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

## Simplified System Command Call Notation

System command calls are so fundamental, that they get the most basic notation that still fits within the language, even though it can get a more complex notation when fully drawn out.

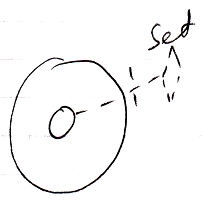
There are system commands that apply to object references and to the object itself. System commands, that apply to an object reference are displayed here with a parent around the targeted reference:



System commands, that apply to the object itself are displayed without a parent around the targeted object.



Here is an example of the simplified system command call notation:



The line type indicates the aspect.

A word like Get, Set or Use indicates assignment direction.

The access mark indicates the object of which the system command is called.

Leaving out the access mark may make the notation even simpler. The access mark may not be required in this particular notation, but other notations are later derived from this notation which require the access mark for disambiguation.

Here follow the basic notations for all the system commands.

|  |  |
| --- | --- |
| Object Get | Object Set |
|  |  |
|  |  |
|  |  |
| Use As Class | Use Reference As Class |
|  |  |
|  |  |
| Class Set |  |
|  |  |
|  |  |
| Object-Class Get | Reference-Class Get |
|  |  |
|  |  |
|  |  |
| Value Get | Value Set |
|  |  |
|  |  |
| Clone (2) Get | Clone (2) Set |
|  |  |
|  |  |
|  |  |
| New | Annul |
|  |  |
|  |  |
| Execute |  |
|  |  |
|  |  |
|  |  |
| Add | Remove |
|  |  |

Usually a Get command or Use command is immediately followed by a Set command. But in the overview above you see isolated Get, Use and Set calls. You will probably not see this happen in actual programs, but the notation above is used later as a base for other notations.