


# Input and Output of Activities

In other programming languages, what you call a "parameter" is drawn as an activity parameter node in UML.

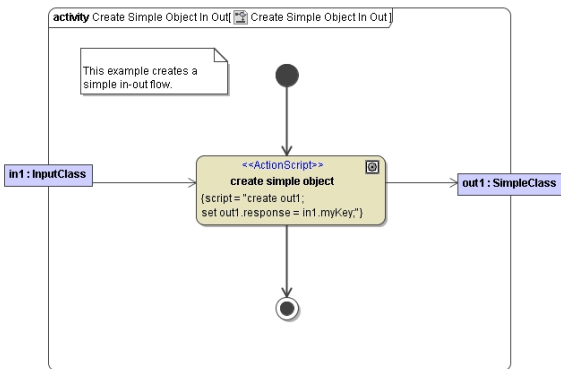
### Example File (Builder project Basic Modeling/Flows):



<your example path>\Basic Modeling\Flows\uml\basicFlows.xml

The Bridge allows you to define multiple output parameters for a single request. This is very useful as it allows you to return an arbitrary number of data items, even if they have different classes. For example, a zoo Service could have an operation that returns all animals in the zoo in an array and an integer stating the total weight of all animals.

Figure: Input and Output Activity Parameter Nodes




Input and output parameters of an operation must be modeled as parameter nodes in your diagrams.

## Transformations

Frequently, just drawing an input or output transition from an object node to an action node specifies the input and output of an activity or an action. However, if the calling object node names are different to the called object node names one has to map object flow state names. Usage of explicit parameter mapping is often required when re-using requests of activities or when applying an adapter.

In this case we use pins or the stereotype `<<Transformation>>` to define the mapping.

### Example File (Builder project Basic Modeling/Data):



<your example path>\Basic Modeling\Data\uml\transformations.xml

## Transformation Using Pins

Pins are placed directly on the action node. There are three different kinds of pins:

Type of Pin	Description
Input Pins	An input pin is a pin that holds input values to be consumed by an action node. Any type is valid. The pin deduces the type from the calling object node and must not be specified.

### On this Page:

- Transformations
  - Transformation Using Pins
  - Transformation Using the Stereotype

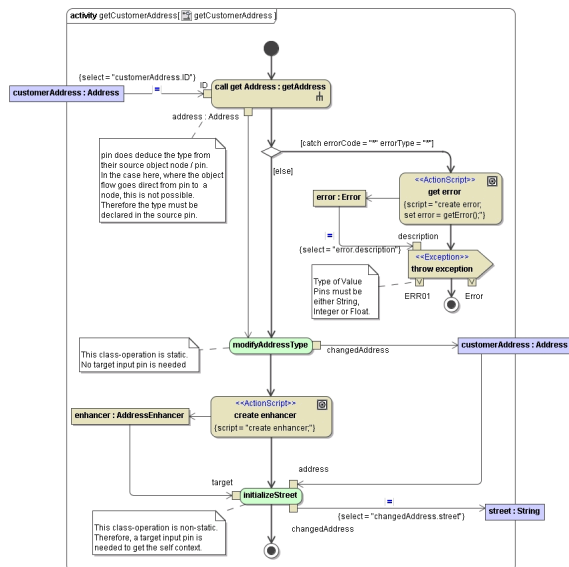
<b>Output Pins</b>	An output pin is a pin that holds output values produced by an action node. Any type is valid. The pin deduces the type from the calling object node and must not be specified.
<b>Value Pins</b>	A value pin is an input pin that provides a value to an action that does not come from an incoming object flow edge. The value is set statically on the pin. Type must be either String, Integer or Float. The type must be specified.

See in the figure below different possibilities how to apply pins.

## Transformation Using the Stereotype

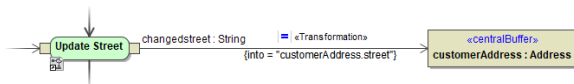
The **<<Transformation>>** stereotype is applied directly on the object flow. It extracts the value of a single attribute from a complex type object node respectively output pin, and transports the value to an input pin respectively an object node.

See in the figure below different possibilities how to apply transformations on an object flow.



The source of the transformation is specified via tag **select**. By selecting the transformation flow and pressing **Ctrl+Enter**, you can open an Action Script window that will help you with specifying the transformation source.

If the target of the transformation is a complex type, you can specify a target attribute in the **into** tag, e.g. `customerAddress.street`. Multiple transformational object flows to the same target object are allowed.



However, due to technical limitations, multiple transformation flows to the same input **pin** are not allowed.

If you use multiple transformations to the same complex target, make sure that the target properties are unique for each transformation.



If two or more flows do change the same property, even indirectly, e.g. like `customer.address` and `customer.address.city`, the behavior is explicitly undefined.