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

## Inactive Command Class Redirection

Inactive command class redirection is less common than inactive command object redirection. Inactive command *object* redirection is a reference to a command definition. Inactive command *class* redirection creates a new command definition for which another command is the prototype.

Inactive command *class* redirection makes a command definition take over all aspects of another command definition, but creates a *new* command definition object, for which the other command definition is the prototype. The new command definition can only redefine the other definition’s *default values*. That’s all that is additionally possible, compared to inactive command *object* redirection.

Inactive command class redirection is analogus to having a class redirect its class, making one object the prototype for another prototype.

An inactive command class redirection can also point to an active command, making an *executable* command object the prototype for the new command definition. That is even less common, but possible all the same.

Do remember, that the new prototype can *not* be executed by pointing to it with an active command reference, because the new prototype is a new *inactive* command object, and an inactive command object can not be executed.

### In a Diagram



Inactive command symbol A is a square because it is an *inactive* command. Inactive command symbol A redirects its class to the inactive command symbol B. Symbol B is also a square because it is also an *inactive* command. The redirection from command A to command B is displayed as a dashed line, which is a class line. The direction of the line is indicated by a line dissector, close to symbol B.

Symbol B could also be replaced by a diamond, making an *executable* command the prototype for a new command object. That is less common, but possible all the same.

