

Digital Business Automation Blue Demos 2018

Lab instructions for 5. ODM sub-scenario

Scenario designers: Guilhem Molines, Marie-Françoise Lim-Meffre, Nicolas Mériau

Instruction authors: Nicolas Mériau, Margaux Chatelain, Jeffrey Mariasine

Instruction editors: Nicolas Mériau, Moussa Yattara, Marie-Françoise Lim-Meffre,
Olaf Hahnl

Version: 1.0

Date: 1-Aug-2018 – 4:00 PM

Table of Content

| | | |
|-------|--|----|
| 0 | Introduction..... | 5 |
| 0.1 | Audience..... | 6 |
| 0.2 | Prerequisites | 6 |
| 0.3 | Objectives..... | 6 |
| 0.4 | About this Sub-scenario | 7 |
| 1 | Exercise: Get started with ODM and Decision Composer | 8 |
| 1.1 | Get started with ODM and Decision Composer – Introduction..... | 8 |
| 1.2 | Get started with ODM and Decision Composer – Step by Step Instructions | 8 |
| 1.2.1 | What is Operational Decision Manager? | 8 |
| 1.2.2 | What is Decision Server?..... | 9 |
| 1.2.3 | What is Decision Center?..... | 10 |
| 1.2.4 | What is IBM Decision Composer? | 12 |
| 1.2.5 | How do all these components interact? | 14 |
| 1.3 | Get started with ODM and Decision Composer – Verification Instructions | 15 |
| 1.4 | Get started with ODM and Decision Composer – Summary | 15 |
| 2 | Exercise: Get Started with this Sub-Scenario | 16 |
| 2.1 | Get Started with this Sub-Scenario – Introduction | 16 |
| 2.2 | Get Started with this Sub-Scenario – Step by Step Instructions | 17 |
| 2.3 | Get Started with this Sub-Scenario – Verification Instructions | 31 |
| 2.4 | Get Started with this Sub-Scenario – Summary..... | 32 |
| 2.5 | Get Started with this Sub-Scenario – Troubleshooting..... | 33 |
| 2.5.1 | Issue with starting the Sample Server..... | 33 |
| 3 | Exercise: Set up DBA Jam 1 Decision Service in Rule Designer | 35 |
| 3.1 | Set up DBA Jam 1 Decision Service in Rule Designer – Introduction..... | 35 |
| 3.2 | Set up DBA Jam 1 Decision Service in Rule Designer – Step by Step Instructions..... | 36 |
| 3.3 | Set up DBA Jam 1 Decision Service in Rule Designer – Verification Instructions..... | 41 |
| 3.4 | Set up DBA Jam 1 Decision Service in Rule Designer – Summary | 42 |
| 3.5 | Set up DBA Jam 1 Decision Service in Rule Designer – Troubleshooting | 42 |
| 3.5.1 | DBA Jam 1 > deployment > Local_Deployment_Configuration is in error | |
| | 42 | |
| 4 | Exercise: Finalize DBA Jam 1 Decision Service in Business console | 44 |
| 4.1 | Finalize DBA Jam 1 Decision Service in Business console – Introduction..... | 44 |
| 4.2 | Finalize DBA Jam 1 Decision Service in Business console – Step by Step Instructions..... | 44 |
| 4.2.1 | Understand the Business console environment..... | 44 |
| 4.2.2 | Edit the ruleflow | 50 |
| 4.2.3 | Create a rule and edit a decision table | 51 |
| 4.2.4 | Test your rules in the Business console | 53 |
| 4.2.5 | Deploy the decision service to Rule Execution Server | 54 |
| 4.2.6 | Verify the deployment in Rule Execution Server | 57 |
| 4.2.7 | (Optional) Create a RuleApp archive for your DBA Jam 1 decision service | |
| | 58 | |

| | | |
|-------|---|-----|
| 4.2.8 | Test the deployed DBA Jam 1 decision service in Rule Execution Server | 59 |
| 4.3 | Finalize DBA Jam 1 Decision Service in Business console – Verification Instructions..... | 60 |
| 4.4 | Finalize DBA Jam 1 Decision Service in Business console – Summary | 61 |
| 4.5 | Finalize DBA Jam 1 Decision Service in Business console – Troubleshooting | 61 |
| 4.5.1 | Error when deploying the decision service to RES | 61 |
| 4.5.2 | Clicking a tab of the Business console does not open the tab but instead downloads an HTML file | 62 |
| 5 | Exercise: Define DBA Jam 2 Decision Service in Decision Composer | 63 |
| 5.1 | Define DBA Jam 2 Decision Service in Decision Composer – Introduction | 63 |
| 5.2 | Define DBA Jam 2 Decision Service in Decision Composer – Step by Step Instructions..... | 63 |
| 5.2.1 | Explore Decision Composer | 63 |
| 5.2.2 | Test the current state of the decision service | 64 |
| 5.2.3 | Update the decision service in Decision Composer..... | 65 |
| 5.2.4 | Test your DBA Jam 2 decision service in Decision Composer | 72 |
| 5.2.5 | (Optional) Create a RuleApp archive for your DBA Jam 2 decision service | 74 |
| 5.2.6 | (Optional) Deploy your DBA Jam 2 decision service in Rule Execution Server | 75 |
| 5.3 | Define DBA Jam 2 Decision Service in Decision Composer – Verification Instructions..... | 77 |
| 5.4 | Define DBA Jam 2 Decision Service in Decision Composer – Summary..... | 77 |
| 5.5 | Define DBA Jam 2 Decision Service in Decision Composer – Troubleshooting | 77 |
| 5.5.1 | Warning icons visible near "Design" or in a decision node..... | 77 |
| 6 | Exercise: Integrate within the Workflow solution | 79 |
| 6.1 | Integrate within the Workflow solution - Introduction | 79 |
| 6.1.1 | Move to another environment | 79 |
| 6.1.2 | Stay in your current environment | 79 |
| 6.2 | Integrate within the Workflow solution – Step by Step Instructions – in VM 5 - ODM | 81 |
| 6.3 | Integrate within the Workflow solution – Step by Step Instructions – in VM 3 – Workflow | 83 |
| 6.3.1 | Resume the VM 1 – ECM and VM 3 – Workflow virtual machines | 83 |
| 6.3.2 | Delete the existing External Services | 84 |
| 6.3.3 | Using ODM JSON files, create two new external services | 86 |
| 6.3.4 | In the Service Flow, use the new external services | 90 |
| 6.3.5 | Take a snapshot and activate it | 95 |
| 6.3.6 | Use your process from within the Case solution | 97 |
| 6.4 | Integrate the Rule Apps with Workflow – Verification Instructions | 99 |
| 6.5 | Integrate within the Workflow solution – Summary | 103 |
| 6.6 | Integrate within the Workflow solution – Troubleshooting..... | 103 |
| 6.6.1 | Troubleshooting connection issues in BlueDemos | 103 |
| 7 | Files available as helpers | 105 |

| | | |
|-------|--|-----|
| 7.1 | C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario folder of the virtual machine | 105 |
| 7.1.1 | __Backup | 105 |
| 7.1.2 | Jam-1-Start..... | 105 |
| 7.1.3 | Jam-1-Answer | 105 |
| 7.1.4 | Jam-2..... | 106 |
| 7.2 | Box folder 5. ODM sub-scenario | 106 |
| 7.3 | Box folder "5. ODM Sub-Scenario – Backup" | 106 |
| 7.3.1 | Helper.txt (Box) | 106 |
| 7.3.2 | Jam-1 (Box) | 107 |
| 7.3.3 | Jam-2 (Box) | 108 |

0 Introduction

Within the ODM sub-scenario, you will create two decision services required to automate some parts that are used in the overall Mortgage Application solution. These services are consumed by the Service Flow that is implemented as part of the Business Automation Workflow or the short Workflow sub-scenario.

These two decision services will allow the Workflow solution to determine some scores about a requested loan and whether this loan can be approved, and at which conditions.

- **DBA Jam 1:** This first decision service will compute a score for the borrower and the loan, based on their basic data provided by the requestor, to help with approving the loan (with DBA Jam 2 decision service). You will set up this computational decision service by following the [bottom-up approach](#) with ODM.

In details: This decision service uses the customer's name, age, yearly income, and the loan's number of monthly payments, amount and loan to value ratio as inputs, to provide the customer's credit scores and corporate scores and the loan's monthly payment as outputs. The customer's corporate score and loan's monthly payment values will be used as input to the DBA Jam 2 decision service to determine the approval of the mortgage application.

- **DBA Jam 2:** This second decision service will decide whether the requested loan is approved, and if yes, what would be the insurance rate and whether this insurance is mandatory. You will set up this decision service by following the [top-down approach](#) with Decision Composer.

In details: This decision service uses the customer's name, corporate score, yearly income, and age, and the loan's property zip code, amount and monthly repayment values as inputs, to determine if the mortgage should be approved or rejected. If rejected, it also provides an explanation message.

These two services will be integrated into the Workflow solution, in the Workflow sub-scenario or in the End-to-End scenario, by replacing the internal decision within a Service Flow with REST calls to the two decision services defined in this scenario:

- The Service Flow will initially call the first decision service via a REST call, providing the basic data about the borrower and the loan. The execution of this decision service will return score data.
- The Service Flow will then call the second decision service, providing the basic data about the borrower and the loan plus the score data obtained with the first decision service. The execution of this decision service will return the final decisions: Is the loan approved? What is the insurance rate? Is the assurance mandatory?

Duration: 4 to 8 hours

All exercises in this sub-scenario are optional.

Depending on your level of interest in the individual components, you can opt, for each exercise, to either start from its determined 'starting point', or to skip it and instead use its 'solution point' to move on to the next exercise.

If you opt to skip an exercise, you must at minimum read its **Introduction** to understand what it is about and where you stand in the overall scenario. You could then go to the **Verification Instructions** part of the exercise to discover the 'solution point'. The exact duration of this sub-scenario might therefore highly vary, depending on your choices.

Important: Solutions for the two decision services are pre-deployed in the demo environment. It means that, even if you do not fully perform all the exercises in this ODM sub-scenario, you could still integrate these two decision services with the Workflow solution, by following the instructions that are given in the [Exercise: Integrate within the Workflow solution](#).

0.1 Audience

Everyone interested in the decision management space, especially technical people with IBM Operational Decision Manager, but also people without any background but interested in IBM Operational Decision Manager or IBM Decision Composer.

0.2 Prerequisites

To perform this ODM sub-scenario, you need:

- Access to Blue Demos
- Access to the Internet
- IBM ID
- Some experience in IBM Operational Decision Manager preferred but not mandatory

0.3 Objectives

This sub-scenario is subdivided into multiple exercises. Each exercise enables you to gain knowledge and skills with different parts of IBM Operational Decision Manager (ODM) and IBM Decision Composer:

1. Get familiar with IBM Operational Decision Manager (ODM) and with IBM Decision Composer (see [Exercise: Get started with ODM and Decision Composer](#)).
2. Get familiar with the various modules and major user-interfaces of ODM (Rule Designer, Rule Execution Server, Decision Center > Business console) and of Decision Composer (see [Exercise: Get started](#))
3. In Rule Designer, act like Paul, the IT person, and set up a complex rule-based decision project so Bea, the (non-technical) business user can later author all required decisions by using natural language (see [Exercise: Set up DBA Jam 1 Decision Service in Rule Designer](#)).
4. In Decision Center > Business console, act like Bea, the (non-technical) business user, and author decisions by using natural language and test these decisions by using real-case data, based on Paul's work (see [Exercise: Finalize DBA Jam 1 Decision Service in Business console](#)).

5. In Decision Composer, act like Bea, the (non-technical) business user, to create a simple rule model and simple rule decisions by using natural language, and test these decisions by using real-case data, without the need for Paul's help (see [Exercise: Define DBA Jam 2 Decision Service in Decision Composer](#)).
6. In Rule Execution Server console, identify the information that you need to integrate these rule-based decision services, created either with Operational Decision Manager or Decision Composer, within a solution that is based on IBM Business Automation Workflow (see [Exercise: Integrate within the Workflow solution](#)).

0.4 About this Sub-scenario

This sub-scenario is subdivided into multiple exercises. Each exercise consists of the following sections:

- 1) **Introduction:** Describes what you will learn and complete in this exercise, but also lists all needed data for the exercise so that experienced attendees can complete the exercise by only using this introduction section.
- 2) **Step by Step Instructions:** Guides you in easy to understand steps through the whole exercise. To be used by less experienced attendees.
- 3) **Verification Instructions:** Guides you through how to test and verify your implementation.
- 4) **Summary:** Summarizes what you have completed & learned.
- 5) (optional) **Troubleshooting:** Lists common potential issues for this exercise, and their resolutions.

Note: If you face an issue during an exercise, please have a look into the troubleshooting sections first. If your issue is not mentioned there, pls. reach out to the demo owners. Please make sure to indicate the sub-scenario (here: ODM), the exercise and section number (ex: 1.2 – Get Started with this Sub-Scenario – Step by Step Instructions), and the step where you have the issue (ex: step 1 – Access Blue Demos), so to ease identification and resolution of the issue. Thanks!

1 Exercise: Get started with ODM and Decision Composer

1.1 Get started with ODM and Decision Composer – Introduction

In this exercise, you will learn the main concepts in decision management, how they are applied with IBM Operational Decision Manager (ODM) and IBM Decision Composer, and identify the various modules of ODM.

1.2 Get started with ODM and Decision Composer – Step by Step Instructions

This exercise is mainly composed of lectures, which you can skip if you are familiar with decision management, ODM and Decision Composer.

If you are not familiar with decision management, take time now to read this introduction and learn about the main concepts in decision management, how they are applied with IBM Operational Decision Manager (ODM) and IBM Decision Composer, and identify the various modules of ODM.

You do not have to spend too much time on this introduction as this sub-scenario also contains many exercises where these concepts are reviewed and put in action. Instead, read it to know what it contains and later use this exercise as reference information during the sub-scenario as needed.

Most important terms for this sub-scenario are written in red and highlighted in yellow.

1.2.1 What is Operational Decision Manager?

IBM® Operational Decision Manager (ODM) is a comprehensive **decision automation platform** that helps you capture, analyze, automate and govern **rules-based business decisions**. Anyone can work with it—from IT to business-line leaders. In seconds, it can authorize a loan, decide on a promotion or detect a cross-sell opportunity with high precision and customization. With ODM, your applications will continuously remain up to date and well aligned with the changing business objectives of your organization. It is available in editions for both cloud and private cloud environments. For more details, see [Introducing Operational Decision Manager](#) in the IBM Knowledge Center.

ODM has several components, including the following two ones related to rule/decision management:

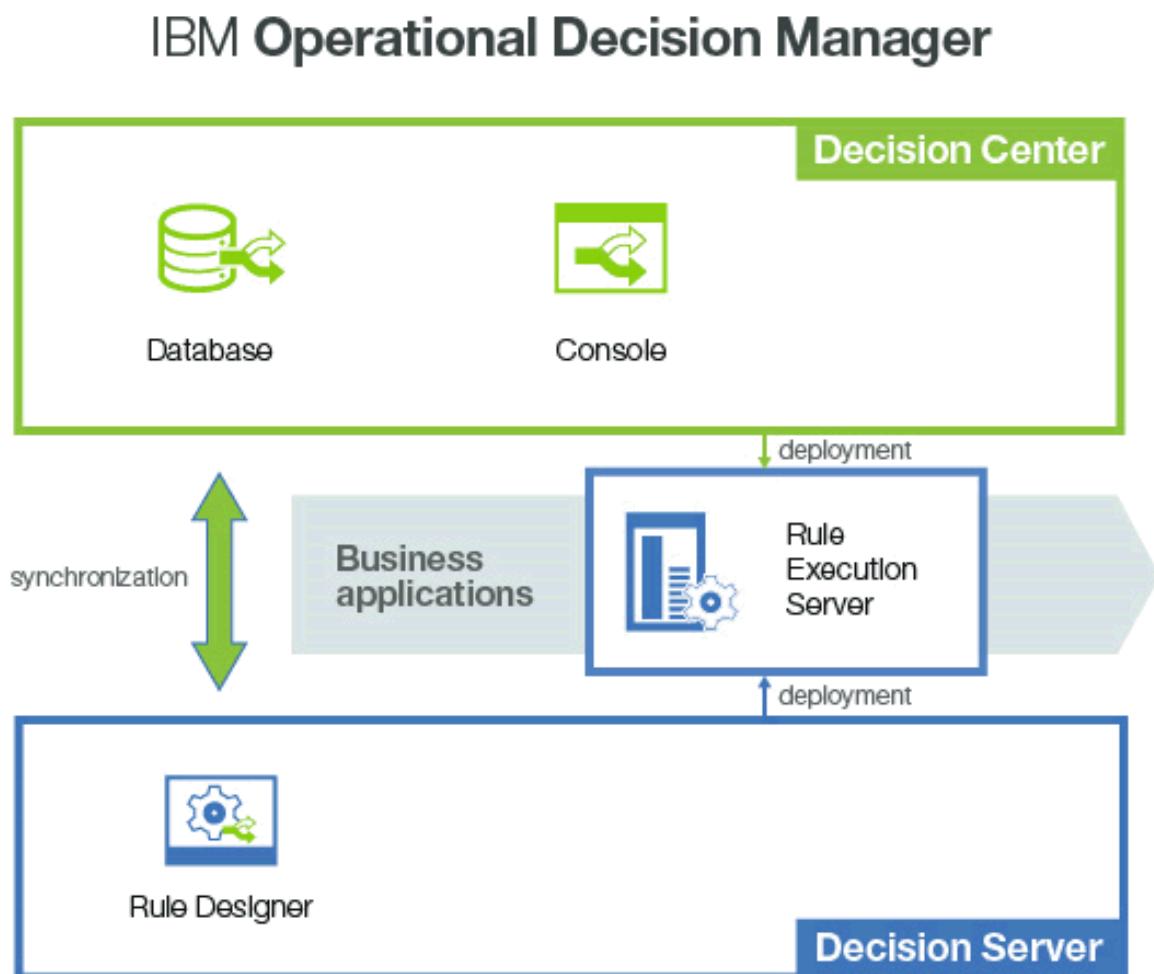
- **Decision Server (Rules):** Development and runtime components for a rule-based solution that automates the response of highly variable decisions required by client applications. Decision Server includes:
 - **Rule Designer:** An Eclipse-based development environment for designing, authoring, testing, and deploying decision services.

- **Rule Execution Server**: The runtime environment for running and monitoring decision services.
- **Decision Center**: Rule repository and collaborative web consoles for business users to author, manage, validate, and deploy rules, that is, to manage decisions that are directly based on organizational knowledge and best practices, with limited dependence on the IT department.

1.2.2 What is Decision Server?

At the core of a rule-based solution, you have a client application requesting a decision from a **decision service**. There can be many decision points required of the decision service by the client application. At each decision point, **business rules** packaged as **rulesets** are used to express policies for how decisions are made. A decision service is **deployed** to Rule Execution Server as a **RuleApp**. Each RuleApp contains one or more rulesets, each corresponding to a decision.

The following diagram illustrates how the different components interact:



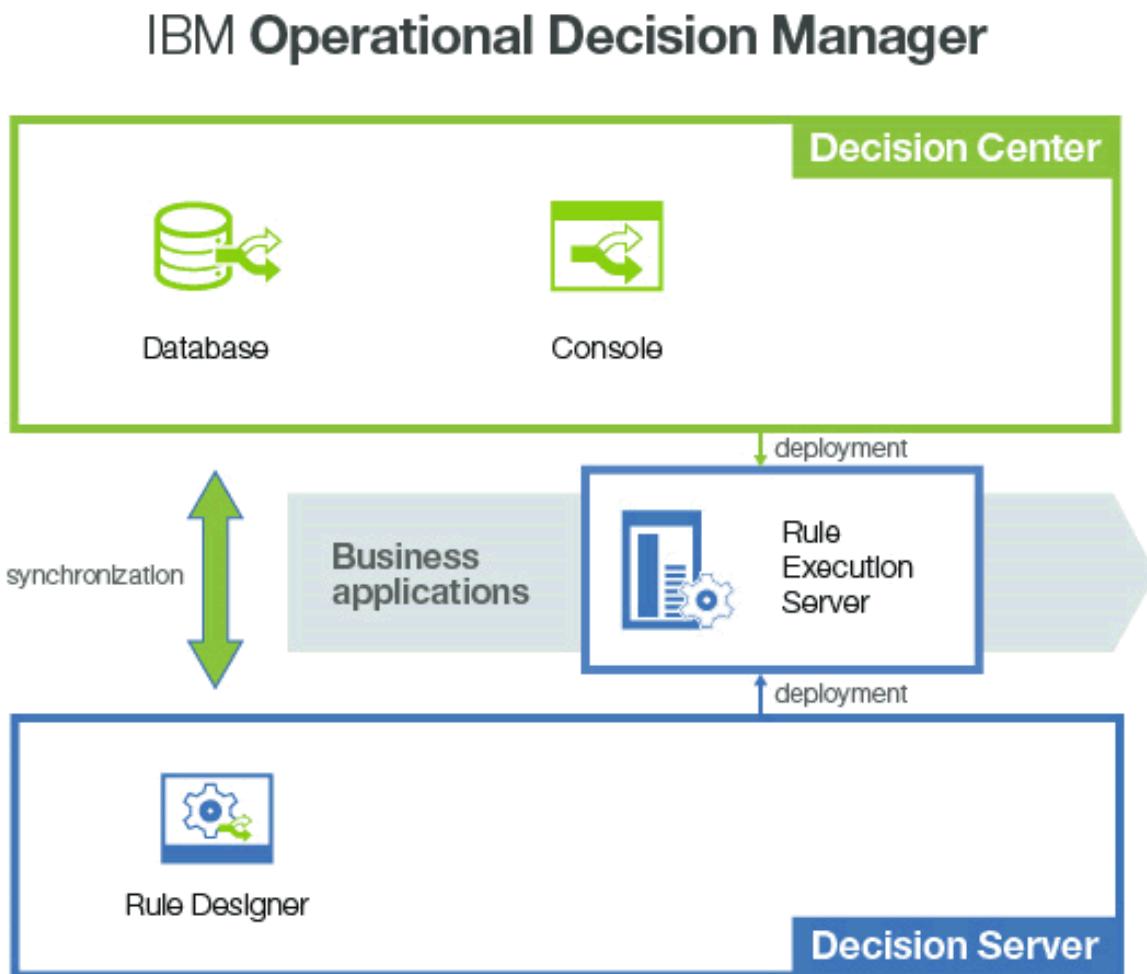
Decision Server also interacts with *Decision Center*, as detailed next.

1.2.3 What is Decision Center?

With Decision Center, you can perform the following operations:

- **Rule editing:** You **author rules** by using a **natural language syntax** in the rule editors that are provided in the Decision Center consoles. The editors make sure that you use the correct **business vocabulary** and comply with proper business rule syntax. There are two consoles available:
 - **Business console** - The preferred environment for business users to take advantage of change management and deployment of decision services (see [Governing rules with the Business console](#)).
 - **Enterprise console** - The environment for some advanced administrative features (see [Managing business rules with the Enterprise console](#)).
- **Synchronization:** IT and business users work on rules in different environments and save them to different locations. IT users typically work in source code control and business users save to the Decision Center repository. To enable collaboration between the development cycle and the business cycles of rules, you must **synchronize** the work that is done by both types of users. You must use the synchronization tool in Rule Designer.
- **Deploying rules:** When the development and testing of rules is complete, you can **deploy to the Rule Execution Server runtime environment**. In Decision Center, you can deploy as follows:
 - In the Business console, users with configuration manager rights can create and edit **deployment configurations in decision services**. These deployment configurations can then be used to deploy releases, change activities, and regular branches of the decision service.
 - In the Enterprise console, users with configuration manager rights can create and deploy RuleApps for classic rule projects. Other users can deploy existing RuleApps.
 - In the Decision Center REST API, you can build, download, or deploy a RuleApp for a deployment configuration.

The following diagram illustrates how Decision Center interacts with Decision Server (through synchronization) and the business applications (through deployment to Rule Execution Server):



- **Decision governance framework:** Decision Center offers a ready-to-use approach to change management and governance that is based on decision services, **releases**, and **activities**. You can also manage your own releases by working directly on **projects** and **branches**.
Note: Decision governance framework is a rich framework that this sub-scenario will not address. In this sub-scenario, you will instead manage your own releases directly with projects and branches.
- **Validating rules:** You can validate the behavior of rules or assess the effects of potential rule changes. Decision Center contains features to run tests and simulations:
 - **Tests:** Compare expected results with the actual results from applying rules to usage scenarios.
 - **Simulations:** Determine how changes to rules affect business.
- **Security access and permissions:** Decision Center provides a security mechanism to control access to branches of decision services and enable permissions on artifacts. This mechanism is based on Decision Center groups.

- **Administrator responsibilities:** Administrator tasks include setting up and administering the database, managing users, and improving the performance of Decision Center.
- **Customizing:** You can customize the Decision Center consoles, the rule model, and rule authoring extensions to adapt them to the needs of your business users.

1.2.4 What is IBM Decision Composer?

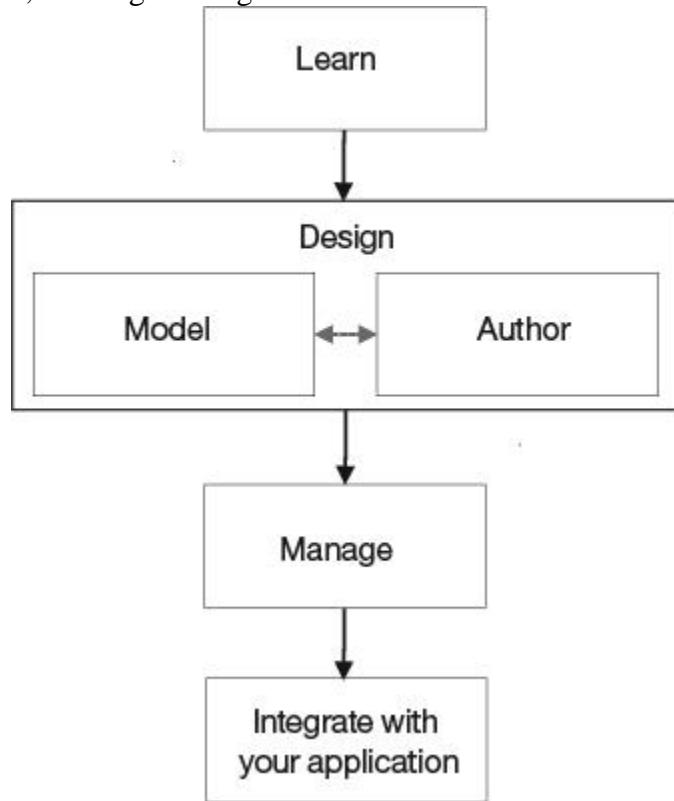
Decision Composer is a **web application** that you will use to **express your business logic in the form of diagrams** that are inspired from the [Decision Model and Notation \(DMN\)](#) standard.

As a difference to ODM, where everything starts with IT people implementing the business model and defining everything for business users to author their rules (**bottom-up approach**), in Decision Composer, a business expert can model, author and test a decision project that describes a decision to automate. IT specialists come into play only later, to deploy this project so it can be called by the client application where the decision must be made (**top-down approach**).

You can therefore use Decision Composer to create decision projects, and design and test your business logic for automation:

- After you learned the basics on the application and decision logic design, you can create the model that represents your business logic and write rules.
- You then test the project and iterate on its design until you are satisfied with it.
- You eventually download and share the project, and deploy it to use it in your client application.

The following diagram shows the typical work flow of work that you, as a user of Decision Composer, would go through.



In Decision Composer, a **decision project** has one **dependency diagram** that contains the decision logic. The **decision nodes** in the diagram apply to elements of the **object model** of this project.

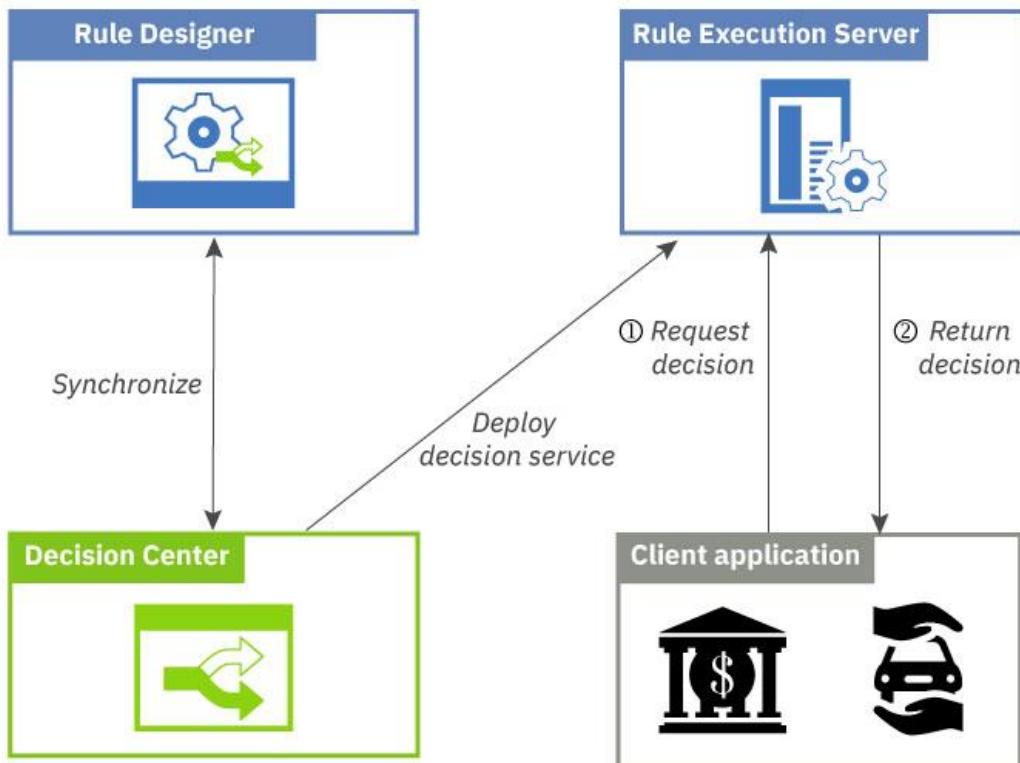
You manage projects all from the home page of Decision Composer.

To use a decision project in a client application, you must execute it. If you have a Rule Execution Server in your own environment, you can deploy the project to this server and execute it there. Otherwise, you can execute the project directly from Decision Composer.

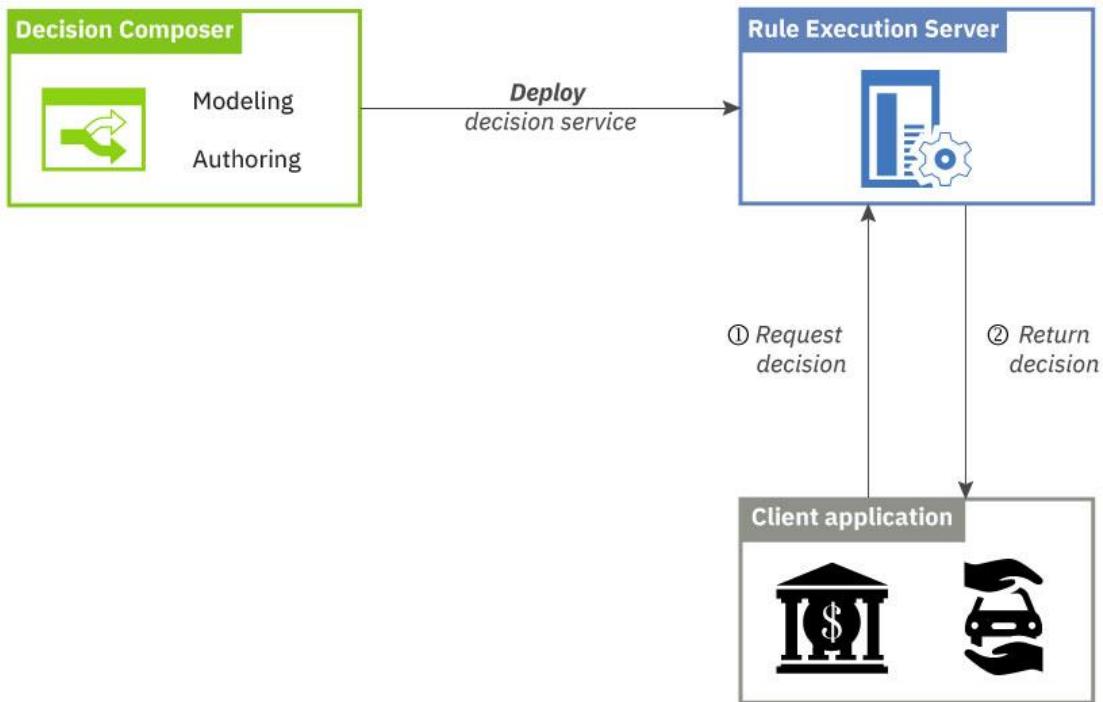
1.2.5 How do all these components interact?

The following diagrams recap how the different components and modules interact.

1.2.5.1 Bottom-up approach: Rule Designer → Decision Center → Rule Execution Server → Customer application



1.2.5.2 Top-down approach: Decision Composer → Rule Execution Server → Customer application



1.3 Get started with ODM and Decision Composer – Verification Instructions

Take some time to reflect on the various concepts and components that you discovered during this lecture.

1.4 Get started with ODM and Decision Composer – Summary

In this exercise, you learned about the major features of ODM and of Decision Composer, and in which cases you would use them.

You are now ready to move to the next exercise.

2 Exercise: Get Started with this Sub-Scenario

2.1 Get Started with this Sub-Scenario – Introduction

In this exercise, you complete the required steps to get started with the Operational Decision Manager (ODM) sub-scenario.

To perform this ODM sub-scenario, like for any other sub-scenarios, you will use an environment that you will create from a Template.

Prerequisites:

- Access to <https://bluedemos.com> with your IBM ID.
- Link to the **shared box folder**
<https://ibm.box.com/s/aiwhfcfs7ms9spnrijwg8bzmk7ad6lsf>.

To perform the ODM sub-scenario, you must select one of the following templates:

- **Template 5:** If you want to perform the ODM sub-scenario only and then (optionally) integrate either the decision services that you create with the ODM sub-scenario or the pre-defined decision services as part of a Workflow solution.
- **Template 7:** If you want to perform all sub-scenarios (including the ODM one) and then (optionally) integrate either the decision services that you create with the ODM sub-scenario or the pre-defined decision services as part of a Workflow solution.

High-level steps: (Detailed steps in section 2.2)

- Either you (start or) continue to work with your existing environment based on **Template 7**
-- or --
- You create your environment based on **Template 5 – Implement ODM sub-scenario only**.
- In your environment, ensure that you only start **VM 5 – ODM** for now. Make sure to **keep the other VMs suspended**. You might work with other virtual machines later during this scenario, and, if that happens, you will be instructed which ones to use and how.
- To work with your environment, you will require the following IDs and passwords:

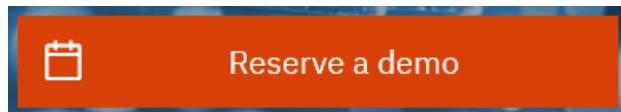
| Tool | User ID | Password |
|---------------------------------|-------------------------|---------------------|
| Windows | Administrator | passw0rd |
| Decision Center (IT user) | Paul | Paul |
| Decision Center (business user) | Bea | Bea |
| Rule Execution Server | resAdmin | resAdmin |
| Decision Composer | your IBMId ¹ | your IBMId password |

¹ If you do not have an IBMId and password yet, you will have the ability to create one during this sub-scenario

2.2 Get Started with this Sub-Scenario – Step by Step Instructions

Reminder: The following two templates are available to perform the ODM sub-scenario:

- a. **Template 5 – "IBM Digital Business Automation Blue Demos 2018 - Template 5 – Implement ODM sub-scenario only"**, if you want to:
 - i. Perform the ODM sub-scenario only (and not try any integration of decision services with a Workflow solution).
 - ii. Perform the ODM sub-scenario and then use the decision services that you obtain to see how you can invoke them in a Workflow solution.
 - iii. Perform the ODM sub-scenario and then use the decision services solution to this ODM sub-scenario (and pre-deployed) to see how you can invoke them in a Workflow solution.
 - b. **Template 7 – "IBM Digital Business Automation Blue Demos 2018 - Template 7 - Implement end-to-end scenario"**, if you want to:
 - i. Perform all scenarios, including the ODM sub-scenario.
1. To perform the ODM sub-scenario with Template 7:
 - a. If you have started with **Template 7** already for another sub-scenario, you have registered for a session based on **Template 7**: Resume your existing environment and continue with step 3 below.
 - b. If you are starting with **Template 7** now, for this ODM sub-scenario, and have not already registered for a session based on **Template 7** before, perform the step 2 by using Template 7 instead of Template 5, and then proceed with step 3.
 2. To perform the ODM sub-scenario with **Template 5** (resp. 7):
 - a. Open <https://bluedemos.com> and login with your IBM ID.
 - b. Search for **IBM DBA Blue Demos 2018** and select the template **IBM DBA Blue Demos 2018 - Template 5 - Implement ODM sub-scenario only** (resp. **IBM DBA Blue Demos 2018 - Template 7 - Implement end-to-end scenario**).
 - c. Click **Reserve a demo**.



- d.** Provide the necessary information and click **Reserve demo**.

| | | | |
|--|-----------------------------|------------------------|------------------------|
| Start date: [*] | Start time: [*] | End date: [*] | End time: [*] |
| 2018-07-26 | 1:00 pm | 2018-07-27 | 6:00 pm |
| Timezone: [*] | | Region: [*] | |
| Europe/Berlin (CEST) | | EMEA | |
| User email address: [*] | | Additional email: | |
| | | | |
| Demo purpose: [*] | Customer name: [*] | Sales Connect ID: | |
| Practice / Self-Education | NONE | | |
| Comments: | | | |
| <div style="border: 1px solid #ccc; height: 40px; margin-bottom: 10px;"></div> <p>This demo can be reserved for a maximum of 1500 hours and has an approximate duration of 30+ minutes</p> | | | |

Reserve demo

Close

- e.** After you click **Reserve demo**, you receive two emails.

The second email is sent only after your reservation is **active** according to the information you provided in step **d**, and contains the **link** and the **password** to access your demo environment:

Reservation Confirmation

IBM Blue Demos

<your_email> (<your_email>)

Demo: IBM Digital Business Automation Blue Demos 2018 - Template 5 - Implement ODM sub-scenario only [EMEA]
07/26/2018 - 07/27/2018 01:00 PM CEST - 06:00 PM CEST

Your reserved demo has started. Use your web browser to access the demo at the following link.

<https://cloud.skytap.com/vms/726cfbf96f785cae502fb4f918669efd/desktops>

Password: <your_password>

- f.** Open the link and enter your password to access your environment.

Virtual machine access

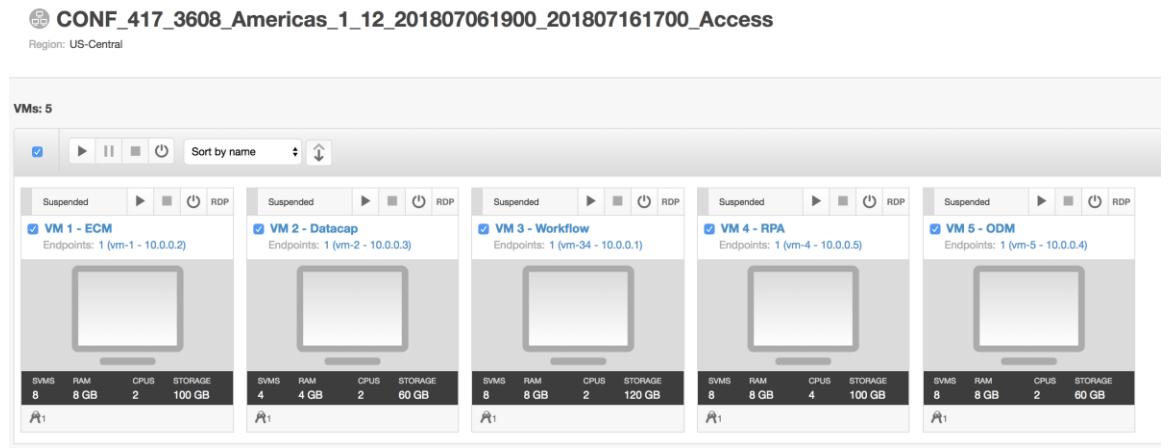
Please enter the supplied password to access this virtual machine. If you need the password, contact your session administrator.

Enter password

Submit

- g.** Click **Submit**.

- h.** Your demo lab environment opens, which looks like the following screen:



- i.** It consists of **five virtual machines (VMs)**:

VM 1 – ECM: This VM hosts an IBM Filenet Content Repository and LDAP. In addition, an IBM Content Navigator is installed.

VM 2 – Datacap: This VM hosts all Datacap tools.

VM 3 – Workflow: This VM hosts IBM Business Automation Workflow.

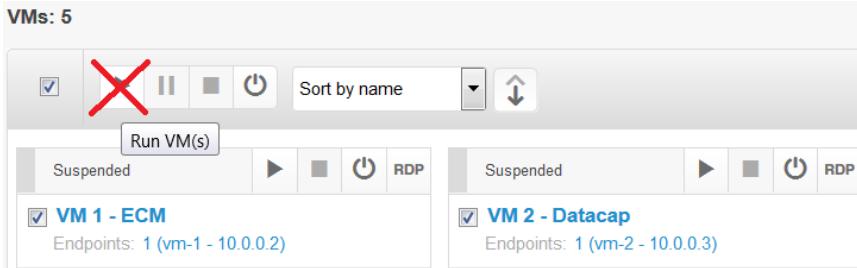
VM4 – RPA: This VM hosts the RPA Bot Designer and RPA Bot Runner.

VM 5 – ODM: This VM hosts ODM.

- j.** At this moment, all VMs are in suspended state.

You must not start all virtual machines at once. Always make sure to start one virtual machine at a time (as and when instructed).

Important: DO NOT use the Run VM(s) button.



Hint: If the icons are not shown or resuming / suspending of a VM takes too long, refresh your Browser window.

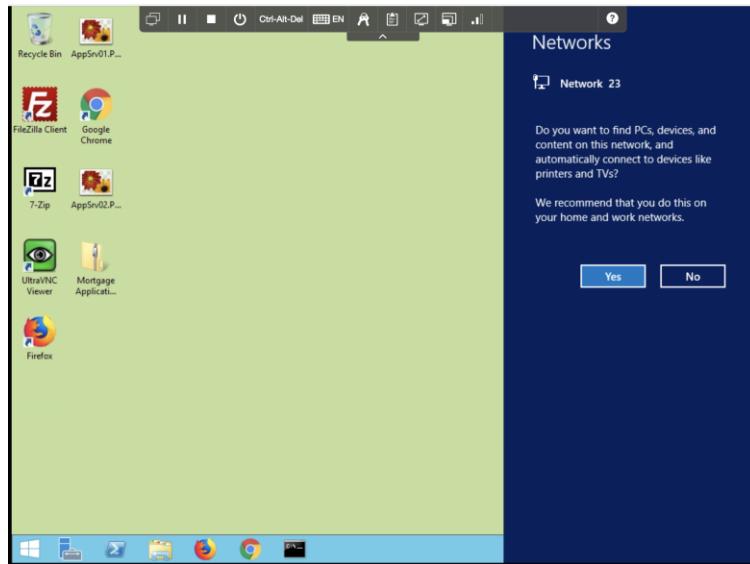
- k. To later **resume or suspend a single virtual machine**, click the **Play** icon or the **Pause** icon next to **Suspended** or **Running** of the mentioned **virtual machine**, here for example **VM 1 – ECM**.



Note: To save resources, you have to **suspend your entire environment when you are not using it for a longer period.**

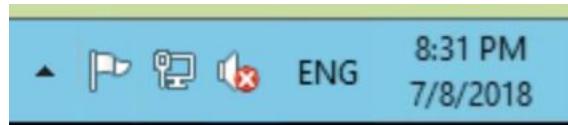
Note: You can also close the Browser tab / window on your local machine and even turn off your computer and **still resume and continue** your session at a later time. To resume your session repeat steps 2e – 2h.

- l. After you resume one of the suspended virtual machines, **always open the virtual machine's desktop** after it is reported as **Running**. You can do this by clicking on the screen display of the virtual machine. This opens the virtual machine in your browser:



If the **Networks** blue bar shows on the right-hand side as shown in the screen above, click **No** in it.

- m.** Next, make sure the virtual machine is **correctly connected to the network** before you proceed using it or starting any other virtual machine. Check that the network icon at the bottom of the desktop shows connected.



Connected:  vs Not connected: 



- n.** The **Windows username / password** for the environment is **Administrator / passw0rd**. Enter it in case you get prompted for it.
- o.** To change the resolution of the Remote Desktop to your liking, **right-click the Desktop**, select **Screen resolution** and update the resolution accordingly.

Congratulations! You successfully created your environment from your template.

You are now ready to run through a quick walk through of the UIs you will require during this sub-scenario.

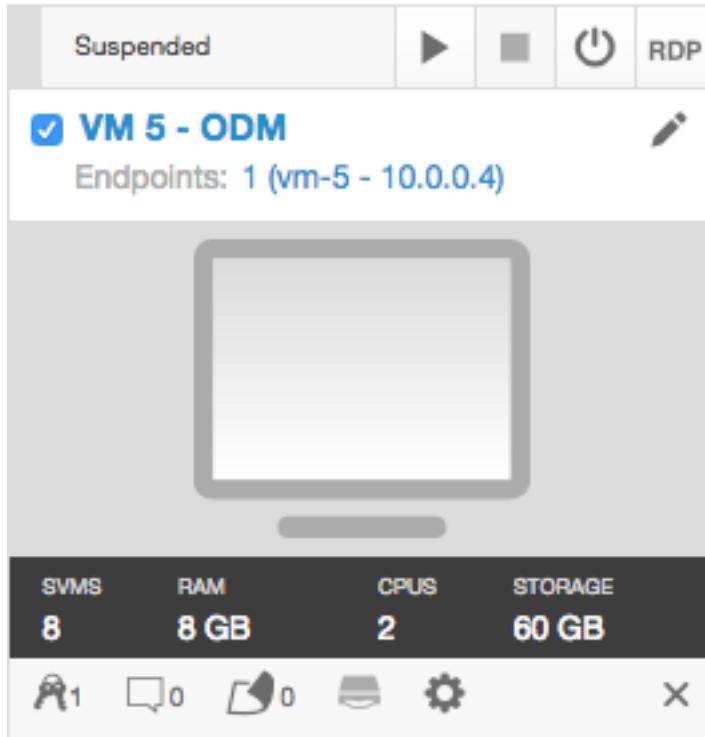
For this scenario, you only need the **VM 5 – ODM** virtual machine (for now).

3. Identify the **VM 5 – ODM** virtual machine in your environment.
This virtual machine should be in the "**Suspended**" state.

- a. If the status of the VM 5 – ODM virtual machine is "Suspended" or "Power off" (that is, not "Running"), start the VM 5 – ODM virtual machine now, by clicking



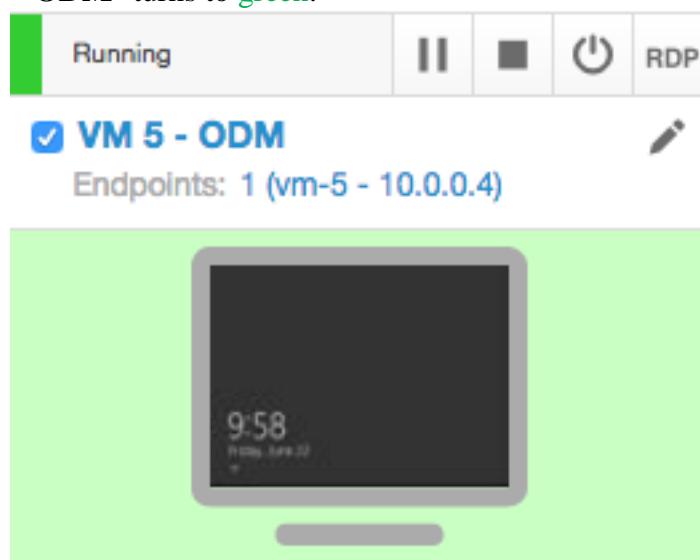
the "Run this VM" button in its menu.



Starting the virtual machine might take some time.

When the virtual machine is started:

- The status of the virtual machine turns to "Running".
- The background of the image of the virtual machine under the title "VM 5 – ODM" turns to green.



Important: The VM 5 – ODM virtual machine contains the materials required for you to perform this ODM sub-scenario and create the two decision services described in [Introduction](#).

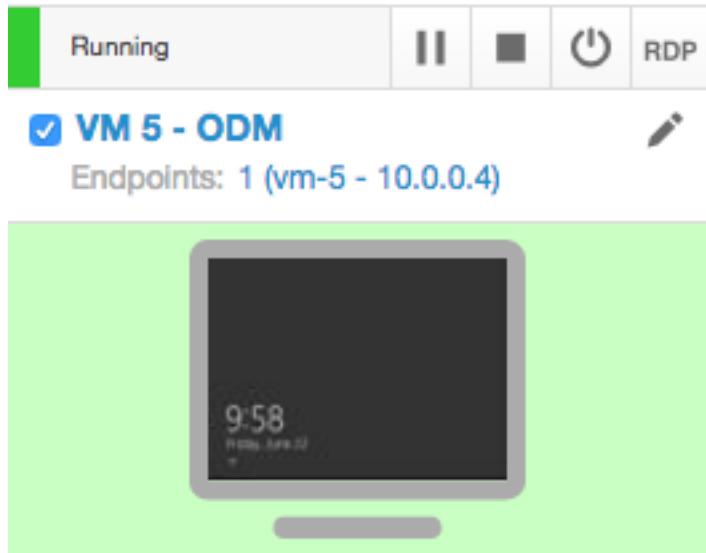
The VM 5 – ODM virtual machine also contains the solution for these two decision services, already pre-deployed in Rule Execution Server (as version 1.0).

Forward looking: If you want to integrate the decision services that you will create during this ODM sub-scenario with the Workflow solution at the end of this ODM sub-scenario, or the decision services that are pre-deployed in the VM 5 – ODM virtual machine (version 1.0), you would also need to start some other virtual machines. You will have instructions on how to do so during the [Exercise: Integrate within the Workflow solution](#).

Important: To minimum resource usage and allow all participants to be properly served, please make sure to suspend all virtual machines in your environment that you do not use.

4. For each virtual machine other than the **VM 5 – ODM** virtual machine:
 - a. Check the status of each virtual machine in its top-bar.
Status may be "**Running**", "**Suspended**" or "**Powered off**".

 - b. For each VM with a "**Running**" status, click the **Suspend this VM** button.
 - c. Wait until the VM reaches the "**Suspended**" status before suspending the next one.
5. When or if the status of the **VM 5 – ODM** virtual machine is **Running**, access the desktop of the **VM 5 – ODM** virtual machine as follows:
 - a. To access the virtual machine from your default web-browser, click the **image** of the virtual machine under the "**VM 5 – ODM**" title.



- b. Alternatively, to view the virtual machine in Remote Desktop Connection, click **RDP** in the button-bar above the "**VM 5 – ODM**" title.

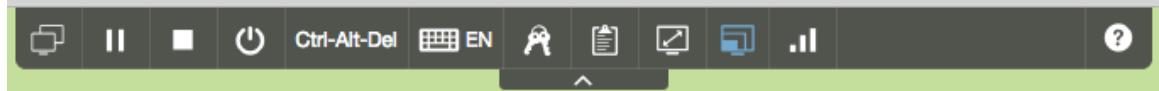
Clicking **RDP** downloads an RDP file to your machine that you can open directly to view the environment.

In both cases, the desktop of the VM 5 – ODM virtual machine opens.

6. On the virtual machine, set the proper resolution and keyboard:
 - a. To select the resolution that matches your need, click **Change Resolution**

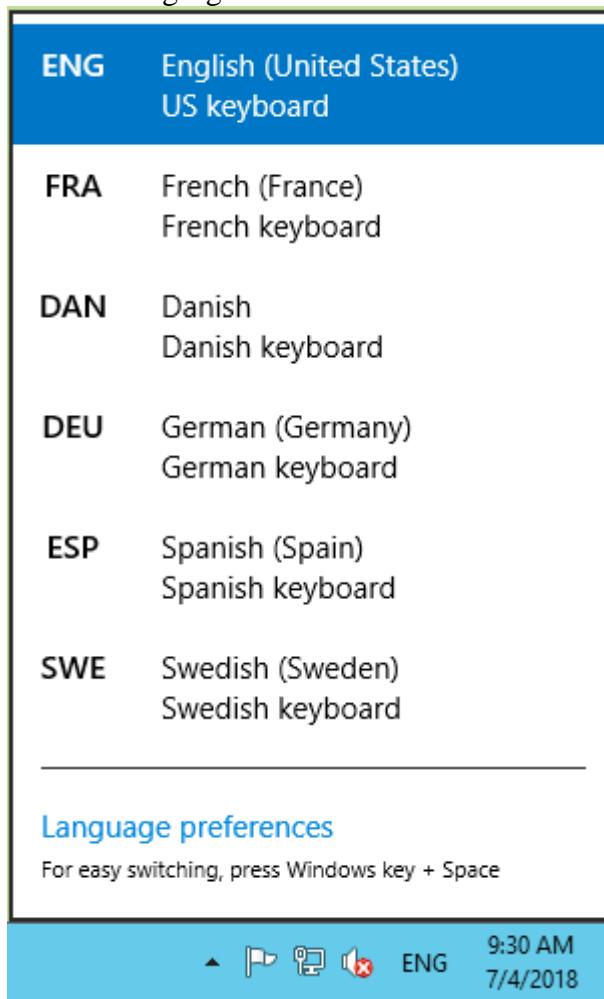


on the top-bar of the virtual machine:



- b. In the bottom-bar of the virtual machine, make sure that the keyboard configuration matches your own keyboard, by clicking the **locale** name and by selecting the appropriate locale.

For example, select ENG if your keyboard is an EN-US keyboard, as shown on the following figure:



Reminder: At any time, if you do not use a virtual machine in your environment, please suspend it.

7. Log in to the VM 5 – ODM virtual machine by using the following **Windows** credentials:

- User-name: **Administrator**
- Password: **passw0rd**

You can check the characters that you enter for the password by clicking the

Eye icon next to them.



Starting this point, and unless otherwise specified, steps are to be performed in the VM 5 – ODM virtual machine.

8. Take some time to discover the multiple shortcuts that are available for you on the desktop:
 - a. **ODM892 – InstallDir:** to open the folder that contains the ODM 8.9.2 product files in a Windows Explorer.
Note: This folder is referred to as <InstallDir> in this document as well as in the ODM user documentation.
 - b. **Start Sample Server:** to start the ODM Sample Server, if it is not started yet.
Note: When started, the ODM Sample Server uses the port **9090**.
 - c. **Stop Sample Server:** to stop the ODM Sample Server, if it is started
 - d. **Rule Designer:** to launch the ODM Rule Designer (RD).
 - e. **DC BC:** to start the ODM Decision Center (DC) Business console (BC)
 - f. **DC EC:** to start the ODM Decision Center (DC) Enterprise console (EC)
 - g. **RES Console:** to start the ODM Rule Execution Server (RES) console
 - h. **Start KC (local):** to start a local Knowledge Center (KC) server, if it is not started yet, so to access the local ODM user documentation from a web-browser.
Important: If you want to use both the Sample Server and the local ODM product documentation with a local KC server, make sure that you start the Sample Server first and only then the local KC Server. This order is important to prevent the local KC server from using the port **9090** that the Sample Server needs to start. When started in the proper order (Sample Server and then local KC server), the local KC server uses a port that is different from **9090**, and might change if you stop and re-start it.
 - i. **Stop KC (local):** to stop the local Knowledge Center (KC), if it is started.
Note: Stopping the local KC server does not close the web-browser window that opened when you started the local KC server. After you stopped the local KC server, the current page in this web-browser window remains available but you can no longer navigate in the local ODM user documentation.
 - j. **Samples Console:** To start the Sample console, and run samples and tutorials.

- k. **Chrome:** To open a Chrome window to visualize the ODM user documentation or run Decision Composer.
9. Check that ODM **Sample Server** is started:
- Double-click **DC BC** on the desktop to open the Business console.
A command window quickly runs that opens a Chrome window and then closes.
 - Check that the Chrome window that opens shows the Business console page at:
<localhost:9090/decisioncenter/login>
- If the Business console page shows, the ODM Sample Server is started: Skip the next step.
- If the Business console page does not show, the ODM Sample Server is not started: Perform the next step.

10. Start the ODM **Sample Server**:

- Double-click "**Start Sample Server**" on the desktop.
A "*Start Sample Server*" command window opens that shows the log messages of the sample server startup.
- Wait for the end of the command execution.
Starting the Sample Server might take some time.
- The Sample Server started successfully when you can see the following message at the end of the command execution window:

```
[samples.echo] INFO: GBRPS0029I: start.server is completed.  
BUILD SUCCESSFUL  
Total time: ... minutes ... seconds  
Press any key to continue . . .
```

If you have another message, please check the [Issue with starting the Sample Server](#) section at the end of this exercise for assistance.
- In the "*Start Sample Server*" command window, press any key to close the command window and continue.

Congratulations! You successfully started the Sample Server.

11. Start the local ODM **Knowledge Center (KC) Server** and access the local ODM user documentation:
- Double-click "**Start KC (local)**" on the desktop.
A "*Start KC (local)*" command window opens that shows the traces of the local KC server startup, and opens a Chrome window with the local ODM user documentation.
The command window automatically closes when this startup ends.
 - Check that the Chrome window that opened shows the Operational Decision Manager Version 8.9.2 documentation page, at:
10.0.0.4:<port>/kc/SSQP76/welcome_local_help.html
Note: The <port> might vary from one execution to the other.
 - Do not perform the steps that are given by this Welcome page.**

Although you are invited to read these steps to discover how the ODM user documentation has been locally installed, these steps were already done on this machine and you should not redo them.

Note: You can also access the product documentation in the public Knowledge Center (KC), without using the local KC Server, by opening a web-browser window at:
https://www.ibm.com/support/knowledgecenter/en/SSQP76_8.9.2/welcome/kc_welcome_odmV.html

Use the local KC Server if you need to read the documentation locally, without Internet access.

12. Stop the local ODM Knowledge Center (KC) Server:

- a. Double-click "**Stop KC (local)**" on the desktop.

A command window opens that stops the local KC server. This action might take some time.

When the command window closes, the local KC server is stopped.

Important: If you want to start the local KC server again later in this sub-scenario, must make sure that you do so only if you have the Sample Server already started. If you start the local KC server when the Sample Server is not running, the local KC server will otherwise use the port 9090 and you will no longer be able to start the Sample Server [unless by tweaking its configuration... see [Issue with step 11 \(Start Sample Server\)](#) in the Troubleshooting section for more details].

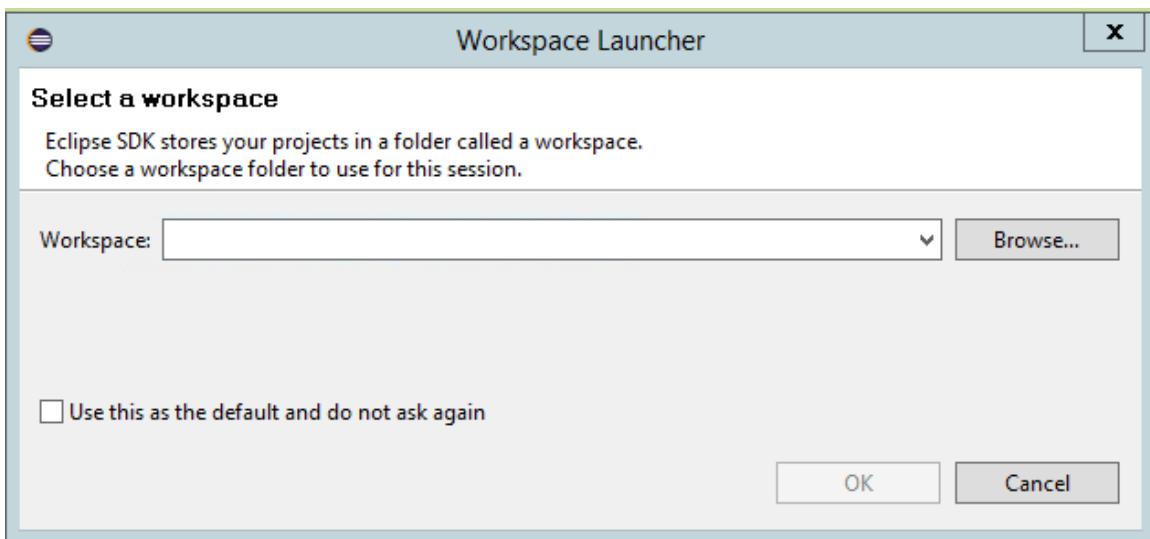
As a rule, start the Sample Server, and keep it running during the whole demo; and then start the local KC server if you need to access the local product documentation. Keep the local KC server running while needed, and then stop it. Stop the Sample Server only when you're done with all ODM hands-on entirely.

Congratulations! You successfully started the local KC server and accessed the local ODM 8.9.2 user documentation.

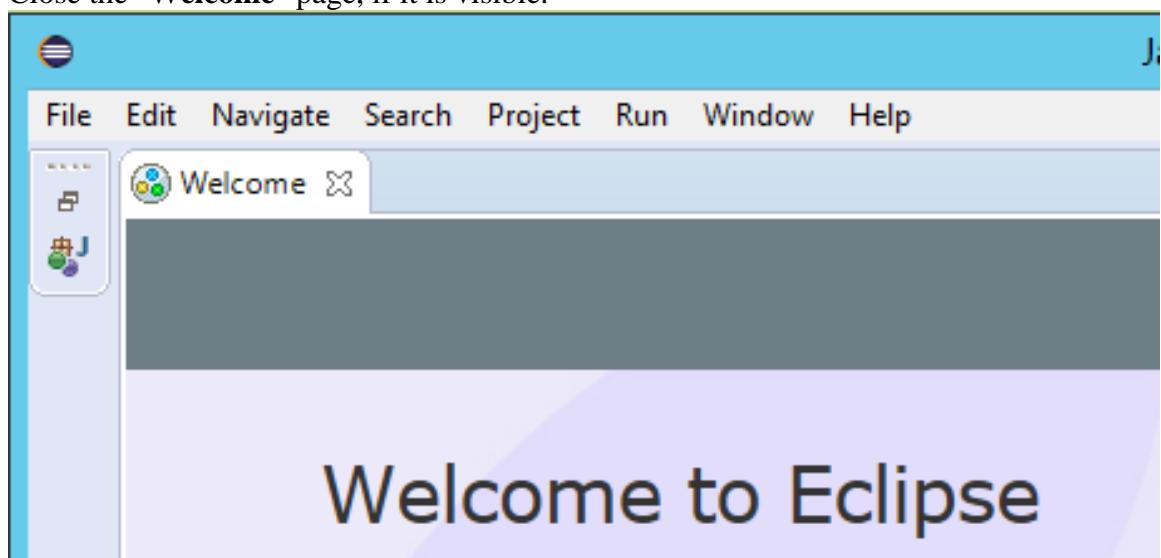
13. Start ODM Rule Designer, a plugin for Eclipse that allows you to create rule projects, publish them to Decision Center, and deploy them to Rule Execution Server:

- b. Double-click **Rule Designer** on the desktop.

The "**Workspace Launcher**" Eclipse window opens, with its **Workspace** field populated with the last used workspace.



- c. Ensure that the following path is entered in the **Workspace** field, if that is not yet the case: **C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-1-Start**
- d. Click **OK**.
- Eclipse opens.
- e. Close the "**Welcome**" page, if it is visible.



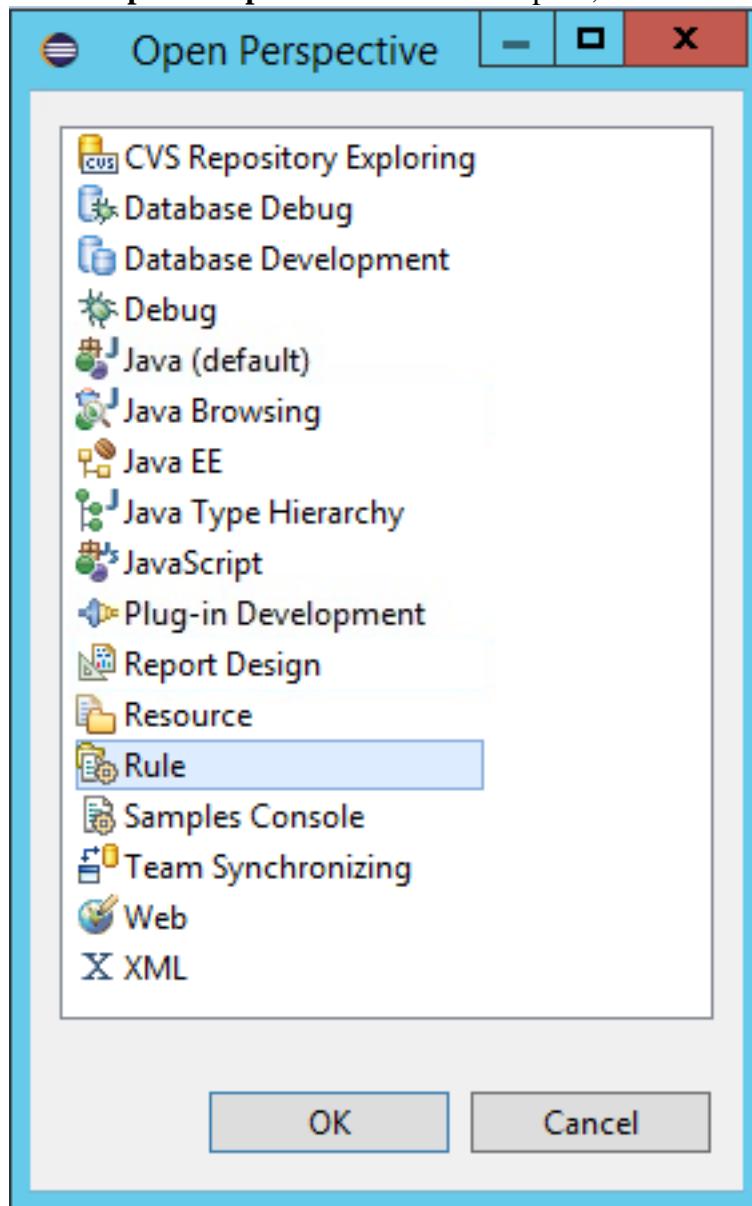
- f. Check that the **Rule** perspective selected in the right-top corner of Eclipse.



- g. If the **Rule** perspective is not shown and not selected:

- Click **Open Perspective** .

- On the **Open Perspective** window that opens, select **Rule** and click **OK**.



- Take a few minutes to explore the Rule perspective, in particular the Rule Explorer tab.
You will however learn more about Rule Designer in [Exercise: Set up DBA Jam 1 Decision Service in Rule Designer](#), so do not spend too much time on this module now.
- Close the Rule Designer (Eclipse) window.

Congratulations! You successfully started Rule Designer and had a first view of what an ODM Rules project looks like.

14. Check that you can access **IBM Decision Composer**:
 - Double-click **Chrome** on the desktop.

- A Chrome window opens, where you can see the "**IBM Decision Composer**" bookmark  in the bookmark bar.
- b. Click the "**IBM Decision Composer**" bookmark to open IBM Decision Composer page at: <https://decision-composer.mybluemix.net>
 - If the IBM Decision Composer page does not open, check the troubleshooting section or contact the demo owner.
 - c. On the IBM Decision Composer page, click **Log in**.
 - d. If you do not have an IBMId, click "**Create an IBMId**" to create your IBMId and password.
 - e. Enter your **IBMid**, select the "**Remember me**" check-box as needed, and then click **Continue**.
 - f. Enter your IBMId **password** and click **Sign in**.
- The **IBM Decision Composer (experimental)** page opens, and a "**Use of Cookies**" dialog opens on top of it.
- g. Read this dialog as needed.
 - h. Click **Ok** to leave the "**Use of Cookies**" dialog.
 - i. Close or reduce the Chrome window. You do not need it for now.

15. Check that you can access the ODM backup materials that is available in Box for you:
 - a. In Chrome, open the page at the following URL:
<https://ibm.box.com/s/aiwhfcfs7ms9spnrijwg8bzmk7ad61sf>
The "**DBA Blue Demos 2018 - Material for Participants**" Box folder opens.
 - b. In the "**DBA Blue Demos 2018 - Material for Participants**" Box folder, click "**5. ODM sub-scenario**".
The "**5. ODM sub-scenario**" Box folder opens.
 - c. Check that you can see the content of the "**5. ODM sub-scenario**" Box folder and that it matches the content listed [in the Box folder 5. ODM sub-scenario](#) section.
 - d. Take time now to identify the files in this Box folder, which are mainly for backup purpose.
The instructions in this scenario do not make use these files.
You would use it if you need to restore a file on the virtual machine for any reasons (for instance: if you delete it by error or if you have overridden it by your work and you want to restore its original state).

2.3 Get Started with this Sub-Scenario – Verification Instructions

To verify successful completion of this exercise, check that you can see the various consoles of the ODM modules that are hosted by the Sample Server:

- Decision Center > Business console (later referred to as Business console, DC BC, or simply BC)
- Decision Center > Enterprise console (later referred to as Enterprise console, DC EC, or simply EC)

- Rule Execution Server console (later referred to as RES Console)
1. Check that you have access to the **Business console**:
 - a. Double-click **DC BC** on the desktop to open the Business console.
A command window quickly runs that opens a Chrome window and then closes.
 - b. Check that the Chrome window that opened shows the Business console page at localhost:9090/decisioncenter/login.
 - c. If the **Privacy** dialog shows on top of the Business console page, read its content as needed, and click "**Agree and Proceed**" to close it.
 - d. As needed², enter the following username and password:
 - Username = rtsAdmin
 - Password = rtsAdmin
 - e. Click **Login In**.
 - f. You will learn more about the Business console in subsequent exercises, so do not spend too much time on this module now.
 - g. Close or reduce the Chrome window with the Decision Center > Business console page open. You do not need it for now.
 2. Check that you have access to the **Rule Execution Server console**:
 - a. Double-click **RES Console** on the desktop to open the RES console.
A command window quickly runs that opens a Chrome window and then closes.
 - b. Check that the Chrome window that opened shows the Rule Execution Server console page at: <http://localhost:9090/res>
 - c. If the **Privacy** dialog shows on top of the Enterprise console page, read its content as needed, and click "**Agree and Proceed**" to close it.
 - d. Ignore the "This user does not have the correct role" message.
 - e. As needed, enter the following username and password:
 - Username = resAdmin
 - Password = resAdmin
 - f. Click **Sign In**.
 - g. You will learn more about the RES console in subsequent exercises, so do not spend too much time on this module now.
 - h. Close or reduce the Chrome window with the RES Console page opened. You do not need it for now.

Congratulations! You successfully accessed all the ODM tools.

2.4 Get Started with this Sub-Scenario – Summary

In this exercise, you have performed the following action:

- 1) Created your environment for the ODM sub-scenario on BlueDemos

² Because of single sign-off, if you are currently logged in to the Enterprise console, you should be automatically logged in to the Business console, and vice-versa.

- 2) Accessed your environment, and started the VM 5 – ODM machine that contains the materials required for this ODM sub-scenario.
- 3) Learned which tools to be used in this sub-scenario and how to start them.
- 4) Verified that you can start or access these tools.

You are now ready to move to the next exercise.

2.5 Get Started with this Sub-Scenario – Troubleshooting

2.5.1 Issue with starting the Sample Server

When starting the Sample Server, if you have the following SEVERE GBRPS0040E message, the port 9090 is already in use and this situation prevents the Sample Server from starting.

[samples.echo] SEVERE: GBRPS0040E (...)

In most cases, this situation means that the Sample Server is already started or that the local KC server was started before you tried to start the Sample Server.

To resolve this issue, perform the following steps:

1. Check whether the Sample Server is already started:
 - a. Double-click **DC BC** on the desktop to start Decision Center > Business console.
A command window quickly runs, opens a Chrome window at the Decision Center > Business console page at localhost:9090/decisioncenter/login, and then closes.
 - b. Check that the Chrome window that opened shows the Decision Center > Business console page at localhost:9090/decisioncenter/login.

If this page is visible, the Sample Server is well started and therefore you can ignore the SEVERE GBRPS0040E message. You're set!

If this page is not visible, proceed with the next step.
2. Check whether the local KC server is already started:
 - a. Double-click Chrome to open a web-browser.
 - b. Enter the following address:
10.0.0.4:9090/kc/SSQP76/welcome_local_help.html
 - c. If the Operational Decision Manager Version 8.9.2 documentation page is visible, the local KC server is started and uses the port 9090.

This usage is the reason why the Sample Server cannot start. You must either:

 - i. Stop the local KC server to free the port 9090, as indicated in "Stop the local KC server on port 9090" next.
This short and simple solution is preferred.
 - or --
 - ii. Reconfigure the Sample Server so it uses a different port number. This solution is more complex. Instructions for it exist, which you can read in the traces that go with the error message. However, for this guided scenario, all instructions and solutions are given with the Sample

Server running on port 9090, so do not try to modify this Sample Server port; instead, please proceed with "*Stop the local KC server on port 9090*" next.

3. Stop the local KC server on port 9090:
 - a. When the local KC server is already started with port 9090, stop it by double-clicking **Stop KC (local)** on the desktop.
A "**Stop KC (local)**" command window opens that stops the local KC server. When the "**Stop KC (local)**" command window closes, the local KC server is no longer started and the port 9090 is free for the Sample Server.
 - b. Retry now step "Start the ODM Sample Server" in [Get Started with this Sub-Scenario – Step by Step Instructions](#).
 - c. If this step "Start the ODM Sample Server" in [Get Started with this Sub-Scenario – Step by Step Instructions](#) fails again, contact the demo owner.

3 Exercise: Set up DBA Jam 1 Decision Service in Rule Designer

3.1 Set up DBA Jam 1 Decision Service in Rule Designer – Introduction

This exercise is the first of two exercises where you define the Decision Service **DBA Jam 1**, described in [Introduction](#).

DBA Jam 1: This first decision service will compute some scores about the borrower and the loan, based on their basic data provided by the requestor. You will set up this computational decision service by following the [bottom-up approach](#) with ODM: Rule Designer → Decision Center → Rule Execution Server → Customer application.

In this exercise, you start with **Rule Designer**, playing the role of Paul, an IT person who sets up the decision project, which is the foundation of the decision service, and then provides it to Bea, a (non-technical) business user who will then be responsible for authoring the decisions.

In Rule Designer, you, as Paul, the IT person, put in place the necessary infrastructure for editing rules and producing one or several **rulesets**, you define the contract with the client application, and you design the models and the vocabulary for authoring business rules:

- You define the set of **business rules** that are put together as one executable decision unit called a **ruleset**. The ruleset uses **input and output parameters** to pass data to and from the client application.
- You define each ruleset and its parameters with a unique **signature** of input and output parameters.
- In decision services, you create **decision operations**, which defines the content and signature of the ruleset.
- You define the **vocabulary** that is used in business rules.
- You develop the **business object model (BOM)** which defines the elements and relationships in the vocabulary.
- You develop the **Execution Object Model (XOM)**, which is the executable code behind the business rule artifacts.
- You organize rules within **rule packages** in your rule projects, and define a **ruleflow** to specify their flow of execution at run-time.

You can learn more about the role of IT people when working in Rule Designer by reading the following topic in the product documentation:

https://www.ibm.com/support/knowledgecenter/SSQP76_8.9.2/com.ibm.odm.dserver.rules.overview/topics/odm_dserver_rules_overview.html

Note: You can also access the product documentation in the local Knowledge Center (KC). See [Get Started with this Sub-Scenario – Step by Step Instructions](#) for instructions on how to start the local KC and access the product documentation locally.

This sub-scenario is not meant for you to learn how to create Java™ code or do basic operations with ODM such as the creation of the Executable Object Model (XOM), the creation of the Business Object Model (BOM) or the verbalization of the BOM to create the vocabulary required by business users to author business rules and decision tables.

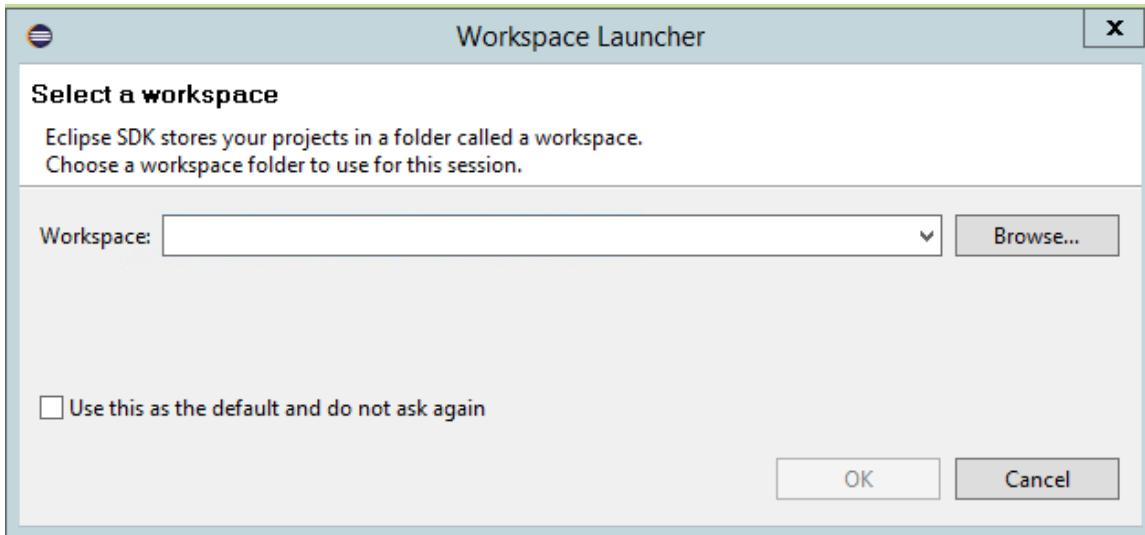
This exercise therefore starts with a project that already contains this minimum information, and you will not re-create it. Instead, you will focus on:

1. Understanding how the decision service is designed.
 - Look at the Execution Object Model.
 - Look at the Business Object Model.
 - Look at the available vocabulary.
2. Defining the signature of the decision service, which is what matters when calling it: input parameters, output parameters.
3. Learning how to synchronize with Decision Center, to hand-over the work done in Rule Designer to business users working in Decision Center.

3.2 Set up DBA Jam 1 Decision Service in Rule Designer – Step by Step Instructions

If you already did this exercise and want to redo it, restore its 'starting point' as follows: Open a Windows Explorer, delete the existing C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-1-Start folder, and then re-create it by using the [Jam-1-Start.zip](#) file in Box.

1. If you are not yet logged in to the VM 5 – ODM virtual machine, do so now by 2. using the **Windows** credentials indicated in [Get Started with this Sub-Scenario – Introduction](#).
3. Check that the Sample Server is already by following step 10 of [Get Started with this Sub-Scenario – Step by Step Instructions](#).
4. If the Sample Server is not started yet, start it now by following step 11 of [Get Started with this Sub-Scenario – Step by Step Instructions](#).
5. Start ODM Rule Designer:
 - a. Double-click **Rule Designer** on the desktop.
The "Workspace Launcher" Eclipse window opens, with its **Workspace** field populated with the last used workspace.



- b. Ensure that the following path is entered in the Workspace field, if that is not yet the case: **C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-1-Start**
 - c. Click OK.
Eclipse opens.
 - d. Close the "Welcome" page, if it is visible.
6. Ensure that with the **Rule** perspective is selected, by following the instructions in step **14.f** of [Get Started with this Sub-Scenario – Step by Step Instructions](#).

Examine the DBA Jam 1 decision service by performing the following steps in the **Rule Explorer** view of the **Rule** perspective.

7. Examine how the DBA Jam 1 decision service is organized:
 - a. Expand the "**loan-validation-xom**" entry, which contains the eXecutable Object Model (XOM), a series of Java classes that are required to execute the decisions at run time.
 - b. Explore the classes in the XOM.
*You can ignore or delete *.bak files – These files are only backup files and have no impact on the project.*
 - c. Expand the **DBA Jam 1 project > bom** folder, which contains the Business Object Model (BOM) that is derived from the XOM and used to define the business vocabulary required to author business rules and decision tables
*You can ignore or delete *.bak files – These files are only backup files and have no impact on the project.*
 - d. Explore the **model** in this BOM.
Take time to explore the BOM, and the vocabulary that it defines to author rules. For example, you might want to:
 - Double-click the **Borrower** entry, which gives all the fields related to the borrower.

- Double-click its **age** field to see how the **age** field is verbalized that is, what vocabulary can be used to denote the borrower's age in rule artifacts: {age} of {this}
 - Go back to the **Class** tab and double-click the **creditScore** field to see how one can update this **creditScore** field in a rule by using the vocabulary: set the credit score of {this} to {credit score}
8. Examine how the DBA Jam 1 decision service is defined:
- Expand **DBA Jam 1 > deployment**.
Under deployment, you can see:
 - **computeLoanRequestParameters**, which is the name of the decision service
 - **Local_Deployment_Configuration**, which defines the Rule Execution Server(s) that you can deploy your decision service to. Currently, only one local Rule Execution Server is defined (**Local_RES**).
 - Double-click **computeLoanRequestParameters** to define the decision service signature.
 - Look at the decision service, which is defined by a signature (that is, a series of input, input-output and output parameters), an associated ruleset name and a specified main ruleflow.
 - Currently, the decision service does not have any parameter in its signature.
 - Currently, the decision service is associated with the ruleset **computeLoanRequestParameters**.
 - Currently, the decision service has the **mainRuleflow** ruleflow defined as its main ruleflow.

9. Define the signature of the DBA Jam 1 service:

- In the **Signature** area, click **Define** to define the proper parameters.

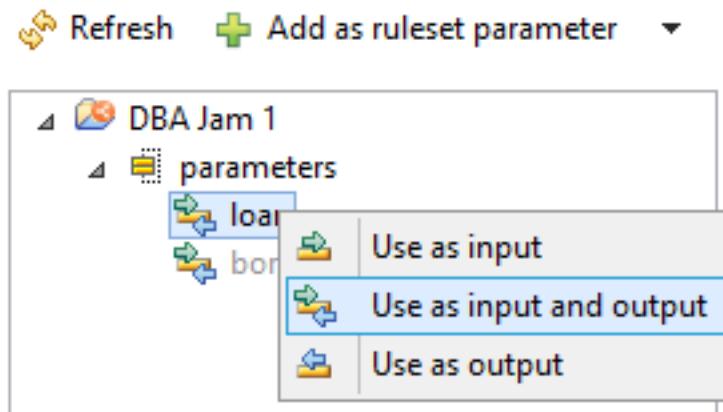
Signature

 [Define the input and output parameters for the ruleset.](#)

- Add an input-output parameter called "loan", of type "loan.Loan" and with verbalization "**the loan**".
For this purpose, right-click the **loan** ruleset variable in the "**Eligible variables**" area, and select "**Use as input and output**".

Eligible variables

Select the ruleset variables that you want to use as parameters for the decision operation. Ruleset variables are defined in variable sets.



- c. Similarly, add an input-output parameter called "borrower", of type "loan.Borrower" and with verbalization "the borrower".
- d. Check that you obtain the following result:

Signature

[Define](#) the input and output parameters for the ruleset.

Input: none

Input - output: borrower,loan

Output: none

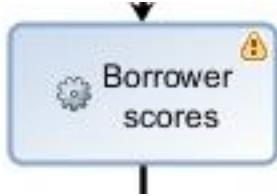
10. Look at the main ruleflow of the DBA Jam 1 service:

- a. In the Rule Explorer view, double-click **mainRuleFlow** under "DBA Jam 1" to edit the **mainRuleFlow** ruleflow.

This ruleflow contains a series of tasks between a starting-point and an ending-point, indicating which operations are executed, and in which order.

Tasks marked with a lightning are action tasks, that is, they contain code to execute.

Tasks marked with the engine symbol are rule tasks, that is, they indicate the rule packages in the project to execute. For example, the "**Borrower scores**" task is here to indicate that the rules in the "compute borrower scores" rule package must be called at this point of the decision.



- b. The two rule tasks are not yet defined (a warning sign indicates so because they do not have any rule packages associated with them. You will not define them in Rule Designer. In this sub-scenario, it is not your role as an IT person to define the ruleflow in use for the decision service. As an IT person, you define the structure of the decision service and the underlying code required to execute it. It is now time to hand-over the decision service project to business users, who will finalize and test it in Decision Center (Business console).)

In "normal" life, as an IT person, you would also publish (synchronize) the DBA Jam 1 decision service from Rule Designer to Decision Center so to hand it over to business users.

However, in this ODM sub-scenario, this publication is already performed, so DO NOT do it; doing so would override the starting-point project for the next exercise in Decision Center, which would prevent you from doing this exercise.

If you are interested in seeing what it means to publish (synchronize) the decision service from Rule Designer to Decision Center, you can however look at the following steps.

Only read but do not perform these steps, so not to override the starting-point project of the next exercise in Decision Center > Business console.

11. Understand the steps required in Rule Designer to publish a decision service to Decision Center:

- a. Look at the **Problems** view to first check that your decision service has no errors.

You should only have warning messages, no errors.

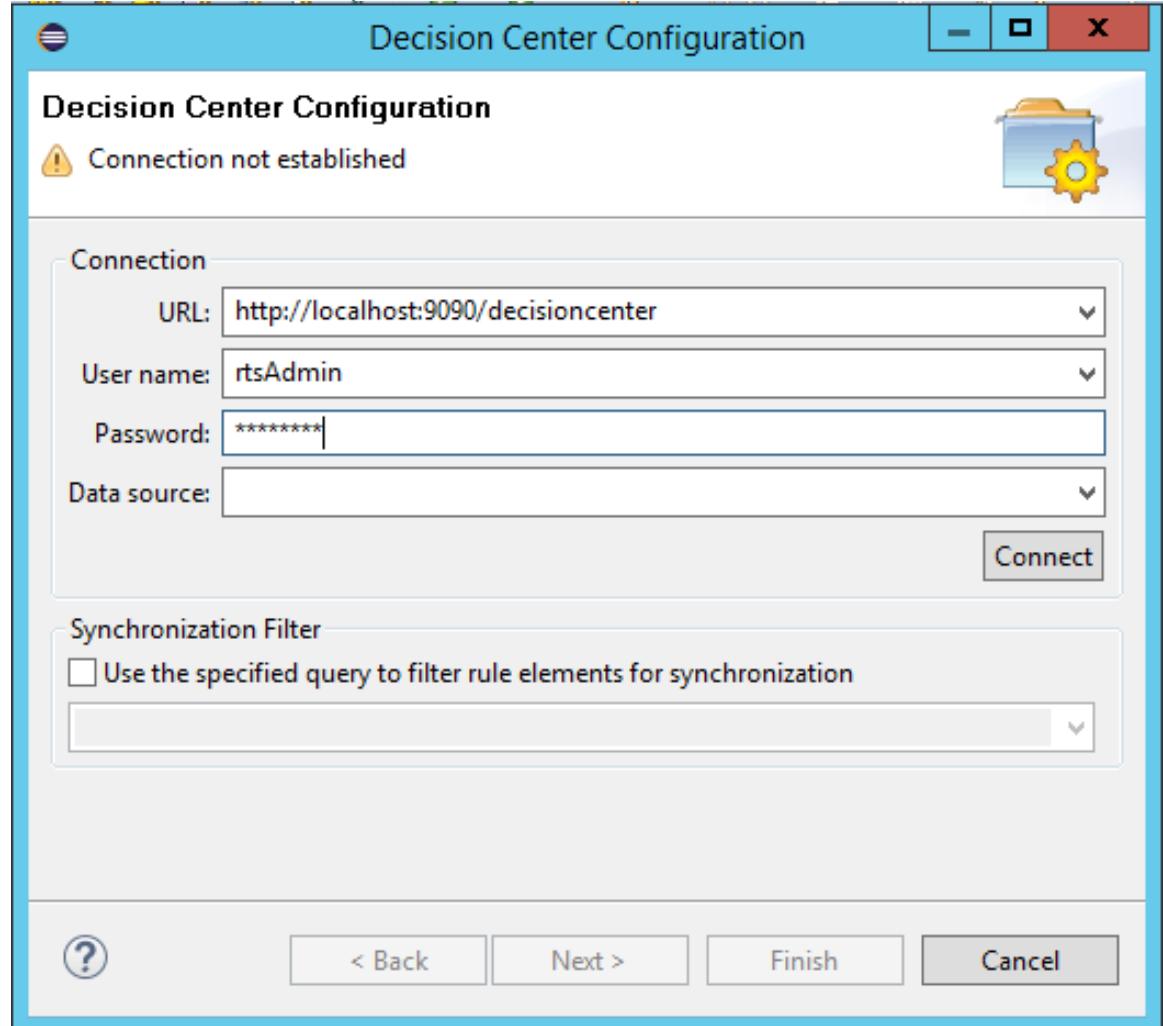
If you have an error in the deployment >

Local_Deployment_Configuration part of your decision service, please check the [DBA Jam 1 > deployment > Local Deployment Configuration is in error](#) troubleshooting section at the end of this exercise for help.

The warning messages, such as that the decision table has gaps and that the rule tasks are empty, are OK, and they indicate some work for the business users. Ignore them.

- b. When all errors are corrected, check that the Sample Server is started and that Decision Center is available, by following steps 10 and 11 in [Get Started with this Sub-Scenario – Verification Instructions](#).
- c. Right-click **DBA Jam 1** in the Rule Explorer view, and select **Decision Center > Connect**.
- d. In the Decision Center Configuration window that opens, enter the following connection parameters:

- URL = <http://localhost:9090/decisioncenter>
- User name = **rtsAdmin**
- Password = **rtsAdmin**
- Data source = (*leave this field void*)



If you have an error such as **Authentication failed for user "rtsAdmin" on (...)**, double-check the **Password** that you entered.

e. **STOP.**

Click Cancel.

At this point, the IT person would normally click **Finish** to perform the publication/synchronization, see any differences, adapt and so on...

Again, please do **not** do so for the ODM sub-scenario. Instead:

3.3 Set up DBA Jam 1 Decision Service in Rule Designer – Verification Instructions

The exercise is completed when you have successfully examined the DBA Jam 1 decision service in Rule Designer, have created its signature, and have understood what is missing (a decision rule, a decision table, some elements of the main workflow) so to be

able to finalize it, acting as a business user in Decision Center > Business console during the next [Exercise: Finalize Decision Service](#).

To see the solution for this exercise, start Rule Designer by selecting the following workspace: C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-1-Answer

If you must restore this solution workspace: Open a Windows Explorer, delete the existing C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-1-Answer folder, and then re-create it by using the [Jam-1-Answer.zip](#) file in Box.

3.4 Set up DBA Jam 1 Decision Service in Rule Designer – Summary

In this exercise, you acted as an IT person to ensure that the decision service 1 is properly structured.

1. You ensured that the executable part of the service exists (the XOM).
2. You ensured that the vocabulary of the rule project exists (the BOM), so that business users can author rules for their decision service.
3. You ensure that the main ruleflow structure exists, leaving it empty so business users can select the rule packages that they want for their decision service.
4. You defined the decision service signature, based on the rule project ruleset parameters (input, input/output or output).
5. You assured that the rule project is available in Decision Center for authoring and finalization by business users in the Business console, which is the task of the next exercise.

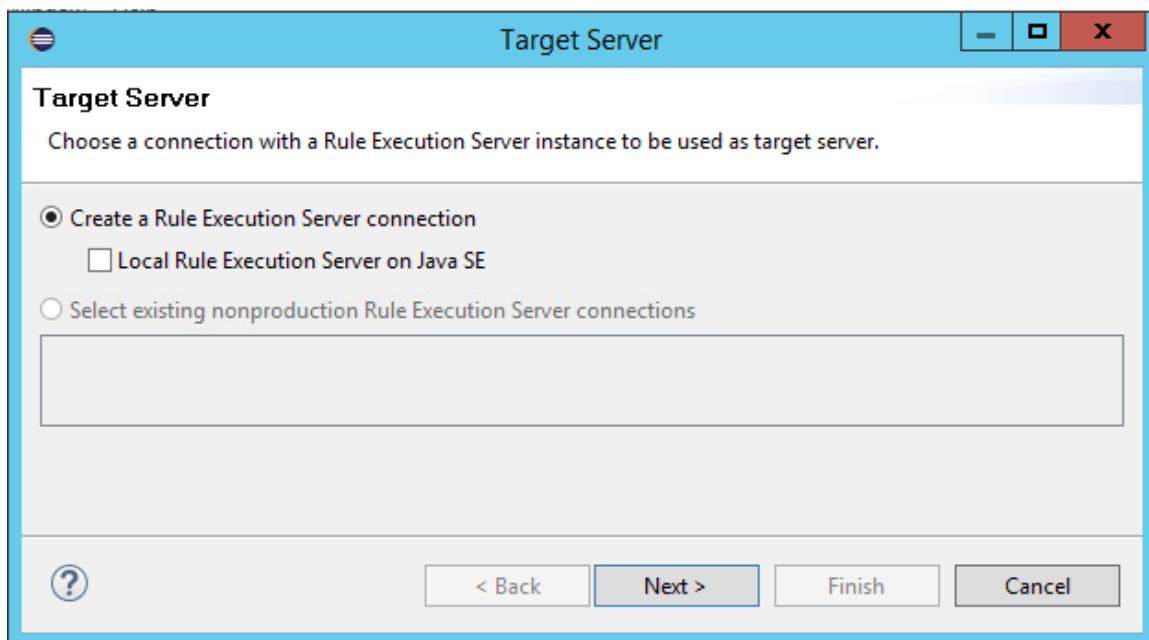
You are now ready to move to the next exercise.

3.5 Set up DBA Jam 1 Decision Service in Rule Designer – Troubleshooting

3.5.1 DBA Jam 1 > deployment > Local_Deployment_Configuration is in error

The diagnostic is that the RES configuration is badly defined or undefined. Resolve this issue as follows:

1. Double-click **Local_Deployment_Configuration**.
2. Go to **Target Servers**.
3. Check whether the **Local_RES** target server exists.
4. If yes, you might want to try to see what is wrong and correct. Alternatively, delete it and then re-create it, as follows.
5. If **Local_RES** did not exist or if you deleted **Local_RES**, create a new **Local_RES** now:
 - a. Click "+".
 - b. Select **Create a Rule Execution Server connection**.
 - c. Clear the **Local Rule Execution Server on Java SE** check-box.



- d. Click **Next**.
 - e. Set the **Name** as "**Local_RES**".
 - f. Set the **URL** as "<http://localhost:9090/res/login.jsf>".
 - g. Set the **Username** as "**resAdmin**".
 - h. Click **Finish**.
6. If you have a RES called "**Bluemix RES**", delete it.

4 Exercise: Finalize DBA Jam 1 Decision Service in Business console

4.1 *Finalize DBA Jam 1 Decision Service in Business console – Introduction*

This exercise is the second (and last) of two exercises where you define the Decision Service **DBA Jam 1**, described in [Introduction](#).

In this exercise, you are introduced to the Business console, which is the collaborative environment used by business users to author, manage, test, and deploy rules.

You will act as Paul, the IT specialist and project administrator, and then as Bea, the business users who will author the rules. Paul looks at the decision service in the Business console, tests it and discovers errors. He asks Bea to make the necessary changes. When Bea is done, Paul goes back to the Business console and deploy the decision service in Rule Execution Server so it can be called by the customer application.

Important: A solution for this DBA Jam 1 decision service is already pre-deployed in the Rule Execution Server of the VM 5 – ODM virtual machine. It means that, even if you do not fully perform this exercise, you could still integrate this DBA Jam 1 decision service with the Workflow solution, by following the instructions that are given in the [Exercise: Integrate within the Workflow solution](#).

4.2 *Finalize DBA Jam 1 Decision Service in Business console – Step by Step Instructions*

If you already did this exercise and want to redo it, restore its 'starting point' as follows: Go to Business console, delete the existing DBA Jam 1 project, and then import [DC-Jam-1-Start.zip](#) file in Box.

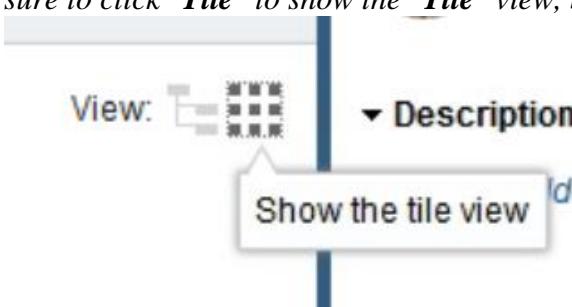
4.2.1 Understand the Business console environment

1. If you have not yet started the Sample Server, do so by following step 12 of [Get Started with this Sub-Scenario – Step by Step Instructions](#).
2. At any time, if you want to read the local user documentation, start the local KC server, by following step 13 of [Get Started with this Sub-Scenario – Step by Step Instructions](#).
3. Start ODM **Business console**:
 - a. Double-click **DC BC** on the desktop.
 - b. If you are already logged in, log out.
 - c. Log in by entering the following credentials:
 - login: **Paul**
 - password: **Paul**

4. Click the **Library** tab at the top of the screen to open the **Library** page.

If clicking **Library** does not open the **Library** tab but instead apparently downloads this page as an HTML file, use **Shift+Click** instead of (simple) **Click** to open it.
The same advice applies to all tabs (**Home**, **Library**, **Work**, **Administration**...). For more information, see [Clicking a tab of the Business console does not open the tab but instead downloads an HTML file](#) in the troubleshooting section.
5. Click **DBA Jam 1**, to get to the view that shows the Releases and Branches for the decision service.
6. Click **main** to open the **DBA Jam 1** decision service's **main** branch.

Known limitation: If your web-browser is Firefox, you can click the DBA Jam 1 **main** branch to open it only if it is shown in the "**Tile**" view. On Firefox, make sure to click "**Tile**" to show the "**Tile**" view, before clicking **main** on the **Tile** view.



The **main** branch of the DBA Jam 1 decision service opens:

The screenshot shows the SAP Fiori Decision Center interface. At the top, there are navigation links: 'Decision Center' (highlighted in blue), 'HOME' (with a house icon), 'LIBRARY' (with a folder icon), and 'WORK' (with a checkmark icon). Below these, a breadcrumb trail says 'All Decision Services >'. The main title is 'DBA Jam 1' with a gear icon. Below the title are two tabs: 'Releases' (selected) and 'Branches'. A large '+' button is visible. In the center, a card displays the 'main' branch, which was created by Paul on Jun 27, 2018. It features an upward arrow icon.

7. Go to the **Decision Artifacts** tab, if not yet in: You can see the list of projects that you have seen in the previous exercise in Rule Designer.
8. Click **Types** and select **All Types** to make sure that all elements, including ruleflows and operations, are visible:

The screenshot shows the 'Types' filter dialog. At the top, there are buttons for 'All Projects' and 'Types (1 / 5)'. The 'Types (1 / 5)' button has a close icon. Below these are two columns of checkboxes:

| | |
|--|---|
| <input checked="" type="checkbox"/> All Projects | <input checked="" type="checkbox"/> All Types |
| <input checked="" type="checkbox"/> DBA Jam 1 | <input checked="" type="checkbox"/> Rules and Decision Tables |
| | <input checked="" type="checkbox"/> Ruleflows |
| | <input checked="" type="checkbox"/> Operations |
| | <input checked="" type="checkbox"/> Variable Sets |
| | <input checked="" type="checkbox"/> Resources |

At the bottom right of the dialog is a blue 'Apply' button.

9. Look at the elements in the decision service.

- a. Select **Operations > computeLoanRequestParameters**.

You see the overview of this operation, with the ruleset name and its ruleflow. The ruleflow is used to order the sequencing of the rules. You also see the input/output parameters that IT has set for this operation.

- b. Check that the decision service signature is well defined:

Compare the ruleset parameters...

| Name | Type | Verbalization | Initial Value |
|----------|----------|---------------|---------------|
| borrower | borrower | the borrower | |
| loan | loan | the loan | |

With the decision service signature:

Operations > **computeLoanRequestParameters**

This decision operation has no description.

Overview

Source Project: DBA Jam 1

Extracted Rules: Default Validator

Ruleset Name: computeLoanRequestParameters

Main Ruleflow: mainRuleFlow

Inputs and Outputs

Input:

Input/Output:
DBA Jam 1
(x) parameters
the borrower
the loan

Output:

c. Select **mainRuleFlow**.
You see the order in which the rules must be executed, but two of the rule tasks have a warning icon.
d. Click each node to read the warning message, which informs you that these rule tasks do not contain any set of rules to be executed.

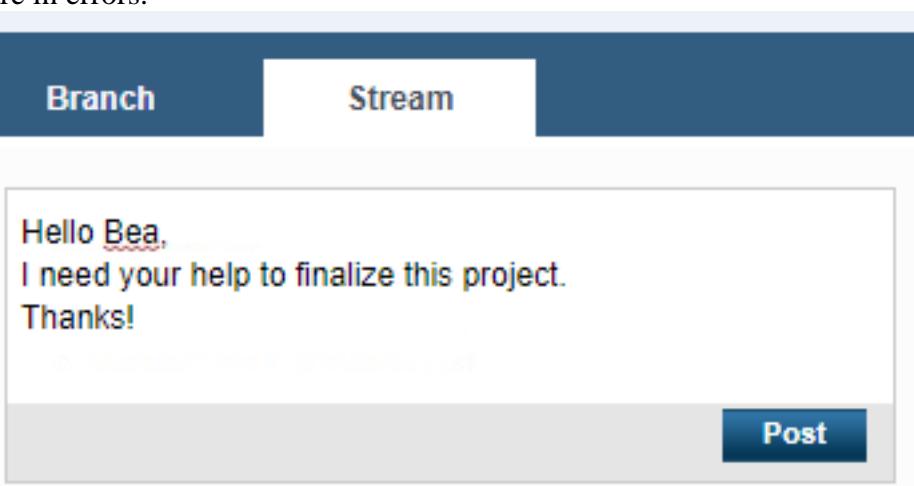
↓


Loan features

Empty rule task

×

▼

- e. Go to the **compute loan features** folder and select the **duration** rule.
You see that this rule is empty.
 - f. Open the **yearly interest rate** decision table.
This decision table shows errors, and you can see that some rows are missing.
10. Go to the **Tests** tab.
A test suite has already been created.
You can click it to see the details:
- **Operation:** the set of rules that you are testing
 - **File to use:** the scenario file containing your expected results. A scenario file is provided for you.
- 
11. Click **Run** to run the test.
When the test is finished, you see that the status indicates errors.
12. Click the report.
In the report, you see that some rules failed.
Expand the report to see which rules failed, and try to identify why...
Answer: The test fails because the results specified by Bea under "Expected" are the ones to obtain after all the rules, the decision tables and the ruleflow are ready. For now, because these artifacts are missing or are badly defined, the results (**bankruptcy score**, **corporate score**, and so on) are all undefined and default to 0.0, which is not the expected results...
You will ask Bea, a rule author in your team, to correct these elements.
13. Close the report.
14. Go back to the **Decision artifacts** tab.
15. In the right-hand panel, select the **Stream** tab.
16. **(Optional)** Post a comment to ask for Bea's help correcting the rule artifacts that are in errors.
- 

17. Log out of the Business console.

Your role, as Paul, is now over. You must now hand over the decision service to Bea, the business user, who will author and complete the decision service with proper definition of the ruleflow and author the missing rules and rows in the decision table.

Note: The `testsuite.xlsx` spreadsheet is available in the `Backup` folder, sub-folder `Jam-1`. You cannot open it because there is no MS Office Excel on the virtual machine, but you can use it for the tests.

4.2.2 Edit the ruleflow

You now act as Bea to correct the issues found by Paul.

1. Log in to the Business console with the following credentials:
 - username: **Bea**
 - password: **Bea**
2. Go to the **Home** tab, then **Stream** to check the comments.
Because Bea follows the stream (she selected the gold start), you can see here that Paul has asked you to make some changes. Otherwise, you cannot see any comment.

▼ New Comments in Activity Stream



Paul created a new post

Hello Bea,
I need your help to finalize this project.

Thanks!

Start by editing the ruleflow.

3. Open the DBA Jam 1 "**main**" branch.
4. Select **mainRuleFlow**.

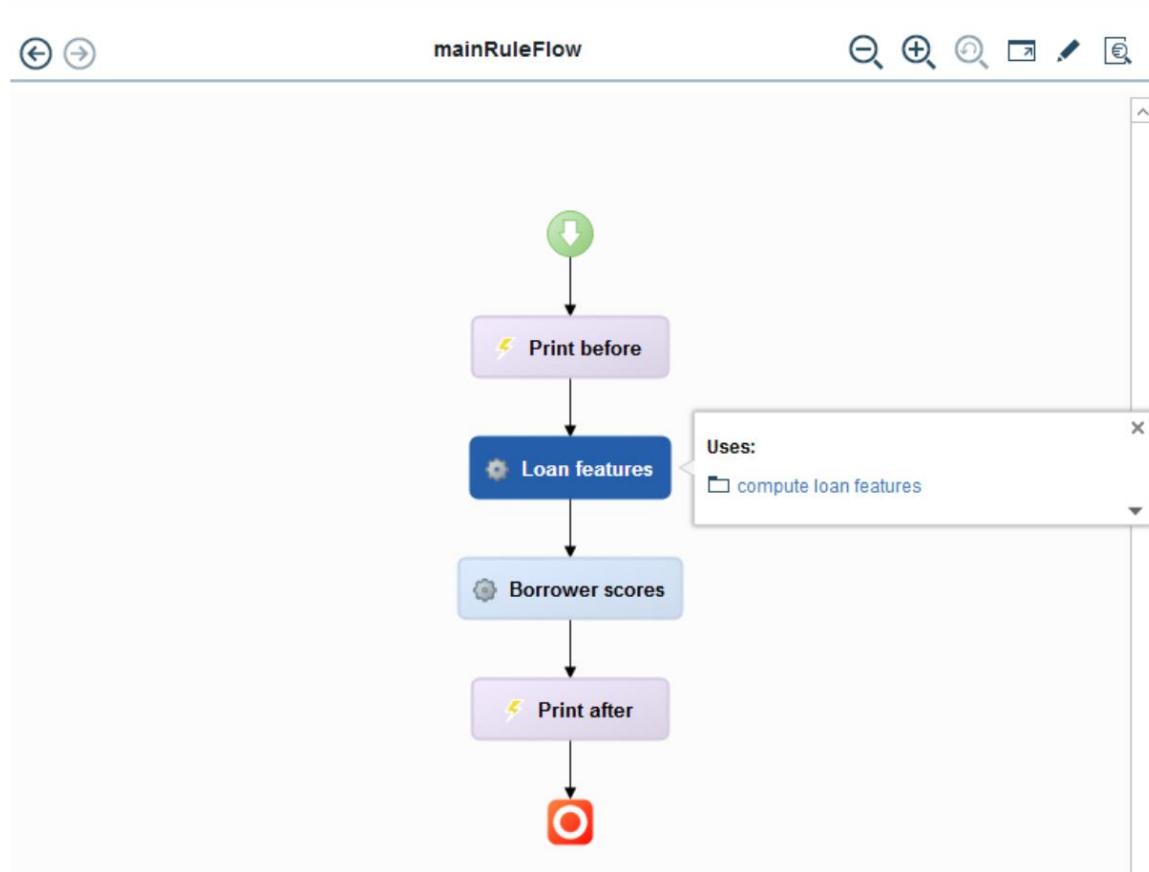
Note: If ruleflows are not visible, check that all types are selected to be displayed. See step 6 in part 4.2.1.



5. To edit the ruleflow, click **Edit** .
You can see that two tasks have warnings.
You must attach the correct rule packages to them.
6. Click **Loan features**, then click **Edit** next to **Uses** to select the rule package.
7. Select **Compute loan features** and add it to the selected artifacts with the **Add** button.
8. Click **OK**.

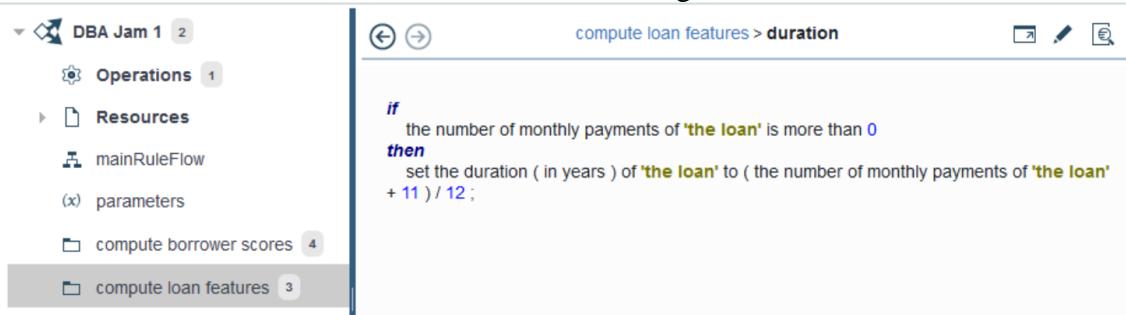
9. Similarly, click the **Borrower scores** task and add the **Compute borrower scores** to the task.
10. Click **Save** in the top toolbar.
11. Click **Create new version**.

Your ruleflow should look like this:

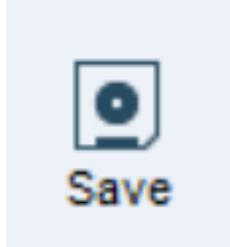


4.2.3 Create a rule and edit a decision table

1. In the **Decision Artifacts** tab, select the **Compute loan features** rule package.
2. Hover over the **duration** rule and click **Edit**.
3. Author the **duration** rule, so it looks like the following rule:



4. You can also copy the content for this rule, found in the [Helper.txt \(Box\)](#) file, and paste it in the editor.



5. When done, click **Save** and **Create a new version**.
6. Go back to the **main** branch.
7. Hover over the **yearly interest rate** decision table and click **Edit**.
8. Choose automatic row order.

Choose how you want the row order to be set



This decision table contains errors, represented by orange icons. You can hover over the icons to have more details. In this case, the range of values in the duration column has gaps: there are no rows for values between 9 and 12 (the rows at the beginning of the table, up to row #8 cover values that are lower than or equal to 8, while the row #9 and the next ones up to the end of the table, cover values that are greater than or equal to 13). To complete the table, add rows to consider values between 9 and 12:

9. Select the four rows from row #5 to row #8.
10. Right-click the column with the rows numbers and select **Copy**.
11. Right-click the row #9 and select **Insert copied rows**.
The four copied rows are added to the table after the row #8; these copied rows are now numbered row #9 to row #12. The row that was previously row #9 is now row #13.
12. In the **duration** column, in the newly added rows #9 to #12, change the minimum value to 9 and the maximum value to 12.

13. In the column **Yearly interest rate**, in these rows #9 to #12, change the values as follows:

| | Duration | | Loan to value | | Yearly interest rate |
|----|----------|-----|---------------|-----|----------------------|
| | min | max | min | max | |
| 3 | < 5 | | 0.8 | 0.9 | 0.053 |
| 4 | < 5 | | ≥ 0.9 | | 0.055 |
| 5 | 5 | 8 | 0 | 0.7 | 0.056 |
| 6 | 5 | 8 | 0.7 | 0.8 | 0.057 |
| 7 | 5 | 8 | 0.8 | 0.9 | 0.058 |
| 8 | 5 | 8 | ≥ 0.9 | | 0.059 |
| 9 | 9 | 12 | 0 | 0.7 | 0.06 |
| 10 | 9 | 12 | 0.7 | 0.8 | 0.061 |
| 11 | 9 | 12 | 0.8 | 0.9 | 0.062 |
| 12 | 9 | 12 | ≥ 0.9 | | 0.063 |
| 13 | 13 | 17 | 0 | 0.7 | 0.064 |
| 14 | 13 | 17 | 0.7 | 0.8 | 0.065 |
| 15 | 13 | 17 | 0.8 | 0.9 | 0.066 |
| 16 | 13 | 17 | ≥ 0.9 | | 0.067 |
| 17 | 18 | 25 | 0 | 0.7 | 0.068 |
| 18 | 18 | 25 | 0.7 | 0.8 | 0.069 |
| 19 | 18 | 25 | 0.8 | 0.9 | 0.07 |
| 20 | 18 | 25 | ≥ 0.9 | | 0.08 |

14. Save your decision table (and create a new version).
 15. (Optional) Go to the **Stream** tab in the **main** branch and post a comment to Paul, telling him that you made the requested changes.
 16. Log out.

4.2.4 Test your rules in the Business console

Acting as Paul again, you will now validate the rules by testing them against an Excel scenario file where you provided the values you expect to have. You then run the test in the Business console to compare the actual results with your expected results.

Log in again as Paul to test the decision service and make sure that all errors have been corrected and that the changes did not introduce unexpected behavior.

1. Log in with the username **Paul** and password **Paul**.

2. In the main branch of the decision service **DBA Jam**, select **Timeline**



in the top toolbar. You can see the changes that have been made in this branch by you and your collaborators. Here, you see that Bea has made some changes.

The screenshot shows a timeline entry for a user named Bea. The entry details a change made 'Today, 4:55:58 PM' where Bea created a version of the 'yearly interest rate rule'. The version number is listed as [v1.1]. To the right of the entry is a large double-headed arrow icon, indicating a comparison or history function.

3. Click **Exit Timeline** in the top toolbar to exit the timeline.

4. Go to the **Tests** tab.

5. Run the test suite.

After the test finishes running, you see that the status is green and shows no errors.

6. Open your latest test report to see the details.

The screenshot displays a test summary report. At the top, it shows a green circular progress bar with '100%' in the center, indicating a success rate. Below this, under 'Scenario success rate', it shows '2 Succeeded' and '0 Failed'. The report lists the following details:

| Test suite | Test Suite (latest) |
|-------------------|---|
| Operation | computeLoanRequestParameters |
| Output | None |
| Decimal precision | 2 decimal places |
| Run by | Paul on Simulation |
| Rule source | Rules from the branch ' main ' using the extractor ' ci ' |
| Starting point | The ruleflow task: mainRuleFlow |

Below the summary, there is a 'Results' section with a table:

| Status | Test | Result | Expected |
|--------|---|--------|----------|
| | Scenario Paul | -100.0 | -100.0 |
| | the bankruptcy score of the borrower equals | -100.0 | -100.0 |

7. In the section **Results**, expand the list of fired rules.
You see the results for each rule.
8. Click **Details** to see the collections that were tested and the expected results.
9. Close the report.

4.2.5 Deploy the decision service to Rule Execution Server

Now that you have validated your rules, you can deploy the project by using a deployment configuration listed in the **Deployment** tab.

Note: Only users with configuration manager or administrator rights can create deployment configurations. Paul is one of these users, but not Bea.

Note: During this step, you will deploy your RuleApp / ruleset. Because the version v1.0 of this RuleApp already exists in Rule Execution Server, you will create a new version of these artifacts in Rule Execution Server. In all cases, you could always roll-back to the version v1.0 as a backup solution either by deleting all other versions in the Rule Execution Server console, or by using version 1.0 in the ruleset path when invoking the decision service.

In the Business console, deploy your RuleApp:

1. Double-click the **DC BC** icon on the desktop to open the **Business console**, and log in as Paul.
2. Go to the **main** branch of the **DBA Jam 1** decision service (from the Library tab)
3. Go to the **Deployment** tab.
4. Click the existing **LocalRES**.

The LocalRES deployment configuration opens.

This configuration is wrong as, among other points, the RuleApp name is "**DBA_Jam**" while the Workflow solution expects it to be "**DBA_Jam_1**".



5. Click **Edit** to modify this configuration, and ensure that its properties are as follows:
 - a. **General:** This tab provides an overview of the deployment configuration, including the name, type, and RuleApp name and base version number.
 - Rename the configuration name from "**LocalRES**" to "**Local_Deployment_Configuration**".
 - Change the **RuleApp name** into "**DBA_Jam_1**" ("*DBA_Jam*" is not the name that is expected by the Workflow solution).
 - b. **Operations:** This tab lists the decision operations to deploy.
 - Make sure that the **computeLoanRequestParameters** operation is selected for deployment.Decision operations define how the rules you are changing are used in specific rulesets for deployment. Hover over the decision operation name to see its content.

The screenshot shows the Oracle JAM Deployment Configuration interface. The top navigation bar includes tabs for General, Operations, Targets, Ruleset Properties, Groups, and Deployment Snapshot. The 'Operations' tab is currently selected. On the left, a sidebar lists 'Operations' with two items checked: 'Operations' and 'computeLoanRequestParameters'. To the right, a detailed view of the selected rule set 'computeLoanRequestParameters' is shown, listing properties such as Source Project (DBA Jam 1), Extracted Rules (Query: none, Validator: Default Validator), Ruleset Name (computeLoanRequestParameters), Main Ruleflow (mainRuleFlow), Input (none), Input/Output (the borrower, the loan), and Output (none).

- c. **Targets:** This tab lists where the rules can be deployed.
→ Make sure that the check-box for the **Local_RES** (JAM deployment server) is selected and that all other check-boxes are cleared.
 - d. **Ruleset Properties:** This tab defines the versioning policy for each deployment.
→ Keep "**Increment minor ruleset version numbers**" as the policy.
 - e. **Groups:** This tab lets the administrator choose which groups can deploy using this deployment configuration.
As an administrator, Paul can see all the deployment configurations.
→ *All is properly set, so no action is needed here.*
 - f. **Deployment Snapshot:** This tab defines whether a snapshot is taken when the deployment is done.
→ Keep "**Automatically create a deployment snapshot**" selected.
 - g. Click **Save**.
 - h. Click **Create New Version**.
6. Select the **Local_Deployment_Configuration** target server.



- 7. Click **Deploy** from the menu-bar.
A **Deploy main** window opens.
- 8. In the **Deploy main** window:
 - a. Double-check that the **Target** is set to **Local_RES** and not to RuleApp archive; check that the other characteristics of the deployment are as defined during step 5.
 - b. Click **Deploy**.
The Deployment status window opens.
 - c. Check that the deployment succeeded, and then click **OK** to close the Deployment status window.

If you have any errors during this deployment, see [Error when deploying the decision service to RES](#).

9. Back in the **Deployments** tab, you can see a new deployment report:

The screenshot shows a software interface with a top navigation bar containing tabs: Decision Artifacts, Queries, Tests, Simulations, and Deployments. The Deployments tab is selected, indicated by a blue background. Below the tabs is a secondary navigation bar with Configurations and Reports. The main content area contains a table header with columns: Name, Status, Run By, and Report. A single row is visible in the table, showing a deployment named "Report 2018-07-05_06-23-0...", which is successful (indicated by a green checkmark) and run by "Paul". The table also shows "Total: 1 Selected: 0" and a page number "1" with navigation arrows.

| Name | Status | Run By | Report |
|---|-------------------------------------|--------|--------|
| <input type="checkbox"/> Report 2018-07-05_06-23-0... | <input checked="" type="checkbox"/> | Paul | 7/ |

10. Click the deployment report to see what was deployed and how.

The deployment report includes the target server, the configuration name, the ruleset, the deployment time, the version of the ruleset, and the deployment snapshot.

11. Take some time to reflect on why you have the indicated versions.

4.2.6 Verify the deployment in Rule Execution Server

1. On the desktop, click the **RES Console** shortcut.
2. Log in to the RES console by using the following credentials:
 - a. login: **resAdmin**
 - b. password: **resAdmin**
3. In the **Explorer**, you see the list of deployed rule apps.
4. Click **DBA_Jam_1**.

You see the list of deployed rulesets: **computeLoanRequestParameters** is deployed (at least twice, if you did [Deploy the decision service to Rule Execution Server](#)).

5. Click the "**computeLoanRequestParameters**" ruleset to view it.

Its version is **1.***, where * might be any number > 0.

6. Note the exact ruleset path of your deployed decision service. You need it to integrate with Workflow.

This ruleset path is something like: **DBA_Jam_1/<version>/computeLoanRequestParameters/<version>**

When invoking the DBA Jam 1 decision service from the Workflow solution, you can either:

- Use the ruleset path that you identified here
- Remove **<version>** (in which case, you would use: **DBA_Jam_1/computeLoanRequestParameters**)
- Replace **<version>** with **/latest** (in which case, you would use: **DBA_Jam_1/latest/computeLoanRequestParameters/latest**)

-- or --

- Replace <version> with 1.0, if you want to use the pre-deployed RuleApp.

Depending on your choice:

- If you do not specify any version in the ruleset path (DBA_Jam_1/computeLoanRequestParameters) or if you specify /latest in the ruleset path (DBA_Jam_1/latest/computeLoanRequestParameters/latest), the most recently deployed decision service will be used.
- If you specify a version in the ruleset path, for example: DBA_Jam_1/1.0/computeLoanRequestParameters/1.1, the corresponding versions of the RuleApp and of the ruleset will be used (they must exist or an error is raised).
- If you specify 1.0 for both the RuleApp version and the ruleset version in the ruleset path, that is: DBA_Jam_1/1.0/computeLoanRequestParameters/1.0, the pre-deployed RuleApp will be used.

4.2.7 (Optional) Create a RuleApp archive for your DBA Jam 1 decision service

In this optional part of the exercise, you see how to create an RuleApp archive for the DBA Jam 1 decision service that you created.

Create such a RuleApp archive if, after being done with this ODM sub-scenario, you plan to eventually use the DBA Jam 1 Decision Service that you created in your current BlueDemos environment into another BlueDemos environment.

For example: You developed the DBA Jam 1 decision service in an environment created from Template 5, you followed this ODM sub-scenario up to here, and you plan to later use your created DBA Jam 1 decision service in an environment created from Template 7 (for integration purpose).

To create a RuleApp archive for the DBA Jam 1 decision service, perform the following steps:

1. Log in to the RES console and open the RuleApp view on the "**DBA_Jam_1**" RuleApp as indicated in steps 1 – 3 of [Verify the deployment in Rule Execution Server](#).

RuleApp View

The screenshot shows a user interface for managing RuleApps. At the top, there is a horizontal toolbar with four buttons: 'Add Ruleset' (with a plus sign icon), 'Add Property' (with a plus sign icon), 'Download Archive' (with a green arrow icon), and 'Edit' (with a pencil icon). Below the toolbar, there is a list of ruleapps. The first item in the list is '/DBA_Jam_1/1.0', which is highlighted. To the left of this item is a small blue cube icon.

2. In the RuleApp View for DBA_Jam_1/1.0, click **Download Archive**



3. Save the downloaded RuleApp archive (a JAR file) in the location of your choice, with the name of your choice (for example: DBA_Jam_1_RuleApp.jar).

4.2.8 Test the deployed DBA Jam 1 decision service in Rule Execution Server

1. Log in to the RES console and open the Ruleset view on the "computeLoanRequestParameters" ruleset as indicated in steps 1 – 5 of [Verify the deployment in Rule Execution Server](#)
2. On the Ruleset View for the `computeLoanRequestParameters` ruleset, click "Retrieve HTDS Description File".
3. On the Retrieve HTDS Description File window that opens:
 - Select "REST" as the "Service protocol type".
 - Select "OpenAPI – JSON" as the Format.
 - Select both the **Latest ruleset version** check-box and the **Latest RuleApp version** check-box.

By selecting these two check-boxes, you make sure that the HTDS description file corresponds to the latest version of this decision service (the most recently deployed), including potential updates of this decision service.

The screenshot shows the 'Retrieve HTDS Description File' dialog box. At the top, it displays the path `/DBA_Jam_1/computeLoanRequestParameters`. Below this, there is a section labeled 'Service protocol type' with a radio button for 'REST' selected. Under the 'Format' label, a dropdown menu shows 'OpenAPI - JSON'. A group of checkboxes is present, with 'Latest ruleset version' and 'Latest RuleApp version' checked. There are also other unchecked options: 'Decision trace information' and 'Proxy for API Connect'. At the bottom of the dialog are four buttons: 'Cancel', 'View', 'Download', and 'Test'.

4. Click **Test** to test the decision service.
The UI to test the decision service opens.
The UI generates the decision service signature (inputs and output parameters) as JSON.
5. In this UI, perform a series of tests, with valid or invalid data, to test the ruleset.

- i. Modify the parameters with some values, such as the ones in the **Jam-1-RequestAnswer-in-RES.txt** file in **C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Backup\Jam-1** folder.
 - ii. Click "Execute Request".
 - iii. Look at the Server Response and check that it matches your input parameters.
4. Back to the Rule Execution Server console > **Retrieve HTDS Description File** window:
 - a. Click **View** to see the Swagger definition file of the decision service.
 - b. Look at its definition and relate its content to the content of the decision service.
 - c. Take note of the URL for this file, which looks like the following one:
localhost:9090/DecisionService/rest/V1/DBA_Jam_1/1.0/computeLoanRequestParameters/1.1/OPENAPI?format=JSON

This Swagger definition file is what you need to invoke the decision service from an external application.

In the Workflow sub-scenario or in the End-to-End scenario, you will have the opportunity to see how to use this Swagger definition file to call the DBA Jam 1 decision service from a Service Flow.

Note: The URL to the Swagger file, when read locally, starts with `localhost`. You must replace this `localhost` with the host name as known by the calling application. For this sub-scenario, the VM 5 – ODM virtual machine has `vm-5.example.com` as its host name, so the URL to the Swagger file would start with `vm-5.example.com:9090/DecisionService/rest...`

Congratulations! You have a tested decision service, and identified the Swagger definition file that you can later use to integrate within a Workflow solution.

4.3 Finalize DBA Jam 1 Decision Service in Business console – Verification Instructions

You have successfully completed this exercise when you have used the Business console to edit and author rule artifacts, looked at the different capabilities offered to business users, then tested and deployed your decision service.

You have also understood how different team members, for example project manager and rule authors, work together and interact within the Business console.

To see the solution for this exercise: Open the Business console, delete the existing DBA Jam 1 project, and then import the [DC-Jam-1-Full-Answer.zip](#) file in Box.

If you need to redeploy the RuleApp for the DBA Jam 1 solution in RES: Open the RES console and deploy the [DBA_Jam_1_RuleApp.jar](#) file in Box.

4.4 Finalize DBA Jam 1 Decision Service in Business console – Summary

During this exercise, you finalized the DBA Jam 1 Decision Service:

- You created some rules and rows in a decision table.
- You tested it locally.
- You deployed it in Rule Execution Server.

The DBA Jam 1 Decision Service is therefore now available for invocation from the Workflow solution.

You are now ready to move to the next exercise.

4.5 Finalize DBA Jam 1 Decision Service in Business console – Troubleshooting

4.5.1 Error when deploying the decision service to RES

If you have an error when deploying the decision service to RES, check that you have properly defined the server where to deploy. This system activity is performed in the Enterprise console:

1. Access to the **Enterprise console**:
 - a. Double-click **DC EC** on the desktop to open the Enterprise console.
A command window quickly runs that opens a Chrome window and then closes.
 - b. Check that the Chrome window that opened shows the Decision Center > Enterprise console page at: <http://localhost:9090/teamserver>
 - c. If the **Privacy** dialog shows on top of the Enterprise console page, read its content as needed, and click "**Agree and Proceed**" to close it.
 - d. As needed³, enter the following username and password:
 - Username = Paul
 - Password = Paul
 - e. Click **Sign In**.
1. Go to the **Configure** tab.
2. Click **Manage Servers** under **Deployment**.
3. Look at the list of defined servers:
 - a. Select **Local_RES**.
 - b. Click **Edit** to check its content.
 - c. Ensure that **Local_RES** points to the local Rule Execution Server at: <http://localhost:9090/res>
 - d. Ensure that **Local_RES** has correct credentials (**resAdmin**) stored.
 - e. If you did any change, click **Save**, otherwise, click **Cancel**.
 - f. Click **Test** to test that this server is available for deployment.

³ Because of single sign-off, if you are currently logged in to the Business console, you should be automatically logged in to the Enterprise console, and vice-versa.

- g. If there is an error, re-do the above-listed checks or contact the demo owner for further help.
4. **Sign out** of the Enterprise console.
5. Close the Decision Center > Enterprise console page. You no longer need it.

4.5.2 Clicking a tab of the Business console does not open the tab but instead downloads an HTML file

If you switch between the desktop of your own machine (where you run the virtual machine) to the desktop of the virtual machine (where Decision Center > Business console is open) and then click **HOME**, **LIBRARY** or **WORK**, the Business console will not open the corresponding tab, instead, it will download the corresponding page as an HTML file.

To avoid this issue, use **Shift + Click** instead of **Click** when you switch from the desktop of your computer to the desktop of the virtual machine with the Business console opened.

5 Exercise: Define DBA Jam 2 Decision Service in Decision Composer

5.1 Define DBA Jam 2 Decision Service in Decision Composer – Introduction

In this exercise, you use Decision Composer to set up the second DBA Jam 2 decision service. This decision service determines whether a loan request from a customer is approved.

Acting as Bea, the (non-technical) business user, you start with an initial project for this second decision service. You start by running a simple test on this provided (but incomplete) decision service, and then augment the service to determine whether insurance is required, and if so at what rate.

Decision Composer presents a visual representation of the decision service where you can act on the decisions and the data fed in and out of the decisions. Work your way through the steps to learn more about Decision Composer now...

Important: A solution for this DBA Jam 2 decision service is already pre-deployed in the Rule Execution Server of the VM 5 – ODM virtual machine. It means that, even if you do not fully perform this exercise, you could still integrate this DBA Jam 2 decision service with the Workflow solution, by following the instructions that are given in the [Exercise: Integrate within the Workflow solution](#)

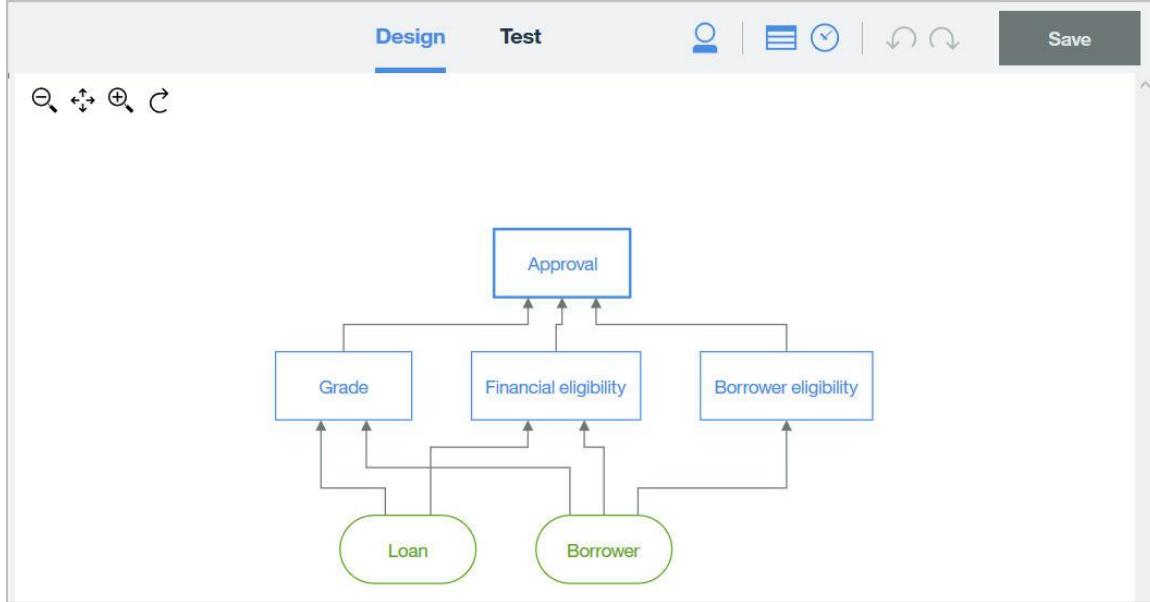
5.2 Define DBA Jam 2 Decision Service in Decision Composer – Step by Step Instructions

If you already did this exercise and want to redo it, restore its 'starting point' as follows: Go to Decision Composer, delete the existing DBA Jam 2 project, and then import the [Jam-2-Start.dproject](#) file in Box.

5.2.1 Explore Decision Composer

1. Open Chrome, and click the "**IBM Decision Composer**" bookmark to open IBM Decision Composer at <https://decision-composer.mybluemix.net>.
2. On the IBM Decision Composer page, click **Log in**.
3. Enter your **IBMid** and then click **Continue**.
 - a. If you do not have an IBMid, you may create one now by clicking "**Create an IBMid**".
4. Enter your IBMid **password** and click **Sign in**.
5. If the "**Use of Cookies**" dialog opens, read its content and then click **Ok** to close it.

- Import a project**
6. Click **Import a project** at the top of Decision Composer, to import the **Jam-2-Start.dproject** available in **C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario\Jam-2**. The **Home / DBA Jam 2 start** page opens.
 7. On the **Home / DBA Jam 2 start** page, you can see the decision service in its visual form, at its starting point for the work you are going to do in this exercise:



5.2.2 Test the current state of the decision service

Before adding insurance capabilities to the decision service, test the current state against sample data:

1. Click the **Test** tab.
2. In the **Test** tab, notice that a dataset is already available for Jane Doe.
3. Click **Run**. The decision service is tested with this dataset and returns a message saying that the loan has been approved:

The screenshot shows the Decision Composer interface in the 'Test' tab. On the left, under 'Input data', there is a 'Data sets' section with a 'Jane Doe' entry. The main area displays 'Input - Jane Doe' with a 'Run' button. Below it is an 'Edit as JSON' section showing a nested object 'Borrower' with fields: name (Jane Doe), corporate score (320), yearly income (100000), age (40), and zip code (12345). To the right, the 'Output' section shows 'Output - Jane Doe' from a run on 7/4/2018 at 9:42:13 AM. It lists a single item: { "approved": true, "message": "Congratulations! Your loan has been approved" }. Below the output are sections for 'Decision' (Approval), 'Messages' (No output messages to display), and 'Statistics' (2 rules, 1 ms execution time, 764 compilation time, 356 ms build time).

4. Check that wrong data lead to rejection, by changing the age of the borrower to 340 and rerun the test.
Under those conditions the loan is rejected, with the message "The borrower's age is not valid.", which is what you expect.
5. Reset the borrower's age to 40, so to restore a valid test dataset.
6. Save your work.

5.2.3 Update the decision service in Decision Composer

To add the insurance decision to your service, you will:

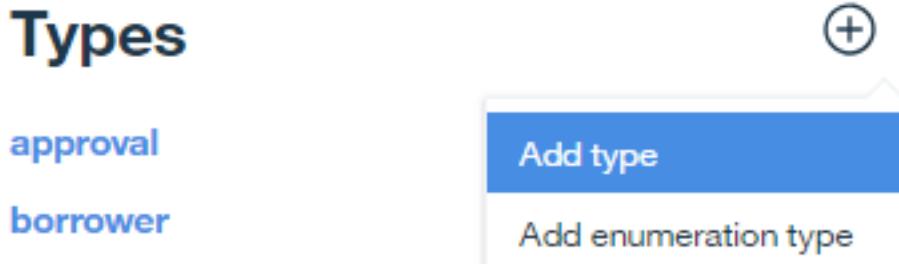
- Add a custom data type to handle the decision and the rate for the insurance. This is for convenience, instead of creating individual data nodes for each data entry.
- Create a new decision node in the diagram.
- Create the decision logic in the decision node: a rule and a decision table to obtain maximum coverage for possible input data.

To create the custom data type:

1. Return to the **Design** tab.
2. Click the **Show/Hide types** button to display the custom types.

The screenshot shows the Decision Composer interface in the 'Design' tab. At the top, there is a toolbar with tabs for 'Design' (which is selected) and 'Test'. Below the toolbar is a 'Types' section. A 'Show/Hide types' button is highlighted with a mouse cursor. The 'Types' section lists four custom data types: approval, borrower, grade, and loan. There is also a '+' button to add more types.

- In the **Types** panel, click the + sign and select ‘Add type’.



- Name your custom type ‘insurance’, and then click the + sign to create the following attributes:

The image shows a screenshot of a type configuration screen for 'insurance'. At the top, it says 'Name:' followed by 'insurance'. Below that is a table titled 'Attributes:' with two rows. The first row has 'Name' 'required', 'Type' 'boolean', and 'List' with a checkbox and a trash icon. The second row has 'Name' 'rate', 'Type' 'number', and 'List' with a checkbox and a trash icon. There is also a '+' button at the top right of the attributes table.

| Name | Type | List |
|----------|---------|---|
| required | boolean | <input type="checkbox"/> trash |
| rate | number | <input type="checkbox"/> trash |

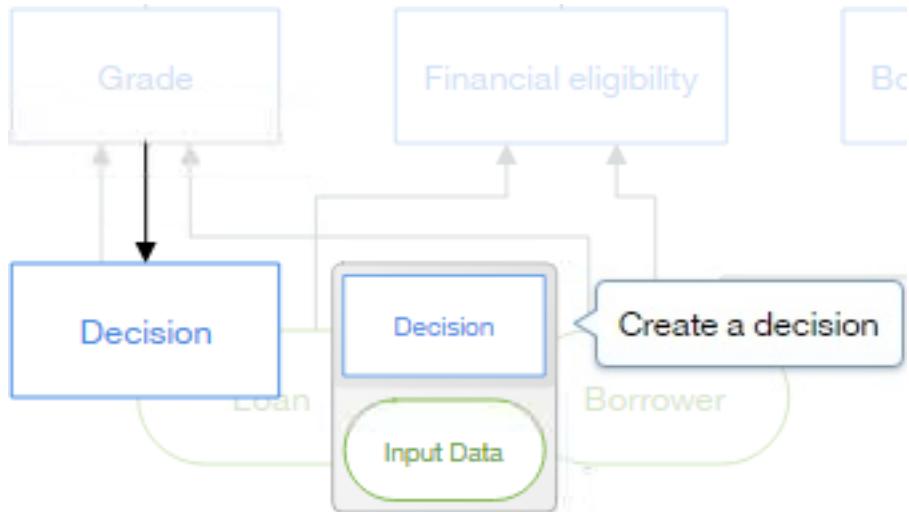
Important: When you edit a type, and add an attribute, the default type for the attribute is string. Make sure to change it to match the need – here, either use boolean and number – and then save your work.

Known issue: With Decision Composer (Experimental), saving a type might not well work. After you change any type, make sure that your change is saved. To do so, hide the Types panel, save your work, reopen the Types panel, and check that your changes are really considered. If not, please redo your changes and recheck them.

- Hide the **Types** panel and save your work.

To create a new decision node in the diagram:

- Hover over the **Grade** decision node, click the + that appears, and select **New node > Decision**.



2. Select this new node.
- Its description shows on the left panel of Decision Composer.
3. Set the **Decision name** for this new node as **Insurance**. For its **Type**, select in the list the **insurance** type that you created.

Insurance

Decision name:

Insurance

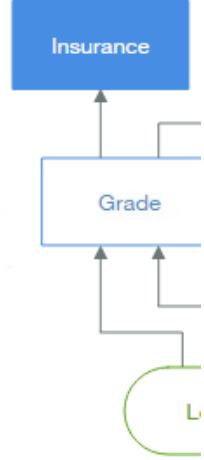
Type:



List

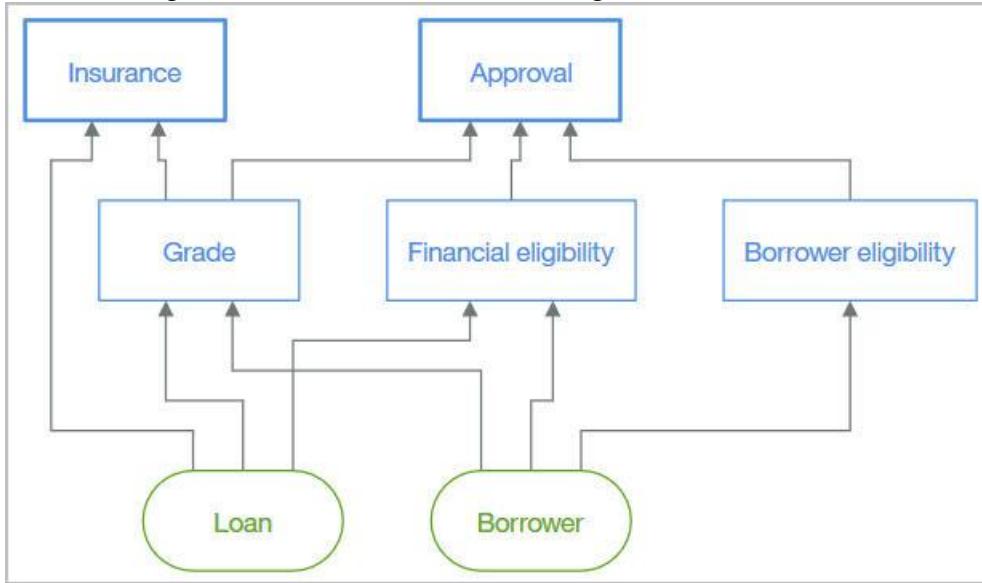
insurance

- Please select
- System types**
- boolean
- date
- date & time
- day of week
- integer
- month of year
- number
- string
- time
- year
- Custom types**
- approval
- borrower
- grade
- insurance**
- loan



4. Your new decision node takes input from both the **Grade** decision node and from the **Loan** data node. To create this link, hover over the **Loan** data node, click & hold the + sign that appears, and drag & drop on the **Insurance** node.

Your diagram now looks like the following one:



To create the decision logic of the **Insurance** node:

1. Click the **Insurance** node in the diagram, and click **Edit decision logic** on the left panel of Decision Composer.
2. In the new panel, click the + sign beside **Tables and rules**, and select **Add business rule**.

[← DBA Jam 2 start / Insurance](#)

Decision logic

[⚙️](#)

Tables and rules [+](#)

Click the + button or business rule [Add decision table](#) [Add business rule](#)

3. Click **Create rule**.
4. Click the name of the rule, and rename it as: **default-rule**
5. Edit its content so it is as follows:

```
set decision to a new insurance where
the rate is 0.02,
required is true ;
```

Important: Do not copy the content of this rule from this document, as it might contain characters that Decision Composer will not accept. Instead, you might want to copy the content of this rule from the [Helper.txt \(Box\)](#) file that is available from the Box folder for participants.

6. Click **Save**.
7. Click the + sign beside **Tables and rules**, and select **Add decision table**.
8. Select the following filters for your columns and then click **Create table**.

Filter...

| | |
|--|--------|
| <input checked="" type="checkbox"/> the grade of 'the grade' | string |
| <input type="checkbox"/> the message of 'the grade' | string |
| <input checked="" type="checkbox"/> the amount of 'the loan' | number |

Preview your decision table

Drag column headers in the preview below to change the order of the condition columns.

| grade | amount | | Insurance |
|-------|--------|-----|-----------|
| | min | max | |
| abc | 123 | 456 | |
| | | | |
| | | | |

Create table

9. Rename your decision table to **insurance**.
10. In the decision table, right-click the column headers and rename **grade** to **Grade**, and **amount** to **Loan amount**.
11. Enter the data to obtain the following table:

| | Grade | Loan amount | | Insurance | |
|----|-------|-------------|---------|-----------|-------------------------------------|
| | | min | max | rate | required |
| 1 | A | < 100,000 | | 0 | <input type="checkbox"/> |
| 2 | A | 100,000 | 300,000 | 0.001 | <input checked="" type="checkbox"/> |
| 3 | A | 300,000 | 600,000 | 0.003 | <input checked="" type="checkbox"/> |
| 4 | A | ≥ 600,000 | | 0.005 | <input checked="" type="checkbox"/> |
| 5 | B | < 100,000 | | 0 | <input type="checkbox"/> |
| 6 | B | 100,000 | 300,000 | 0.003 | <input checked="" type="checkbox"/> |
| 7 | B | 300,000 | 600,000 | 0.002 | <input checked="" type="checkbox"/> |
| 8 | B | ≥ 600,000 | | 0.007 | <input checked="" type="checkbox"/> |
| 9 | C | < 100,000 | | 0.007 | <input checked="" type="checkbox"/> |
| 10 | C | 100,000 | 300,000 | 0.006 | <input checked="" type="checkbox"/> |
| 11 | C | 300,000 | 600,000 | 0.009 | <input checked="" type="checkbox"/> |
| 12 | C | ≥ 600,000 | | 0.015 | <input checked="" type="checkbox"/> |

Some hints to improve the experience:

- Double-click in the cell to edit its content.
- For the **Loan amount**, enter 100000 in the cell and then right-click the cell, and choose **Change operator...** and select "<"
- You can select groups of similar cells when they are done, right-click and select **Copy**, and then paste the content to the equivalent cells.

You might have warning messages that indicate that a series of rows (example: row #1 to row #4) are overlaps of each other.

Edit preconditions

| | Grade |
|---|---|
| 1 | the grade of 'the grade' is <an object> |
| 2 | Errors |
| 3 | Line 1 overlaps with line(s) 2, 3, 4 |
| 4 | Line 2 overlaps with line(s) 1, 3, 4 |
| 4 | Line 3 overlaps with line(s) 1, 2, 4 |
| 4 | Line 4 overlaps with line(s) 1, 2, 3 |

12. To remove these warning messages, you must group cells for which you entered the same Grade, as follows:

- a. Select the cells in the "Grade" column that contain the same value (*in the given example, all four cells that contain "A"*).
- b. Right-click to get the selection's menu and select "**Group**".

- c. By doing so, you group the four rows and ensure that the four "A" are considered the same value.

Edit preconditions

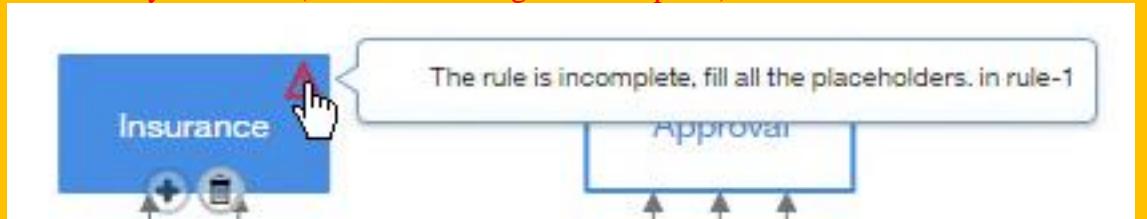
| | Grade | Loc |
|---|-------|-----|
| 1 | A | min |
| 2 | A | |
| 3 | A | |
| 4 | A | |

A context menu is open over the selected row (row 1). The menu items are: Cut, Copy, Paste, Insert copied cells, Insert row, Clear, Split, and Group. The 'Group' option is highlighted.

The warning messages are no longer visible.

- d. Repeat steps a – d for all groups of cells with the same Grade.

Decision Composer regularly validates your project, and, in case of issues, it displays a warning icon near "Design" (top bar), as well as in the decision node(s) that cannot be successfully validated (see the following screen capture).

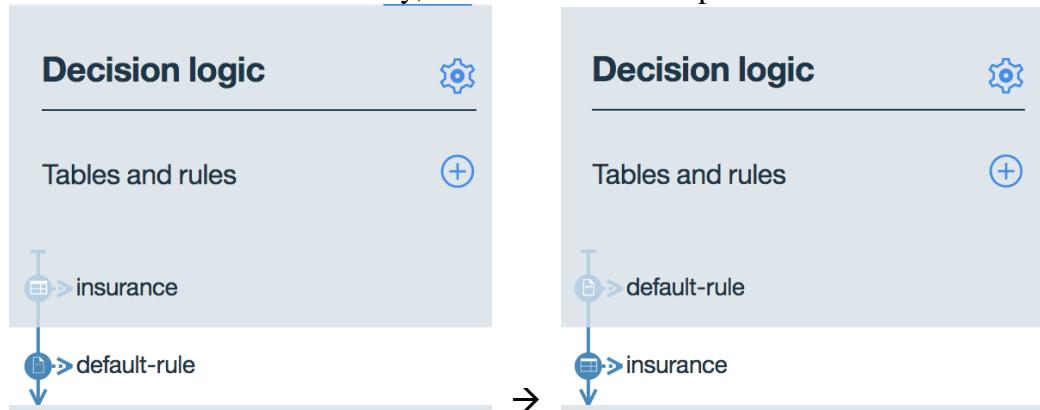


If you see such a warning icon, please check for the nodes in errors, as detailed in [Warning icons visible near "Design" or in a decision node](#).

Next step is to ensure that the order of the artifacts is correct in the **Decision Logic** panel on the left-hand side. Currently, the **insurance** decision table is above the **default-rule**

rule. It means that, at run time, the **insurance** decision table will be run first, and then the **default-rule** rule. This behavior is not the one that you need (the insurance would indeed always be required with a rate of 2%). Instead, you must have the **default-rule** rule run first, and then only the **insurance** decision table, so exchange the order of the two artifacts as follows:

13. In the **Decision Logic** panel on the left-hand side, drag the **default-rule** entry above the **insurance** decision table entry, so it is now on the top.



14. Save your work.

5.2.4 Test your DBA Jam 2 decision service in Decision Composer

Now that your decision service is complete, you need to:

- Run some tests using typical input data to make sure that it is working correctly.
- Obtain the information required for the Workflow solution to call your decision service.

To test your decision service:

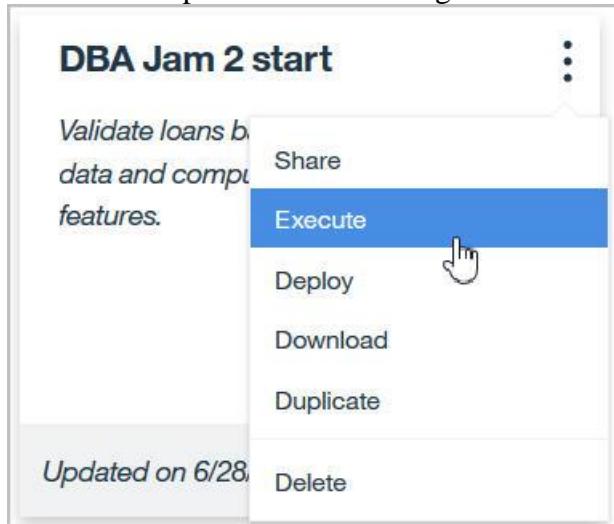
1. Click the **Test** tab.
In the **Test** tab, the age in the Jane Doe dataset is 40 so that your tests should be successful.
2. Click **Run**. Your decision service correctly provides a decision on the insurance and the rate:

The screenshot shows the Decision Composer interface. On the left, the **Input** section for "Jane Doe" displays four fields under the "Borrower" category: name (Jane Doe), corporate score (320), yearly income (100000), and age (40). A "Run" button is at the top right. On the right, the **Output** section shows two decisions: "Approval" (status: true, message: "Congratulations! Your loan has been approved") and "Insurance" (rate: 0.006, required: true). Below the decisions is a "Messages" section with a note: "No output messages to display."

3. Try some other typical input data until you are satisfied that your decision service works correctly in different situations.
4. Save your work.

To complete your work in Decision Composer, recuperate the information needed by Workflow to call your decision service:

1. Return to the **Home** tab.
2. Select the drop-down list to the right of **DBA Jam 2 start** and click **Execute**:



3. Copy the last line of the curl example and save it somewhere in a text file. **This line corresponds to the execution URL for use in the Workflow solution:**

Execute DBA Jam 2 start

Invoke your Decisions via simple REST calls ! Here's an example using curl :

```
curl ^
-H "Content-Type: application/json" ^
-H "Authorization: ApiKey %MY_DCOMP_API_KI%
-d "{\"Borrower\":{\"age\":0.0,\"corporateScore\"
"https://decision-composer.mybluemix.net/rest/p
< >
```

[Swagger API](#)

Alternatively, you can use our 'Decision Form' generator in order to run the decision manually, or automate its execution via a *Robotic Process Automation* tool

[Decision Form](#)

4. Click **Swagger API** to generate the definition of the types required as input/output so that the Workflow solution can call the decision service from a Service Flow.
5. Save the result as a **JSON** file.

5.2.5 (Optional) Create a RuleApp archive for your DBA Jam 2 decision service

In this optional part of the exercise, you see how to create an RuleApp archive for the DBA Jam 2 decision service that you created.

Create such a RuleApp archive if, after being done with this ODM sub-scenario, you plan to eventually use the DBA Jam 2 Decision Service that you created in your current BlueDemos environment into another BlueDemos environment.

For example: You developed the DBA Jam 2 decision service in an environment created from Template 5, you followed this ODM sub-scenario up to here, and you plan to later use your created DBA Jam 2 decision service in an environment created from Template 7 (for integration purpose).

To create a RuleApp archive for the DBA Jam 2 decision service, perform the following steps:

1. Return to the **Home** tab.
2. From the drop-down list to the right of **DBA Jam 2 start**, click **Deploy**.

DBA Jam 2 start



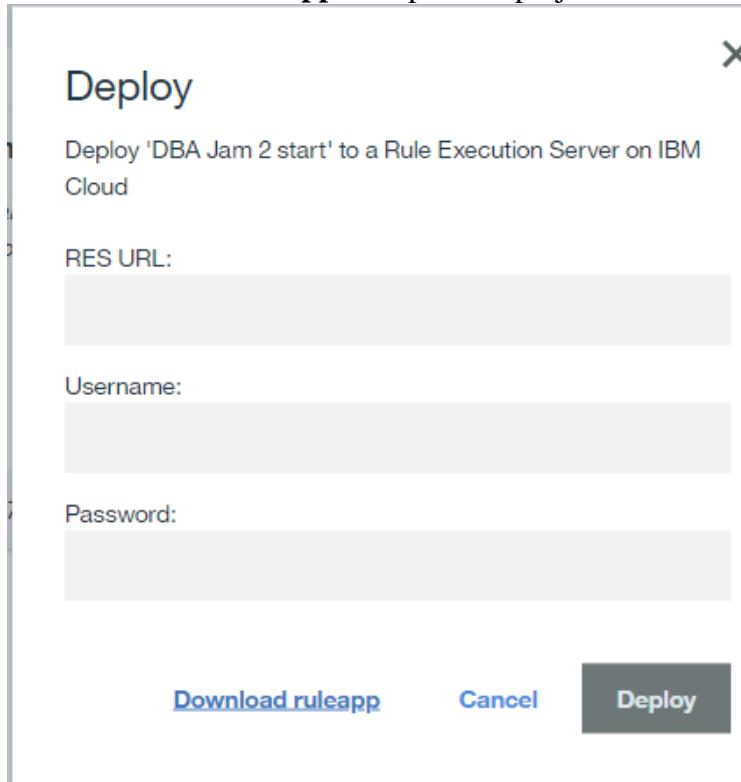
Validate loans by
data and compute
features.

Share

Execute

Deploy

3. Click **Download ruleapp** to export the project as a RuleApp:



4. Save the downloaded RuleApp archive (a JAR file) in the location of your choice, with the name of your choice (for example: DBA_Jam_2_RuleApp.jar).
5. Click **Cancel** (*not Deploy*).

5.2.6 (Optional) Deploy your DBA Jam 2 decision service in Rule Execution Server

In this optional part of the exercise, you see how to deploy the RuleApp for the DBA Jam 2 decision service in to the Rule Execution Server that is hosted on the virtual machine.

This step is optional because Decision Composer is backed with a Rule Execution Server (RES) on the Cloud that executes the decision service. However, you may want to perform this deployment for various reasons, including to:

- Learn how to do so.
- Have all your decision services within the same RES.
- Have a backup solution.

To deploy the RuleApp for the DBA Jam 2 decision service in to the same Rule Execution Server as where the RuleApp for the DBA Jam 1 decision service is deployed, perform the following steps:

1. If not done yet, create a RuleApp archive for the DBA Jam 2 decision service, as indicated in [\(Optional\) Create a RuleApp archive for your DBA Jam 2 decision service](#).
2. Open the Rule Execution Server console by double-clicking the **RES Console** icon on the desktop.
3. Log in to the Rule Execution Server console (`resAdmin / resAdmin`).
4. Deploy the RuleApp that you created in step 1 in to Rule Execution Server:
 - a. Go to the **Explorer** tab.
 - b. Click **RuleApps** under **Navigator** (left panel) to go to **RuleApp** View.
 - c. Click **Deploy RuleApp Archive**.
 - d. Click **Choose File** and browse your disk to the JAR file that you created in step 4.
 - e. Examine the versioning policies that you might use, and select the ones that matches your needs (or keep the default ones).
 - f. Click **Deploy**.
5. Note the versions of the deployed ruleset (RuleApp version, ruleset version).
6. The ruleset path is **DBAJam2_RuleApp/<version-A>/DBAJam2/<version-B>**
Where the <version-A> and <version-B> depend on the RuleApps that were already deployed in the RES before you deploy the new one, and the versioning policies that you selected in step 8.e.
7. Test the DBA Jam 2 decision service, similarly to how you tested the RuleApp for DBA Jam 1 decision service, in [Test the deployed DBA Jam 1 decision service in Rule Execution Server](#).
 - a. As you test the DBA Jam 2 decision service, take note of the URL to its Swagger definition file, similarly to how you identified it for DBA Jam 1 decision service, in [Test the deployed decision service in Rule Execution Server](#).

This Swagger definition file is what you need to invoke the DBA Jam 2 decision service from an external application.

In the Workflow sub-scenario or in the End-to-End scenario, you will have the opportunity to see how to use this Swagger definition file to call the DBA Jam 2 decision service from a Workflow Service Flow.

Note: The URL to the Swagger file, when read locally, starts with `localhost`. Naturally, you need to replace this `localhost` with the host name as known by the calling application. For this sub-scenario, the VM 5 – ODM virtual machine has **vm-**

5.example.com as its host name, so the URL to the Swagger file would start with **vm-5.example.com:9090/DecisionService/rest...**

5.3 Define DBA Jam 2 Decision Service in Decision Composer – Verification Instructions

You have successfully completed this exercise after you have successfully created the decision service 2 in Decision Composer and tested its execution.

To see the solution for this exercise: Open Decision Composer, delete the existing DBA Jam 2 project, and then import the [Jam-2-Answer.dproject](#) file in Box.

If you need to redeploy the RuleApp for the DBA Jam 2 solution in RES: Open the RES console and deploy the [DBA_Jam_2_RuleApp.jar](#) file in Box.

5.4 Define DBA Jam 2 Decision Service in Decision Composer – Summary

During this exercise, you created and tested the DBA Jam 2 Decision Service all in Decision Composer.

- You created parts of the data model.
- You created parts of the decision model (some rules and rows in a decision table).
- You tested the decision service locally.
- You deployed it to Rule Execution Server.

The DBA Jam 2 Decision Service is therefore available for invocation by the Workflow solution.

You are now ready to move to the next exercise.

5.5 Define DBA Jam 2 Decision Service in Decision Composer – Troubleshooting

5.5.1 Warning icons visible near "Design" or in a decision node

If you have a warning icon that is visible near "**Design**" in Decision Composer (see the top menu in the following figure), or in your decision node (see "**Insurance**" node in the following figure), Decision Composer faces an error when validating your project.

← Home / DBA Jam 2 start ⚪

Design ⚪ Test

User | List | ⏴ | ⏵ | ⏵ | ⏵ | Save

Insurance

Decision name:

Insurance

Type:

insurance

+ List

Edit decision logic

Description:

Describe the node (optional)

The diagram illustrates a decision tree structure. At the top level, there are two main nodes: "Insurance" (blue rectangle) and "Approval" (blue rectangle). The "Insurance" node has three children: "Grade" (blue rectangle), "Financial eligibility" (blue rectangle), and "Borrower eligibility" (blue rectangle). The "Approval" node also has three children: "Grade", "Financial eligibility", and "Borrower eligibility". At the bottom level, there are two green ovals: "Loan" on the left and "Borrower" on the right. Arrows point from each of the three "eligibility" nodes down to both the "Loan" and "Borrower" ovals, indicating that both factors are considered for both loan types.

To resolve this issue:

1. Click **Save** (top-right side of Decision Composer) to save your project.
 2. Click **← Home** (top-left side of Decision Composer) to go back to the **Home** page.
 3. Reopen your project.
 4. Check again.
 5. If warning icons remain, hover over each of them and read the detailed error messages to see how to resolve them.

6 Exercise: Integrate within the Workflow solution

6.1 *Integrate within the Workflow solution - Introduction*

In this exercise, you select the next step. You might indeed want to:

- Move to another environment to perform another sub-scenario, and stop running this ODM sub-scenario. See [Move to another environment](#)
- Stay in your current environment to integrate the two decision services that you created within the Workflow solution that is provided in your environment.
 - If you are in an environment based on Template 5, see [Stay in your environment from Template 5](#).
 - If you are in an environment based on Template 7, see [Stay in your environment from Template 7](#).

6.1.1 Move to another environment

If you want to move to another environment to perform another sub-scenario (for example: move to Template 2 to perform Datacap sub-scenario), you might still want to keep the work that you did so far with ODM.

For this purpose:

1. Back up your DBA Jam 1 RuleApp as instructed in **(Optional) Create a RuleApp archive for your decision service (DBA Jam 1)**.
2. Back up your DBA Jam 1 RuleApp as instructed in **(Optional) Create a RuleApp archive for your decision service (DBA Jam 2)**.

Congratulations! You're now done with this ODM sub-scenario. You can continue the demo by using the instructions of the sub-scenario that you selected. Please ignore the rest of this document.

6.1.2 Stay in your current environment

You stay in your current environment if you want to integrate the two decision services that you created as part of a Service Flow. For this purpose, you will gather the information about these two DBA Jam 1 and DBA Jam 2 decision services so to be able to integrate these. You will learn how to do so in this ODM sub-scenario, in the Workflow sub-scenario or in the End-to-End scenario, depending on your choice (see next).

Important: The Workflow solution in the VM 3 – Workflow virtual machine in your environment (based on Template 5 or on Template 7) is configured by default to use the latest version of the DBA Jam 1 and DBA Jam 2 decision services that are deployed in your VM 5 – ODM virtual machine, that is, the decision services that you just created by following this ODM sub-scenario.

6.1.2.1 Stay in your environment from Template 5

If you are currently using an environment created from Template 5, you now have three options:

1. If you want to integrate the ODM decision services that you created during this ODM sub-scenario within the Workflow solution of the VM3 of your Template 5, then you can use the Workflow solution that is available in VM 3 – Workflow in your environment (from Template 5). For this purpose, perform the step [Integrate within the Workflow solution – Step by Step Instructions – in VM 5 - ODM](#) and then either the step [Integrate the Rule Apps with Workflow – Verification Instructions](#) (if you did not modify the signature of any of the decision services) or the step [Integrate within the Workflow solution – Step by Step Instructions – in VM 3 – Workflow](#) (otherwise).
2. Although Template 5 is not really meant for this purpose, note that, if you want to ignore your work and instead integrate the decision services that are pre-deployed in the VM 5 – ODM virtual machine within the Workflow solution of your Template 5, you can do so as follows:
 - a. Restore these pre-deployed decision services as the latest versions in the Rule Execution Server. For this purpose, open the RES console, and make sure that only the RuleApp and rulesets with version 1.0 remain in RES – Delete all the others.
 - b. When the pre-deployed RuleApps with version 1.0 are the only RuleApps left, they are also the most recent ones, and you can integrate them within the Workflow solution of Template 5, by following the same steps as in the previous step 1.
3. If you want to stop your work, that is, do not want to integrate your decision services within a Workflow solution –neither the decision services that you created nor the ones that are pre-deployed in the VM 5 – ODM virtual machine -, then you're done with this ODM sub-scenario. **Congratulations! You're now done with this ODM sub-scenario. Please ignore the rest of this document.**

6.1.2.2 Stay in your environment from Template 7

If you are currently using an environment created from Template 7, you now have three options:

1. If you want to integrate the ODM decision services that you created during this ODM sub-scenario within the Workflow solution of the VM3 of your Template 7, then perform the step [Integrate within the Workflow solution – Step by Step Instructions – in VM 5 - ODM](#), and then proceed with the **Workflow sub-scenario** that you can find in the [**DBA Blue Demos 2018 – Material for Participants > 3. Workflow sub-scenario**](#) Box folder.
2. Although Template 7 is not really meant for this purpose, note that, if you want to ignore your work and instead integrate the decision services that are pre-deployed in the VM 5 – ODM virtual machine within the Workflow solution of Template 7, you can do so as follows:

- a. Restore these pre-deployed decision services as the latest versions in the Rule Execution Server. For this purpose, open the RES console, and make sure that only the RuleApp and rulesets with version 1.0 remain in RES – Delete all the others.
 - b. When the pre-deployed RuleApps with version 1.0 are the only RuleApps left, they are also the most recent ones, and you can integrate them within the Workflow solution of Template 7, by following the same steps as in the previous step 1.
3. If you want to stop your work, that is, do not want to integrate decision services within a Workflow solution –neither the decision services that you created nor the ones that are pre-deployed in the VM 5 – ODM virtual machine -, then you're done with this ODM sub-scenario. **Congratulations! You're now done with this ODM sub-scenario. Please ignore the rest of this document.**
-

Note: Other Templates, 1, 2, 3, 4 and 6, are not to be used with this ODM sub-scenario, and their usages are therefore not covered by this document.

6.2 Integrate within the Workflow solution – Step by Step Instructions – in VM 5 - ODM

Use these step-by-step instructions in the **VM 5 – ODM** virtual machine to prepare your work if you want to integrate the decision services DBA Jam 1 and/or DBA Jam 2 that you created during this ODM sub-scenario within the Workflow solution of the **VM 3 – Workflow**.

1. Make sure you know the URL for the Swagger file definition and the ruleset path of the DBA Jam 1 decision service.
 - a. If you performed the [Exercise: Finalize DBA Jam 1 Decision Service in Business console](#), this URL and this ruleset path were identified during this exercise's section: [Test the deployed DBA Jam 1 decision service in Rule Execution Server](#)
 - b. If you did **not** perform the [Exercise: Finalize DBA Jam 1 Decision Service in Business console](#), you will use the pre-deployed RuleApp for DBA Jam 1.
When using this pre-deployed RuleApp for DBA Jam 1:
 - The URL to the Swagger definition file is: http://vm-5.example.com:9090/DecisionService/rest/V1/DBA_Jam_1/1.0/computeLoanRequestParameters/1.0/OPEN_API?format=JSON
 - The ruleset path is:
`DBA_Jam_1/1.0/computeLoanRequestParameters/1.0`
2. Download the Swagger file definition for DBA Jam 1 decision service and store it in the location of your choice.
3. Make sure you know the URL for the Swagger file definition and the ruleset path of the DBA Jam 2 decision service:

- a. If you performed the [Exercise: Define DBA Jam 2 Decision Service in Decision Composer](#), this URL and this ruleset path are as defined during this exercise's section: [Test your DBA Jam 2 decision service in Decision Composer](#)
 - b. If you performed the final [\(Optional\) Deploy your DBA Jam 2 decision service in Rule Execution Server](#) section of the [Exercise: Define DBA Jam 2 Decision Service in Decision Composer](#), you may also opt to use the DBA Jam 2 decision service that you deployed in the local Rule Execution Server. In this case, use the URL and the ruleset path that you defined during this final [\(Optional\) Deploy your DBA Jam 2 decision service in Rule Execution Server](#) section.
 - c. If you did **not** perform the [Exercise: Define DBA Jam 2 Decision Service in Decision Composer](#), you will use the pre-deployed RuleApp for DBA Jam 2 in Rule Execution Server. When using this pre-deployed RuleApp for DBA Jam 2:
 - The URL to the Swagger definition file is: http://vm-5.example.com:9090/DecisionService/rest/V1/DBAJam2_RuleApp/1.0/DBAJam2/1.0/OPENAPI?format=JSON
 - The ruleset path is: /DBAJam2_RuleApp/1.0/DBAJam2/1.0
4. Download the Swagger file definition for DBA Jam 2 decision service and store it in the location of your choice.

What next?

A) If you created your environment from **Template 7**, continue with the **Workflow sub-scenario** that you can find in the [DBA Blue Demos 2018 – Material for Participants > 3. Workflow sub-scenario](#) Box folder (as indicated in [Staying in your environment from Template 7](#)). **Ignore the rest of this document.**

B) If you created your environment from **Template 5** (as indicated in [Staying in your environment from Template 5](#)), and want to integrate your decision services within the Workflow solution, you will continue your work in the VM 3 – Workflow virtual machine.

Remember that the VM 3 – Workflow virtual machine is set up in such a way that the Workflow solution on VM 3 – Workflow uses by default the latest decision services that are deployed in the VM 5 – ODM virtual machine.

If the decision services that you created and deployed have the signatures that the pre-deployed Workflow solution needs, calling them from the Workflow solution should therefore require no action from you. That is the magic of ODM! Business users can update their rules and the calling application (here the Workflow solution) requires no update to consider the most recently authored rules.

If you strictly follow the instructions in this ODM sub-scenario, the signature (that is, the input parameters, input-output parameters and output parameters) of the DBA Jam 1

decision service and of the DBA Jam 2 decision service should be what the Workflow solution needs.

As a consequence:

1. If you did not modify the signatures of the two DBA Jam 1 and DBA Jam 2 decision services, you can see how your updated decision services integrate smoothly within the Workflow solution, without any action from you on this Workflow solution, by proceeding with [Integrate the Rule Apps with Workflow – Verification Instructions](#).
2. If you modified the signatures of any of the two DBA Jam 1 and DBA Jam 2 decision services, the pre-deployed Workflow solution of the VM 3 – Workflow will not work as is: You will have to delete the services built in this Workflow solution, and re-create them based on the two ODM decision services that you created with ODM. For this purpose, proceed with [Integrate within the Workflow solution – Step by Step Instructions – in VM 3 – Workflow](#)

6.3 Integrate within the Workflow solution – Step by Step Instructions – in VM 3 – Workflow

Follow the instructions in this section only if you created your environment from the Template 5 and if you modified the signature of the DBA Jam 1 decision service and/or of the signature of the DBA Jam 2 decision service.

If you have not changed the signatures of any of the decision services, you can instead directly proceed with [Integrate the Rule Apps with Workflow – Verification Instructions](#).

If you modified the signatures of any of the two DBA Jam 1 and DBA Jam 2 decision services, the pre-deployed Workflow solution of the VM 3 – Workflow will not work as is. Instead, you must modify this pre-deployed Workflow solution and integrate your ODM decision services within the Workflow solution.

For this purpose, follow the next series of steps:

1. [Resume the VM 1 – ECM and VM 3 – Workflow virtual machines](#)
2. [Delete the existing External Services](#)
3. [Using ODM JSON files, create two new external services](#)
4. [In the Service Flow, use the new external services](#)
5. [Take a snapshot and activate it](#)
6. [Use your process from within the Case solution](#)

6.3.1 Resume the VM 1 – ECM and VM 3 – Workflow virtual machines

In your environment, if not done yet, resume the following two virtual machines and ensure that they are properly connected:

1. Resume VM 1 – ECM.
2. Make sure it's properly connected (see bottom-right corner "connection" icon).
3. Resume VM 3 – Workflow.

4. Make sure it's properly connected.

Starting this point, and unless otherwise specified, steps are to be performed in the VM 3 – Workflow virtual machine.

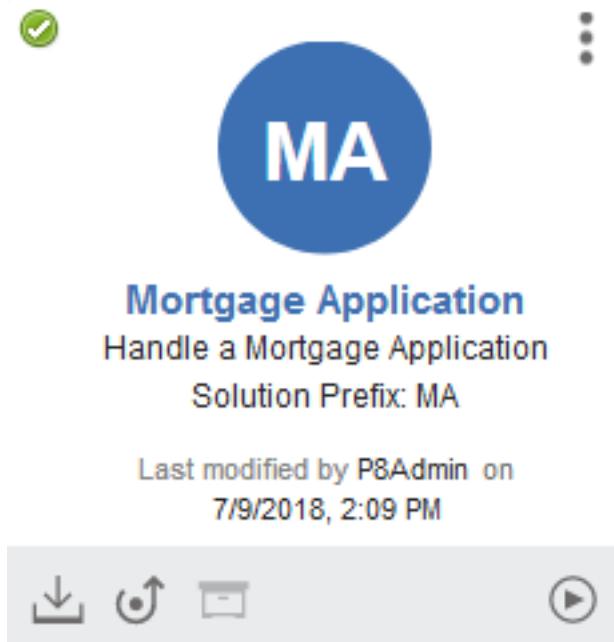
6.3.2 Delete the existing External Services

On the VM 3 – Workflow virtual machine, external services are already defined that are based on the pre-deployed versions of the two decision services. You must delete these two external services to use the ones that you created instead.

1. Connect to **VM 3 – Workflow** (Administrator / passw0rd).

Note: As a difference from the VM 5 – ODM (where you use Chrome), make sure you use **Firefox** web-browser when you work on the VM 3 – Workflow, because bookmarks are pre-defined for this web-browser to the various Workflow modules. On the VM 3 – Workflow virtual machine, if you use a web-browser other than Firefox, you will have to enter the corresponding URLs manually.

2. Open **Firefox**, expand the **Workflow** bookmark and click **Case Builder** to open the page at: <https://vm-34.example.com:9443/CaseBuilder>
3. Log in by using:
 - a. User name = P8Admin
 - b. Password = Think4meIf logging fails, check that the VM 1 – ECM virtual machine is **Running**.
4. Click the **Mortgage Application** tile to access it.



5. Go to Case Types.

The screenshot shows a top navigation bar with several tabs: Properties, Roles, In-baskets, Documents, Business Objects, Pages, and Case Types. The 'Case Types' tab is highlighted with a blue border. Below the tabs is a button labeled 'Add Case Type' with a question mark icon. The main content area has a title 'New Mortgage Application'.

6. Click New Mortgage Application.

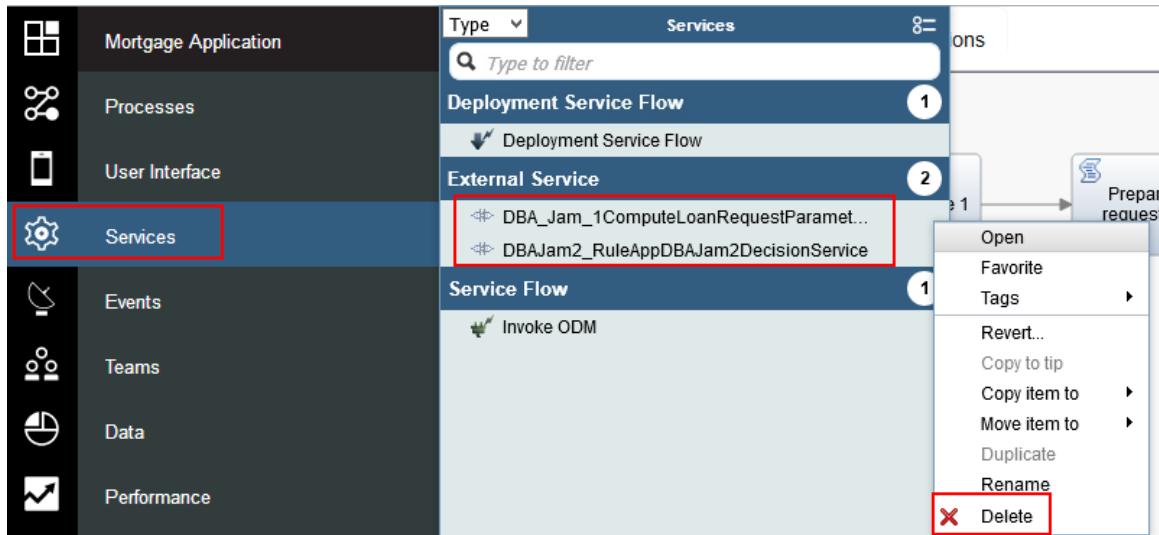
The screenshot shows the 'IBM Business Automation Workflow Case Builder' interface. The left sidebar has a 'Case Type' section with options: Properties, Views, Case Folders, Stages, Rules, and Tasks. The 'Tasks' option is selected. The main panel is titled 'Case Type Attributes' and contains fields for 'Case type name' (set to 'New Mortgage Application') and 'Case type unique identifier' (set to 'MA_NewMortgageApplication'). There is also a 'Case type description' field which is empty.

7. Go to Tasks.

8. Hover over "Finalize Loan Agreement" and click "Open Web Process Designer".

The screenshot shows a task card for 'Finalize Loan Agreement'. The card includes sections for 'Get decision from BPM', 'Precondition: Property expression: Is Application...', and 'Set: <None>'. To the right of the card is a context menu with an option 'Open Web Process Designer' highlighted with a blue box. Other menu items include 'Edit', 'Delete', and 'Properties'.

9. Click the Services section and delete the two existing External Services by clicking on the context menu icon and selecting Delete.



You have deleted the integration with the default decision services. You will integrate with the decision services of your choice now.

6.3.3 Using ODM JSON files, create two new external services

1. Retrieve the OpenAPI / Swagger JSON files for the two RuleApps that you want to integrate with, by using the URL that you saved in [Integrate within the Workflow solution – Step by Step Instructions – in VM 5 - ODM](#).
 - a. Alternatively, open the RES console from the VM 3 – Workflow image (in your Firefox web-browser, click the **Rule Execution Server** bookmark, or enter <http://vm-5.example.com:9090/res> in the address field), and retrieve the OpenAPI / Swagger JSON files directly, as you did in:
 - i. [Test the deployed DBA Jam 1 decision service in Rule Execution Server](#) for DBA Jam 1, from RES console
 - ii. [Test your DBA Jam 2 decision service in Decision Composer](#) for DBA Jam 2 from Decision Composer
 - iii. [\(Optional\) Deploy your DBA Jam 2 decision service in Rule Execution Server](#) for DBA Jam 2 from RES
2. Back to **Case Builder**, in the **Library** pane, hover over the **Services** section and click the + button and select **External Service**.

- In the **New External Service** dialog, select the option **Java, REST or Web Service** and click **Next >**.

New External Service

An external service lets you call a service or application that is external to IBM Business Automation Workflow.
[Learn More](#)



Java, REST or Web service
 Create an external service by discovering a Java, REST or Web service.

External Implementation
 Create an external service with an External Implementation binding.

The next page opens for the next step **Select a method to discover the service**.

- Click the browse icon and select the **DBA_Jam_1ComputeLoanRequestParametersDecisionService.json** JSON file downloaded for the first RuleApp.

Browse local files (Swagger) ▾

External service name:

File name:

Selecting the file auto-populates the **External service name**.

- Click **Next >**.

This page shows the operations included in the file. As the RuleApp contains one ruleset, you have a single operation in the list that is the name of the ruleset.

| <input checked="" type="checkbox"/> Operation Name |
|--|
| <input checked="" type="checkbox"/> callcomputeLoanRequestParametersDecisionServiceOperation |

- Click **Next >** again.

This page shows the options to create a server that contains the information such as the hostname, username, password, etc. Because there is an existing server, from the External Services created previously, select any of the servers in the list as they all point to the same ODM server.

By default, the wizard has the option "**Create a new server**" selected, with a default name; make sure to "**Select an existing server**" instead.

Select a server for the external service.



7. Click **Finish**.

Finishing the wizard opens the **External Service** editor.

Expanding the operation in the editor shows the inputs and outputs defined for the operation.

The screenshot shows the "External Service" editor. It displays a tree structure of operations. The root node is "DBA_Jam_1ComputeLoanRequestParametersDecisionService". It has two children: "callcomputeLoanRequestParametersDecisionServiceOperation" and "Input". The "Input" node has two children: "Request (Request)" and "Output". The "Output" node has two children: "callcomputeLoanRequestParametersDecisionServiceOperation_200 (Response)" and "callcomputeLoanRequestParametersDecisionServiceOperation_default (Error)". To the right of the tree, there are four buttons: up, down, add, and delete.

8. The Business Objects for the inputs and outputs are created automatically upon creation of the External Service.

The screenshot shows a list of business objects. There are six items listed: "Bankruptcy (Discovered)", "Borrower (Discovered)", "Error (Discovered)", "Loan (Discovered)", "Request (Discovered)", and "Response (Discovered)". In the top right corner, there is a circular badge with the number "6".

Explore these Business Objects by opening them up from the **Library** pane, category **Data**.

9. Switch back to the **External Service** and click the **DBA_Jam_1ComputeLoanRequestParametersDecisionService** title in the **External Service** section.

10. Switch to **Binding** tab in the column on the right.

The **Binding** tab shows the Binding type as REST because you created the external service using an OPEN API file describing a REST interface.

The tab also lists the server created or re-used during the creation of the External Service.

11. Click the link to the **DBA_Jam_1ComputeLoanRequestParametersDecisionServiceServer** Server. The **Process App Settings** editor opens, with the server selected and the details of the server as shown in the column on the right.

12. For the **Authentication** section, ensure that the following details exist:

- a. **Host name:** vm-5.example.com
- b. **Port:** 9090
- c. **Authentication:** Username and password (basic authentication)
- d. **User name:** resAdmin
- e. **Password:** resAdmin

| | |
|-------------------------|--|
| Host name: | vm-5.example.com |
| Port: | 9090 |
| Secure server: | <input type="checkbox"/> |
| SSL configuration: | |
| ▼ Authentication | |
| Authentication: | Username and password (basic authentication) |
| User name: | resAdmin |
| Password: | ***** |



13. Click **Finish Editing** on the top menu:

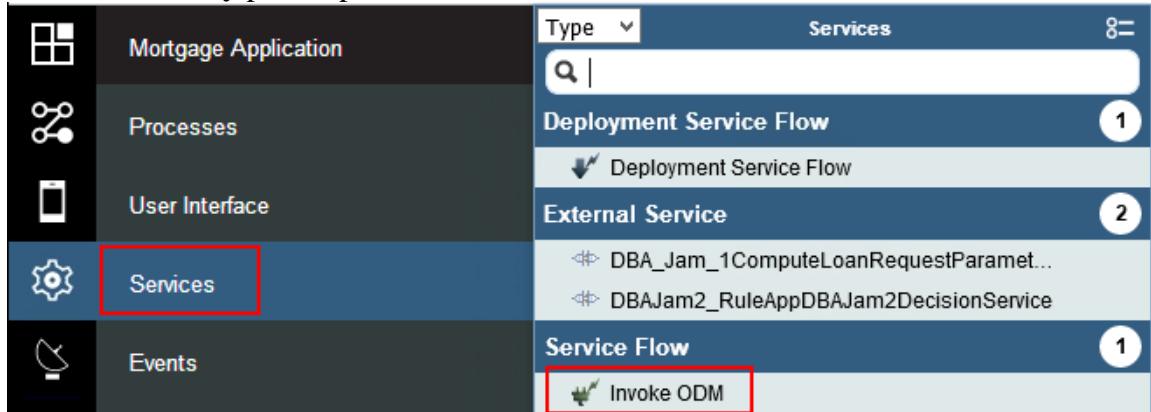
14. Repeat the steps used to create the external service for the 1st RuleApp again for the 2nd RuleApp using the 2nd JSON file downloaded.

During this creation, in the last step of the External Service creation wizard, continue to select the existing server.

You should now have two External Services in your library and many Business Objects. The Business Objects are tied to the External Services and will be deleted when the External Service is deleted.

6.3.4 In the Service Flow, use the new external services

- From the **Library** pane, open the **Invoke ODM** Service Flow.



- Click the **Variables** tab and delete all private variables that show type **unavailable**.

Variables

Type to filter

- Input**
 - customerName (String)
 - dateOfBirth (Date)
 - yearlyIncome (Decimal)
 - loanAmount (Decimal)
 - propertyAddress (String)
- Output**
 - evaluationResult (Boolean)
 - decisionServiceMessage (String)
- Private**
 - age (Decimal)

Request (unavailable)

callcomputeLoanRequestParametersDecisionServiceOperation_200 (unavailable)

callcomputeLoanRequestParametersDecisionServiceOperation_default (unavailable)

Request_1 (unavailable)

callDBAJam2DecisionServiceOperation_200 (unavailable)

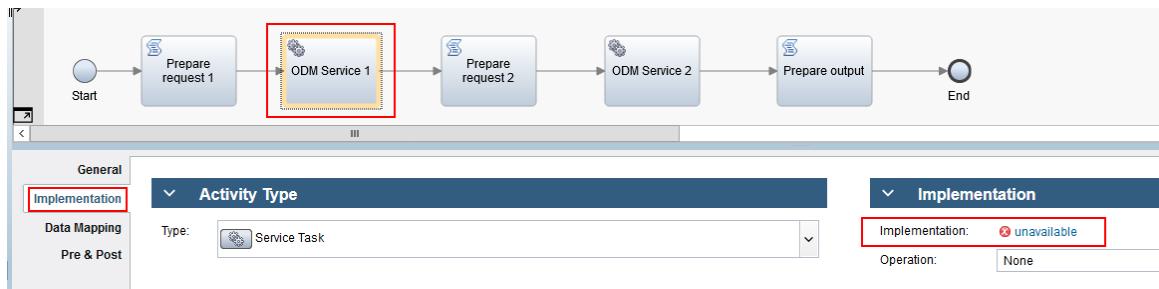
callDBAJam2DecisionServiceOperation_default (unavailable)

Exposed Process Variables

Localization Resources

3. Switch back to the **Diagram** tab.

4. Click the **ODM Service 1** node and switch to the **Implementation** tab.



5. In the **Implementation** section, click **Select...** and select the External Service previously created.

Implementation

| | | |
|-----------------|--|--------|
| Implementation: | <input type="button" value="Select..."/> | New... |
| Operation: | <input type="button" value="None"/> | |

6. Select the single operation "**callcomputeLoanRequestParametersDecisionServiceOperation**" from the External Service in the **Operation** dropdown.

The screenshot shows the "Implementation" tab with the following details:

- Implementation:** DBA_Jam_1ComputeLoanRequestParametersDecisionService
- Operation:** callcomputeLoanRequestParametersDecisionServiceOperation

7. Switch to the **Data Mapping** tab.
8. Click **auto-map** (\leftrightarrow) on the right-side of the **Input Mapping** section (see figure):

The screenshot shows the "Input Mapping" tab with the following details:

- A red box highlights the **auto-map** button (\leftrightarrow) located on the right side of the section.
- Below the button, there is a status message: "Request (Request)" with icons for a person and a file.

The **Variable Creation** dialog opens that allows you to create variables automatically.

Variable Creation

Create variables where no matching variable exists. The new variables are automatically mapped. Existing mappings are not overwritten. Existing variables with the same name but different types are omitted.



Select the variables to be created and auto-mapped. By default, the variables are created as private variables. To create them as input, output, or input and output variables, select the check box beside the variable.

| Variable Name | Variable Type | Input | Output |
|---|---------------|--------------------------|--------------------------|
| <input checked="" type="checkbox"/> Request | Request | <input type="checkbox"/> | <input type="checkbox"/> |

9. On the Variable Creation dialog, click **Finish** (at the bottom).

The screenshot shows the "Variable Creation" dialog with the following details:

- The "Request" variable is selected in the list.
- The "Request" variable is listed under "Variable Name" and "Variable Type".
- The "Input" and "Output" checkboxes are both unchecked.
- At the bottom, the "FINISH" button is highlighted with a red box.

10. Repeat the auto-mapping steps for the **Output Mapping** section.

11. Similarly, set the operation in the **DBAJam2_RuleAppDBAJam2DecisionService** node to **callDBAJam2DecisionServiceOperation**.

Important: The input for the 2nd RuleApp cannot be auto-mapped as the name **Request** conflicts with the 1st RuleApp. To work-around this conflict, create a **Request_1** private variable as follows.

- a. Go to **Variables** tab, and click "+" that is next to **Private**.

The Details page opens to create your variable:

Details

| | |
|--|----------------------------|
| Name: | variable1 |
| Documentation: | B I U |
| | |
| Is list: | <input type="checkbox"/> |
| Variable type: | String |
| System Data Select... New... | |

b. Define the new variable as follows:

- **Name:** Enter "Request_1".
- **Variable type:** Click **Select** and select **Request_1** type.
- **Click Finish Editing** (top menu)

The screenshot shows the Variables tab in a process editor. On the left, there is a tree view of variables categorized into Input, Output, Private, and Exposed Process Variables. Under Private, there is a node labeled 'Request (Request)'. On the right, there is a Details panel for this 'Request' variable. The Details panel has fields for Name (set to 'Request_1'), Documentation (empty), Is list (unchecked), Variable type (set to 'Request_1'), and a Default Value field containing the value '1'.

c. Now, return to the **Diagram > ODM Service 2 > Data Mapping**, and then

map **Request_1** manually: click the Variable Picker to select the "**Request_1**" variable as a replacement of the "**Request**" variable.

12. Click the **Prepare request 1** node.

13. Switch to the **Script** tab and update the existing Script to match the Business Object structures that you created in ODM.

Note: If you followed the instructions in this ODM sub-scenario step-by-step, then the Business Object structures should be the same as in the existing script. If you do need to update the script, an easy way to get the starting script is the following:

- a. Switch to the **Variables** tab.
- b. Select the **Request** variable.

- c. Click the **Has default** check-box.

Variable type: Request

Select... New...

Default Value

Has default:

```

1 var autoObject = new tw.object.Request();
2 autoObject._DecisionID_ = "";
3 autoObject.borrower = new tw.object.Borrower();
4 autoObject.borrower.name = "";
5 autoObject.borrower.age = 0.0;
6 autoObject.borrower.yearlyIncome = 0.0;
7 autoObject.borrower.creditScore = 0.0;
8 autoObject.borrower.salaryScore = 0.0;
9 autoObject.borrower.bankruptcyScore = 0.0;
10 autoObject.borrower.corporateScore = 0.0;
11 autoObject.borrower.latestBankruptcy = new tw.object.Bankruptcy();
12 autoObject.borrower.latestBankruptcy.date = new TWDate();
13 autoObject.borrower.latestBankruptcy.chapter = 0;
14 autoObject.borrower.latestBankruptcy.reason = "";
15 autoObject.loans = new tw.object.Bank();

```

- d. Replace **autoObject** in the script with **tw.local.Request** and enter values for the fields in the script.

14. Update the script for the **Prepare request 2** and **Prepare output** node as needed.

15. Update the script for the **Prepare output** node as needed.



16. Click **Finish Editing** to unlock the editor.

6.3.5 Take a snapshot and activate it

1. Click the arrow besides the **Snapshots** icon and select **Create a new snapshot**.



2. In the **Take Snapshot** dialog enter the name of the new snapshot, for example **v0.5**. Click **Finish**.
3. In the **Snapshot Created** dialog, take a note of the new name, and click **OK**.
4. Now, open the **Process Admin Console**.
- Open Firefox and click the **Workflow > Process Admin Console** bookmark to open the page at: <https://vm-34.example.com:9443/ProcessAdmin>
5. Open the **Installed Apps** page, and switch to the **All** view.

No items for the filter 'Active' were found.

6. Click the newly created snapshot **Mortgage Application (MA) – v0.5**.

- Finalize Loan Agreement - 0 instances
- Request Property Evaluation - 0 instances
- Review Loan Agreement Decision - 0 instances

7. On the right-hand side, click **Activate Application**.

- Activate Application
- Migrate Inflight Data
- Sync Settings
- Update Tracking Definitions

8. Click **OK** in the dialog that opens.
9. On the right-hand side, click **Make Default Version**, and then confirm the dialog by clicking **OK**.

- Deactivate Application
- Migrate Inflight Data
- Sync Settings
- Make Default Version
- Update Tracking Definitions

10. Go back to **Installed Apps**.

11. Select **Active**.

You can see the snapshot Mortgage Application (MA) – v0.5 that you created.

12. De-activate each other snapshot for the Mortgage Application that is also active, (such as the snapshot v1.0.3 shown on the following example screen capture) as follows:

- Click the snapshot to select a snapshot.
- In the right-hand side, click **Deactivate Application**.
- Click **OK**.

When you are done, in the **Installed Apps > Active** view, you should have only two active applications: **Land Property Register Simulator** (LCRC) and the **Mortgage Application** (MA) that you just activated.

You have now completed the implementation of the process, activated it, and deactivated useless ones.

You are now ready to use your process from within the Case solution.

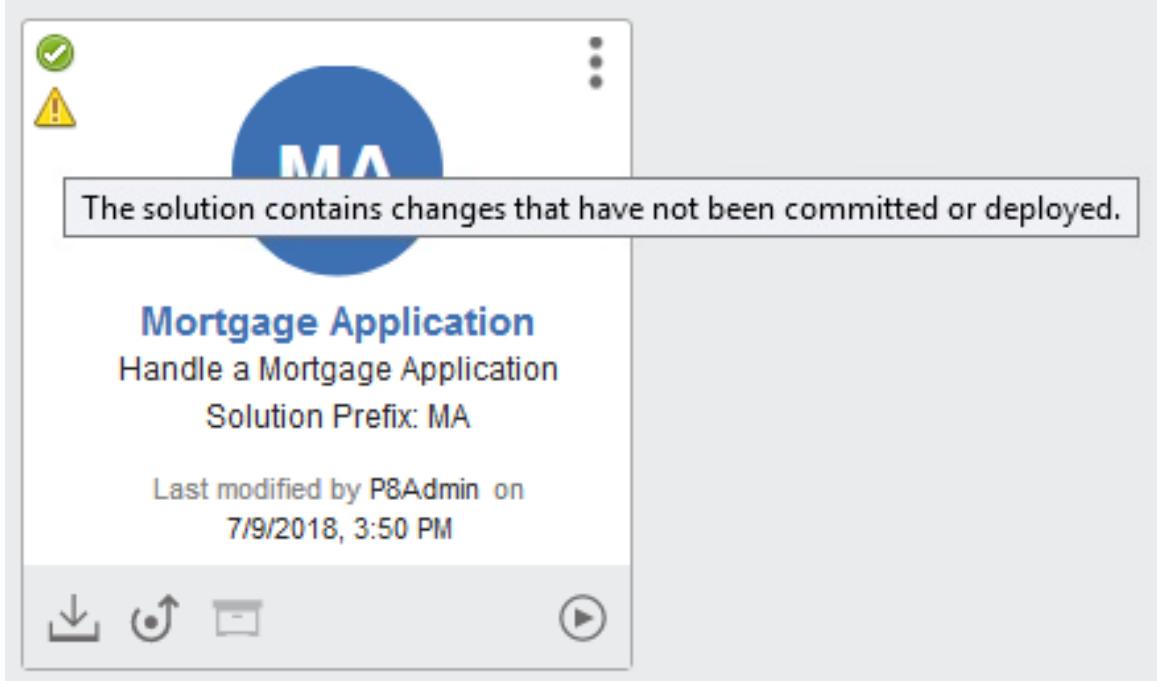
6.3.6 Use your process from within the Case solution

- Go back to **Case Builder > Mortgage Application > Tasks tab**, and click **Validate**.
- In the **Confirmation** dialog that opens, click **Save Changes**. The message "**The case type was validated successfully.**" shows at the bottom of the screen.



3. Click **Save and Close**.

You are back to the **Mortgage Application** page, where you can now see a warning sign that says: "**The solution contains changes that have not been committed or deployed.**"



The last steps now are therefore to commit the changes and the re-deploy the solution.

4. Commit the changes:

- a. Click **Commit** at the bottom of the Mortgage Application:



- b. On the Confirmation window that opens, click **Commit My Changes**.

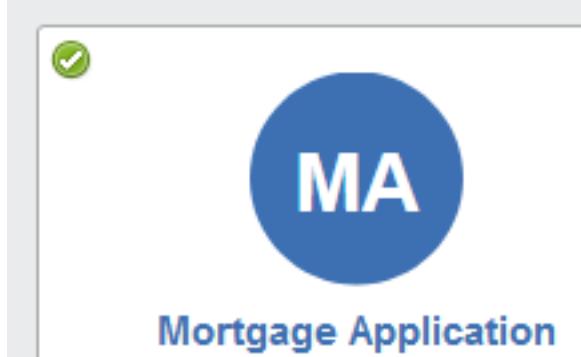
5. Re-deploy the solution.

- a. Click **Deploy** at the bottom of the Mortgage Application:



Deploying the solution might take time.

- b. Wait until the warning sign is no longer visible and the message at the bottom of the page that says "**The solution Mortgage Application was deployed successfully.**".



After you have successfully completed all the steps in VM 3 – Workflow, verify your work through the next section.

6.4 Integrate the Rule Apps with Workflow – Verification Instructions

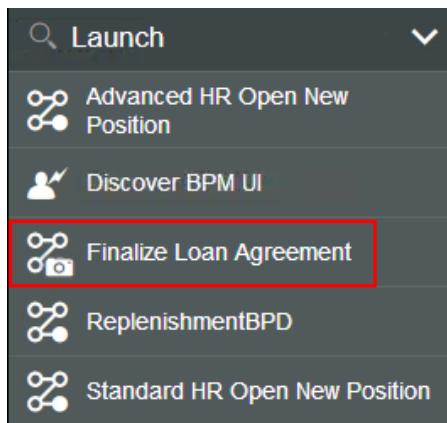
To verify successful completion of this exercise, complete the following steps:

1. If not done yet, resume the VM 1 – ECM and the VM 3 – Workflow virtual machine, as indicated in [Resume the VM 1 – ECM and VM 3 – Workflow virtual machines](#).

You now work in the VM 3 – Workflow virtual machine.

Note: As a difference from the VM 5 – ODM (where you use Chrome), make sure you use **Firefox** web-browser when you work on the VM 3 – Workflow, because bookmarks are pre-defined for this web-browser to the various Workflow modules. On the VM 3 – Workflow virtual machine, if you use a web-browser other than Firefox, you will have to enter the corresponding URLs manually.

2. In Firefox, select to the **Workflow** bookmark, and then select **Process Portal**.
Note: If you use a web-browser other than Firefox on the VM 3 – Workflow virtual machine, this bookmark is not predefined and you need to enter the URL manually.
1. If not already logged in, log-in as user **P8Admin** and password **Think4me**.
2. In the **Launch** section on the left, select **Finalize Loan Agreement**.



This opens a page with the fields defined during creation of the Launch UI.

3. Enter the following information in the fields:
 - a. **Customer Name:** Jane Doe
 - b. **Date of Birth:** 04/07/1979
 - c. **Monthly Income:** 8334
 - d. **Loan Amount:** 100000
 - e. **Purchasing Price:** 143000
 - f. **Property Address:** 304 East 10nd Street, 45202 Cincinnati, OH
4. Click **OK**.
5. In a new Firefox tab, go to the bookmark **Workflow** folder and select **IBM Workflow Center**.
6. Click the name of the **Mortgage Application (MA)** Process Application.
7. For the **latest** snapshot, for example **v1.0.6**, click **Open in Designer**.



Note: You can safely ignore the message that says that this snapshot is not yet installed to Workflow Server.

8. Switch to the **Inspector** view.



9. Click the **Search** icon.



10. Click the **Search** button.

The screenshot shows a search form for 'Person'. It includes fields for 'Name or user name', date ranges ('From Date' and 'To Date') with time pickers, and a red box around the 'Search' button.

11. Click the **latest Process Instance**.



The details of that instance are now visible in the right-hand side pane.

12. Verify that the **Status** is **Completed**, and then expand the **Data** section.

The screenshot shows the details of the 'Finalize Loan Agreement:9' instance. The status is listed as 'Completed'. The 'Data' section is expanded, showing a red box around the 'Data' tab. Other sections like 'Tasks (1)' and 'Locations' are also visible.

13. In the **Data** section, verify the following properties:

- Customer Name (String): Jane Doe
- Date of Birth (Date): 04/07/1979
- Loan Amount (Decimal): 100000
- Monthly Income (Decimal): 8334
- Purchasing Price (Decimal): 143000
- Property Address (String): 304 East 10nd Street, 45202 Cincinnati, OH

- g. decisionServiceMessage (String): Congratulations! Your loan has been approved
- h. evaluationResult (Boolean): True
- i. isComplete (Boolean): True

You can see that the **evaluationResult** is **True** and the **decisionServiceMessage** is "**Congratulations! Your loan has been approved.**"

Note: If you have changed the rules during your implementation of the ODM sub-scenario, the result might naturally differ. Verify that the results correspond to your implementation.

14. Launch the **Finalize Loan Agreement** process again and enter different values to test all the decisions. Verify the results in the **Inspector** view of the Process Designer. You can use the table below to provide sample values to the Process and compare the results:

| Customer Name | Date of Birth | Loan Amount | Monthly Income | Purchasing Price | Property Address | Result | Message |
|---------------|---------------|-------------|----------------|------------------|--|--------|--|
| Jane Doe | 04/07/1979 | 100000 | 8334 | 143000 | 304 East 10nd Street, 45202 Cincinnati, OH | true | Congratulations! Your loan has been approved |
| Paul Smith | 09/12/1973 | 150000 | 15000 | 215000 | Lundweg 1, 24941 Flensburg 215000 | true | Congratulations! Your loan has been approved |
| Betty Weber | 12/28/2001 | 50000 | 16668 | 720000 | 14 Rue Crespin du Gast, 75011 Paris | false | The borrower's age is not valid |
| John Moore | 06/06/1972 | 150000 | 4167 | 215000 | 80 Trinity St, 07860 Newton, NJ | false | Too big Debt/Income ratio: 0.3853632 |

You have now successfully completed this exercise as you have identified all the information about the two decision services that you need for calling them from the Workflow Service Flow, and made sure that the Workflow solution within the VM 3 – Workflow virtual machine is able to call them.

6.5 Integrate within the Workflow solution – Summary

In this exercise, you have:

1. Used the OpenAPI / Swagger definition files built with ODM to create External Services in the **Mortgage Application** Process Application.
2. Used the External Services in the **Finalize Loan Agreement** Process.
3. Tested the invocation of rule-based decisions created with Operational Decision Manager and Decision Composer in processes managed within a Workflow Service Flow.

Because you are using an environment from Template 5, you may now proceed with the end-to-end scenario by following the instructions in the [DBA Blue Demos 2018 - Material for Participants > 6. Platform end-to-end scenario](#).

Congratulations! You completed the ODM sub-scenario, and integrated and tested the execution of the decision services that you created within a Workflow solution.

6.6 Integrate within the Workflow solution – Troubleshooting

6.6.1 Troubleshooting connection issues in BlueDemos

If you experience issues in your BlueDemos environments with the connections between the virtual machines, please try the following steps (especially when you want to run the End-to-End scenario with Template 7, or are working on sub-scenarios that have a dependency on another VM, for example when integrating the ODM decision services

within the Workflow solution by using the "VM 3 – Workflow" virtual machine in an environment based on Template 5.

Resume a virtual machine:

1. Connect to it with Remote Desktop and click **No** in the blue bar on the right-hand side.



2. Check that this icon (bottom-right of the screen) turns into . What that is the case, the virtual machine is correctly connected to the network. You can then resume the next virtual machine and repeat the steps above.

- - - The ODM sub-scenario ends here. - - -

7 Files available as helpers

Some files are available to help you perform the ODM sub-scenario. These files are available in the [C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario](#) folder of the VM 5 – ODM virtual machine and in the [Box folder for the ODM sub-scenario](#).

7.1 C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario folder of the virtual machine

This folder on the **VM 5 – ODM** virtual machine contains the files that you need to perform the ODM sub-scenario:

- **__Backup** folder
- **Jam-1-Start** folder
- **Jam-1-Answer** folder
- **Jam-2** folder

7.1.1 __Backup

This **__Backup** folder contains the same content as in the [Box folder "5. ODM Sub-Scenario – Backup"](#).

It is available for you pre-uploaded on the virtual machine, in case you cannot connect to the Box folder.

7.1.2 Jam-1-Start

This folder is the Eclipse workspace that is used to start Rule Designer with, when starting Exercise 3.

It contains the DBA Jam 1 decision service project that corresponds to the initial state when demo participants start exercise 3.

This project also corresponds to the decision service that is available at start in the Decision Center > Business console.

It therefore contains a decision service project that has a XOM, a BOM, but is missing its signature and is also incomplete: a rule and some rows in a decision table are still missing. Participants will create the decision service signature during **Exercise: Set up DBA Jam 1 Decision Service in Rule Designer**.

7.1.3 Jam-1-Answer

This folder is the Eclipse workspace that participants can use to start Rule Designer with so to see the DBA Jam 1 decision service in its final state after demo participants have performed exercise 3.

This project therefore corresponds to the decision service that is available at start in the Decision Center > Business console, that is, it contains a decision service project that has a correct structure (including its signature and the local deployment configuration) but that is incomplete in terms of decision because a rule and some rows in a decision table are missing.

Participants will author these missing pieces during the **Exercise: Set up DBA Jam 1 Decision Service in Rule Designer**.

7.1.4 Jam-2

This folder contains all the materials to define the DBA Jam 2 decision service in Decision Composer.

- **Jam-2-Start.dproject:** The DBA Jam 2 decision service project that you can import in Decision Composer to start with the **Exercise: Define DBA Jam 2 Decision Service in Decision Composer**.
- **Jam-2-Answer.dproject:** The DBA Jam 2 decision service project that you can import in Decision Composer to see the solution of the **Exercise: Define DBA Jam 2 Decision Service in Decision Composer**.
- **DBA_Jam_2.json:** This JSON file corresponds to the DBA Jam 2 decision service that is pre-deployed in the Rule Execution Server. You can use it to integrate with Workflow. Its content matches the **Jam-2-Answer.dproject**, above.

7.2 Box folder 5. ODM sub-scenario

The [**DBA Blue Demos 2018 – Material for Participants**](#) Box folder's subfolder "5. ODM sub-scenario" contains the following sub-folder and files:

- **DBA Blue Demos Jam 2018 – ODM Lab Instructions v1.0.pdf:** This document.
- **5. ODM Sub-Scenario – Backup:** This folder contains files that you might need in case of issues. See [next](#) for more details.
- **5. ODM Sub-Scenario – Backup.zip:** This ZIP file is a compressed version of the [5. ODM Sub-Scenario – Backup](#) sub-folder, compressed for ease of download purpose.

7.3 Box folder "5. ODM Sub-Scenario – Backup"

You can find the following materials for the ODM sub-scenario in the **5. ODM sub-scenario\5. ODM Sub-Scenario - Backup** folder of the [**DBA Blue Demos 2018 – Material for Participants**](#) Box folder.

- A **Helper.txt** file
- A **Jam-1** folder
- A **Jam-2** folder

Note: This content is pre-downloaded on the virtual machine, available in the [Backup](#) folder of the [C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario](#) folder.

7.3.1 Helper.txt (Box)

This ASCII file contains some text that you can copy & paste to create the content of your decision logic. It is recommended to copy content from this ASCII file rather than to copy content from this PDF file, so to avoid un-recognized characters.

7.3.2 Jam-1 (Box)

This folder contains all the materials to define the DBA Jam 1 decision service in Operational Decision Manager.

7.3.2.1 DBA_Jam_1_RuleApp.jar (Box)

This JAR file is a copy of the RuleApp archive for the DBA Jam 1 decision service that is pre-deployed in the ODM 8.9.2 RES installed on the VM 5 - ODM virtual machine, for backup purpose.

Note: You can use this pre-deployed RuleApp as a backup solution for the integration within the Workflow solution for this DBA Jam 1 service, if, for any reasons, the RuleApp for this service already deployed in the RES gets corrupted. If you use this pre-deployed RuleApp, note that the DBA Jam 1 decision service pre-deployed in the RES has the following ruleset path: DBA_Jam_1

/1.0/computeLoanRequestParameters/1.0 (with versions), or DBA_Jam_1 /computeLoanRequestParameters (without versions).

7.3.2.2 DC-Jam-1-Full-Answer.zip (Box)

This zip file is an export from the Business console of the DBA Jam 1 decision service as it should be at the end of the **Exercise: Finalize DBA Jam 1 Decision Service in Business Console**. Participants can import this zip file from the Business console if their DBA Jam 1 decision service project gets corrupted in Decision Center and/or if they want to re-create it in a state that allows integration with the Workflow solution.

7.3.2.3 DC-Jam-1-Start.zip (Box)

This zip file is an export from the Business console of the DBA Jam 1 decision service at the start of the **Exercise: Finalize Decision Service in Business Console**. Participants can import this zip file from the Business console if their DBA Jam 1 decision service project gets corrupted in Decision Center and/or if they want to re-run the **Exercise: Finalize Decision Service in Business Console**.

7.3.2.4 Jam-1-Answer.zip (Box)

This compressed folder is a copy of the homonymous Eclipse workspace in C:\DBA_SWAT_JAM_2018\5. ODM Sub-Scenario, for backup purpose.

7.3.2.5 Jam-1-ComputeLoanRequestParametersDecisionService.json (Box)

This JSON file corresponds to the DBA Jam 1 decision service that is pre-deployed in the Rule Execution Server as the RuleApp DBA_Jam_1/1.0/computeLoanRequestParameters/1.0. You can use it to integrate the DBA Jam 1 decision service with the Workflow solution.

7.3.2.6 Jam-1-ExpectedOutputAnswer-in-RD.txt (Box)

This text file contains some input data and the expected answer (output data) that you can use to test the solution for the DBA Jam 1 decision service, as fully implemented in the [Jam-1-Full-Answer](#) workspace of Rule Designer.

7.3.2.7 Jam-1-Full-Answer.zip (Box)

This compressed folder is an Eclipse workspace that can be used to start Rule Designer with so to see the DBA Jam 1 decision service successfully completed after **Exercise: Set up Decision Service in Rule Designer** and then **Exercise: Finalize Decision Service in Business Console**.

It therefore contains the DBA Jam 1 decision service project fully implemented, that is: with a decision signature, the local deployment configuration, and all rules and decision table rows defined.

You can use this workspace as needed to re-create (re-deploy) a RuleApp to Rule Execution Server that matches the decision service expected by the Workflow solution. Note that the version 1.0 of the RuleApp for this project is already deployed in the RES, and its ruleset is: DBA_Jam_1/1.0/computeLoanRequestParameters/1.0 This RuleApp can be used if participants do not do **Exercise: Set up Decision Service in Rule Designer** and then **Exercise: Finalize Decision Service in Business Console**, and still want to call the DBA Jam 1 decision service from Workflow.

7.3.2.8 Jam-1-RequestAnswer-in-RES.txt (Box)

This text file contains some input data and the expected answer (output data) that you can use to test the DBA Jam 1 decision service when deployed in Rule Execution Server (RES).

7.3.2.9 Jam-1-Start.zip (Box)

This compressed folder is a copy of the homonymous Eclipse workspace in the C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario folder of the virtual machine, for backup purpose.

7.3.2.10 testsuite.xlsx (Box)

This Microsoft Office Excel spreadsheet contains test data that you, as Bea, can use to create a test suite and test the DBA Jam 1 decision service in Business console.

7.3.3 Jam-2 (Box)

This folder contains all the materials to define the DBA Jam 2 decision service in Decision Composer.

7.3.3.1 Jam-2-Answer.dproject (Box)

This *.dproject file is a copy of the homonymous file in the C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario/Jam-2 folder of the virtual machine, for backup purpose.

7.3.3.2 *Jam-2-Start.dproject (Box)*

This *.dproject file is a copy of the homonymous file in the C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario/Jam-2 folder of the virtual machine, for backup purpose.

7.3.3.3 *DBA_Jam_2.json (Box)*

This JSON file is a copy of the homonymous file in the C:/DBA_SWAT_JAM_2018/5. ODM Sub-Scenario/Jam-2 folder of the virtual machine, for backup purpose.

7.3.3.4 *DBA_Jam_2_RuleApp.jar (Box)*

This JAR file is a copy of the RuleApp archive for the DBA Jam 2 decision service that is pre-deployed in the ODM 8.9.2 RES installed on the VM 5 - ODM virtual machine, for backup purpose.

Note: You can use this pre-deployed RuleApp as a backup solution for the integration within the Workflow solution for this DBA Jam 2 service, if calling the RES that backs Decision Composer fails for any reasons. If you use this pre-deployed RuleApp, note that the DBA Jam 2 decision service pre-deployed in the RES has the following ruleset path:
DBAJam2_RuleApp/1.0/DBAJam2/1.0 (with versions), or
DBAJam2_RuleApp/DBAJam2 (without versions).