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

## Target Command Object

A command reference can point to another command reference, which points to another command reference and so on. The first command found in this redirection, that does not refer to another command again, is called the *target command object*. Even though any of the command *references* can be used like it is the command object itself, the *target command object* is considered the real command object and not just a reference to it.

The term target command is also used to denote the direct reference target, not necessarily the final target. What kind of target is denoted, will be clear from the context.

## Ideas

### Out of the original Symbol documentation

###### Execution Trace

To find the execution you do the following.



What you do is that you follow reference lines until you encounter a symbol without a reference line. The last symbol in the trace to be a *diamond* is the target execution.

Tracing the execution target only seems to be relevant when the source symbol is a diamond. It usually is. Usually you’ll only do this trace for a diamond. However, the target execution is also relevant for a square, because if a square’s redirection has a diamond in it, any call to the square is a call to the same execution.

##### Tracing Procedure Aspects

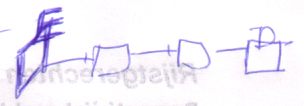
Even though there are only *two* line types for procedure symbols, there are *three* aspects of a procedure symbol to trace:

* Execution
* Definition
* Interface

A definition is a lot like the type of the procedure, while an execution is an instance of the procedure.

###### Redirecting to an Execution

A lot of times you will redirect an execution, but you won’t redirect *to* an execution.



By redirecting to an execution, it is possible for multiple symbols to represent the same single execution.



Also note that when redirecting to an execution, the definition is always the same execution.