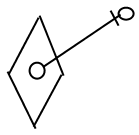
|  |
| --- |
| Circle Language Spec: System Objects |

## System Command Call with Argument Notation

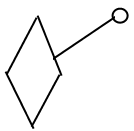
*Basic System Command Calls* showed the basic system command call notation, but most system command calls are incomplete without an argument.

System commands can use a simplified argument notation.

Explicit notation of an argument of a command:



can be simplified for system commands:



The simplified notation is only used for system commands, because system commands are so fundamental that they need a notation that is as simple as possible.

The general use of this notation will be explained with Object Get as an example. Object Get will probably never use this notation, but it is easier to explain this way.

Here is the basic notation of an Object Get system command:

|  |
| --- |
| Object Get |
|  |

The Object Get command has a return value: the object. It is an *output* argument. It is not displayed in the picture above, but it should be there to transfer the Object aspect to another symbol. This is drawn out as follows, using the simplified argument notation for system commands:

|  |
| --- |
| Object Get |
|  |

In the picture above the output argument of the Object Get call is filled in.

Object Get is called upon the smaller circle contained by the larger circle.

But next, an Object Set is *implicitly* called upon the smaller circle in the corner.

The diagram above implictly causes an Object Set to be executed upon the smaller circle in the corner.

Something similar happens in case of an Object Set command. Here is the basic notation of an Object Set command call:

|  |
| --- |
| Object Set |
|  |

And here the argument of the Object Set command is filled in:

|  |
| --- |
| Object Set |
|  |

This causes an implicit Object Get on the smaller circle in the top-right corner, followed by the explicit Object Set, targeting the smaller circle contained by the larger circle.

So the basic display of a basic system command call with an argument filled in, implictly causes another system command to be called as well.

Below you will find an overview of all the system commands with their argument filled in using the simplified argument notation for system commands. Which system commands are implicitly called, is also made clear in the overview.

|  |  |  |
| --- | --- | --- |
| Object Get | Object Set |  |
|  |  |  |
| Object Get 🡨  (Object Set 🡪) | (Object Get 🡪)  Object Set 🡨 |  |

Yielding over Class aspect:

|  |  |  |
| --- | --- | --- |
| Use As Class | Use Reference As Class | Class Set |
|  |  |  |
| Use As Class 🡨 (Class Set 🡪) | Use Reference As Class 🡨  (Class Set 🡪) | (Class Get 🡪)  Class Set 🡨 |
|  |  |  |
| Object-Class Get | Reference-Class Get |  |
|  |  |  |
| Object-Class Get 🡨  (Class Set 🡪) | Reference-Class Get 🡨  (Class Set 🡪) |  |

Yielding over between Class aspect and Object aspect:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class Set | Reference-Class Get | Object-Class Get |  | |
|  |  |  |  | |
| (Use As Class 🡪)  Class Set 🡨 | Reference Class Get 🡨  (Object Set 🡪) | Object-Class Get 🡨  (Object Set 🡪) |  | |
|  |  |  |  | |
|  |  |  |  | |
| Value Get | Value Set |  | |
|  |  |  | |
| Value Get 🡨  (Value Set 🡪) | (Value Get 🡪)  Value Set 🡨 |  | |
|  |  |  | |
| Clone (2) Get | Clone (2) Set |  | |
|  |  |  | |
| Clone (2) Get 🡨  (Clone (2) Set 🡪) | (Clone (2) Get 🡪)  Clone (2) Set 🡨 |  | |
|  |  |  | |
|  |  |  | |
| New |  |  | |
|  |  |  | |
| (Use As Class 🡪)  New 🡨 |  |  | |
|  |  |  | |
|  |  |  | |
| Add |  |  | |
|  |  |  | |
| (Object Get 🡪)  Add 🡨 |  |  | |

Most of the notations above will never be used. Only the notation for the New and Add arguments are common.

System command calls to Get, Set or Use are often not visible, because they are usually implied by assignment calls. But for alternative types of assignment, that do not have a standard assignment notation, you can explicitly display Get, Set and Use calls. Only when a simpler notation does not suffice, a Get, Use or Set command are explicitly displayed. This basically only happens in unusual cross-aspect assignments. In abstract diagram expression though, you can always use the explicit notations to simply make a point. When you look at the aspects introduced so far, the only time you might need the ‘system command with argument’ notation for Gets and Sets, is in a Class Get – Class Set assignment. This creates the following notations of the different ways to use a class:

|  |
| --- |
| Class Assignment |
|  |
| Use As Class 🡨  Class Set 🡪 |
|  |
| Class to Object Assignment |
|  |
| Class Get 🡨  Object Set 🡪 |
|  |
| Class Get – Class Set Assignment: |
|  |
|  |
| Class Get 🡨  (Class Set 🡪) |
|  |
|  |
| (Class Get 🡨)  Class Set 🡪 |

The Class Get notation and the Class Set notation both have the same effect, only in one case it is the Get that is explicitly displayed and in the other case it is the Set that is explicitly displayed. It is not yet clear which is the best one.