|  |
| --- |
| Circle Language Spec: Commands |

## Command Reference

Concept

A command reference is a command, that *redirects* to another command object. This is done with *object* redirection.

A command reference is usually inactive. But a command reference can also be active.

The handy thing about command reference, is that it makes you able keep the operation to execute variable. The target of the command reference is variable. When you *call* a command reference, then the target of the command reference determines which command is called. So calling a command reference means calling a variable command definition.

A command reference represents the same object as the object pointed to. So the command reference’s contents are the exact same objects as that of the command object it points to.

A command reference can also redirect to yet another command reference, creating multiple command object redirections. The target of the last command reference determines the definition of the first command reference.

Both the command reference and its target can be either active or inactive.

An *active* reference to an *inactive* command can *not* be executed. The final target of object redirections is the object itself, and when it is inactive, the command object can *not* be executed, but only function as a prototype.

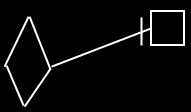
You can not execute an *active* command object through an *inactive* command reference either. But an active reference to an inactive reference to an active command *can* be executed.

Diagram Notation

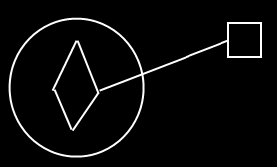
A command reference is displayed in a diagram as a diamond or square connected to another diamond or square with a solid line.



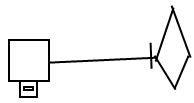
It does not matter whether it is squares or diamonds, because the only difference between a square and a diamond, is that a square can *not* be executed and a diamond *can*.



Direction of the line must be indicated with an access mark, unless the line is going outwards:



You can not execute an active command object through an inactive command reference.



But with an executable reference to an inactive reference to an executable command you *can* execute the command object again.

