Support Tasks	@BACKGROUND_PRIORITY

The priority to launch a process.

Keyword:

@BACKGROUND_PRIORITY

Usage:

Any command or task can be executed in the background. No synchronization or error checking is performed. The command string can be a computed expression

Data Fields:

start_type	code for when to execute the command - options are AT_START, AT_END, AFTER_STABILITY
command	The command string. This can be a computed expression, a literal string(constant), or the label # of an CyFlex variable. This form also supports a priority value used to set the priority of the task started in the background.
kill_option	optional field used to specify whether or not the spawned process should be killed and when that should happen NONE – process is never terminated by gp_test (default) AT_START – process is terminated at the start of the mode if it was spawned in the previous execution of the mode AT_END – process is terminated when the mode is terminated If the process terminates before the mode is executed again, a new copy will be spawned.
priority	value used to set the priority of the task started in the background

Priority Value Information:

- Linux systems use a priority system with 40 priorities, ranging from -20 (highest priority) to 19 (lowest priority). If a real-time priority is needed, the priority argument within the spec file should be a positive number. For example, if 14 is specified as the priority within the spec file, the spawned task will have a real-time priority of -14.
- If the specified priority is less than or equal to 0, the spawned task will have a non-real time priority of 20.

Note that most command strings as used previously are literal strings and should be enclosed in single quotes.

Example Specification:

```
@BACKGROUND_PRIORITY
#start_type            command                  kill_option     priority
AT_START               '/specs/cmds/myscript'   NONE            -2
```

Execute /specs/cmds/myscript command with a non-real time priority of 2.

Notes:

The command string is enclosed in double quotes if it is a computed expression. A literal string is enclosed in single quotes for executing a particular command, or if not enclosed, then it is assumed that the command is a string variable which contains the name of the command to be executed.

Other Examples:

```
@BACKGROUND_PRIORITY
        #start_type         command                      kill_option     priority
        AFTER_STABILITY     ‘/cyflex/bin/do_it’          AT_START        10
        AT_END              my_command                                   -20
        AFTER_START         " ‘/cyflex/cmds’ + myscript "                19
```