

What's new in IBM Operational Decision Manager 8.9 Standard Edition

Release themes

- [User empowerment in the Business Console](#)
- [Improved development and operations \(DevOps\) features](#)
 - Easier integration with API management
 - Simpler ODM integration in DevOps pipelines
 - Better experience for ODM developers
- [Product simplification](#)
 - Removal of some deprecated features
 - Updated list of supported platforms
- [Integration with IBM Cloud Product Insights](#)



User empowerment in the Business Console, for decision services

User empowerment in the Business Console, for decision services

- [Dynamic domain update, with increased control on changes](#)
- [Management of resource files that are included in projects](#)
- [Complete branch management, including compare & merge between branches, releases, and activities](#)
- [Column filtering in decision table editor, to show only rows of interest](#)
- [Support for custom scenario provider for decision service testing](#)
- [Variable sets included in snapshot comparison view](#)
- [Direct links to artifacts](#)



Dynamic domain update, with
increased control on changes

Dynamic domains and resources in the Business Console

- This new capability allows users of the Business Console to:
 - Identify the changes in dynamic domains.
 - Changes in dynamic domains are indicated with icons, colors, and tips.
 - Update the content of dynamic domains.
 - They can identify the changes in dynamic domains in the **Model** tab, and accept or ignore them.
- If a dynamic domain is defined with a Microsoft Excel spreadsheet in the **Resources** folder, they can use the **Decision Artifacts** tab to import this spreadsheet locally, and import it back into the project after their changes, and then use the **Model** tab to manage the dynamic domain changes.



Typical use-case

- The IT specialist created a Microsoft Excel spreadsheet with a series of values, and used it as the basis for a dynamic domain that the business user uses in the Business Console.
- The IT specialist later modified this spreadsheet.
- In the Business Console, the business user can import then new `*.xlsx` file, and see all changes that the IT specialist made.
 - the business user can see the details of the changes.
 - the business user can select which changes to approve or ignore.
 - By rebuilding the project, the business user can then see the decision artifacts to refactor.



Details (1 of 7)

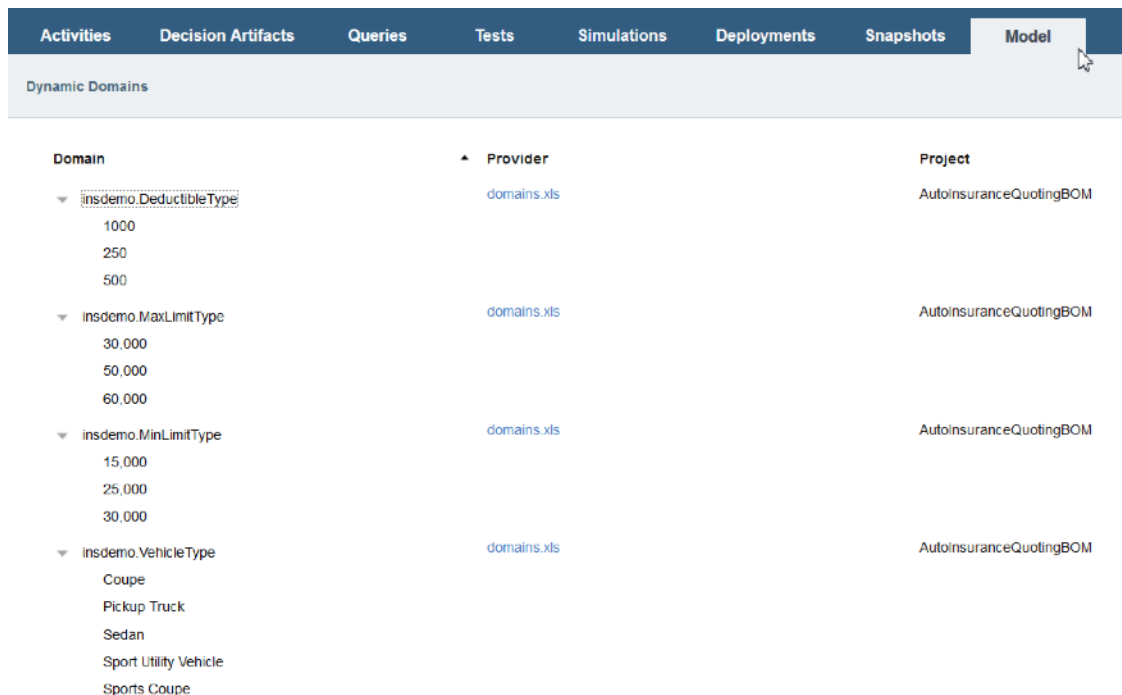
- In the Business Console, open the project that uses dynamic domains, and select an activity or a branch.
 - In *releases*, dynamic models can only be read, not modified.
 - In *activities* and in *branches*, dynamic models can be acted upon.

Domain	Provider	Project
insdemo.DeductibleType	domains.xls	AutoInsuranceQuotingBOM
1000		
250		
500		
insdemo.MaxLimitType	domains.xls	AutoInsuranceQuotingBOM
30,000		
50,000		
60,000		
insdemo.MinLimitType	domains.xls	AutoInsuranceQuotingBOM
15,000		
25,000		
30,000		
insdemo.VehicleType	domains.xls	AutoInsuranceQuotingBOM
Coupe		
Pickup Truck		
Sedan		
Sport Utility Vehicle		
Sports Coupe		



Details (2 of 7)

- In the **Model** page, you can see all dynamic domains in use in the project.
 - In this example, the project contains four dynamic domains, all defined by the **domains.xls** Microsoft Excel spreadsheet that can be found in the **Resources** folder.
- Click the **domains.xls** spreadsheet to modify the dynamic domain values.



The screenshot shows the 'Model' page in the IBM Operational Decision Manager interface. The page title is 'Dynamic Domains'. The table below lists the dynamic domains, their providers, and the project they belong to.

Domain	Provider	Project
insdemo.DeductibleType <ul style="list-style-type: none">1000250500	domains.xls	AutoInsuranceQuotingBOM
insdemo.MaxLimitType <ul style="list-style-type: none">30,00050,00060,000	domains.xls	AutoInsuranceQuotingBOM
insdemo.MinLimitType <ul style="list-style-type: none">15,00025,00030,000	domains.xls	AutoInsuranceQuotingBOM
insdemo.VehicleType <ul style="list-style-type: none">CoupePickup TruckSedanSport Utility VehicleSports Coupe	domains.xls	AutoInsuranceQuotingBOM



Details (3 of 7)

- In the **Decision Artifacts** page, you can see the **Resources** folder of the project.
 - Use the **new Resources** check box under **All Types** to toggle the visibility of this folder.
- You can find the **domains.xls** Microsoft Excel spreadsheet in this **Resources** folder.

The screenshot displays the 'Decision Artifacts' page in IBM Operational Decision Manager. The top navigation bar includes 'Decision Artifacts', 'Queries', 'Tests', and 'Sir'. Below the navigation bar are filters for 'All Projects' and 'All Types'. A list of decision artifacts is displayed, including 'All Projects', 'AutoQuote', 'AutoInsuranceQuotingBOM', 'DataValidation', 'Eligibility', 'Pricing', 'All Types', 'Rules and Decision Tables', 'Ruleflows', 'Operations', 'Variable Sets', and 'Resources'. The 'Resources' checkbox is highlighted with a red box. A red arrow points from this checkbox to the 'Resources' folder in the project tree below. The project tree shows a hierarchy: 'AutoQuote', 'AutoInsuranceQuotingBOM' (with a '1' badge), 'Operations', 'Resources' (with a '1' badge), '(x) Parameters', 'DataValidation' (with a '1' badge), 'Eligibility' (with a '1' badge), and 'Pricing' (with a '3' badge'). The 'Resources' folder is expanded, showing a list of items with a 'Name' checkbox and a 'domains.xls' file, which is also highlighted with a red box. A red arrow points from the 'domains.xls' file to the 'Resources' folder in the project tree.



Details (4 of 7)

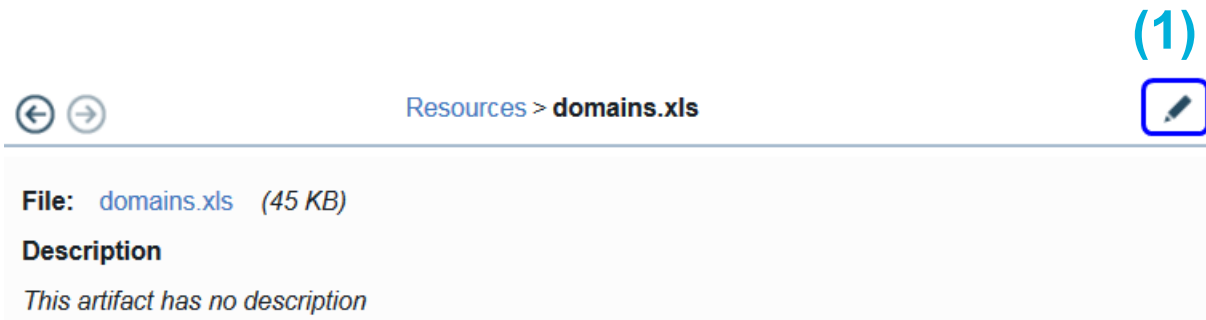
- Click the **domains.xls** spreadsheet to see its description, and then download its content.

The screenshot displays the IBM Operational Decision Manager interface. On the left, a navigation pane shows a tree structure with categories: AutoQuote, DataValidation, AutoInsuranceQuotingBOM, Eligibility, and Pricing. Under 'AutoInsuranceQuotingBOM', the 'Resources' sub-item is selected and highlighted. The main content area shows the details for the file 'domains.xls' (45 KB). A blue arrow points from the file name to a Firefox file opening dialog box. The dialog box contains the following text: 'Opening domains.xls', 'You have chosen to open:', 'domains.xls which is: Microsoft Excel 97-2003 Worksheet (45,0 KB) from: http://jrules1.francelab.fr.ibm.com:12090', and 'What should Firefox do with this file?'. The 'Save File' option is selected, and the 'Do this automatically for files like this from now on.' checkbox is unchecked. 'OK' and 'Cancel' buttons are at the bottom right.



Details (5 of 7)

- In an activity, you can update the spreadsheet content, and upload it back.
 - (1) Click **Edit** to edit the resource.
 - (2) Click **Refresh** to upload the modified spreadsheet.
 - (3) Click **Save** to update the resource.



Resources > domains.xls

File: domains.xls (45 KB)

Description

This artifact has no description



Resources > domains.xls

File: domains.xls (45 KB) Refresh...

Description



Details (6 of 7)

- Back in to the **Model** page, you can see the changes in the dynamic domains.
 - added values: in green, with a plus (“+”) sign (1)
 - modified values: in yellow, with an unequal (“≠”) sign (2)
 - Here a change in the verbalization (B2X)
 - deleted values: in red, with a minus (“-”) sign (3)
- You get a tip about the change by moving the mouse over it. (4)

insdemo.VehicleType	insdemo.VehicleType
-	+ City Car
Coupe	≠ Coupe
Pickup Truck	Pickup Truck
Sedan	- Sedan
Sport Utility Vehicle	Sport Utility Vehicle
Sports Coupe	Sports Coupe

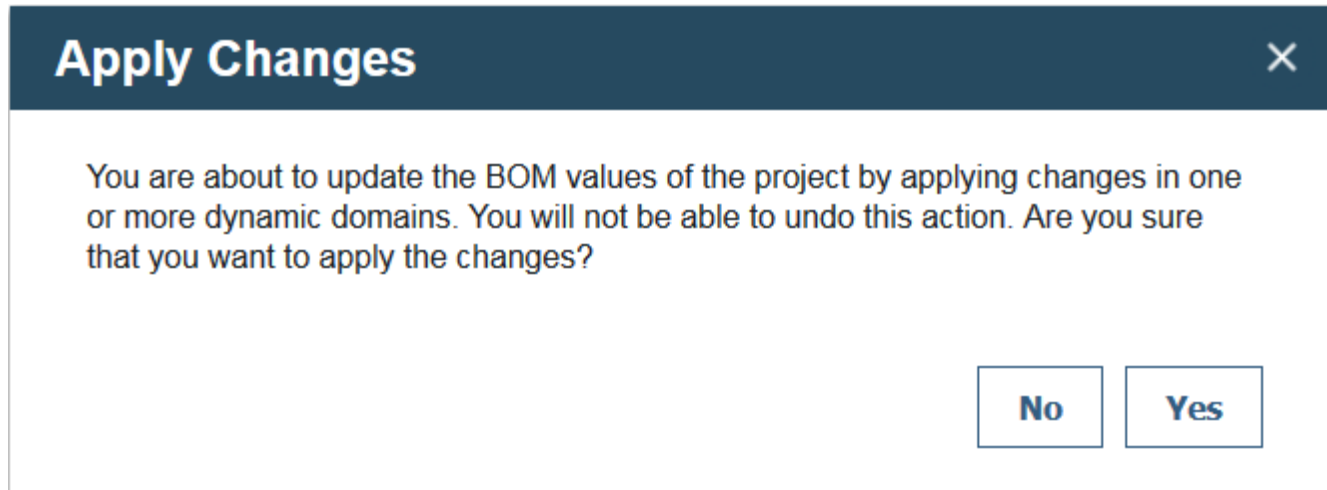
insdemo.VehicleType	insdemo.VehicleType
-	+ City Car
Coupe	≠ Coupe
Pickup Truck	Pickup Truck
Sedan	- Sedan
Sport Utility Vehicle	Sport Utility Vehicle
Sports Coupe	Sports Coupe

insdemo.VehicleType	insdemo.VehicleType
-	+ City Car
Coupe	≠ Coupe
Pickup Truck	Pickup Truck
Sedan	- Sedan
Sport Utility Vehicle	Sport Utility Vehicle
Sports Coupe	Sports Coupe



Details (7 of 7)

- You can select the dynamic domains where to apply changes, and the ones to ignore.
- If you click **Apply changes** to accept the changes, you are warned that the changes cannot be undone.
- When the dynamic domain is updated, you can refactor all rules that use the deleted or modified dynamic domain values.



Documentation

- The documentation for these features is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business Console > Decision artifacts > Updating dynamic domains](#).



Management of resource files
that are included in projects

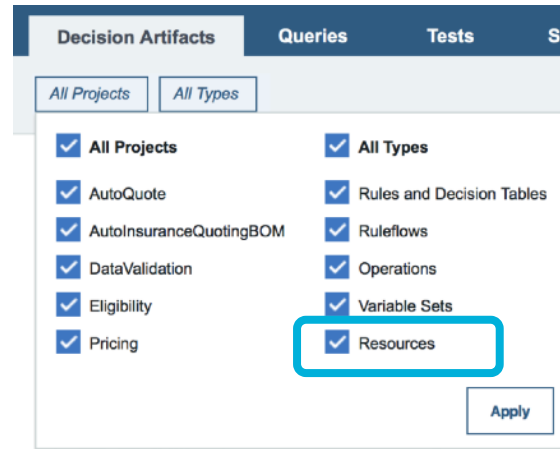
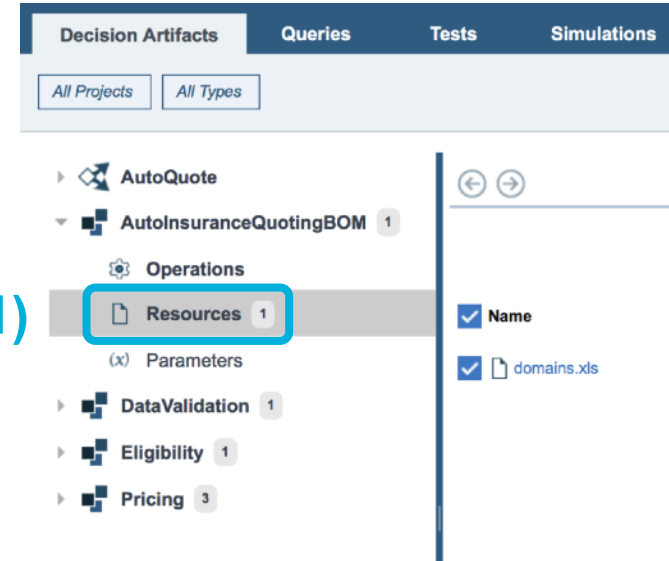
Resources in the Business Console

- This new capability allows users of the Business Console to access the content of the **Resources** folder of their project in the Business Console, Decision Artifacts tab.
 - The content of the **Resources** folder is now visible in the Business Console.
- If a dynamic domain is defined, a Microsoft Excel spreadsheet in the **Resources** folder, they can use the **Decision Artifacts** tab to import this spreadsheet locally, and import it back into the project after your changes.



Resources check box under Decision Artifacts > All Types

- (1) In the **Decision Artifacts** page, you can now see the **Resources** folder of the project.
- (2) Toggle the visibility of the **Resources** folder by selecting, or clearing, the **Resources** check box under **Decision Artifacts > All Types**.



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business Console > Decision artifacts](#).



Complete branch management, with
compare & merge between branches,
releases, and activities

Branch comparison and merge in the Business Console

- This new capability allows users of the Business Console to compare and merge branches of a decision service, and have a view of an entire decision service.
- They can see the differences between two branches, releases or activities.
- They can decide what to do with each difference.
- They can perform all specified actions in one click.



Typical use-cases

1. The business user and the project manager are both working on two releases of the same decision service in the Business Console. Each of them works within their own activity to make various changes in the project artifacts.
2. The business user and the project manager are both working on two (non-governed) branches of the same decision service in the Business Console.
3. The business user and the project manager are working on different decision services that share a common BOM project and want to exchange BOM entries.



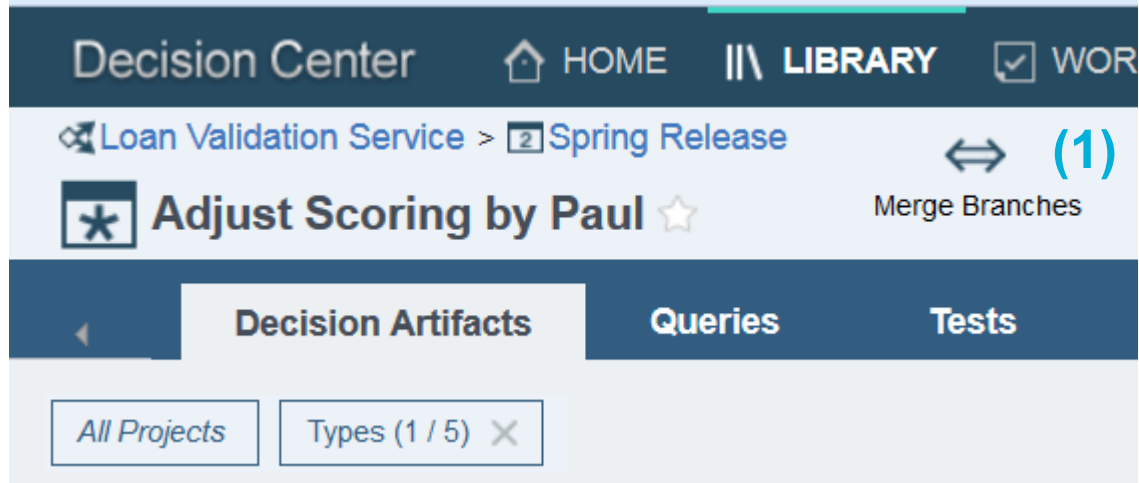
“Merge Branches” for two activities (use-case #1)

- If you have the right to access both activities, you can use the “**Merge Branches**” feature to compare the content of the two branches of this decision service within the Business Console.
 - You can see a summary of the differences.
 - You can see the details of each difference by clicking it.
- You can also use the “**Merge Branches**” feature to merge the contents of these two branches.
 - You can report a change from one branch to the other.
 - You can ignore a change.



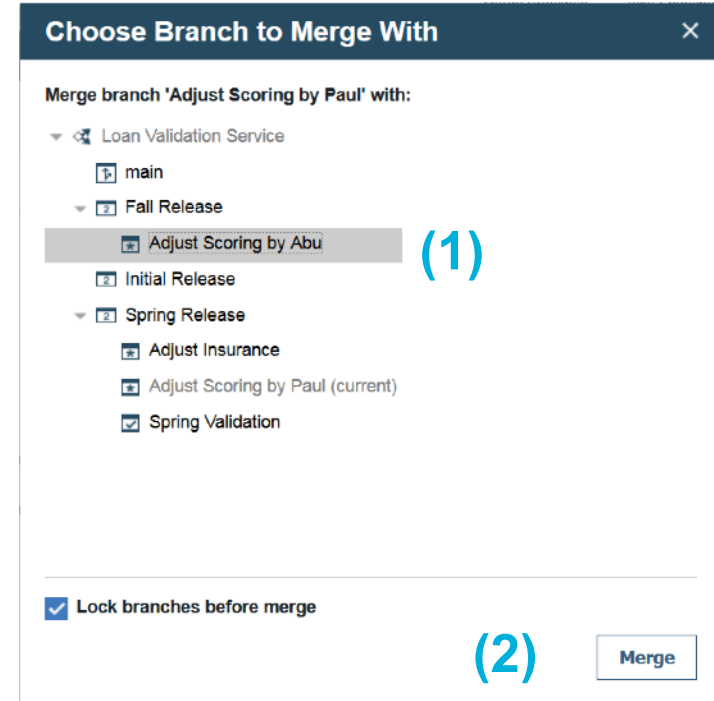
“Merge Branches” button

- You have an activity in the Business Console, and do some changes in your project.
- In this activity, you have a **Merge Branches** button (1) to merge it with the activity or branch of your choice.



“Choose Branch to Merge With”

- Click **Merge Branches** to select the other activity that you want to compare with.
- In the **Choose Branch to Merge With** window that opens, select the branch that you want to merge with **(1)**.
 - You cannot select the current activity (disabled).
- When the selection is done, click **Merge (2)**.



Summary of changes

- In the **Merge** window (1), you can see all the projects in the decision service, with all differences:
 - Added elements (2)
 - Deleted elements (3)
 - Modified elements (4) [possibly marked with (conflict) when modified on both sides]
 - Even if an element is changed identically in both sides, it is considered as modified (conflict) because at least its version differs.

(1)

Decision Center HOME LIBRARY WORK ADMINISTRATION Paul

Loan Validation Service > Spring Release > Adjust Scoring by Paul

↔ Merge Adjust Scoring by Paul with Adjust Scoring by Abu Apply Merge

Merge direction: Both directions Only to Adjust Scori Only to Adjust Scori

Expand all Collapse all Reset all actions Ignore all changes

Review and select the action to perform to merge each difference between the two branches.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu	Action
Loan Validation Service	31 elements added, 2 elements modified	-	add 31 elements to Adjust Scoring by Abu, update 2 elements in Adj...
Loan Validation Base	-	-	-
Loan Validation Check	-	-	-
Loan Validation Determination	1 element added	1 element added, 1 element deleted, 1 element modified	⚠ add 1 element to Adjust Scoring by Paul, update 1 element in Ad...
Loan Validation Scoring	-	-	-

(2) (3) (4)



Default actions

- In the **Merge** window, you can see the possible **default** actions to handle these differences (1).
 - These default actions depend on the “direction” of the merge (see slide “**Direction**”).

The screenshot shows the 'Merge' window in IBM Operational Decision Manager. The breadcrumb trail is: Decision Center > HOME > LIBRARY > WORK > ADMINISTRATION > Loan Validation Service > Spring Release > Adjust Scoring by Paul. The merge is between 'Adjust Scoring by Paul' and 'Adjust Scoring by Abu'. The merge direction is set to 'Both directions'. A table lists the differences between the two branches, with a red circle and the number '1' highlighting the 'Loan Validation Determination' row.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu	Action
Loan Validation Service	31 elements added, 2 elements modified	-	add 31 elements to Adjust Scoring by Abu, update 2 elements in Adj...
Loan Validation Base	-	-	-
Loan Validation Check	-	-	-
Loan Validation Determination	1 element added	1 element added, 1 element deleted, 1 element modified	add 1 element to Adjust Scoring by Paul, update 1 element in Ad...
Loan Validation Scoring	-	-	-



Navigation

- You can now:
 - Expand or collapse all the artifact folders to get more details (1)
 - Expand or collapse each folder (2).
 - Identify the status for each artifact (unchanged, modified, deleted, or added) (3).
 - Move the mouse over a specific change to learn more about it (4).
 - See the actions for each change (5).

The screenshot displays the 'Decision Center' interface for a merge operation. The breadcrumb trail shows 'Loan Validation Service > Spring Release > Adjust Scoring by Paul'. The main heading is 'Merge Adjust Scoring by Paul with Adjust Scoring by Abu'. The interface includes controls for 'Merge direction' (Both directions, Only to Adjust Scoring by Paul, Only to Adjust Scoring by Abu) and buttons for 'Expand all', 'Collapse all', 'Reset all actions', and 'Ignore all changes'. A table lists the artifacts and their status in each branch, with corresponding actions.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu	Action
Loan Validation Service	31 elements added, 2 elements modified	-	add 31 elements to Adjust Scoring by Abu, update 2 elements in Adjust Scoring by Paul
Loan Validation Base	-	-	-
Loan Validation Check	-	-	-
Loan Validation Determination	-	-	-
insurance	-	-	-
defaultinsurance	-	deleted	do not modify
insurance	-	modified	update in Adjust Scoring by Paul
new_rule_by_Abu	-	added	add to Adjust Scoring by Paul
new_rule_by_Paul	added	-	add to Adjust Scoring by Abu
Loan Validation Scoring	-	-	-

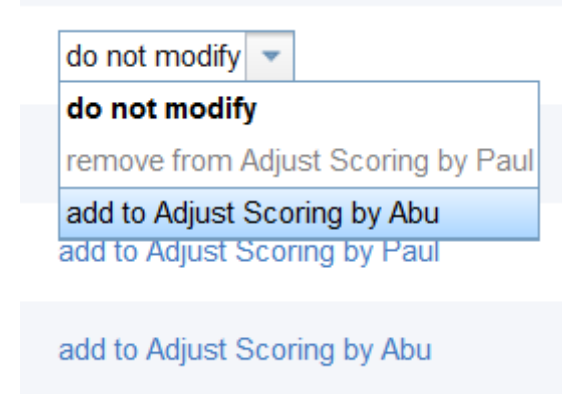
Annotations in the image: (1) points to the merge heading; (2) points to the 'insurance' folder; (3) points to the 'added' status of 'new_rule_by_Paul'; (4) points to the tooltip 'This element was added to Adjust Scoring by Abu'; (5) points to the 'add to Adjust Scoring by Abu' action.



Action selection

- For each action, you need to set the action to perform **(1)**.
- You can override the default action and opt to:
 - Do nothing (**do not modify**)
 - Update in your activity
 - Update in activity


(1)



Compare view

- To identify the proper action, check the differences in the artifact between the two activities.
- To do so, click the double-arrow icon (1) to see the artifact definition in the two activities (2), with differences highlighted (3).

(1)

insurance 

(2)

insurance ✕

Hide summary Content (3) Properties (0)

row 1, column 2: content was changed from < 100 000 to < 80 000

row 1, column 4: content was changed from to null

row 5, column 2: content was changed from < 100 000 to < 50 000

version 4.1 (current) in Adjust Scoring by Paul
Created by Paul on Nov 25, 2016

Preconditions

	Grade	Amount of loan		Insurance requir	Insurance ra
		Min	Max		
1	A	< 100,000		false	0.001
2	A	100,000	300,000	true	0.001
3	A	300,000	600,000	true	0.003
4	A	≥ 600,000		true	0.005
5	B	< 100,000		false	0.003
6	B	100,000	300,000	true	0.003
7	B	300,000	600,000	true	0.005

version 9.0 (current) in Adjust Scoring by Abu
Created by rsAdmin on Dec 6, 2016

Preconditions

	Grade	Amount of loan		Insurance requ	Insurance ra
		Min	Max		
1	A	< 50,000		false	0.001
2	A	100,000	300,000	true	0.001
3	A	300,000	600,000	true	0.003
4	A	≥ 600,000		true	0.005
5	B	< 50,000		false	0.003
6	B	100,000	300,000	true	0.003
7	B	300,000	600,000	true	0.005

(3)



Directions (1 of 2)

- You can also select the “direction” for the merge:
 - By default, the merge is working in both directions (1)
 - Merging in both directions means that, unless the artifact is locked or in conflict, the action is to “synchronize” the changes between the two activities.

The screenshot shows the 'Merge' interface in IBM Operational Decision Manager. The breadcrumb trail is: Decision Center > HOME > LIBRARY > WORK > ADMINISTRATION > Loan Validation Service > Spring Release > Adjust Scoring by Paul. The main heading is 'Merge Adjust Scoring by Paul with Adjust Scoring by Abu'. Below this, there are three buttons for 'Merge direction': 'Both directions' (selected), 'Only to Adjust Scoring by Paul', and 'Only to Adjust Scoring by Abu'. To the right are buttons for 'Expand all', 'Collapse all', 'Reset all actions', and 'Ignore all changes'. A note says 'Review and select the action to perform to merge each difference between the two branches.' Below this is a table with columns: Name, Adjust Scoring by Paul, Adjust Scoring by Abu, and Action. A red circle with the number '1' is drawn around the 'Name' column header. The table lists several artifacts, including 'Loan Validation Service', 'Loan Validation Base', 'Loan Validation Check', 'Loan Validation Determination', and an 'insurance' folder containing 'defaultinsurance', 'insurance', 'new_rule_by_Abu', and 'new_rule_by_Paul'. The 'Action' column shows actions like 'add 31 elements to Adjust Scoring by Abu, update 2 elements in Adj...', 'do not modify', 'update in Adjust Scoring by Paul', and 'add to Adjust Scoring by Paul'. A tooltip is visible over the 'new_rule_by_Paul' row, stating 'This element was added to Adjust Scoring by Abu'.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu	Action
Loan Validation Service	31 elements added, 2 elements modified	-	add 31 elements to Adjust Scoring by Abu, update 2 elements in Adj...
Loan Validation Base	-	-	-
Loan Validation Check	-	-	-
Loan Validation Determination	-	-	-
insurance	-	-	-
defaultinsurance	-	deleted	do not modify
insurance	-	modified	update in Adjust Scoring by Paul
new_rule_by_Abu	-	added	add to Adjust Scoring by Paul
new_rule_by_Paul	added	-	add to Adjust Scoring by Abu
Loan Validation Scoring	-	-	-



Directions (2 of 2)

- You can opt to merge only changes into the first activity (2), or into the second activity (3).
- Modifying the “merge direction” changes the default actions.

The screenshot shows the 'Decision Center' interface for merging two versions of a service. The breadcrumb trail is 'Loan Validation Service > Spring Release > Adjust Scoring by Paul'. The main heading is 'Merge Adjust Scoring by Paul with Adjust Scoring by Abu'. Below this, there are buttons for 'Merge direction': 'Both directions' (selected), 'Only to Adjust Scoring by Paul' (labeled with a blue (2)), and 'Only to Adjust Scoring by Abu' (labeled with a blue (3)). There are also buttons for 'Expand all', 'Collapse all', 'Reset all actions', and 'Ignore all changes', along with an 'Apply Merge' button.

Below the merge direction controls, a table displays the differences between the two versions. The table has columns for 'Name', 'Adjust Scoring by Paul', 'Adjust Scoring by Abu', and 'Action'.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu	Action
Loan Validation Service	31 elements added, 2 elements modified	-	add 31 elements to Adjust Scoring by Abu, update 2 elements in Adj...
Loan Validation Base	-	-	-
Loan Validation Check	-	-	-
Loan Validation Determination	-	-	-
insurance	-	-	-
defaultinsurance	-	deleted	do not modify
insurance	-	modified	update in Adjust Scoring by Paul
new_rule_by_Abu	-	added	add to Adjust Scoring by Paul
new_rule_by_Paul	added	-	add to Adjust Scoring by Abu
Loan Validation Scoring	-	-	-

A tooltip is visible over the 'new_rule_by_Paul' row, stating 'This element was added to Adjust Scoring by Abu.'



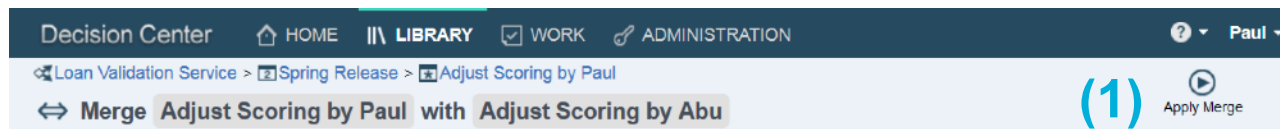
Reset or ignore changes

- After you changed an action to a non-default action, you can also reset all actions back to their default values **(1)**.
- You can also select to ignore all changes **(2)**.
 - Useful to consider only one change and ignore all others.

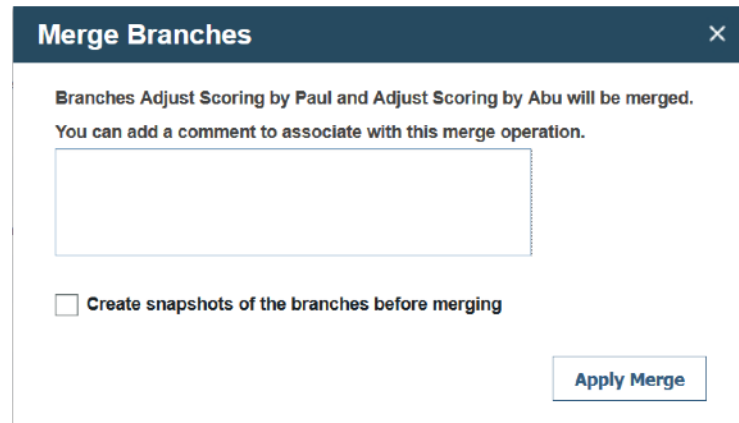


Apply Merge

- When you have set all the needed actions, click **Apply Merge** to request these actions (1).



- In the **Merge Branches** dialog, you can set a comment, and, if needed, take a snapshot of the branches before the merge.
- When you click **Apply Merge**, the actions are done.



Results

- The yellow banner gives the summary of the merge operation (1).
- The main area gives the details of the changes that occurred.

The screenshot displays the IBM Operational Decision Manager interface. At the top, the navigation bar includes 'Decision Center', 'HOME', 'LIBRARY', 'WORK', and 'ADMINISTRATION'. The breadcrumb trail shows 'Loan Validation Service > Spring Release > Adjust Scoring by Paul'. A yellow banner at the top of the main content area reads 'Merge completed successfully (1)' with a blue circled '1'. To the right of the banner are 'Expand all' and 'Collapse all' buttons. Below the banner, the text states 'The following changes have been performed during the merge operation:'. A table follows, comparing changes between 'Adjust Scoring by Paul' and 'Adjust Scoring by Abu'. The table has three columns: 'Name', 'Adjust Scoring by Paul', and 'Adjust Scoring by Abu'. The 'Loan Validation Check' row is highlighted, and the value '-' in the 'Adjust Scoring by Abu' column is circled in blue with a '2'. The 'insurance' folder is expanded, showing a 'new_rule_by_Paul' entry with the value 'added' in the 'Adjust Scoring by Abu' column.

Name	Adjust Scoring by Paul	Adjust Scoring by Abu
▶ Loan Validation Service	-	31 elements added
■ Loan Validation Base	-	-
■ Loan Validation Check	-	- (2)
▼ ■ Loan Validation Determination		
▼ insurance		
■ new_rule_by_Paul	-	added
■ Loan Validation Scoring	-	-



At the end...

- If you refresh the web browser, the yellow banner indicates that there is nothing to merge any more (1).

The screenshot shows the IBM Operational Decision Manager interface. At the top, there is a navigation bar with 'Decision Center', 'HOME', 'LIBRARY', 'WORK', and 'ADMINISTRATION'. Below this, a breadcrumb trail reads 'Loan Validation Service > Spring Release > Adjust Scoring by Paul'. A merge operation is in progress: 'Merge Adjust Scoring by Paul with Adjust Scoring by Abu'. A yellow banner displays the message 'Nothing to merge (no changes). (1)'. Below the banner, there are controls for 'Merge direction' with options 'Both directions', 'Only to Adjust Scori...', and 'Only to Adjust Scori...'. There are also buttons for 'Expand all', 'Collapse all', 'Reset all actions', and 'Ignore all changes'. A message below the controls states 'Review and select the action to perform to merge each difference between the two branches.' A table header is visible with columns: 'Name', 'Adjust Scoring by Paul', 'Adjust Scoring by Abu', and 'Action'. At the bottom of the screenshot, a message reads 'There are no differences between the two branches.'



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business Console > Managing changes > Merging branches](#).



Column filtering in decision table editor, to show only rows of interest

Column filtering in decision table editor

- This new capability helps users of the Business Console to manage and understand Decision Tables with many rows and columns.
- Within the Decision Artifacts tab, they can set filters on columns of a decision table, to reduce the number of rows displayed.
- As a result, they can see a reduced number of decision table rows:
 - This reduced view shortens search time, and makes it faster and easier to edit specific partitions of their decision tables.
 - It also help in better understanding the table structure and rules.



Use-case

- A policy manager is working on a loan validation decision table that has many rows.
- However, the policy manager needs to only work on rows that are for loans with a duration strictly less than 10 years.
- With the new feature, the policy manager can now filter the decision table by selecting the cell value for this column.
 - The policy manager can filter the content of the decision table by value from an enumeration, by text, or use comparison operators and intervals for columns with numbers and dates.
- By doing so, the number of rows shown in the decision table is reduced to only these rows for which the value of the cell in this column matches the selected filter.



Details (1 of 3)

- You can filter all the columns of a decision table by using the **Sort / Filter** menu.
- Filtering is limited when a column has sub-columns (multiple placeholders).
 - In this case, the first sub-column is used for filtering.
 - For example: the **Action** column in **AutoQuote > main > Profile > Driver Profile**

Loan to Value		Yearly inte
Min	Max	
0.7	0.8	0.05
0.8	0.9	0.05

Sort/Filter column

Sort

↑ Ascending ↓ Descending

Filter

[.] 0.72 0.85

Include Otherwise

Clear filter Apply



Details (2 of 3)

- Filtering is semantic, which means that it takes into account all cells for which the expressions include the value of the filtering.
- As examples, if you filter with “[0.72 .. 0.85]”, results include rows with value ranges:
 - 0.7 .. 0.8
 - 0.8 .. 0.9

Loan to Value		▼
Min	Max	
0.7	0.8	
0.8	0.9	
0.7	0.8	
0.8	0.9	
0.7	0.8	
0.8	0.9	
0.7	0.8	
0.8	0.9	
0.7	0.8	
0.8	0.9	
0.7	0.8	
0.8	0.9	



Details (3 of 3)

- You can identify the filtered columns with the “funnel” indicator, which replaces the down-arrow of the **Sort / Filter** menu.



Results

	Loan duration (y)		Loan to Value	
	Min	Max	Min	Max
2	< 5		0.7	0.8
3	< 5		0.8	0.9
4	< 5		≥ 0.9	
5	5	8	0	0.7
6	5	8	0.7	0.8
7	5	8	0.8	0.9
8	5	8	≥ 0.9	
9	9	12	0	0.7
10	9	12	0.7	0.8
11	9	12	0.8	0.9
12	9	12	≥ 0.9	
13	13	17	0	0.7
14	13	17	0.7	0.8
15	13	17	0.8	0.9
16	13	17	≥ 0.9	
17	18	25	0	0.7
18	18	25	0.7	0.8
19	18	25	0.8	0.9
20	18	25	≥ 0.9	
21	≥ 26		0	0.7
22	≥ 26		0.7	0.8
23	≥ 26		0.8	0.9
24	≥ 26		≥ 0.9	

	Loan duration (y)		Loan to Value	
	Min	Max	Min	Max
2	< 5		0.7	0.8
3	< 5		0.8	0.9
6	5	8	0.7	0.8
7	5	8	0.8	0.9
10	9	12	0.7	0.8
11	9	12	0.8	0.9
14	13	17	0.7	0.8
15	13	17	0.8	0.9
18	18	25	0.7	0.8
19	18	25	0.8	0.9
22	≥ 26		0.7	0.8
23	≥ 26		0.8	0.9

before filtering

after filtering



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business Console > Decision artifacts > Decision tables > Decision table editor > Columns](#).



Support for custom scenario
provider for decision service testing

Custom data provider for testing

- This new capability allows users of the Business Console to create test suites input and expected results with means other than with Microsoft Excel spreadsheets.
- This new capability allows users of the Business Console to define custom data providers for testing purpose, not only for simulation purpose.
 - They can describe the tests and expected execution details of your test suite.
 - They can download the template.
 - They can define a test suite description file by using this template to describe your custom data provider.
 - They can use this test suite description file to define the test.
- They can use data from different sources, such as databases.
 - Test suites are no longer limited to using data from Microsoft Excel spreadsheet.



Use-case

- The policy manager has a database that contains a lot of data that can be use for testing purpose, but this data cannot be used because it is not available as a Microsoft Excel spreadsheet.
- The IT specialist can now create a custom data provider to use this data from the database for testing purpose.
- The policy manager can use test suites in:
 - Change activities in governed branches
 - In ungoverned branches
 - In validation activities



Implementation

- In the Business Console, you can now define the test predicates with the same expressiveness as the one used in Microsoft Excel spreadsheets.
- You click **Download** to get the `*.testx` template for the test suite configuration file.
 - This configuration file is an extension of the simulation configuration file (`*.simx`) that adds an XML element to define the test predicates.
- You can complete it with the appropriate calls to your Java™ code.
- When the custom data provider is done, uploads it back in to the Business Console.
- You can now use the input data for testing purpose, as they are in the format known by the custom data provider.



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Tests and simulations for business rules > Setting up custom data providers](#).
- The sample for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Samples > Business rule management extensions for Decision Center > Custom data provider for tests and simulations](#).



Variable sets included in
snapshot comparison view

Variable sets included in snapshot comparison view

- This new capability allows users of the Business Console to see variable sets included in snapshots and in snapshot comparisons.
- They can now see the differences between two versions of a variable set by comparing snapshots of these versions in the Business Console.
- For information about snapshots, see the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business Console > Managing changes > Snapshots](#).



Direct links to artifacts

Direct links to artifacts

- This new capability allows users of the Business Console to externally access the various pages of the Business Console.
- Users of the Business Console can bookmark these URLs in their browser, or reference them from an external source, and navigate directly to a specified branch, release, activity, or project element.
- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Center > Governing rules with the Business console > Introducing the Business console > Overview of the Business console](#).



Improved development and operations (DevOps) features

Improved development and operations (DevOps) features

- Easier integration with API management
 - [Automatic creation of OpenAPI description \(Swagger\) for runtime decision services](#)
- Simpler ODM integration in DevOps pipelines
 - [Lightweight build command tool with Maven support](#)
 - [Managed Java™ XOMs embedded in RuleApp archives](#)
- Better experience for ODM developers
 - [Lightweight Sample Server running with WebSphere Application Server Liberty](#)
 - [Decision engine as the default engine and deprecation of classic rule engine](#)
 - [Automatic exception handling during rule execution with decision engine](#)



Automatic creation of OpenAPI
description (Swagger) for runtime
decision services

Automatic creation of OpenAPI description (Swagger) for runtime decision services

- This new capability allows users of the Rule Execution Server Console to export descriptions of REST API for ODM decision services with YAML and JSON OpenAPI standard formats.
- This new capability helps in integrating with other tools that understand these OpenAPI standard formats, such as IBM Business Process Manager (BPM).
- With this new capability, users of the Rule Execution Server Console can:
 - Generate description files of REST API for ODM decision services with OpenAPI formats (YAML or JSON).
 - Generate OpenAPI description files for IBM API Connect.
- They can then use tools that understand these description files to generate client applications.



Ask for the HTDS description file

- In the Rule Execution Server Console, from the **Explorer** tab, **(1)** select your RuleApp and its ruleset, and then **(2)** click **Retrieve HTDS description file**.

The screenshot shows the Rule Execution Server console interface. On the left, the **Navigator** pane shows a tree view of RuleApps, with the path **/production_validation_production/1.0** selected. A large blue **(1)** is overlaid on this path. On the right, the **Ruleset View** pane shows details for the ruleset **/production_validation_production/1.0/loan_validation_production/1.0**. A large blue **(2)** is overlaid on the **Retrieve HTDS Description File** button in the toolbar. Below the ruleset details, there is a **Ruleset Parameters** table.

Direction	Name	Kind	XOM Type
loan	loan	native	loan.LoanRequest
borrower	borrower	native	loan.Borrower
report	report	native	loan.Report



Select the needed OpenAPI format

- Set the **Service protocol type** as **REST (1)**.
- Select the **OpenAPI** format, among the following **new** options:
 - **OpenAPI JSON**
 - **OpenAPI YAML (2)**
 - *Note: Although not an OpenAPI format, the WADL format is still available as an option, as it was in previous releases.*
- You can also specify some options:
 - Use of the latest version of the RuleApp in the path
 - Use of the latest version of the ruleset in the path
 - Use of decision trace information
 - Creation of a proxy for API Connect **(3)**

Retrieve HTDS Description File

/mydeployment/1.0/Miniloan_ServiceRuleset/1.0

Service protocol type

SOAP **REST (1)**

Format OpenAPI - YAML **(2)**

Latest ruleset version

Latest RuleApp version

Decision trace information

Proxy for API Connect (3)

Cancel View Download Test



View the generated description file

- Click **View** to view the generated description file in a browser
 - Example of URL: http://localhost:9080/DecisionService/rest/v1/mydeployment/1.0/Miniloan_ServiceRuleset/1.0/OPENAPI?format=YAML&extension=apiconnect
- The YAML file describes how to run the service:
 - Which host?
 - What is the base path?
 - What is the REST operation (“post”, “get”, ...) called?
 - What are the input parameters?
 - What are the output parameters?

```
---
swagger: "2.0"
info:
  description: "API to invoke the execution of the ruleset Miniloan_ServiceRuleset."
  version: "1.0.0"
  title: "Miniloan_ServiceRuleset API"
host: "$catalog.host"
basePath: "/DecisionService/rast"
consumes:
- "application/json"
produces:
- "application/json"
paths:
  /mydeployment/1.0/Miniloan_ServiceRuleset/1.0:
    post:
      summary: "Invokes the execution of the ruleset Miniloan_ServiceRuleset."
      description: "Executes the ruleset Miniloan_ServiceRuleset using the ruleset\
        \ path /mydeployment/1.0/Miniloan_ServiceRuleset/1.0."
      operationId: "CallDecisionOperation"
      parameters:
      - in: "body"
        name: "Request"
        description: "Request for the ruleset execution. Contains notably the input\
          \ parameters that are used for the ruleset execution."
        required: true
        schema:
          $ref: "#/definitions/Request"
      responses:
        description: "Error occurring when invoking the ruleset execution."
        schema:
          $ref: "#/definitions/Error"
```



Proxy for API Connect

- If the **Proxy for API Connect** check box is clear, the generated description file indicates the host of the RES with its **host** property (for example: **localhost:9090**).
- If the **Proxy for API Connect** check box is selected, the generated description file has:
 - The **host** property that is defined as the **\$(catalog.host)** placeholder, for IBM API Connect to replace with its own host at run time
 - An **x-ibm-configuration (1)** property added, with properties such as the **title** or the **target-url (2)** for IBM API Connect to later use to run the decision.

```
x-ibm-configuration:
  assembly: (1)
  execute:
    - invoke:
        title: "Calls HTDS decision service operation"
        target-url: http://localhost:9080/DecisionService/rest/mydeployment/Miniloan\_ServiceRuleset
        description: "Executes the ruleset Miniloan_ServiceRuleset using the ruleset
          \ path /mydeployment/Miniloan_ServiceRuleset." (2)
```



Download the generated OpenAPI description file

- Click **Download** to download the generated OpenAPI description file and use it within the software of your choice that needs OpenAPI as inputs data.
- **Note:** IBM API Connect is able to import the OpenAPI description file directly from its URL, as generated by the RES Console.
 - Such as http://localhost:9080/DecisionService/rest/v1/mydeployment/Miniloan_ServiceRuleset/OPENAPI?format=YAML&trace=true&extension=apiconnect



Tip: Check the REST API first

- Before you try to generate an OpenAPI description file, test that the REST API is working, by using the **REST Test** in the Rule Execution Server console.



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Developing client applications > Executing rules by using the REST service](#).
- The “**Executing a hosted transparent decision service**” tutorial in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Tutorials > Rule Execution Server tutorials > Executing a hosted transparent decision service](#) is extended with two tasks related to the OpenAPI support feature:
 - Task 3, about how to retrieve an OpenAPI description and call the decision service in a generated client
 - Task 4, about how to run a decision service through IBM API Connect



Lightweight build command tool
with Maven support

Lightweight build command tool with Maven support

- This new capability offers a command-line mechanism to build rule and Java projects into RuleApps without the need to install Rule Designer and Eclipse in headless mode.
- The ODM Build Command is available as an Apache Maven plugin that can be associated with your continuous build pipeline.
- Users of the ODM Build Command plugin for Apache Maven can automate the creation of RuleApps for their rule projects and their Java projects outside of Rule Designer or Decision Center.

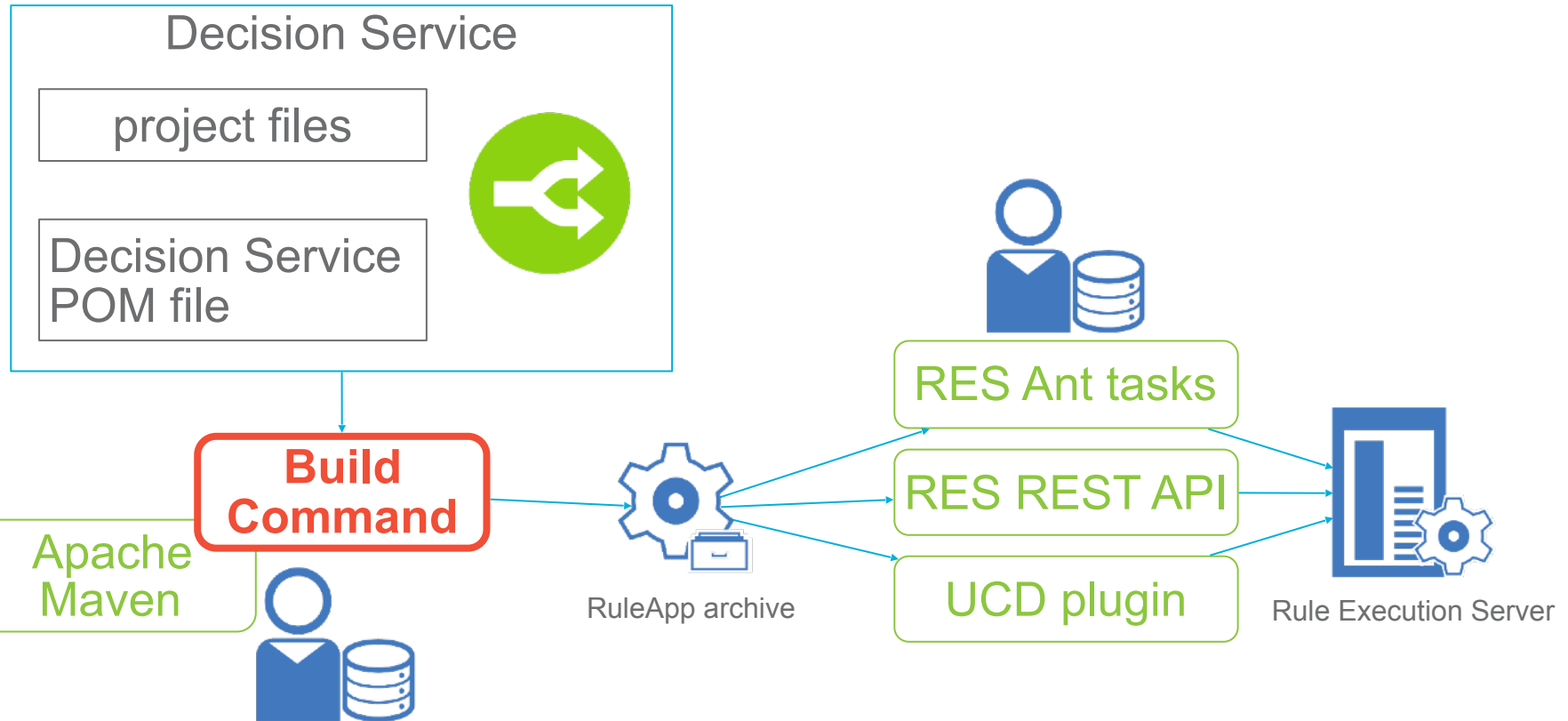


Scope

- The Build Command is focused on generating the decision service executable artifacts (RuleApp and XOM archives).
 - It is not focused on deploying these artifacts directly to Rule Execution Servers.
 - Such deployment can generally be handled with Rule Execution Server REST API, Ant tasks, or an IBM UrbanCode Deploy (UCD) plugin, for example.
 - The Build Command is only for ODM on-premises – it is not available in ODM on Cloud
- This feature is intended at users who are storing their artifacts in a Source Control Management tool and accessing them through Rule Designer.
- You can use Build Command to build **RuleApps from decision services**.
 - You cannot use Build Command to build projects that make use of classic rule projects, of the classic rule engine, or of COBOL.



Architecture



Prerequisites – Install Apache Maven

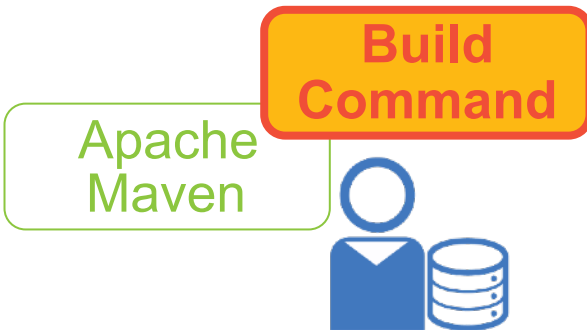
- If not installed yet, install the version of Apache Maven that is required by the ODM Build Command
 - At least version 3.2
 - Current minimum required version as listed in the ODM product documentation
 - Download the needed Apache Maven **binary** file from the Apache Maven download site at: <https://maven.apache.org/download.cgi>
 - Install Apache Maven as indicated on the Apache Maven website at: <https://maven.apache.org/install.html>

Apache
Maven



Prerequisites – Install the ODM “Build Command” plugin

- You must install the Build Command `com/ibm/rules/buildcommand/rules-compiler-maven-plugin` plugin locally in your local Maven repository.
- For instruction on how to install this plugin, read the **Installing the Build command plugin** section of the beta product documentation, in the [Decision Server Rules > Running rules in Rule Designer > Building projects with Build command > Preparing your environment](#) topic.

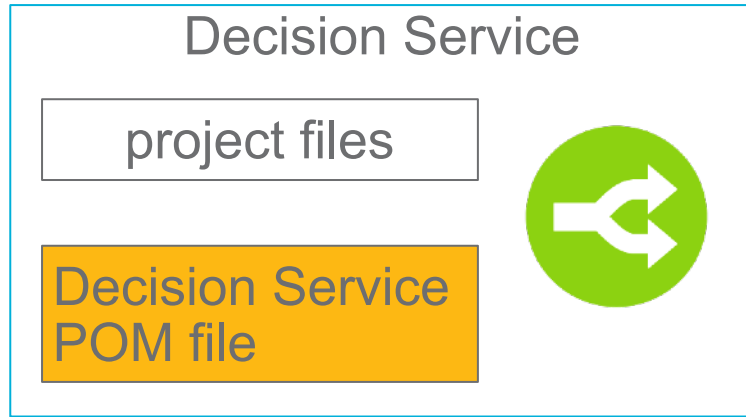


Use-case

- The rule administrator has a series of projects to build.
- The rule administrator first needs to define what to build and structure the projects as a result (Java and rule projects must have their folders at the same level in the file system).
- The rule administrator then creates the corresponding POM files in the projects: Decision Service POM, XOM POM (as needed), and Aggregator POM (optional)
 - A **Project Object Model (POM)** is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details that Maven uses to build the project. It contains default values for most projects.
- When the projects have the supported structure and the corresponding POM files, the rule administrator can run the Build Command to generate one or more RuleApp files (`JAR`) for them.
- The rule administrator can then deploy these `JAR` files to Rule Execution Server as required.



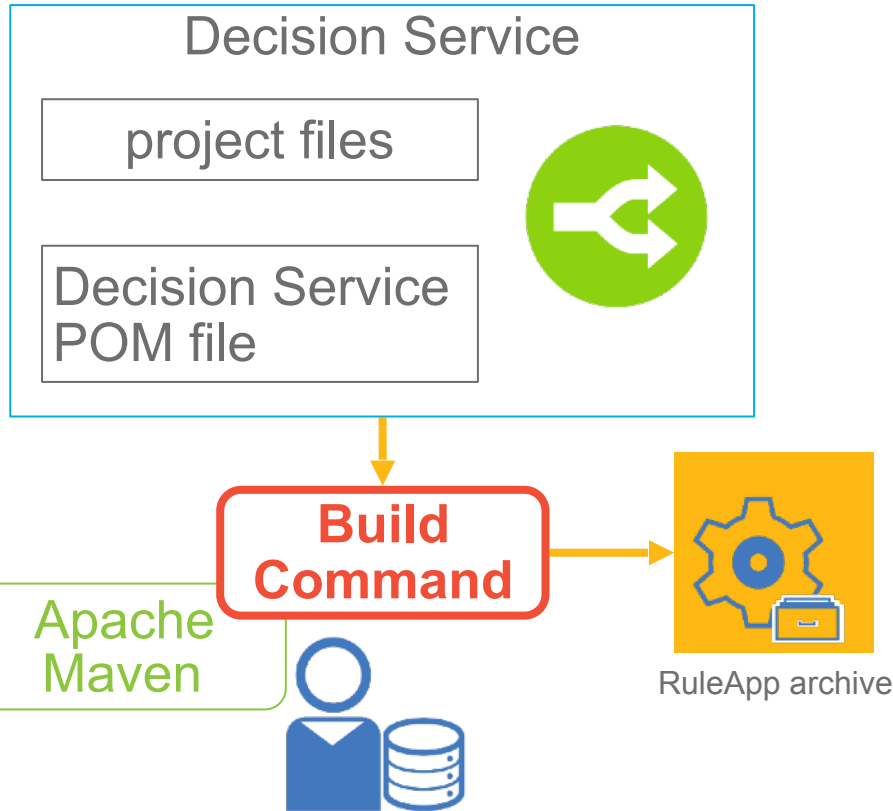
Define the Decision Service POM file



- Define the POM file for your Decision Service.
 - Add the `<configuration>` element in your Decision Service POM file, and set its content as indicated in the **Writing POM files** section of the beta product documentation, in the **Decision Server Rules > Running rules in Rule Designer > Building projects with Build command > Setting up your projects** topic.
- As needed, define XOM POM files for your XOM projects, as well as any required aggregator XOM file, as indicated in the Apache Maven user documentation.



Build the projects

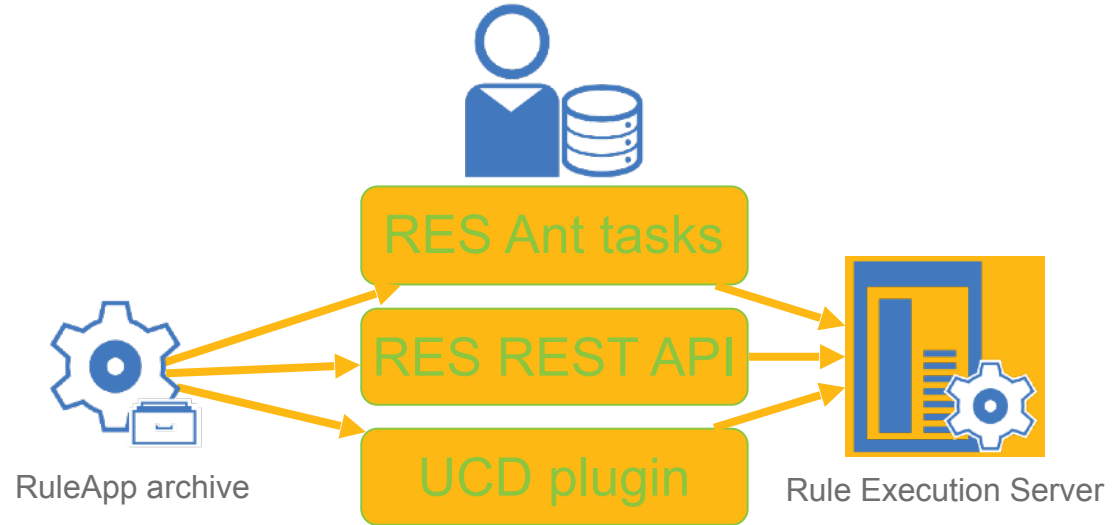


- From the root of your project folder, run the following command:
 - `mvn clean install`
 - The command might take time, and terminates with the **Reactor Summary**, and the **BUILD SUCCESS** message.
- As a result, you have one `target` directory that contains one RuleApp archive (JAR file) per deployment configuration.



Deploy to Rule Execution Server

- You can now deploy these RuleApp archive (JAR) files to Rule Execution Server by using, for example, RES REST API, Ant tasks, or a UCD plugin.



Build Command and projects in Decision Center

- The Build Command works on projects that are locally stored on disk (such as projects managed in Rule Designer or in a SCM tool), and have the supported structure and corresponding POM files.
- The Build Command can therefore not work directly on projects managed in Decision Center, because these projects:
 - Are not locally stored on disk.
 - Cannot have POM files added to them.
- To use the Build Command plugin to build projects that you manage in Decision Center, you might have to perform extra steps, such as:
 - Extract these projects locally on your disk (for example by using the Business Console or Decision Center REST API).
 - Add the needed POM files to these extracted projects.



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Building and running rules > Building projects with Build Command](#).
- For more information about how to define elements of the Build Command POM files, you can also see the sample POM files that are available in the `<ODM_Home>/buildcommand/samples/` folder.



Managed Java XOMs embedded
in RuleApp archives

Managed Java XOMs embedded in RuleApp archives

- This new capability allows use of scripts and API to deploy a RuleApp and its associated managed Java XOMs to RES in a single step. It simplifies the two-step deployment process of previous releases, and removes the risk of inconsistencies between the deployed RuleApp and its deployed managed XOMs.
- With this new capability, users of Rule Execution Server Console, REST APIs, Ant tasks, and Build Command can create or retrieve a RuleApp archive that embeds the managed XOMs of the RuleApp.
 - Self-contained RuleApp archives that contain the ruleset(s) and the required managed XOMs.
- They can deploy a RuleApp archive with its embedded managed XOMs in a single operation.
 - Simplified deployment process, and no risks of inconsistencies between RuleApps and managed XOMs
- This new feature is **optional** and must be explicitly activated (not active by default).



Use-case

- The Rule Execution Server administrator is working on a RuleApp and wants to:
 - Ensure that the proper version of the required XOMs is always deployed at the same time as the RuleApp, what ever its version.
 - Automate the RuleApp deployment and guarantee that the same archive (same RuleApp and managed XOMs) is deployed to all targets, without having to regenerate it each time.
- The Rule Execution Server administrator must therefore ensure that the XOMs of specific RuleApp versions are “*attached*” to these RuleApps, that is, embedded within these RuleApp archives.
- With the new feature, The Rule Execution Server administrator can:
 - **Create or retrieve** RuleApp archives with embedded managed XOMs.
 - **Deploy** a RuleApp archive in one operation, which ensures that the ruleset(s) of the RuleApp and any associated managed XOM(s) are deployed together in one operation.



Possibilities

- You can create/retrieve or deploy embedded managed XOMs in RuleApp archives with the Rule Execution Server Console, the Rule Execution Server REST API, Ant tasks, and Build Command, as follows:

Where	Create/Retrieve	Deploy
Rule Execution Server Console	YES	YES
Rule Execution Server REST API	YES	YES
Ant tasks	NO	YES
Build Command	YES	NO



Creation with Rule Execution Server Console

- As with previous versions of ODM, you can create a RuleApp archive by using the **Download Archive** menu for this RuleApp (on the **Explorer** tab).
- To create RuleApp archives with embedded managed XOMs, first set the `ilog.rules.res.XOM_IN_RULEAPP_ARCHIVE` parameter in the `web.xml` deployment descriptor file of the Rule Execution Server Console to `true`.
 - This parameter is set to `false` by default. By default, the created RuleApp archives do not embed any managed XOMs.
- If this parameter is `true`, when you use the **Download archive** to create a RuleApp archive for a RuleApp with managed XOMs, the created archive embeds these managed XOMs.

RuleApp View



Name	production_deployment
Version	1.0
Creation Date	Feb 7, 2017 4:12:15 PM GMT+01:00
Display Name	
Description	



Creation with Rule Execution Server REST API (1 of 3)

- Open the Rule Execution Server Console, and select the RuleApp you want to create a RuleApp archive with embedded managed XOMs for.

The screenshot displays the IBM Rule Execution Server console interface. The top navigation bar includes 'Home', 'Explorer', 'Decision Warehouse', 'Diagnostics', 'Server Info', and 'REST API'. The breadcrumb path is 'Explorer > RuleApps > RuleApp > Ruleset'. The left pane shows a 'Navigator' tree with the following structure:

- RuleApps (2)
 - /mydeployment/1.0 (1)
 - /Miniloan_ServiceRuleset/1.0
 - /test_deployment/1.0 (2)
 - /loan_validation_production/1.0
 - /loan_validation_with_score_and_grade/1.0
- Resources (2)
 - loan-validation-xom.zip/1.0
 - miniloan-xom.zip/1.0
- Libraries (2)
 - Service Information

The right pane shows the 'Ruleset View' for '/mydeployment/1.0/Miniloan_ServiceRuleset/1.0'. It includes a table of metadata:

Name	Miniloan_ServiceRuleset
Version	1.0
Creation Date	Oct 17, 2016 3:31:13 PM GMT+02:00
Display Name	Miniloan ServiceOperation
Description	
Rule engine	Classic Rule Engine
Status	enabled
Debug	disabled

Below the metadata is a 'Ruleset Parameters' section with a table:

Direction	Name
	borrower
	loan

At the bottom, the 'Managed URIs' section is highlighted with a blue box and contains a table:

Select All	Index	URI
<input type="checkbox"/>	1	rest://mydeployment_1.0/1.0



Creation with Rule Execution Server REST API (2 of 3)

- Go to the **REST API** tab of the RES console.
- (1)** Select the **getRuleAppArchive** REST API command, and enter the needed parameters.
- (2)** Set the **ruleappname** and **ruleappversion** properties, and then set the **xom** property to **true**.
 - Setting this property ensures that the managed XOM is embedded in the RuleApp archive.
- (3)** Click **Call method** to create a RuleApp archive that includes both the ruleset and the XOM.

(1)

(2)

(3)

Parameter Details
Show Parameter Details



Creation with Rule Execution Server REST API (3 of 3)

- A message opens that says:

 The response contains binary data (application/octet-stream).

To download the data, you can submit the request directly by clicking on the following link: </api/v1/ruleapps/<RuleAppname>/<version>/archive>

- You can click the link to retrieve (download) the RuleApp archive, which contains both:
 - the ruleset archive:
`ruleapp\mydeployment\1.0\Miniloan_ServiceRuleset\1.0\ruleset.jar`
 - The XOM archive: `ruleapp\miniloan-xom_1.0.zip`
- You can then deploy the RuleApp archive (with embedded managed XOMs) by the mean of your choice.



Creation with Build Command

- You can create a RuleApp archive with the **Build Command**, as indicated in the [Lightweight build command tool with Maven support](#) section.
- If you want to create a RuleApp archive that embeds the managed XOMs, you must specify the `<embeddedXom>true</embeddedXom>` deployment parameter in the POM file of your project.



Deployment of RuleApp archives with embedded managed XOMs

- As with previous versions of ODM, you can deploy a RuleApp archive by using the Rule Execution Server Console, the Rule Execution Server REST API, and Ant tasks.
- If this RuleApp archive has embedded managed XOMs, **the RuleApp and the embedded managed XOMs are all deployed in the same operation.**
 - If there are no embedded managed XOMs, the deployment is as in previous versions.
- As with previous versions of ODM, when you deploy a RuleApp archive, you must specify the versioning policies for the RuleApp and rulesets.
- With the new feature, **you must also specify the versioning policies for any embedded managed XOMs** (libraries and resources).
 - If there are no embedded managed XOMs, these policies are ignored.



Embedded managed XOMs versioning - Principles

- The deployed versions of the XOMs and of the rulesets are modified (or not) per the specified policies, depending on whether the embedded managed XOMs and the rulesets of the deployed RuleApp archive differ from the ones already deployed.
 - The content of these artifacts are compared.
 - The value of the `ruleset.managedxom.uris` ruleset property is set to the version of the XOM that is really used: the one to deploy or an already deployed one (see next slide).
- If there are no embedded managed XOMs in the RuleApp archive, the resource versioning policy and the library versioning policy are ignored.

Deploy RuleApp Archive

Local path of RuleApp archive No file selected.

RuleApp Versioning Policy

- Increment RuleApp major version
- Increment RuleApp minor version
- Replace RuleApp version
- Increment Ruleset major version
- Increment Ruleset minor version
- Replace Ruleset version

Resource Versioning Policy

- Increment resource major version
- Increment resource minor version

Library Versioning Policy

- Increment library major version
- Increment library minor version



Embedded managed XOMs versioning – Example

Consider a XOM to deploy with version “V”, while a XOM with the same name is already deployed with versions 1.0, 1.1, and 1.4. Using the “minor version” policies, the version of the XOM “V”, after its deployment, is as shown in this table.

V	Already deployed XOM matching XOM “V”?	XOM “V “ really deployed?	XOM version in the ruleset property after deployment?
1.0	1.0	No, the XOM 1.0 is considered.	1.0 (no changes)
	1.1	No, the XOM 1.1 is considered.	1.1 (modified ruleset property)
	1.4	No, the XOM 1.4 is considered.	1.4 (modified ruleset property)
	none of 1.0, 1.1, and 1.4	YES	1.2 (modified ruleset property), because 1.1 already used.
1.1	1.0	No, the XOM 1.0 is considered.	1.0 (modified ruleset property)
	1.1	No, the XOM 1.1 is considered.	1.1 (no changes)
	1.4	No, the XOM 1.4 is considered.	1.4 (modified ruleset property)
	none of 1.0, 1.1, and 1.4	YES	1.2 (modified ruleset property)
1.4	1.0	No, the XOM 1.0 is considered.	1.0 (modified ruleset property)
	1.1	No, the XOM 1.1 is considered.	1.1 (modified ruleset property)
	1.4	No, the XOM 1.4 is considered.	1.4 (no changes)
	none of 1.0, 1.1, and 1.4	YES	1.5 (modified ruleset property)
else	1.0	No, the XOM 1.0 is considered.	1.0 (modified ruleset property)
	1.1	No, the XOM 1.1 is considered.	1.1 (modified ruleset property)
	1.4	No, the XOM 1.4 is considered.	1.4 (modified ruleset property)
	none of 1.0, 1.1, and 1.4	YES	V (no changes)



Documentation

- The documentation for this feature is available in the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Rule Execution Server Console online help > Managing Java XOM resources and libraries > Embedded managed Java XOM.](#)
- The documentation for the versioning policy is available in the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Rule Execution Server Console online help > Managing RuleApps > Versioning policy options.](#)



Lightweight Sample Server running with WebSphere Application Server Liberty

Lightweight Sample Server running with WebSphere Application Server Liberty

- The ODM Sample Server for doing samples and tutorials is now on WebSphere Application Server Liberty.
 - Installing, configuring, and starting Sample Server is lighter, faster, and easier.
- The ODM Sample Server database is now H2, instead of Apache Derby.
 - Database performance with H2 is better, compared to Apache Derby.
- The documentation for this feature is available in the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Samples environment](#).



Decision engine as the default engine
and deprecation
of classic rule engine

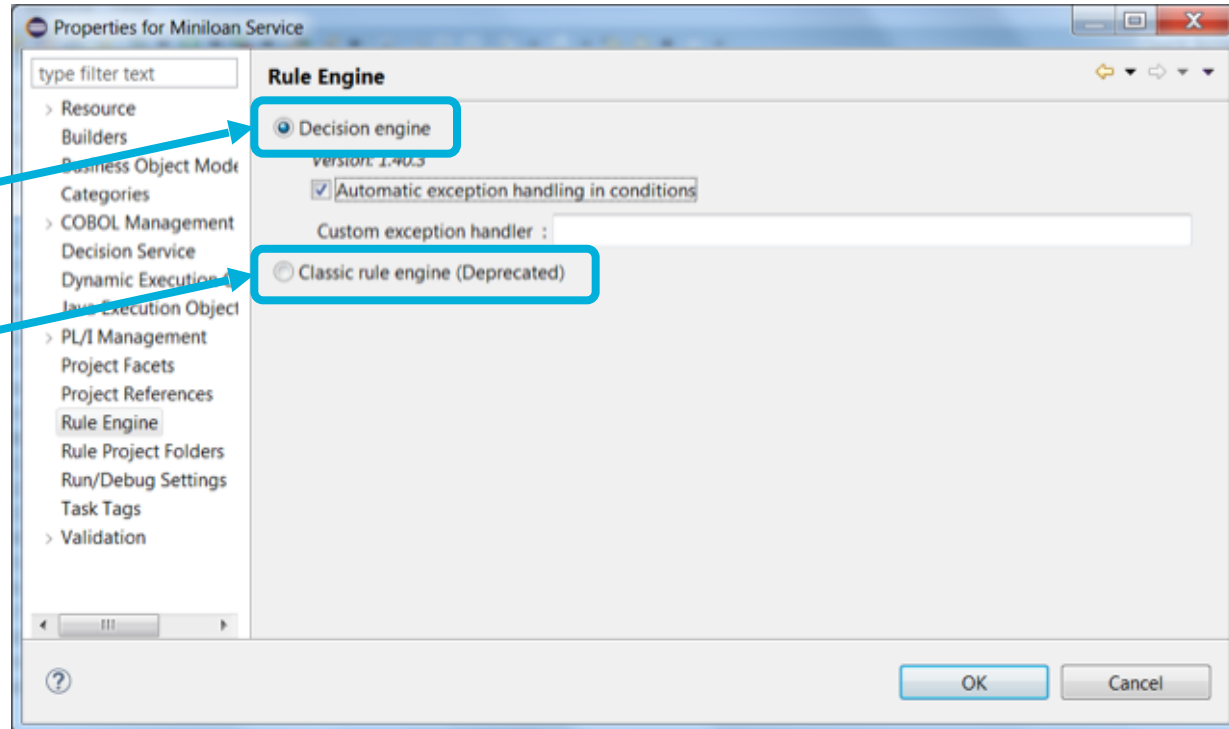
Decision engine as the default engine and deprecation of classic rule engine

- The decision engine replaces the classic rule engine as the default rule engine.
- By using the decision engine, users of ODM benefit from better performance and scalability.



Rule Engine selection in Rule Designer

- In Rule Designer, select the Rule Engine to use in your project:
 - By default: **Decision engine** is selected.
 - Select **Classic rule engine (deprecated)** to revert to the behavior of previous releases.



Documentation

- The documentation for the deprecation of the classic rule engine is available in the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Migrating and updating your applications > Deprecated and removed features](#).
- For information about how to migrate projects from the classic rule engine to the decision engine, see the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Migrating and updating your applications > Migrating Decision Server Rules > Migrating from classic rule engine to decision engine](#).
- For information about execution with the decision engine, see the IBM Knowledge Center for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Building and running rules > Executing rulesets > Executing using the decision engine](#).



Automatic exception handling during
rule execution
with decision engine

Automatic exception handling during rule execution with decision engine

- This new capability allows ruleset execution to continue instead of interrupting even if exceptions occur at run time in its rule conditions.
- This new capability simplifies rule authoring by removing the need for rule authors to add, in their rule conditions, tests against `null` values on the fields of the input parameters.
 - Rule authors can now rely on the decision engine to handle exceptions in rule conditions at run time.
- With this new capability, exceptions in conditions can be considered as unknown values instead of errors.
- This new feature is **optional** and must be explicitly activated (not active by default).



Exceptions seen as “unknown”

- If an exception occurs during the evaluation of a condition expression, this evaluation is considered as unknown.
- The overall condition of the rule artifact (action rule or decision table) is evaluated as **TRUE**, **FALSE**, or unknown, based on the evaluations of its expressions with a 3-valued logic (next slide).
- **No** action parts of a rule artifact are run if its condition is **unknown**.
- The “then” action part of a rule artifact is run only if its condition is **TRUE**.
- The “else” action part of an action rule is run only if its condition is **FALSE**.
 - **The fact that the condition is not TRUE no longer systematically means that the “else” action part is run.** If the condition is not **FALSE** (that is, it is **unknown**), the “else” action part is also not run.
- The “otherwise” action part of a decision table is handled like the “else” action part of action rules.



NOT, AND, and OR “truth” tables in conditions

A	NOT A
TRUE	FALSE
UNKNOWN	UNKNOWN
FALSE	TRUE

A AND B		A		
		TRUE	UNKNOWN	FALSE
B	TRUE	TRUE	UNKNOWN	FALSE
	UNKNOWN	UNKNOWN	UNKNOWN	FALSE
	FALSE	FALSE	FALSE	FALSE

A OR B		A		
		TRUE	UNKNOWN	FALSE
B	TRUE	TRUE	TRUE	TRUE
	UNKNOWN	TRUE	UNKNOWN	UNKNOWN
	FALSE	TRUE	UNKNOWN	FALSE

- **UNKNOWN** → No action parts of a rule artifact are run.



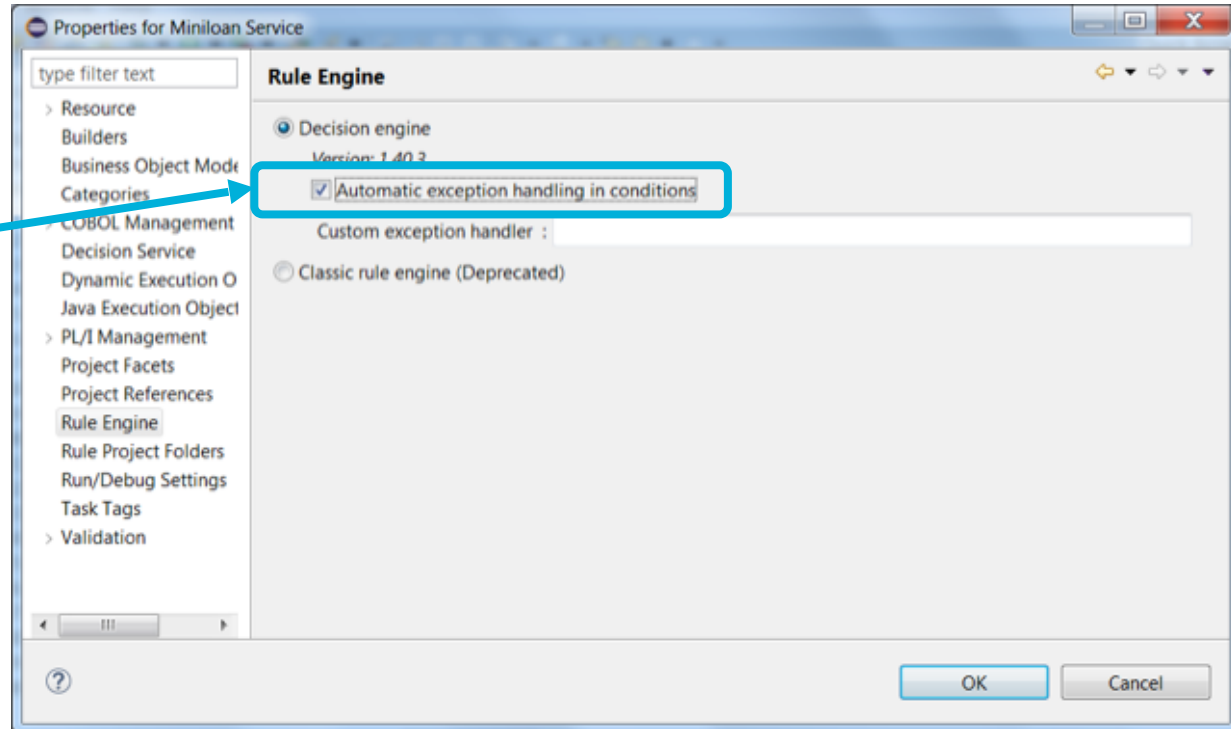
Benefits

- You no longer need to write your business artifacts to test `null` values for the fields of your input parameters.
 - For example: one field in your input form might be optional, and you no longer need to have business users test this field value against `null` in their rule condition parts.
- No exception related to input data (such as `NullPointerException`) is thrown at run time when the condition parts of business artifacts are evaluated.
- Exceptions in the condition parts of business artifacts are handled at the engine level, and considered as unknown values.
- Deal with incomplete or missing data through a consistent logic that is based on the Boolean logic augmented to manage unknown values.



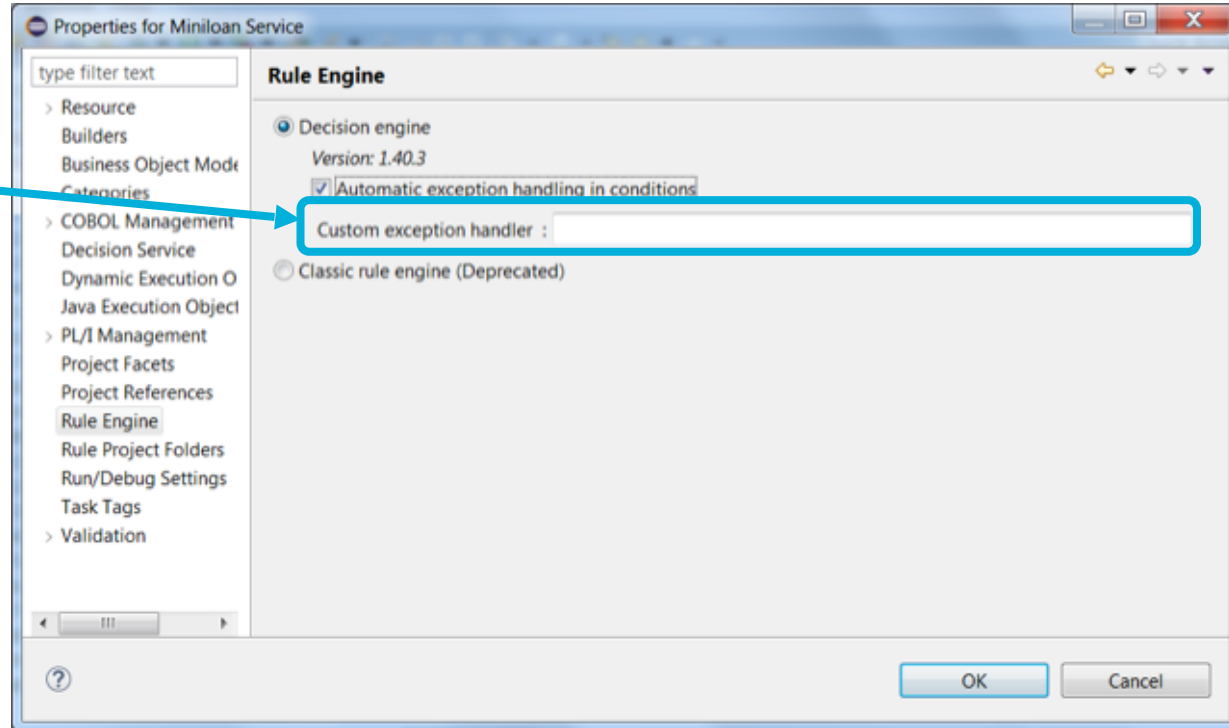
Activation in Rule Designer

- For the Automatic Exception Handling to be considered at build time, select the **Automatic exception handling in conditions** check box.



Automatic exception handling + custom exception handler

- If you select “**Automatic exception handling in conditions**”, and also specify a custom exception handler:
 - Exceptions in conditions handled by *automatic exception handling* are caught and so not seen by the *custom exception handler*.
 - Exceptions in actions are not handled by the *automatic exception handling*.



Logging API

- Use the `com.ibm.rules.engine.aeh` logger name to define the logging level. The minimum level to get logs is FINE.
 - Example: `com.ibm.rules.engine.aeh.level=FINE`

Jan 17, 2017 9:45:17 AM

`com.ibm.rules.generated.ruleflow.miniloan$003epre$002dvalidation.FASTEngine exceptionLogger`

`FINE: AEH_LOG: Java.lang.NullPointerException: In 'customer status' rule condition, at offset 3, and length 61`

`the approval status is false [Too big Debt-To-Income ratio]`

- “**aeh**” = **automatic exception handling**



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Decision Server Rules > Building and running rules > Optimizing execution > Exception handling in rules.](#)



Product simplification

Removals for product simplification

- The following deprecated features are now removed:
 - Events
 - Rule Solutions for Office
 - Scorecard
 - Business Rules Embedded toolkit



Updated list of supported platforms

- The list of supported platforms is available in the IBM Operational Decision Manager detailed system requirements page at: <http://www-01.ibm.com/support/docview.wss?uid=swg27023067>



Integration with IBM Cloud Product Insights

Integration with IBM Cloud Product Insights

- This new capability helps users track ODM instances and report ODM usage metrics.
- This new capability allows to utilize IBM Cloud Product Insights, an IBM Bluemix® service, to monitor Decision Server and Decision Center product usage, such as the number of decisions or the number of managed artifacts.



How to?

- Understand the metrics that you can use, and identify the ones that you want to use.
- Create and configure an IBM Cloud Product Insights instance in Bluemix.
 - Take note of its **API key** required for the connection of your ODM components.
- Use this API Key to set up Decision Server and Decision Center to integrate with IBM Cloud Product Insights:
 - Use the `res-setup` Ant task to set up Decision Server and configure each artifact for which you want to enable or disable metering.
 - Define the `decisioncenter-configuration.properties` file under `<Liberty Home>/usr/servers/odm8900/config/` to set up Decision Center.
- You will have to restart your Liberty application server.



Metrics

- Decision Server metrics:
 - Decisions: number of calls to a decision service
 - Decision Units: indication of the complexity of the decisions taken
 - Rule Evaluated (max) [resp. (min), (sum)]: maximum [resp. minimum, sum of] number of rules and decision table rows evaluated in invoked rulesets
- Decision Center metrics:
 - Artifacts: number of all unique versioned artifacts
 - Registered users: number of users declared in the Decision Center repository
 - Rule Artifacts: number of unique versioned rule artifacts



Documentation

- The documentation for this feature is available in the IBM Knowledge Center collection for ODM at [Operational Decision Manager 8.9.0 > Overview: Operational Decision Manager > Introducing Operational Decision Manager > Integrating with IBM Cloud Product Insights](#) and [Operational Decision Manager 8.9.0 > Configuring > Configuring for IBM Cloud Product Insights](#).
- For more information about IBM Cloud Product Insights, see [Getting started with Product Insights \(Bluemix website\)](#).

