant	Applicant	Delete
downpayment	Down Payment	Integer	Delete
errors	Error details	Validation Error	Delete
mortgageamount	Mortgage amount	Integer	Delete
property	Property	Property	Delete

2.1.3. Viewing the Property data object

Follow these steps to familiarize yourself with the predefined **Property** data object.

Procedure

1. Click **Menu** → **Design** → **Projects**, then click **Mortgage Process**.
2. Input **Property.java** in to the project's asset search box and click the **Property** data object.



3. Review the **Property** data object fields.


Property.java - Data Objects ▾			
Model Overview Source			
Property (Property)			+ add field
Identifier	Label	Type	
address	Address of property	String	Delete
age	Age of property	Integer	Delete
locale	Locale	String	Delete
saleprice	Sale Price	Integer	Delete

2.1.4. Viewing the ValidationErrorDO data object

Follow these steps to familiarize yourself with the predefined **ValidationErrorDO** data object.

Procedure

1. Click **Menu** → **Design** → **Projects**, then click **Mortgage Process**.
2. Input **ValidationErrorDO.java** in to the project's asset search box and click the **ValidationErrorDO** data object.

	ValidationErrorDO src/main/java/com/myspace/mortgage_app/ValidationErrorDO.java	Data Objects
--	---	--------------

3. Review the **Property** data object fields.

ValidationErrorDO.java - Data Objects ▾			
Model Overview Source			
Validation Error (ValidationErrorDO)			+ add field
Identifier	Label	Type	
error	Error and cause	String	Delete

CHAPTER 3. GUIDED RULES

Guided rules are business rules that you create in a UI-based guided rules designer in Business Central that leads you through the rule-creation process. The guided rules designer provides fields and options for acceptable input based on the data objects for the rule being defined. The guided rules that you define are compiled into Drools Rule Language (DRL) rules as with all other rule assets.

All data objects related to a guided rule must be in the same project package as the guided rule. Assets in the same package are imported by default. After you create the necessary data objects and the guided rule, you can use the **Data Objects** tab of the guided rules designer to verify that all required data objects are listed or to import other existing data objects by adding a **New item**.

3.1. DEFINING BUSINESS RULES

Define business rules in Red Hat Process Automation Manager using the Guided Rule wizard.

Related information

For more information about Guided business rules, see [Designing a decision service using guided rules](#).

3.1.1. Creating the Validate Down Payment guided rule


Procedure

1. Log in to Business Central and click **Menu** → **Design** → **Projects**, then **Mortgage Process**.
2. Click **Add Asset** → **Guided Rule**, then enter:
 - **Guided Rule: Validate Down Payment**
 - **Package: com.myspace.mortgage_app**
3. Click **Ok** to open the Guided Rule Editor.


3.1.2. Defining the Validate Down Payment guided rule conditions

The conditions that you specify in this section are used to determine whether or not the down payment meets the specified requirements.

Procedure

1. Click  next to the **WHEN** label to open the **Add a condition to the rule** window. Then, select **Application...** and click **Ok**.
2. Click the **There is an Application** label and select **Any of (Or)** from **Multiple field constraints**.
3. Click the **There is an Application [app] with: any of the following:** label, then select **downpayment** from **Add a restriction on a field**.
4. Click **--- please choose ---**, then select **equal to**.

5. Click  , select **Literal value**, and enter **0**.

6. Click the **There is an Application [app] with: any of the following:** label, then from the **Add a restriction on a field** select **downpayment**.
7. Click **--- please choose ---**, then select **greater than or equal to**.
8. Click  , then click **Expression editor**.
9. From the **Choose...** menu, select **app**.
10. From the **Choose...** menu, select **downpayment**.

3.1.3. Defining the error conditions



You must define the conditions and results of the data that is received and processed. In this section, you will define the following two conditions:

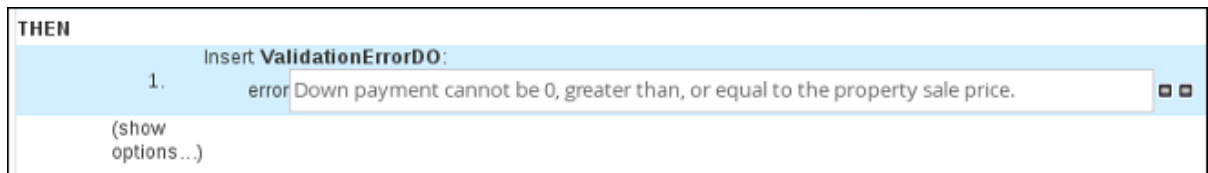
- ValidationErrorDO
- retractValidationErr

3.1.3.1. Defining the ValidationErrorDO

Specify the error message that will be presented to the loan applicant.

Procedure

1. Click  next to the **THEN** label. Select **Insert fact ValidationErrorDO**, and click **Ok**.
2. Click the **Insert ValidationErrorDO** label and select **error** from the **Add field** pull-down menu.
3. Click , then click **Literal value**.
4. In the **error** field, enter: **Down payment cannot be 0, greater than, or equal to the property sale price..**





THEN

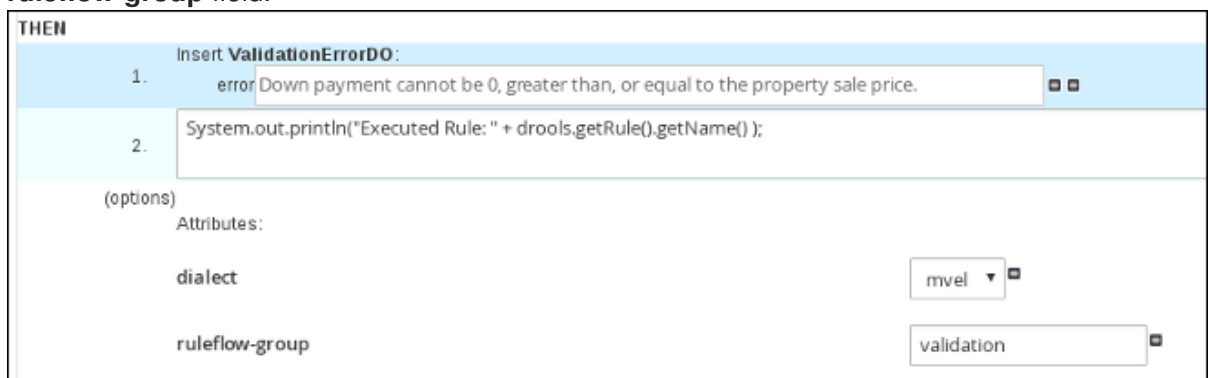
1. Insert **ValidationErrorDO**:

error Down payment cannot be 0, greater than, or equal to the property sale price.

(show options...)

5. Click  next to the **THEN** label. Select **Add free form DRL** and click **Ok**.
6. Enter the following in the *Add free form DRL...* text box:


```
System.out.println("Executed Rule: " + drools.getRule().getName() );
```
7. Click **(show options...)**.
8. Click  next to the **(options)** label to open the **Add an option to the rule** window.
9. Select **ruleflow-group** from the **Attribute** drop-down menu and enter **validation** in the **ruleflow-group** field.



THEN

1. Insert **ValidationErrorDO**:

error Down payment cannot be 0, greater than, or equal to the property sale price.

2. System.out.println("Executed Rule: " + drools.getRule().getName());

(options)

Attributes:



dialect mvel

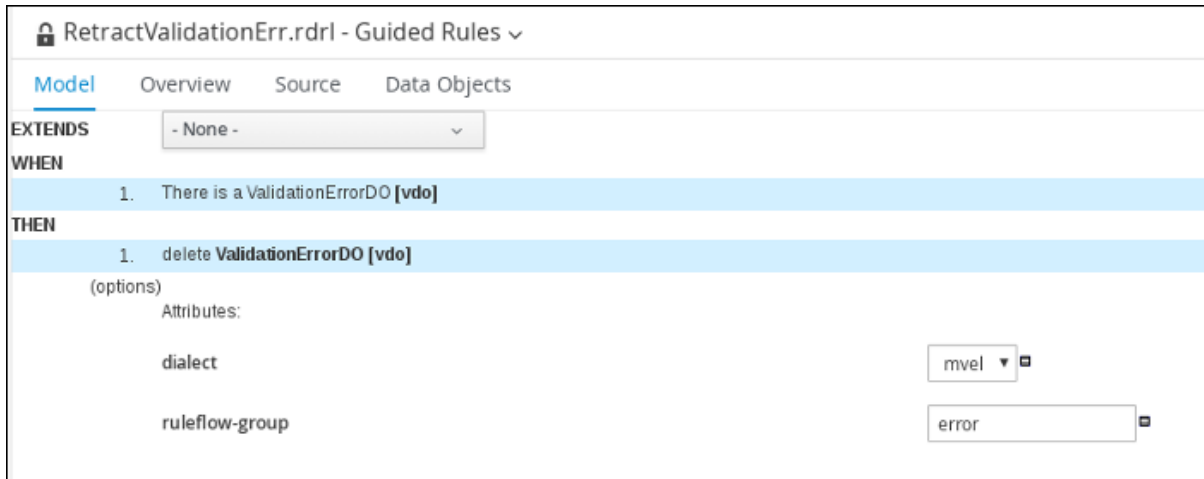
ruleflow-group validation

3.1.3.2. Defining the retractValidationErr

Define the error trigger details.

Procedure

1. Click  next to the **WHEN** label to open the **Add a condition to the rule** window. Then, select **ValidationErrorDO** and click **Ok**.
2. Click **There is a ValidationErrorDO** to open the **Modify constraints for ValidationErrorDO** window, enter **vdo** in the variable name field, and click **Set**.
3. Click  next to the **THEN** label. Select **Delete vdo**, and click **Ok**.
4. Click **(show options)**.
5. Enter **error** in the **ruleflow-group** field.



RetractValidationErr.rdr1 - Guided Rules ▾

Model Overview Source Data Objects

EXTENDS - None - ▾

WHEN

1. There is a ValidationErrorDO [vdo]

THEN

1. delete ValidationErrorDO [vdo]

(options)

Attributes:

dialect mvel ▾

ruleflow-group error

6. Click **Save**, then click **Save** to confirm your changes.

CHAPTER 4. GUIDED DECISION TABLES

Guided decision tables are a wizard-led alternative to uploaded decision table spreadsheets for defining business rules in a tabular format. With guided decision tables, you are led by a UI-based wizard in Business Central that helps you define rule attributes, metadata, conditions, and actions based on specified data objects in your project. After you create your guided decision tables, the rules you defined are compiled into Drools Rule Language (DRL) rules as with all other rule assets.

All data objects related to a guided decision table must be in the same project package as the guided decision table. Assets in the same package are imported by default. After you create the necessary data objects and the guided decision table, you can use the **Data Objects** tab of the guided decision tables designer to verify that all required data objects are listed or to import other existing data objects by adding a **New item**.

4.1. CREATING DECISION TABLES

Decision tables are a way to generate rules driven from the data entered into a spreadsheet. Decision tables are created in Red Hat Process Automation Manager using the Guided Decision Table wizard.

Prerequisites

The business rules have been defined. For more information, see [Section 3.1, “Defining Business Rules”](#).

4.1.1. Viewing the Mortgage Decision Table

The goal of this chapter is to introduce you to the **Mortgage Decision Table**. For this tutorial, you will not create and set the decision table conditions. Instead, review the values and the conditions that are already defined in the **Mortgage Process** sample project’s **MortgageDecisionTable** Guided Decision Tables asset.

Procedure

1. Log in to Business Central and click **Menu** → **Design** → **Projects**, then **Mortgage Process**.
2. Scroll down and click the **MortgageDecisionTable** Guided Decision Tables asset.

Mortgage Decision Table								
#	Description	ruleflow-group	f1 : Applicant		f2 : Property			f3 : Application
			Annual Income (Greater Than)	Annual Income (Less Than or Equal To)	Sale Price (Less Than)	Property Age (Less Than)	Location	Mortgage Amount
1		mortgagecalculation	100000	200000	300000	5	Urban	200000
2		mortgagecalculation	50000	99999	100000	10	Rural	100000

CHAPTER 5. TEST SCENARIOS

Test scenarios in Red Hat Process Automation Manager enable you to validate the functionality of rules, models, and events before deploying them into production. A test scenario uses data for conditions that resemble an instance of your fact or project model. This data is matched against a given set of rules and 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.

After you run all test scenarios, the status of the scenarios is reported in a **Reporting** panel.

Test scenarios can be executed one at a time or as a group. The group execution contains all the scenarios from one package. Test scenarios are independent, so that one scenario cannot affect or modify the other.

5.1. TESTING THE LOAN APPLICATION SCENARIO

Test the loan application scenario using the data that you specified when you created the mortgage guided decision table. For this test, you will input data for a property in an urban location with a sales price of less than \$300000.00 that is under five years old.

Prerequisites

- The **Mortgage Process** project has been created
- The following data objects have been created:
 - Applicant
 - Property
 - ValidationErrorDo
 - Application
- The mortgage guided decision table has been created

5.1.1. Create the Qualify Test Scenario asset

Before you input the fact and test data values, you must create a new Test Scenario asset.


Procedure

1. Log in to Business Central. Click **Menu** → **Design** → **Projects**, then **Mortgage Process**.
2. Click **Projects** → **Add Asset** → **Test Scenario**.
3. In the **Create new Test Scenario window** wizard, enter the following values:
 - a. **Test Scenario:** **Qualify**
 - b. **Package:** select `com.myspace.mortgage_app`
4. Click **Ok**.

5.1.2. Input the Applicant facts and values

Input the Applicant test data based on the guided decision table values.




Procedure

1. Click **+GIVEN** to open the **New input** window.
2. Select **Applicant** from the **Insert a new fact** pull-down menu.
3. Enter **app** in the **Fact name** field and click **Add**.
4. Click **Add a field** to open the **Choose a field to add** window and select **annualincome** from the **Choose a field to add** pull-down menu and click **OK**.
5. Click  next to **annualincome**.
6. Click **Literal value** in the **Field value** window and enter **150000** in the **annualincome** field.

5.1.3. Input the Property facts and values

Input the Property test data based on the guided decision table values.

Procedure


1. Click **+GIVEN** to open the **New input** window.
2. Select **Property** from the **Insert a new fact** pull-down menu.
3. Enter **prop** in the **Fact name** field and click **Add**.
4. Click **Add a field** to open the **Choose a field to add** window and select **saleprice** from the **Choose a field to add** pull-down menu and click **OK**.
5. Click  next to **saleprice**.
6. Click **Literal value** in the **Field value** window and enter **250000** in the **saleprice** field.
7. Click **Insert 'Property'[prop]** and select **age** from the **Choose a field to add** pull-down menu in the **Choose a field to add** window, and click **OK**.
8. Click , then click **Literal value** and enter **4** in the **age** field.
9. Click **Insert 'Property'[prop]** and select **locale** from the **Choose a field to add** pull-down menu in the **Choose a field to add** window, and click **OK**.
10. Click , then click **Literal value** and enter **Urban** in the **type** field.

5.1.4. Input the Application facts and values

Input the Application test data based on the guided decision table values.

Procedure

1. Click **+GIVEN** to open the **New input** window.

2. Select **Application** from the **Insert a new fact** pull-down menu.
3. Enter **application** in the **Fact name** field and click **Add**.
4. Click **Add a field** to open the **Choose a field to add** window and select **mortgageamount** from the **Choose a field to add** pull-down menu and click **OK**.
5. Click  next to **mortgageamount**.
6. Click **Literal value** in the **Field value** window, then enter **250000** in the **mortgageamount** field.

5.1.5. Input the expected values

Input the application's expected values.

Procedure

1. Click **+Expect** to open the **New expectation** window.
2. Expand the **Fact value** pull-down menu, select **Application**, and click **Add**.
3. Click **Application 'application has values:**
4. Select **mortgageamount** from the **Choose a field to add** pull-down menu and click **OK**.
5. Enter **200000** in the **mortgageamount: equals** field.
6. Click **Save**, then click **Save** to confirm your changes.
7. Click **Run scenario**.

Figure 5.1. Application test scenario screen

Qualify.scenario - Test Scenarios

Model

Overview

Data Objects

Settings

Audit log

+ GIVEN

Insert 'Applicant' [app]

annualincome: 150000

'Applicant' facts

Insert 'Property' [prop]

saleprice: 250000

age: 4

locale: Urban

'Property' facts

Insert 'Application' [application]

mortgageamount: 200000

'Application' facts

+ CALL METHOD

Add input data and expectations here.

+ EXPECT

Application 'application' has values:

mortgageamount: equals

200000

'application'

Delete one scenario block above

More...

+ (globals)

CHAPTER 6. ADDITIONAL RESOURCES

- *Designing a decision service using guided rules*
- *Designing a decision service using guided decision tables*
- *Testing a decision service using test scenarios*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Monday, October 1, 2018.