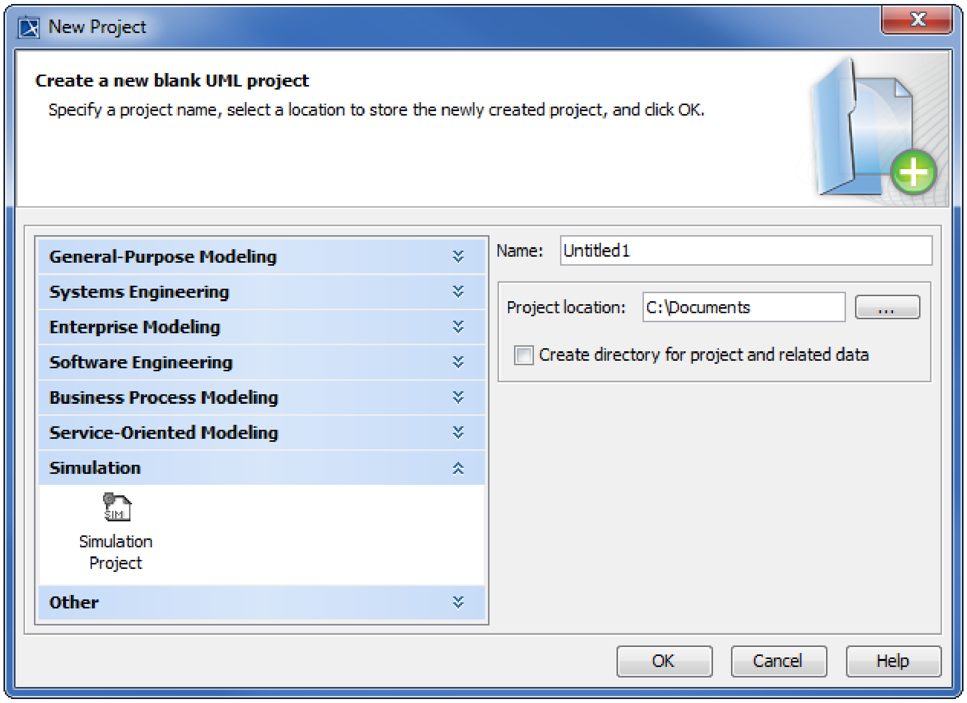
Creating a model for Activity simulation

You can simulate a UML Activity or a Classifier whose Classifier Behavior is defined by an Activity. This section demonstrates how to create a simple but executable Activity model through the following steps

1. Create a Class containing two properties typed by Integers.
2. Create an Activity to print the summation value of the two properties.
3. Assign the Activity as the Classifier Behavior of the created Class.
4. Create an opaque Behavior to print the summation value of two input parameters of type **Integer**.
5. Write a script to print the summation of the given integer values that are referred to by the two input parameters.
6. Complete the Activity diagram of the Class.
7. Create a **ReadSelfAction** to read a runtime object that will be supplied to the input pins of both the **readX** and **readY** Actions.
8. Create an **InstanceSpecification** and assign the values to the slots that correspond to the two created properties.

To create a Class containing two attributes typed by Integers

1. To create a new *UML* project, click **File** > **New Project** on the main menu. The **New Project** dialog will open.



New Project

Create a new blank UML project

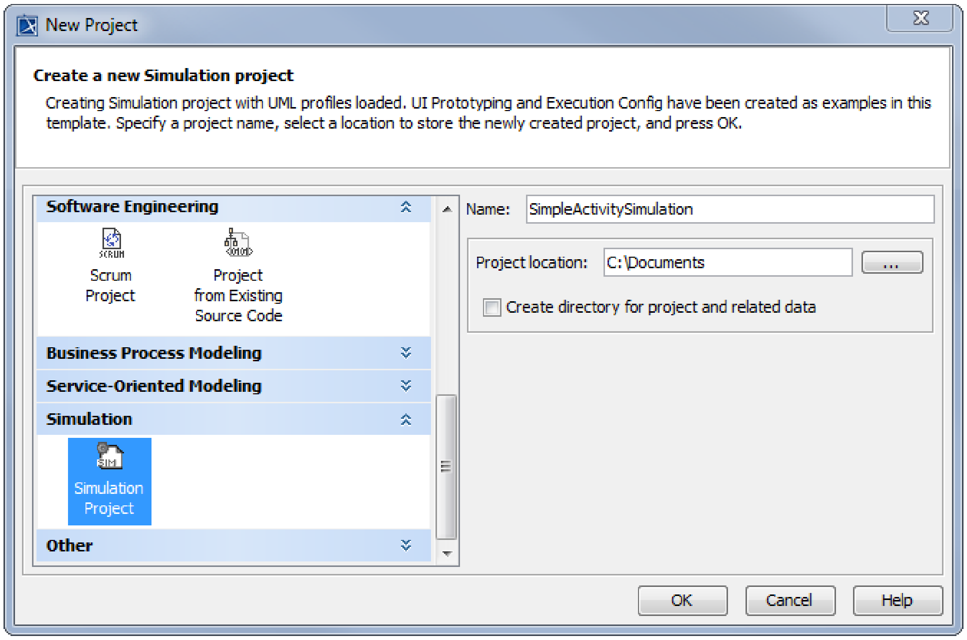
Specify a project name, select a location to store the newly created project, and click OK.

Name: Untitled1

Project location: C:\Documents

□ Create directory for project and related data

2. Select **SimulationProject** from the **Simulation** group and specify the project's name, e.g., "SimpleActivitySimulation".
3. Specify the location where you want to save your project file, and then click **OK**.
4. Right-click the **Model** model in the containment browser and select **Create Element** > **Class**. A new Class element, which is the context of the Activity, will be created in the containment browser. Name the created Class, e.g., "SumPrinter".
5. Add two properties: **x** and **y** of type **Integer**.
Right-click the **SumPrinter** Class and select **Create Element** > **Property**. Type 'x' to name the property (see the first figure below). Right-click **x** and select **Specification** to open its Specification window. Select **Integer** as the property type (see the second figure below).



New Project

Create a new Simulation project

Creating Simulation project with UML profiles loaded. UI Prototyping and Execution Config have been created as examples in this template. Specify a project name, select a location to store the newly created project, and press OK.

Software Engineering: Scrum Project, Project from Existing Source Code

Business Process Modeling

Service-Oriented Modeling

Simulation: Simulation Project

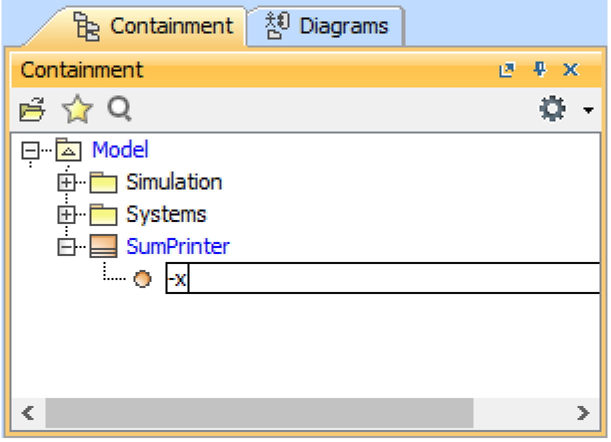
Other

Name: SimpleActivitySimulation

Project location: C:\Documents

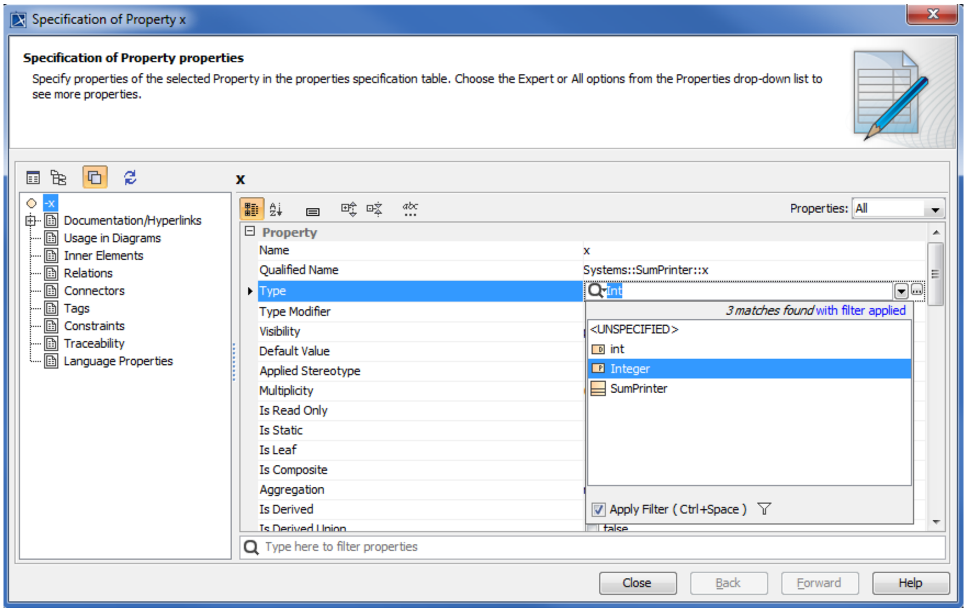
□ Create directory for project and related data

Creating a new Property x for the SumPrinter Class.



Model
- Simulation
- Systems
- SumPrinter
  - -x

The Specification of Property x.



Specification of Property x

Specification of Property properties

Specify properties of the selected Property in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

x

Properties: All

Property	
Name	x
Qualified Name	Systems::SumPrinter::x
Type	int
Type Modifier	
Visibility	
Default Value	
Applied Stereotype	
Multiplicity	
Is Read Only	
Is Static	
Is Leaf	
Is Composite	
Aggregation	
Is Derived	
Is Derived Union	false

3 matches found with filter applied

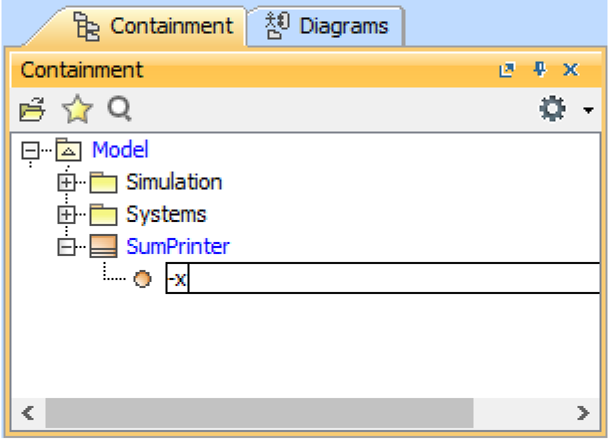
<UNSPECIFIED>
int
Integer
SumPrinter

☑ Apply Filter (Ctrl+Space)

Type here to filter properties

The SumPrinter Class with the Properties X and Y of Integer Type.

6. Repeat Step 5 to create Property **y**.



7. Once the Properties **x** and **y** have been created, define the Behavior of the created Class: Specify the Classifier Behavior of the **SumPrinter** Class with a UML Activity element.

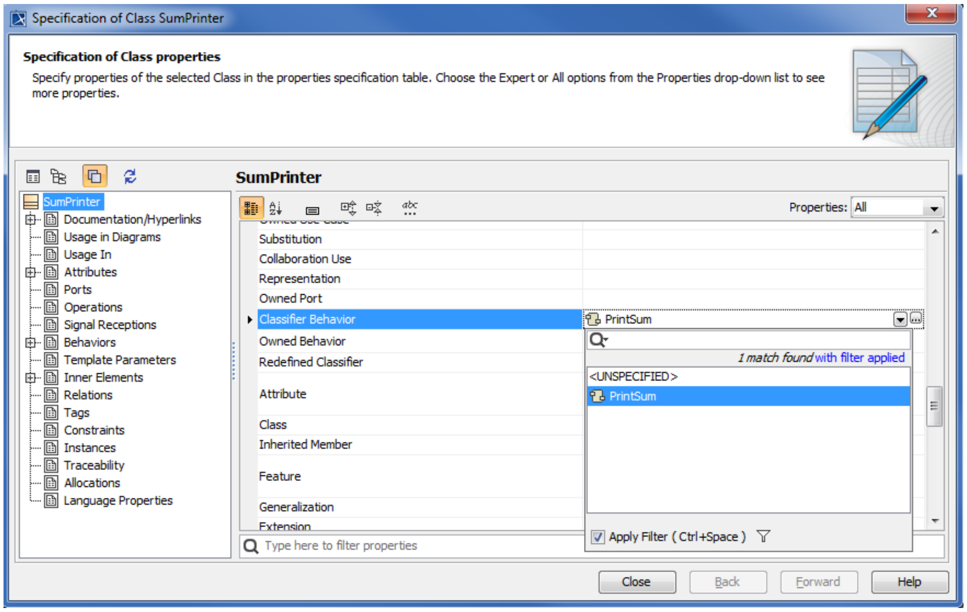
To create an Activity to print the summation value of the two properties

1. Right-click the **SumPrinter** Class in the containment browser and select **New Diagram** > **Activity Diagram** to create a new Activity under it.
2. Name the diagram "PrintSum".

Now that the Activity has been created, you need to assign it as the Classifier Behavior of SumPrinter.

To assign the Activity as the Classifier Behavior of the created Class

1. Right-click the **SumPrinter** Class in the containment browser and select **Specification** to open its Specification window.


Specification of Class SumPrinter

Specification of Class properties

Specify properties of the selected Class in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

SumPrinter

Properties: All

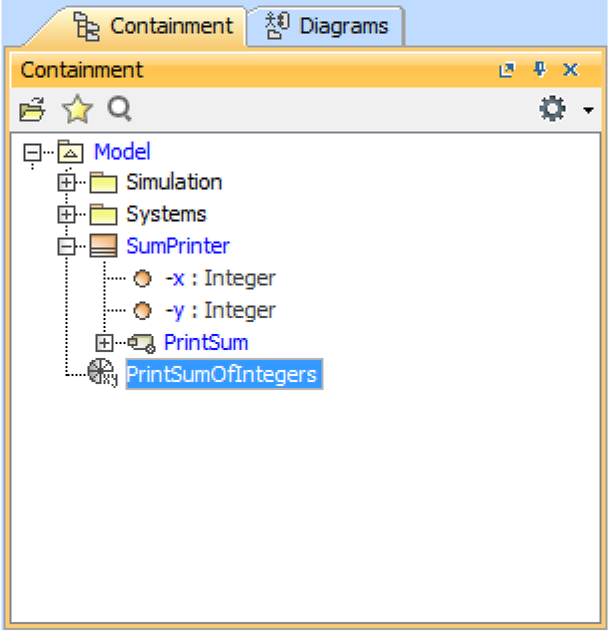
Classifier Behavior: PrintSum

1 match found with filter applied

2. Select **All** from the **Properties** drop-down menu to make sure that all of the properties are listed in the dialog.
3. Click **Classifier Behavior** and select the **PrintSum** Activity from the drop-down list on the right-hand side.

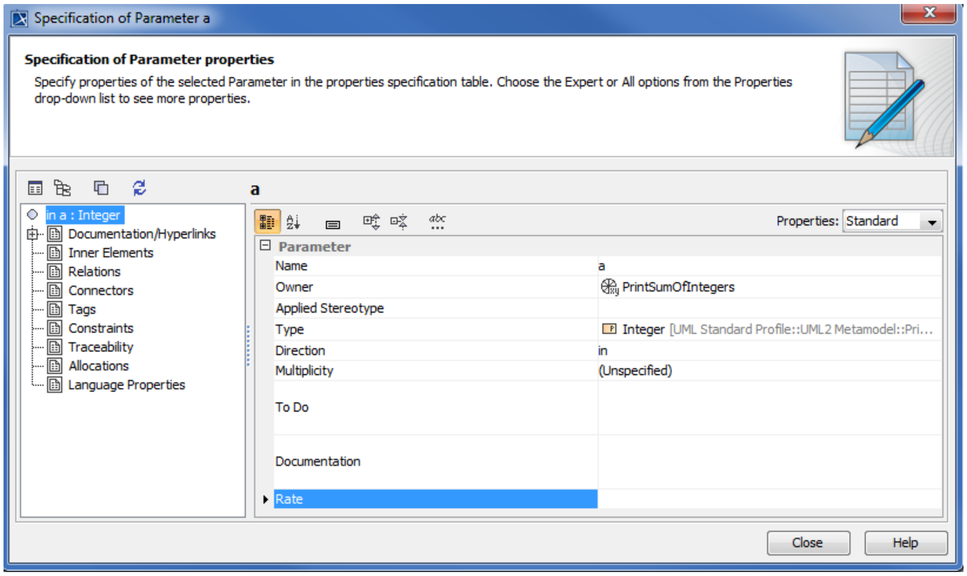
To create an opaque Behavior to print the summation value of the two input parameters of type **Integer**:

1. Right-click the **Model** model in the Containment tree and select **Create Element** > **Opaque Behavior**. A new opaque Behavior will be created under the **Model** model.
2. Name it "PrintSumOfIntegers". The following figure shows the PrintSumOfIntegers Opaque Behavior in the Containment browser.



- Model
  - Simulation
  - Systems
  - SumPrinter
    - -x : Integer
    - -y : Integer
    - PrintSum
  - PrintSumOfIntegers

3. Add two input parameters of type **Integer**: **a** and **b**.
4. Right-click the **PrintSumOfIntegers** opaque Behavior and select **Create Element** > **Parameter**. Name the created parameter 'a' in the name field and select **Integer** as the type of Parameter **a** (see the following figure that shows the Specification of Parameter **a**).



Specification of Parameter properties

Specify properties of the selected Parameter in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

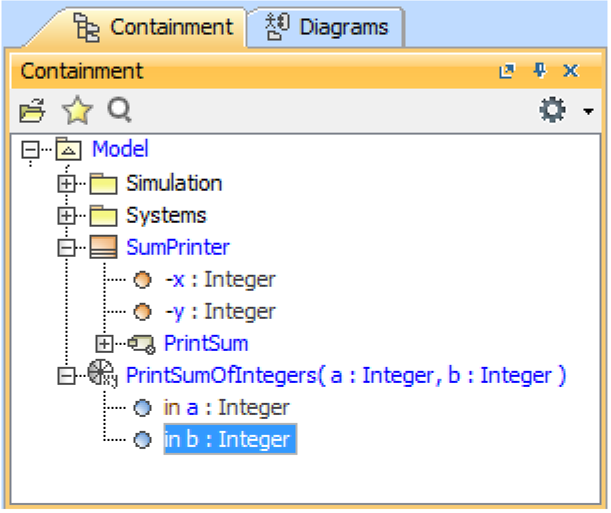
- in a : Integer
  - Documentation/Hyperlinks
  - Inner Elements
  - Relations
  - Connectors
  - Tags
  - Constraints
  - Traceability
  - Allocations
  - Language Properties

a

Properties: Standard

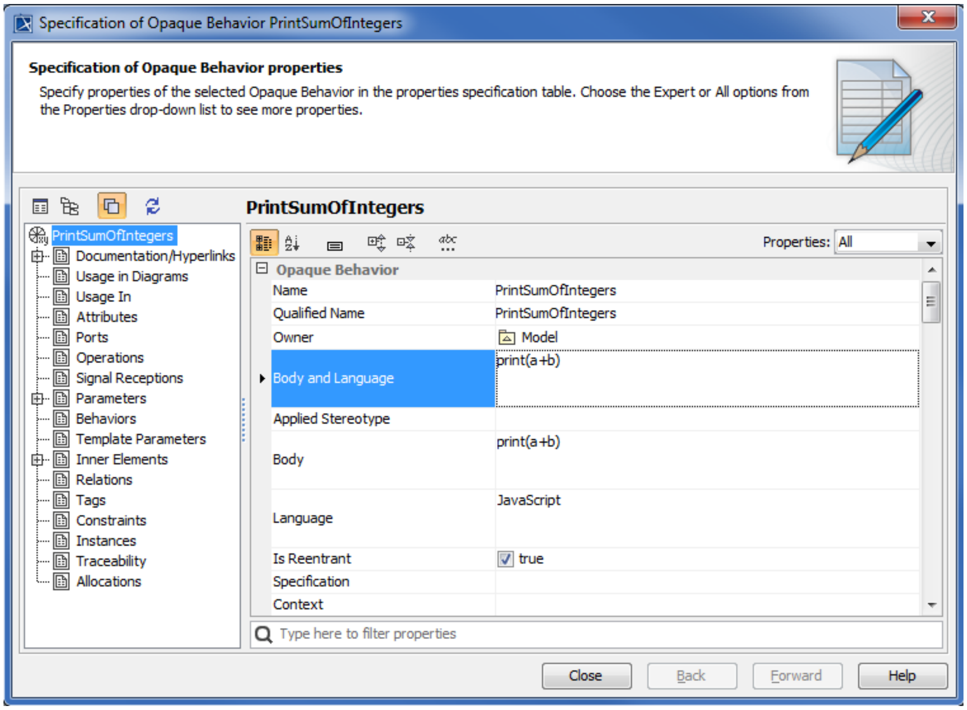
Parameter	
Name	a
Owner	PrintSumOfIntegers
Applied Stereotype	
Type	Integer [UML Standard Profile::UML2 Metamodel::Pri...
Direction	in
Multiplicity	(Unspecified)
To Do	
Documentation	
Rate	

5. Repeat Steps 3-4 to create Parameter **b** as shown in the following figure that shows the PrintSumOfIntegers Opaque Behavior containing Parameters **a** and **b** in the Containment browser.



To write a script to print the summation of the given integer values

- Open the Specification window of the **PrintSumOfIntegers** opaque Behavior and write a script in the **Body** field (You can use any scripting language that is supported by MagicDraw's Macro Engine, e.g., BeanShell, Groovy, JavaScript, Python, or Ruby). In this example, JavaScript will be used to print the summation of the given integer values that are referred to by the parameters **a** and **b**; therefore, the script will be: "print(a+b)".



Specification of Opaque Behavior properties

Specify properties of the selected Opaque Behavior in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

Properties: All

Opaque Behavior	
Name	PrintSumOfIntegers
Qualified Name	PrintSumOfIntegers
Owner	Model
Body and Language	print(a+b)
Applied Stereotype	
Body	print(a+b)
Language	JavaScript
Is Reentrant	☑ true
Specification	
Context	

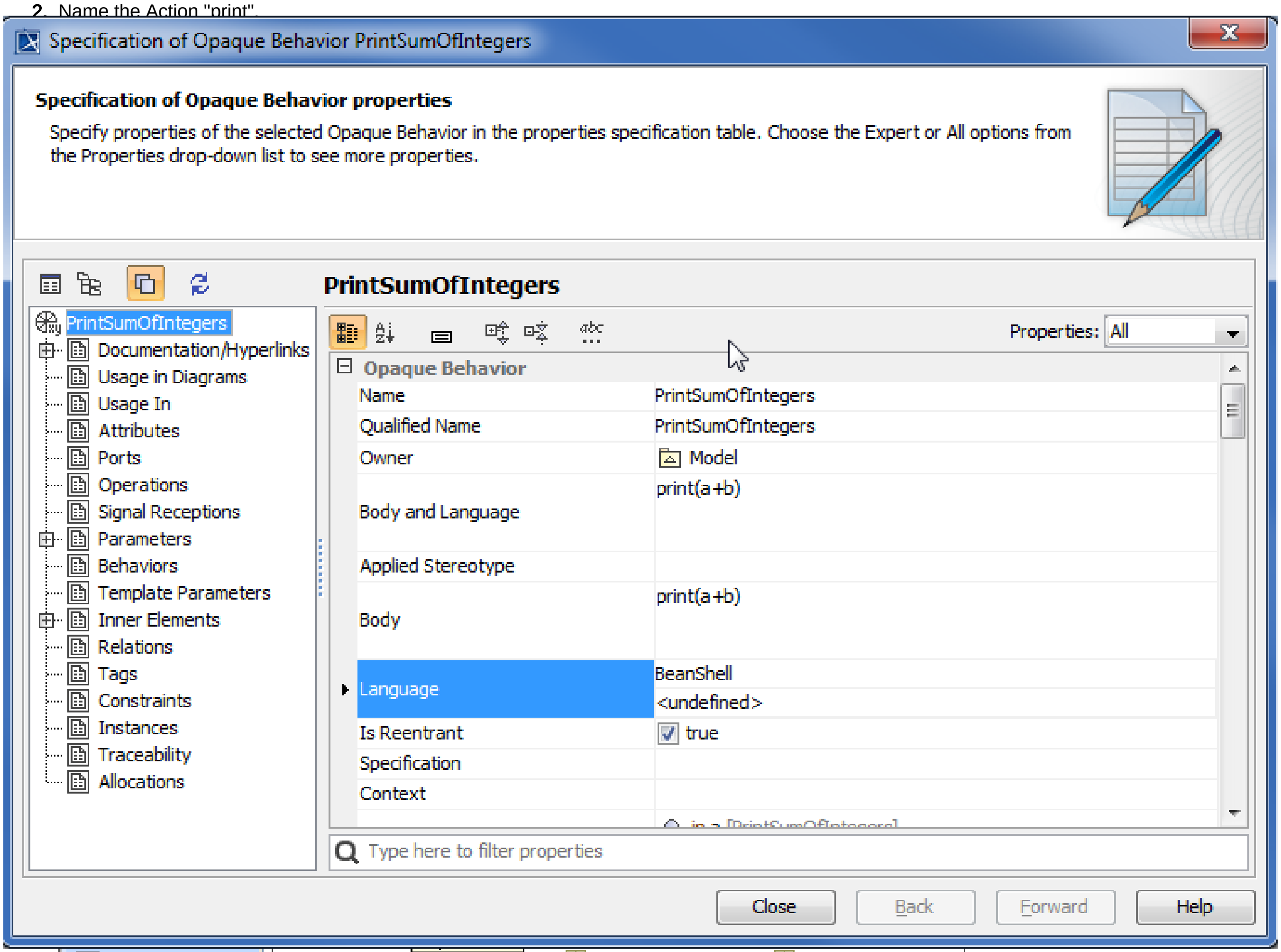
Note

The next step is to complete the **PrintSum** Activity diagram of the **SumPrinter** Class and add a **ReadStructuralFeatureAction** so that the values of properties **x** and **y**, which are owned by the **SumPrinter** Class, are readable. The values of **a** and **b** will later be passed on to the **PrintSumOfIntegers** opaque Behavior as the values of input parameters **a** and **b** respectively.

To complete the Activity diagram of the Class

1. Drag the **PrintSumOfIntegers** opaque Behavior from the containment browser to the **PrintSum** Activity diagram. A new Action of **PrintSumOfIntegers** will be created.
2. Name the Action "print".

If you want to use a scripting language other than JavaScript, you can specify it in the **Language** attribute of the Specification window.



Specification of Opaque Behavior PrintSumOfIntegers

Specification of Opaque Behavior properties

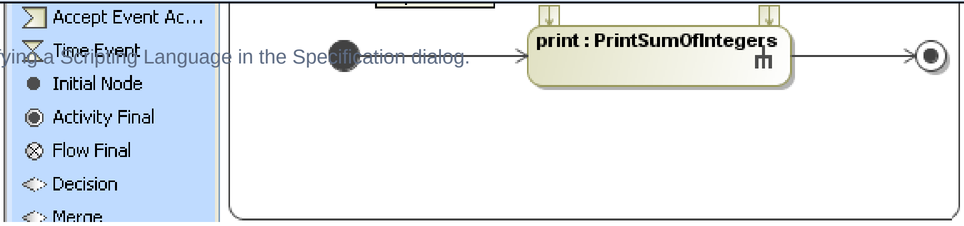
Specify properties of the selected Opaque Behavior in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

PrintSumOfIntegers

Properties: All

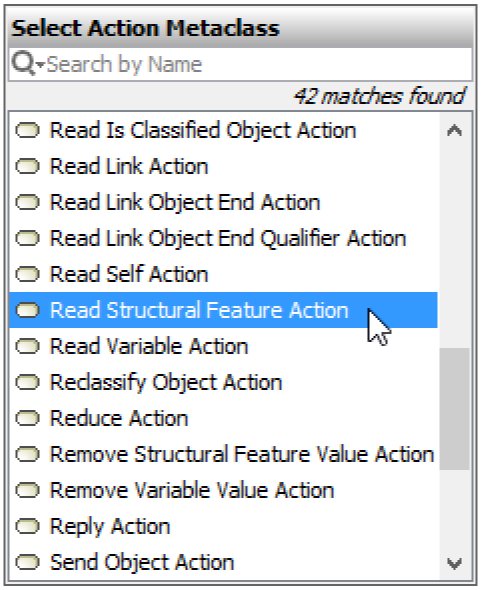
Opaque Behavior

Name	PrintSumOfIntegers
Qualified Name	PrintSumOfIntegers
Owner	Model
Body and Language	print(a+b)
Applied Stereotype	
Body	print(a+b)
Language	BeanShell <undefined>
Is Reentrant	☑ true
Specification	
Context	



Specifying a Scripting Language in the Specification dialog.

a. Select **Read Structural Feature Action** in the **Select Action MetaClass** dialog.



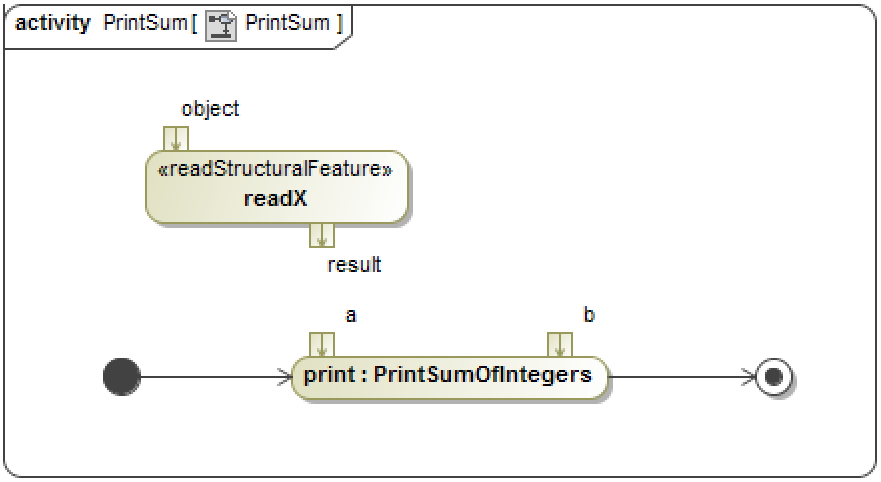
Select Action Metaclass

Search by Name

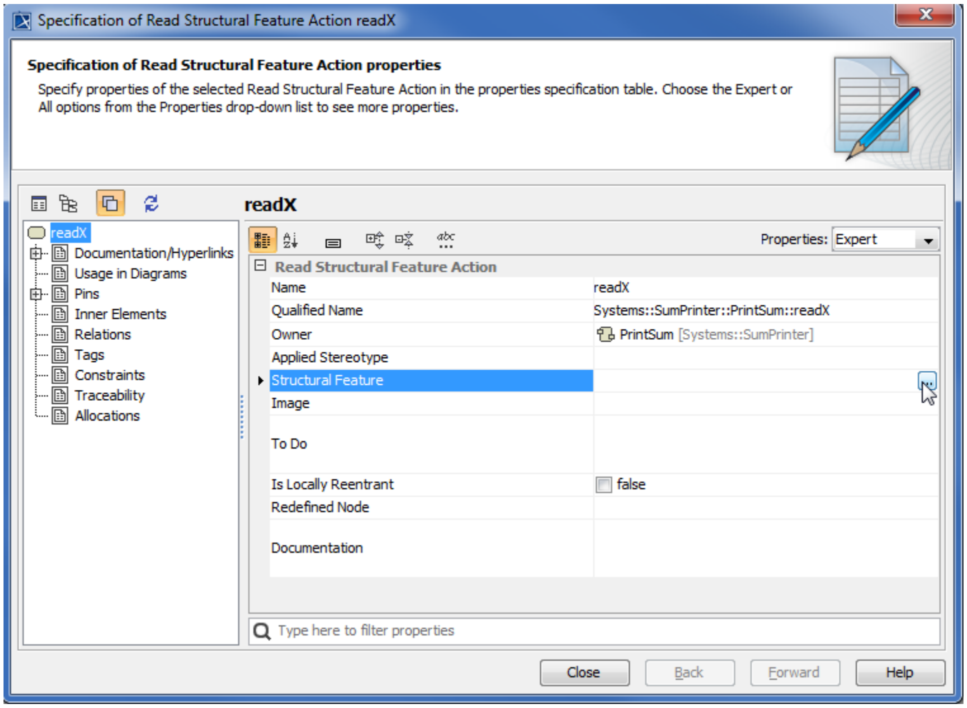
42 matches found

- Read Is Classified Object Action
- Read Link Action
- Read Link Object End Action
- Read Link Object End Qualifier Action
- Read Self Action
- Read Structural Feature Action
- Read Variable Action
- Reclassify Object Action
- Reduce Action
- Remove Structural Feature Value Action
- Remove Variable Value Action
- Reply Action
- Send Object Action

5. Click the **PrintSum** Activity diagram to create the Action and name it "readX" as shown in the following figure



6. Open the Specification window of the Action **readX**.



Specification of Read Structural Feature Action readX

Specification of Read Structural Feature Action properties

Specify properties of the selected Read Structural Feature Action in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

readX

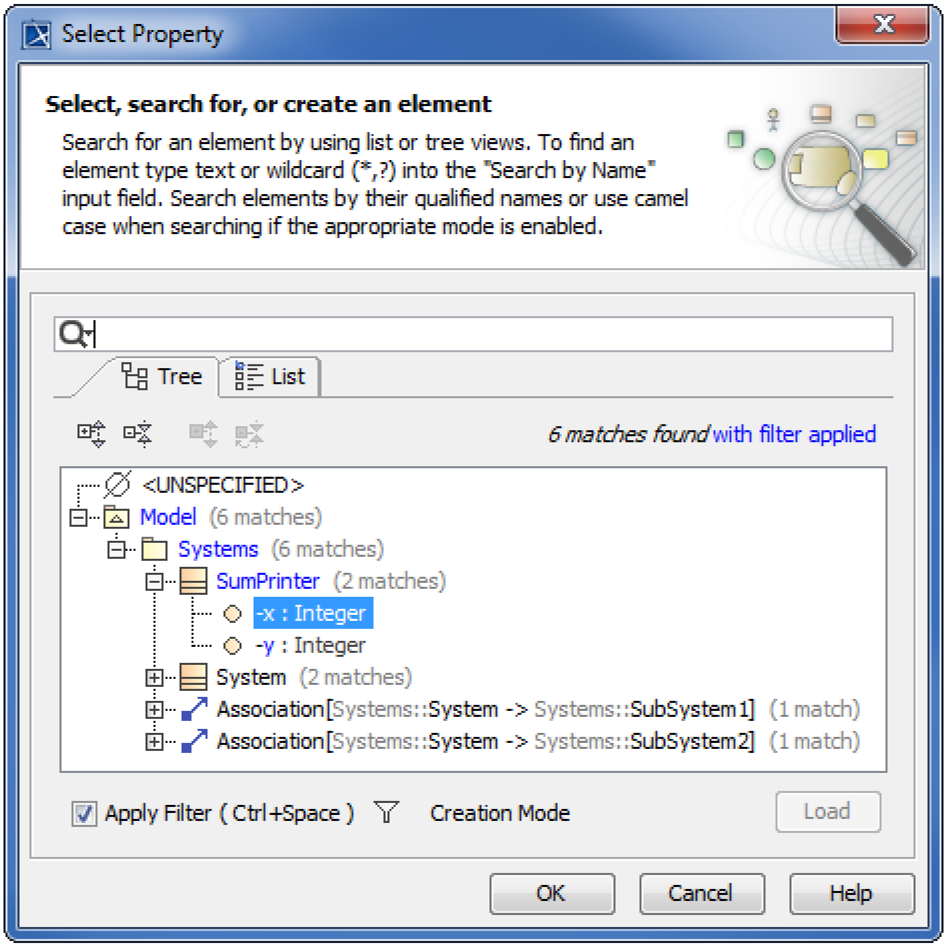
- readX
- Documentation/Hyperlinks
- Usage in Diagrams
- Pins
- Inner Elements
- Relations
- Tags
- Constraints
- Traceability
- Allocations

Properties: Expert

Read Structural Feature Action	
Name	readX
Qualified Name	Systems::SumPrinter::PrintSum::readX
Owner	PrintSum [Systems::SumPrinter]
Applied Stereotype	
Structural Feature	
Image	
To Do	
Is Locally Reentrant	□ false
Redefined Node	
Documentation	

Type here to filter properties

7. Click the **Structural Feature** and the … button to open the **Select Property** dialog to select **Structural Feature**.



Select Property

Select, search for, or create an element

Search for an element by using list or tree views. To find an element type text or wildcard (*,?) into the "Search by Name" input field. Search elements by their qualified names or use camel case when searching if the appropriate mode is enabled.

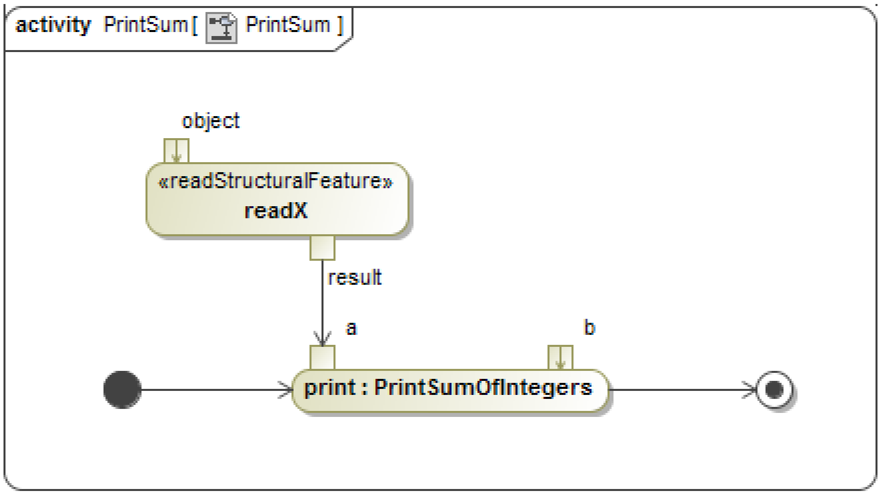
6 matches found with filter applied

- <UNSPECIFIED>
- Model (6 matches)
  - Systems (6 matches)
    - SumPrinter (2 matches)
      - -x : Integer
      - -y : Integer
    - System (2 matches)
    - Association[Systems::System -> Systems::SubSystem1] (1 match)
    - Association[Systems::System -> Systems::SubSystem2] (1 match)

☑ Apply Filter (Ctrl+Space) Creation Mode

8. Select the Property **x** of the **SumPrinter** Class and click **OK**. The **Select Property** dialog will close.

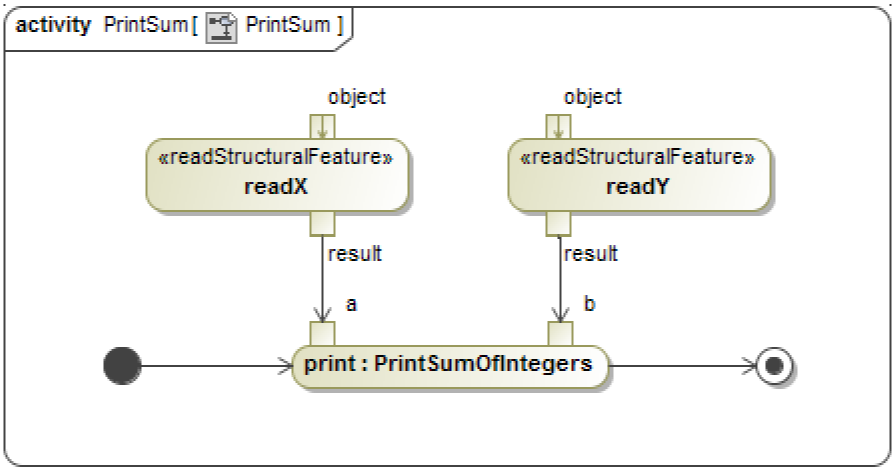
9. Create an object flow from the pin **result** of the **readX** Action to pin **a** of the **print** Action. The following figure shows the Flow between readX and print Actions in the Activity diagram.



10. Repeat Steps 4 to 10 to create another Action named **readY**, which is the **ReadStructuralFeatureAction**, with the following arrangements

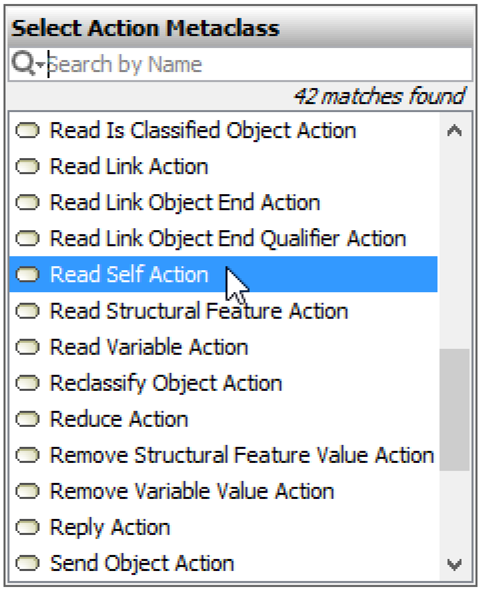
- The name of the Action is "readY".
- The structural feature is the attribute 'y' of the **SumPrinter** Class.
- The name of the output pin of **readY** is 'b'.
- The output pin **b** of **readY** connects to pin **b** of the **print** Action.

The following figure shows the Activity diagram with readX and readY Actions



To create a ReadSelfAction to read a runtime object that will be supplied to the input pins of **readX** and **readY** Actions

1. Click **Action** > **Any Action** on the Activity Diagram toolbar. The **Select Action MetaClass** dialog will open (see the following figure).



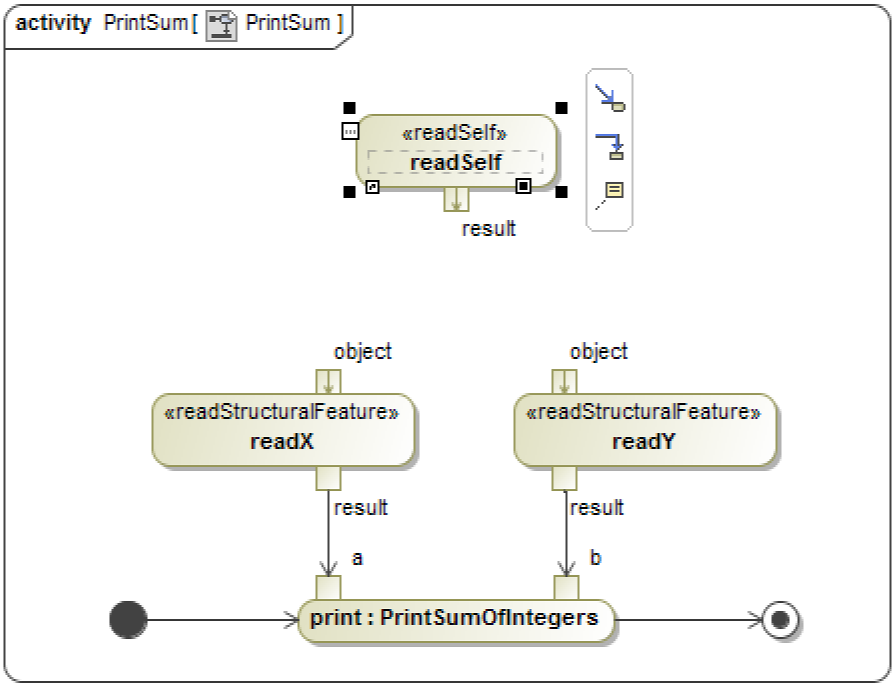
Select Action Metaclass

Search by Name

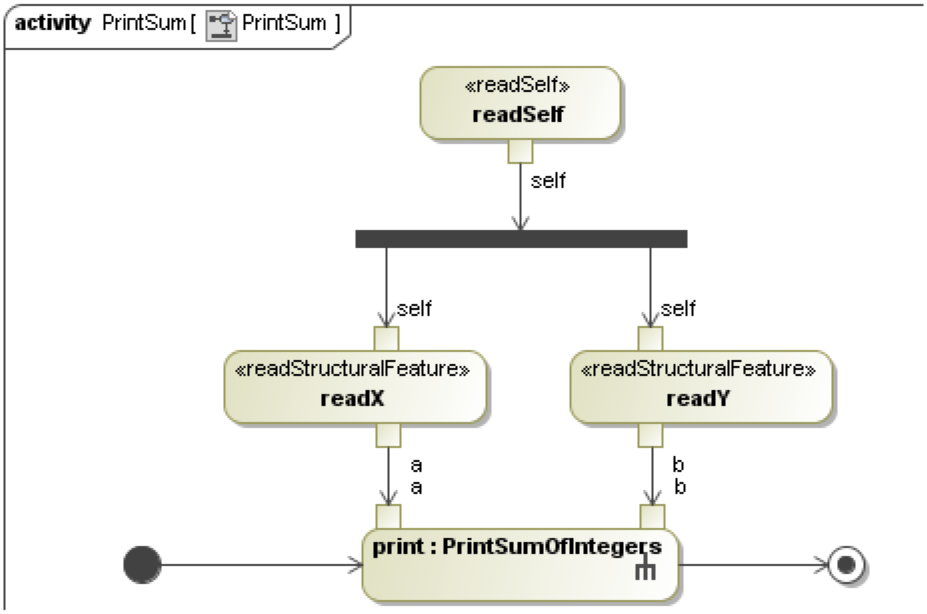
42 matches found

- Read Is Classified Object Action
- Read Link Action
- Read Link Object End Action
- Read Link Object End Qualifier Action
- Read Self Action
- Read Structural Feature Action
- Read Variable Action
- Reclassify Object Action
- Reduce Action
- Remove Structural Feature Value Action
- Remove Variable Value Action
- Reply Action
- Send Object Action

2. Select **Read SelfAction**.
3. Click the **PrintSum** Activity diagram to create an Action and name it, for example, **readSelf.** The following figure shows the Activity diagram with Action readSelf.



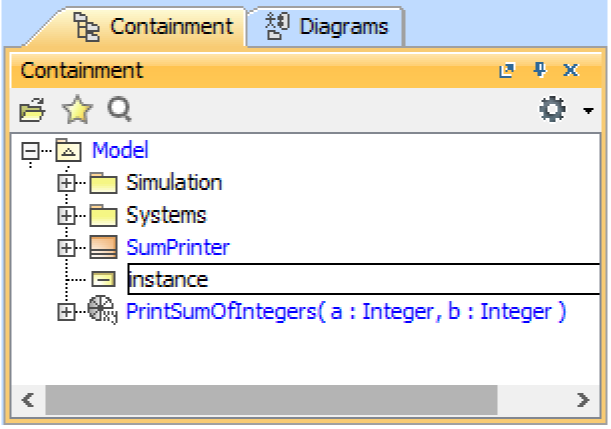
4. Create a **Fork Horizontal** and use **Object Flow** to connect it to the pins of the Actions **readX**, **readY**, and **readSelf** on the diagrams. The following figure shows a complete PrintSum Activity diagram.



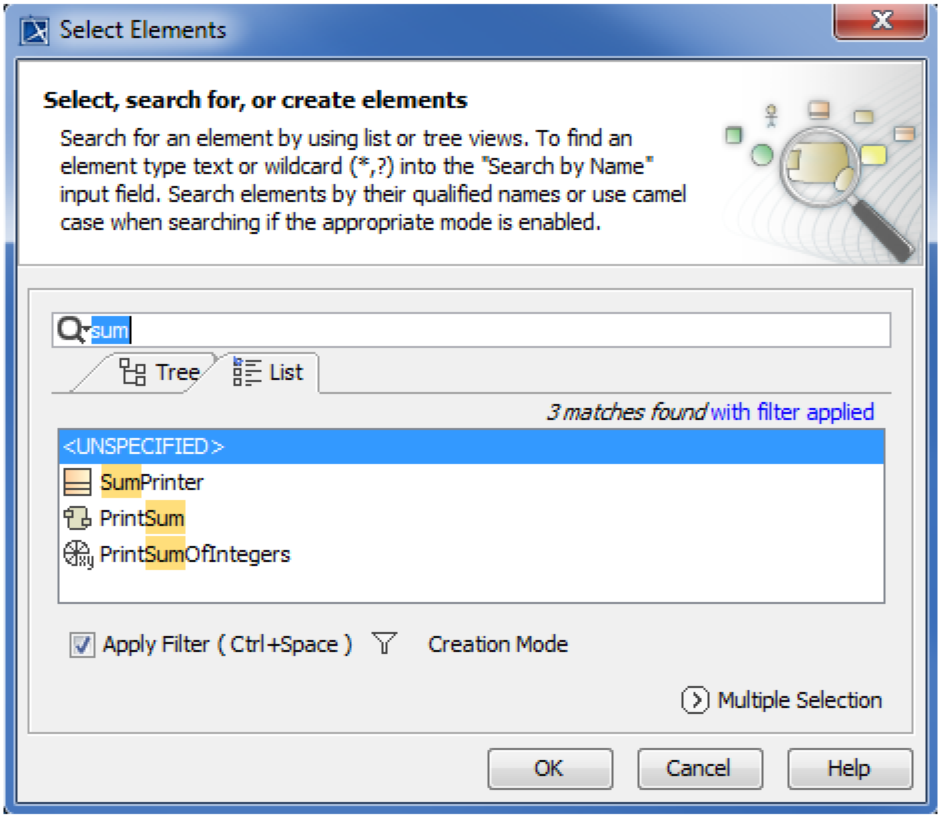
The final step is to create an **InstanceSpecification** whose Classifier is the **SumPrinter** and assign the values to the slots that correspond to the properties **x** and **y**. These values will be used during the simulation.

To create an **InstanceSpecification** whose Classifier is the **SumPrinter** and assign the values to the slots that correspond to the Properties **x** and **y**

1. Right-click the **Model** model and select **Create Element** > **InstanceSpecification**.
2. Name the created InstanceSpecification, e.g., **instance**.



3. Right-click the created **Instance** and open the Specification window of **instance**.
4. Click the Classifier field and the ... button. The **Select Elements** dialog will open.



Select Elements

Select, search for, or create elements

Search for an element by using list or tree views. To find an element type text or wildcard (*,?) into the "Search by Name" input field. Search elements by their qualified names or use camel case when searching if the appropriate mode is enabled.

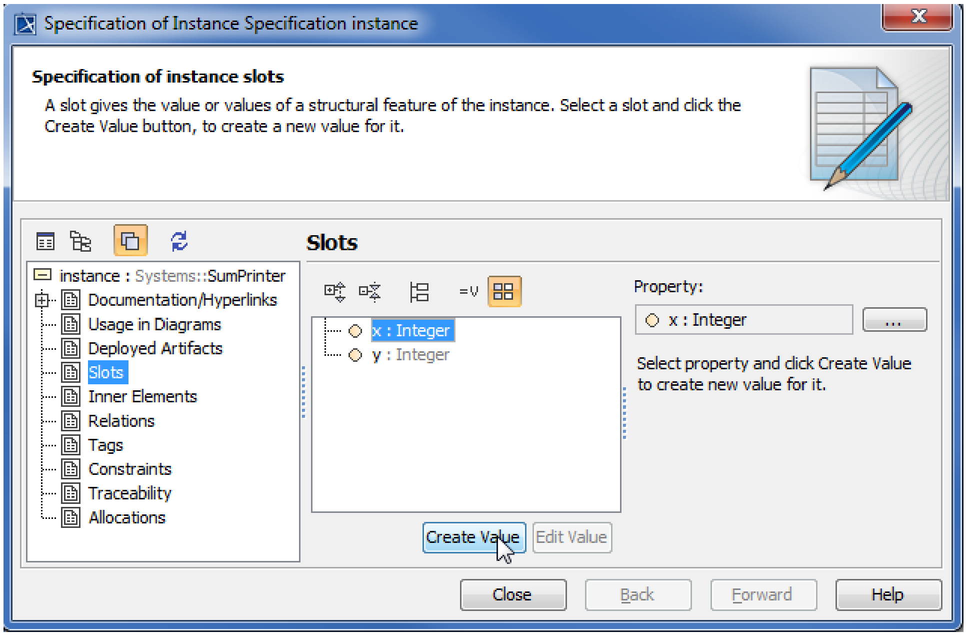
sum

3 matches found with filter applied

<UNSPECIFIED>
SumPrinter
PrintSum
PrintSumOfIntegers

☑ Apply Filter (Ctrl+Space)

5. Select the **SumPrinter** Class to edit the Classifier and click **OK**.
6. Click **Slots** on the left-hand side pane of the Specification window and select **x:Integer**.



Specification of Instance Specification instance

Specification of instance slots

A slot gives the value or values of a structural feature of the instance. Select a slot and click the Create Value button, to create a new value for it.

instance : Systems::SumPrinter
Documentation/Hyperlinks
Usage in Diagrams
Deployed Artifacts
Slots
Inner Elements
Relations
Tags
Constraints
Traceability
Allocations

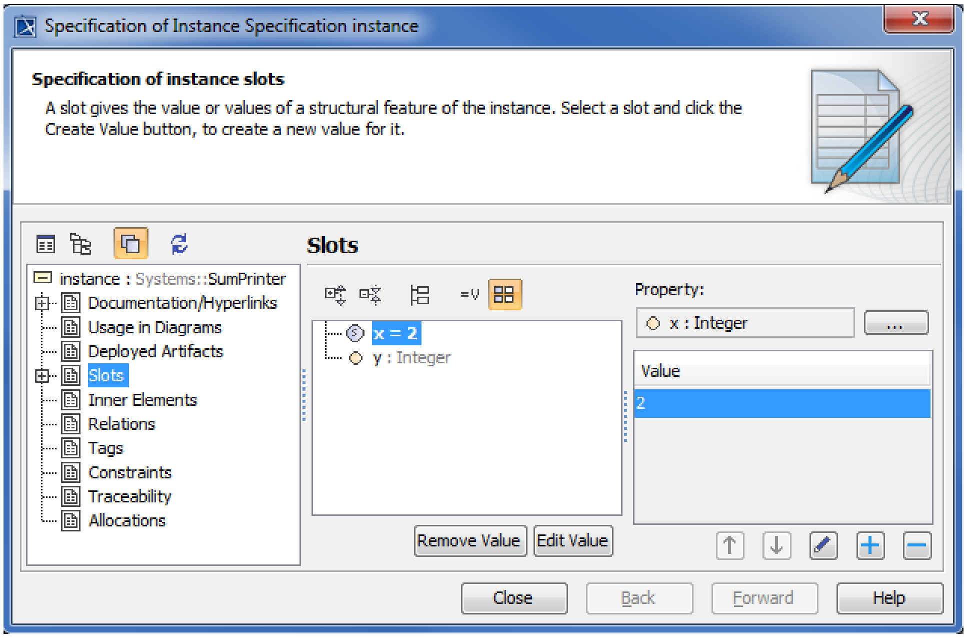
Slots

x : Integer
y : Integer

Property: x : Integer

Select property and click Create Value to create new value for it.

7. Click the **Create Value** button to create a new value of the slot. The **Value** box will open for you to assign the value to Property x slot.



Specification of Instance Specification instance

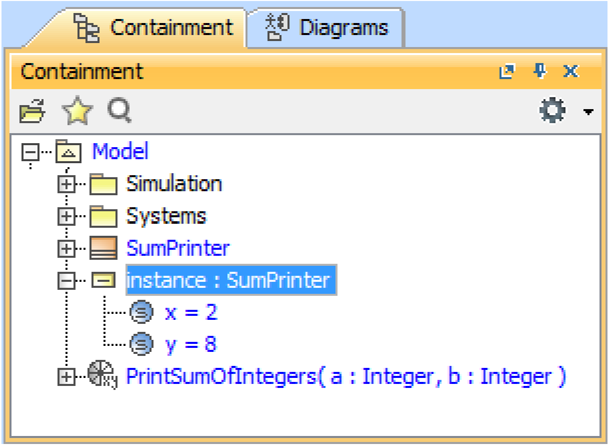
Specification of instance slots

A slot gives the value or values of a structural feature of the instance. Select a slot and click the Create Value button, to create a new value for it.

Property: x : Integer

Value
2

8. Type a number, e.g., 2 as the value of the property **x** slot.
9. Repeat Steps 6 to 8 to assign "8" as the value of the property **y** slot as shown in the following figure and click **Close**.



The model is now ready for you to simulate.

Related pages

- Activity
- Class
- Classifier Behavior property