

Targets and Parameters

Groovy closures by default have a single parameter even if you don't specify one. This means that the Gant target:

```
target ( example : '' ) { println ( it ) }
```

is not an error because *it* is the name of the implicit parameter. When executed the above will print out

| *example, description:*

The implicit parameter is a reference to a map with various key--value pairs, one of which is the name of the target. Gant is using the implicit parameter feature of Groovy to allow access to information about the target. Hence:

```
target ( example : '' ) { println ( it.name ) ; println ( it.description ) }
```

the code in the target closure can use the name and description of the target, in a target name independent way.

In version of Gant prior to 1.5.0, various tricks could be played using the implicit parameter to Gant targets – targets were not nullary functions but single parameter functions. This facility has been withdrawn as of Gant 1.5.0, targets are nullary functions. So prior to Gant 1.5.0, you might have seen things like:

```
target ( example : '' ) { println ( it ) }
target ( callit : '' ) { example ( 'fred' ) }
```

or even:

```
target ( example : '' ) { parameter -> println ( parameter ) }
target ( callit : '' ) { example ( 'fred' ) }
```

but these no longer do what they used to. As of Gant 1.5.0 any parameters to the call of a target is ignored: the parameter to the target always refers to the map containing information about the target.