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

## Command Call

A command definition can be called. A command call is the execution of a command definition.

A command call is an instantiation of a command definition. A command call is an individual object with a class reference to another command object. There is a complete analogy between calls to a definition and objects of a class, with the addition, that a call is *executable*.

A command call always selects another command to function as its definition. Any command object can function as another command’s definition.

Initially, the call will be an exact replica of its definition. However, the data of command definition only functions as a default configuration. The data of the call object can be altered before it runs and altered when it runs. What data of a command can be altered is covered later in the *Parameter* articles.

At first a command call is dormant, so that you get a chance to fill in its parameters. Then you can run the command call. A command call can only be run once (see the article *Execute Once*).

A command call selects its command definition with a *class* redirection, because the definition will be the command call’s *prototype*, and the call will always be its own individual object for which the command definition is a prototype.

When a call is placed inside a parent command, the parent command, when run, automatically executes all the calls it contains. When a call is placed inside an *object*, the command can only be run manually.