

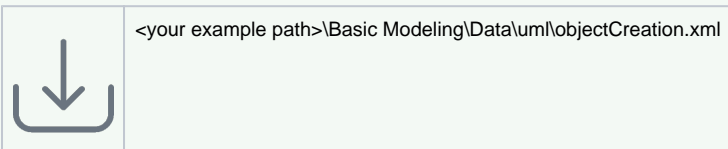
## Creating Objects of Base and Complex Types

The most basic feature of each language is creating instances of classes and setting values. With the `set` assignment statement, it is possible to create objects of base types like strings, integers, booleans, etc. whereas objects of complex types need to be instantiated (created) first with the `create` statement. If you define an output object node having a complex type (opposed to base types like **String**, **Integer**, etc.), it needs to be created first. In object-oriented terminology: class objects need to be instantiated (that is creating an instance of the corresponding class). Once the instance of the class (the object) has been created, it can store data in its attributes.

## Creating Base Types with the set Assignment Statement

The following examples shows how to use the set assignment statement to create base type objects.

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

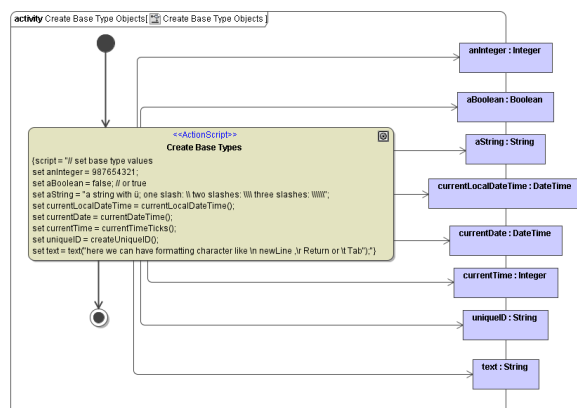


<b>Syntax</b>	<pre>set anObject = aValue;</pre>	
<b>Semantics</b>	Assigns a value to anObject.	
<b>Substitutables</b>	anObject	Can be an object node, an attribute, or an association end.
	aValue	Can be a literal, a base type object node, or an EAL operation or expression returning a base type.
<b>Examples</b>	<pre>set aString = "Hello World!"; set anInteger = 12345; set currentDate = currentDate();</pre>	

The example in the figure below shows how to create base type objects like strings, integers, etc. Base type objects need not to be instantiated, they are created by using the `set` assignment statement to directly assign a value to the variable.

All variables are drawn as object nodes with all objects being of base type (see also [Base Types](#)).

Figure: Creating Base Type Objects



### On this Page:

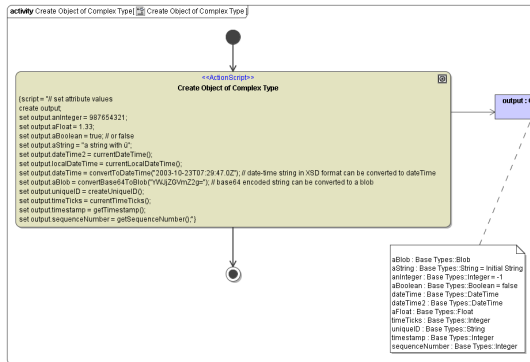
- Creating Base Types with the set Assignment Statement
- Creating Arrays by Appending Items
- Creating Objects of Complex Type

**Related Pages:**

- Base Types
- The Essential Thing: Object References

Attributes of an object are also assigned values by using the `set` assignment statement as shown in the figure below.

Figure: Setting Attribute Values



## Creating Arrays by Appending Items

You can create arrays using the `create` statement (see action script example below):

```

create anArray;
append "Hello World!" to anArray;

```

However, most of the time the E2E Runtime will create the array implicitly on appending the first item. There is one exception to this rule, though: Arrays that contain array elements having a complex type with multiplicity.

Let's assume you have an array of complex type **ArrayElement** and this complex type has a property `subArray` with multiplicity 0..\*.

- What you can do, if `subArray` is NULL:

```
set array1[0].subArray = anotherArray;
```

The reference `subArray` is changed to point to `anotherArray`.

- What you can't do, if `subArray` is NULL:

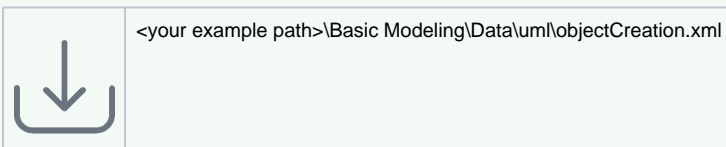
```
append "something" to array1[0].subArray;
```

In this case (get statement on the right side of a statement), the Runtime will throw a get error for `array1[0].subArray`.

## Creating Objects of Complex Type

The following example shows how to use the `create` statement to create objects of complex type.

Example File (Builder project Basic Modeling/Data):



Syntax

```
create anObject;
```

<b>Semantics</b>	Creates an object of complex type. The object reference is stored in <code>anObject</code> . Initial values defined on the class attributes will be set.	
<b>Substitutables</b>	<code>anObject</code>	Can be any valid object name.
<b>Examples</b>	<pre>create simpleObject;</pre>	

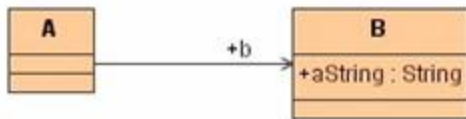
In the following cases, objects need to be created with the `create` statement:

- Creating an object (creating an instance of a class).
- Result objects of an iteration over an action script need to be created (compare section [Iterate Over Action Scripts](#)).

In the following cases, the `create` statement is not necessary:

- Adapters may create objects of any type. Those objects need not to be created explicitly (for instance, an object that is created after an SQL query executed by the SQL adapter).
- When having related classes, the instantiation of intermediate objects is not mandatory. In the example below, class **A** has an attribute **b**, which is of type **B** (see association). This class in turn has a string attribute named **aString**.

Figure: Creating Instances of Related Classes



In order to instantiate all classes and setting the string **aString**, not all class instances need to be created. Only the top-level class needs to be instantiated as shown in the following action script:

```
create objectOfClassA;
set objectOfClassA.b.aString ="Hello World!";
```

Scalar base type objects are never created using the `create` statement.