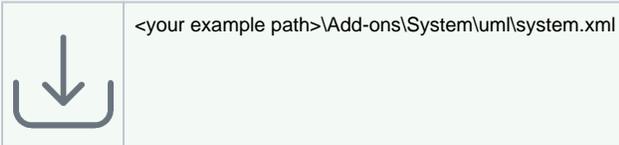


System Adapter

The system adapter can be used for rapid prototyping by enabling the reuse of existing programs (e.g. batch jobs). However, this adapter must be used carefully because the E2E Bridge does not impose any constraints on the executed program. Additionally, the Bridge performance degrades, if the system adapter gets called frequently because each call creates new processes on the host. We recommend using the system adapter for purchased executable software and modules or existing complex executable modules which would be too expensive to re-build with UML.

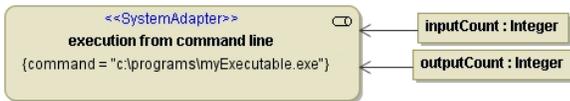
Example File (Builder projectAdd-ons/System):



The system adapter is marked with the stereotype `<<SystemAdapter>>` on the action node. The executable program has to be defined in the tagged value **command** of the action node. There are two mandatory parameters of the `<<SystemAdapter>>` which must be named as follows: **inputCount** and **outputCount** (both of type **Integer**). If the system command requires no parameters, the value has to be set to **0**.

If the program to be executed can be run without parameters, the call is very simple, see the figure below.

Figure: Execution of an External Program without Parameters



If the program has input parameters and output values, which have to be handled by the Bridge, the following objects have to be used:



The name and type must be defined in exactly that way, so that the Bridge can pass the input parameter to the calling executable and map the output of the program to UML classes. The type must be an array of blobs for both objects. The reason for using the type **Blob** is to have the flexibility to have any kind of input (Strings, binaries like used in images, etc.). For example, if the parameter is a **String** value, it has to be transcoded to a **Blob** type first (see [transcodeToBlob\(\) Operation](#)). As it has to be possible to provide more than one input parameter and more than one output value (e.g. one for the value and one for the error message) arrays are used and not simple blobs.

The Bridge expects the input parameters and output values to be files. If the executable does not support a file interface, the whole system command must be prepared by Bridge operations (see [Figure: System call with calculated command line](#)).

UML Object Overview

The following UML objects can be used in combination with the system adapter:

Object	Description	Valid Values	Mandatory
input : Array {arrayElement = Blob}	Input parameter(s) for the program to be executed. Will be passed to the calling executable as input files. The file(s) will be generated by the Bridge and will be automatically deleted after execution.	base64 strings	
output : Array {arrayElement = Blob}	Output value(s) sent back from the executed program. The Bridge expects output files from the called executable. The file(s) will be generated by the Bridge and will be automatically deleted after execution.	base64 strings	

On this Page:

- [UML Object Overview](#)

Related Pages:

- [Invoking a Perl Script by the Use of the System Adapter](#)
- [Executing a Shell Command by the Use of the System Adapter](#)
- [Executing a Calculated Command Line by the Use of the System Adapter](#)

inputCount : Integer	Defines the number of input parameters.	0-n	X
outputCount : Integer	Defines the number of output parameters.	0-n	X
command : String	A string containing the calling statement of the external program as it would be used from the command line. The statement may include the path e.g. D:\programs\myExecutable.bat .	String value	