

# Using Action Script

When creating class operations to your own data model in the **Implementation** folder, you can select from the following different types of implementation:

- **Action Script:** The usage of action script is explained on this page.
- **JavaScript:** For more Information about the usage of java script refer to [Using JavaScript](#).
- **Mapping Diagram:** Refer to [Modeling Data Mapping](#) for more information about mapping diagrams.
- **Activity Diagram:** You can implement class operations by a UML activity diagram. Refer to [Modeling Activities](#) for more information on activity diagrams.

The xUML Action Language is a script like language that you can use in class operations to implement atomic actions like manipulating strings, arrays, and so forth. Such class operations are called action script.



Refer to [Action Script Language](#) for a comprehensive documentation of all elements of the xUML Action Language.

## On this Page:

- [Creating an Operation Containing Action Script](#)
  - [Via a Quick Action](#)
  - [Via the Context Menu](#)
- [Attributes of an Action Script Operation](#)

## Related Pages:

- [Modeling Data Mapping](#)
- [Modeling Activities](#)
- [PAS Designer Developer Guide](#)
  - [Action Script Language](#)

## Creating an Operation Containing Action Script

### Via a Quick Action

The fastest way to create an action script operation is via the quick actions of the related class.

Search Criteria

Assets

Base Types

Connectors

Process

Forms

API

Implementation

Forms

Idea Check

Idea

Libraries

approved: Boolean

employeeName: String

ideaDescription: String

instructionsText: String

personnelNumber: String

mapFromIdeaCheck

mapFromIdeaInput

mapFromInstructionsInput

mapToInstanceList

MappingOperations

Hover over the class you want to add the operation to, and click the **Add Action Script Operation** quick action (AS).

# Using Action Script

When creating class operations to your own data model in the **Implementation** folder, you can select from the following different types of implementation:

- **Action Script:** The usage of action script is explained on this page.
- **JavaScript:** For more Information about the usage of java script refer to [Using JavaScript](#).
- **Mapping Diagram:** Refer to [Modeling Data Mapping](#) for more information about mapping diagrams.
- **Activity Diagram:** You can implement class operations by a UML activity diagram. Refer to [Modeling Activities](#) for more information on activity diagrams.

The xUML Action Language is a script like language that you can use in class operations to implement atomic actions like manipulating strings, arrays, and so forth. Such class operations are called action script.

Refer to [Action Script Language](#) for a comprehensive documentation of all elements of the xUML Action Language.

## On this Page:

- [Creating an Operation Containing Action Script](#)
  - [Via a Quick Action](#)
  - [Via the Context Menu](#)
- [Attributes of an Action Script Operation](#)

## Related Pages:

- [Modeling Data Mapping](#)
- [Modeling Activities](#)
- [PAS Designer Developer Guide](#)
  - [Action Script Language](#)

## Creating an Operation Containing Action Script

### Via a Quick Action

The fastest way to create an action script operation is via the quick actions of the related class.

Search Criteria

Assets

Base Types

Connectors

Process

Forms

API

Implementation

Forms

Idea Check

Idea

Libraries

approved: Boolean

employeeName: String

ideaDescription: String

instructionsText: String

personnelNumber: String

mapFromIdeaCheck

mapFromIdeaInput

mapFromInstructionsInput

mapToInstanceList

MappingOperations

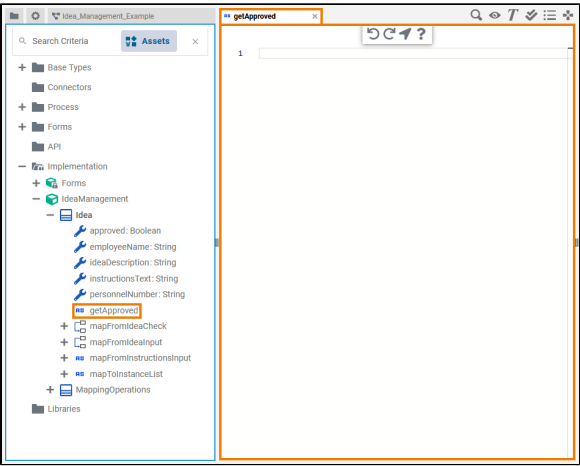
Hover over the class you want to add the operation to, and click the **Add Action Script Operation** quick action (AS).

# Add Action Script Operation

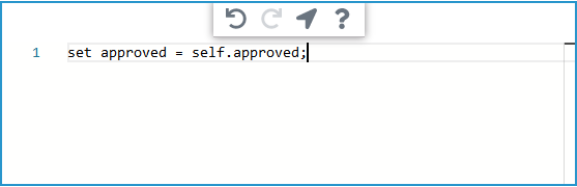
getApproved

Save Cancel


Assign a name to your new action script and click **Save**.



The Action Script Editor opens in a new tab.



You can now add action script.

 **Expert Advice**

Click  
the  
help  
icon (



)  
to  
jump  
to the  
docum  
entatio  
n of  
the [xU  
ML  
Action  
Langu  
age](#)  
for  
detaile  
d  
inform  
ation  
about  
the  
usage  
of  
action  
script.  
This  
docum  
entatio  
n  
contai  
ns  
helpful  
basic  
inform  
ation  
such  
as

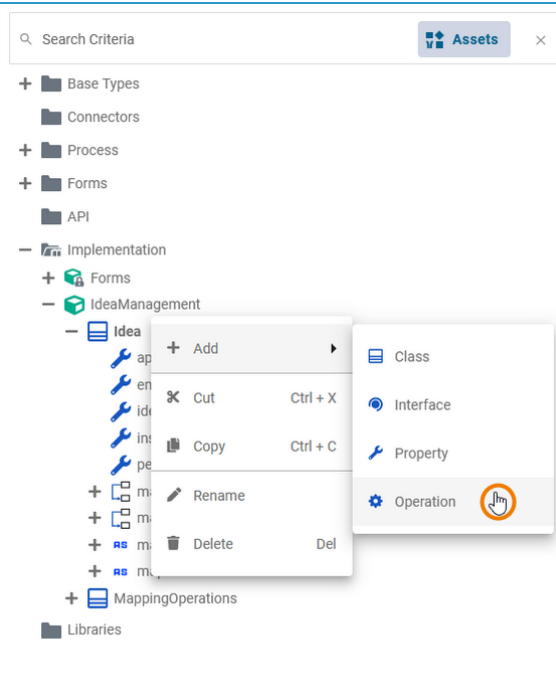
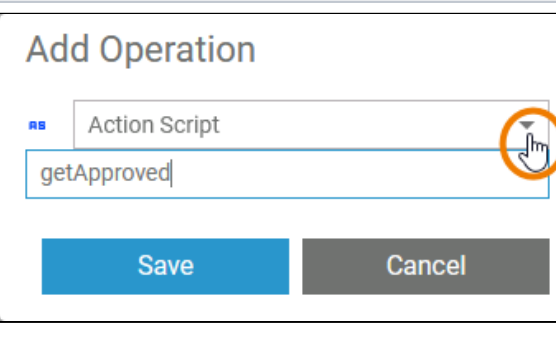
- [b  
a  
s  
i  
c  
s  
o  
f  
t  
h  
e  
x  
U  
M  
L  
A  
c  
t  
i  
o  
n  
L  
a  
n  
g  
u  
a  
g  
e](#)

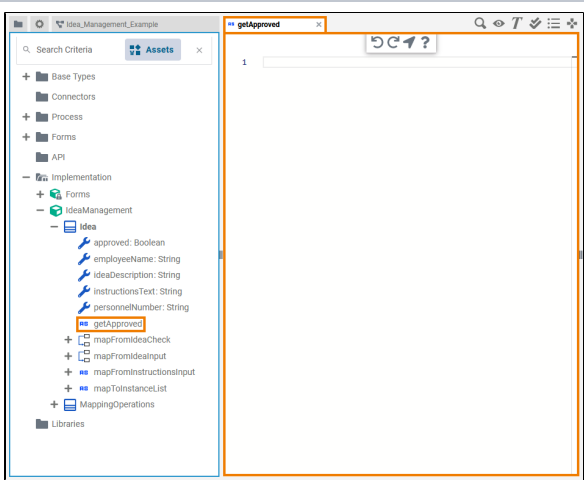
- syntax scheme of the UML Action Language
- are reference for all kinds of operations

The script you have entered is saved and validated in the background.

## Via the Context Menu

Alternatively, you can add a class operation via the context menu of a class, and create an implementation afterwards.

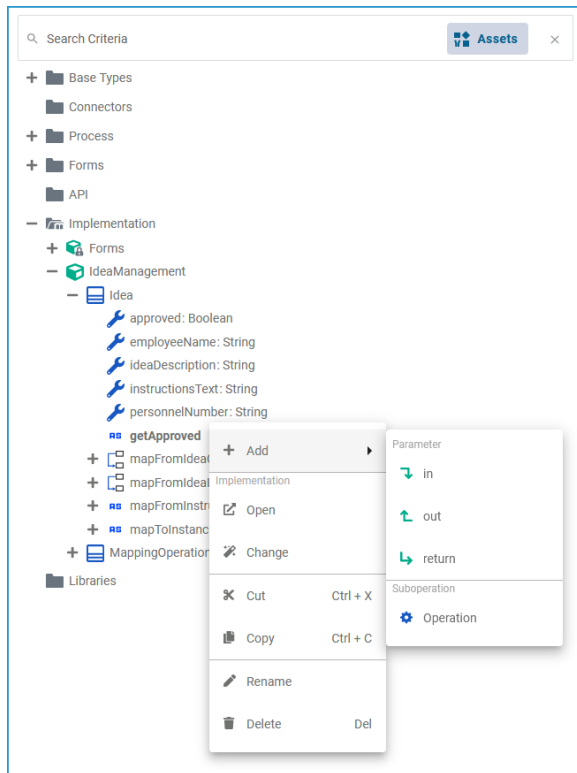
	<p>Right-click the class you want to add an operation to and select <b>Add Operation</b> from the context menu.</p>
	<p>The dialog <b>Add Operation</b> opens.</p> <p>Select <b>Action Script</b> from the drop-down list, enter a name for the operation and click <b>Save</b>.</p>



The new operation has been added to the class. The Action Script Editor opens automatically in a new Designer tab, and you can start entering action script (see [further above](#)).

Go to Working with the Action Script Editor for detailed information.

Once the operation has been created, you can use the quick actions and the context menu to manage it. You can:



- add a parameter
  - in
  - out
  - return
- add a suboperation
- open the implementation of the action script
- change the implementation from action script to activity diagram or mapping diagram
- cut the action script operation
- copy the action script operation
- paste the action script operation (available if **Copy** or **Cut** option have been used before)
- rename the action script operation
- delete the action script operation

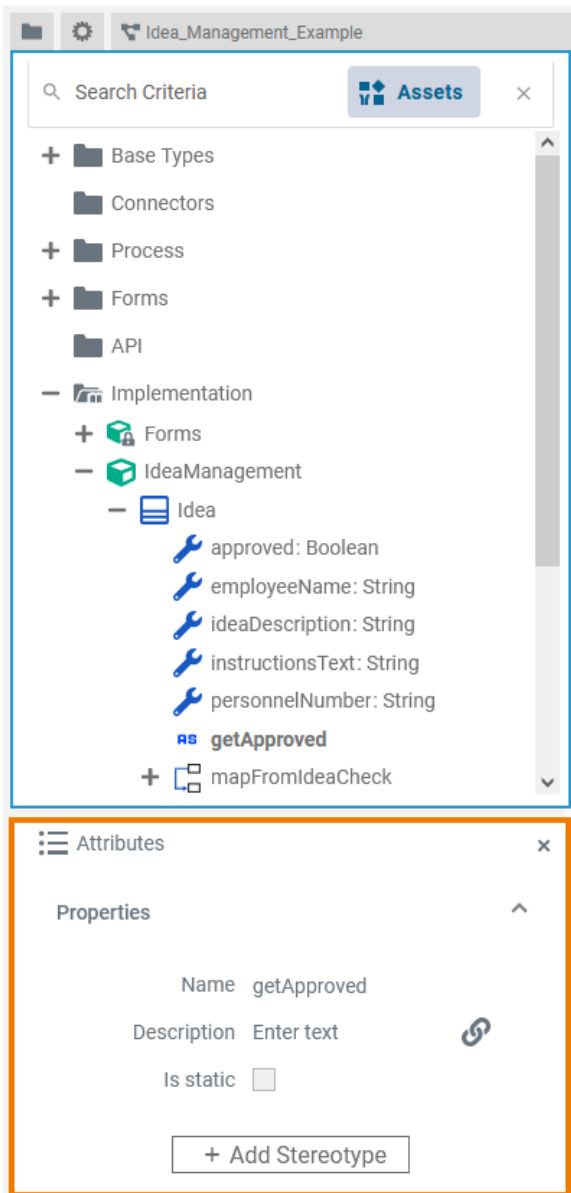
✓ Refer to Implementation and Modeling Data Structures for more information on your options here.

## Attributes of an Action Script Operation

Select an action script operation in the **Implementation** folder of the Service panel to display its attributes in the **Attributes** panel. You can also edit them there.

Action script operations have the following attributes:





Attribute	Description	Possible Values / Example	
<b>Name</b>	<p>Click here to change the <b>Name</b> of the related element.</p> <p>Action script operation names must follow certain naming rules. They</p> <ul style="list-style-type: none"><li>• must not contain blanks</li><li>• must not start with a number</li><li>• must not contain special characters</li></ul>	getApproved	
<b>Description</b>	<p>If you want to insert or change a description for the respective action script operation, click here to open a text editor where you can enter and format your text.</p>		
<b>Is static</b>	<p>Specify if the operation is static (default) or not.</p> <ul style="list-style-type: none"><li>• <b>Static</b> action script operations can be called without creating an instance of the related class. They get all necessary data via their input parameters.</li><li>• Wanting to call a <b>non-static</b> action script operation, you need to create a local instance of the related class, and call the operation on that object. This is called <b>self context</b>.</li></ul> <p>For more information, also refer to <a href="#">Adding Operation Calls</a>.</p>	<b>true</b>	The action script operation is static (default) and can be used outside the context of the related class.
		<b>false</b>	The action script operation is non-static and needs a self object as an input.
<b>Stereotype</b>	<p>Via <b>Add Stereotype</b>, you can add a stereotype to a action script operation. By adding a stereotype, you can extend the attributes of a action script operation with additional properties.</p>	<b>REST</b>	

⋮ Attributes

×

Properties

^

Name

getApproved

Type

IdeaManagement.Id...

When you click in the Action Script Editor, the following attributes of the current action script are displayed in the **Attributes** panel. All attributes are read-only and cannot be edited there.

Attribute	Description	Example
Name	Displays the name of the current action script.	getApproved
Type	Path within the implementation folder where the corresponding action script operation resides.	IdeaManagement.Idea