Execution Control Commands

This section describes the direct commands the debugger provides for controlling the program flow during a debugging session. For a summary of all commands available with the debugger, refer to *Command Summary and Syntax*.

The commands listed below only apply when the debugger interrupts program execution.

- ESCAPE BOTTOM
- ESCAPE ROUTINE
- EXIT
- GO
- NEXT
- RUN
- STEP
- STEP SKIPSUBLEVEL
- STEP SKIPSUBLEVEL n
- STOP

ESCAPE BOTTOM

This command can only be used when a Natural object has been interrupted within a processing loop.

When you enter this command, the interrupted Natural object will be continued with the first statement following the processing loop.

Note:
This command can be disallowed by Natural Security as described in *Components of an Environment Profile* in the *Natural Security* documentation.

ESCAPE ROUTINE

When you enter this command, processing of the interrupted Natural object will be stopped and processing will continue with the object from which the interrupted Natural object was invoked; it will continue with the statement following the corresponding `CALLNAT`, `PERFORM` or `FETCH RETURN` statement.

If you apply the command `ESCAPE ROUTINE` to a main program, Natural ends the program and returns to the command mode.

Note:
This command can be disallowed by Natural Security as described in *Components of an Environment Profile* in the *Natural Security* documentation.

EXIT

If you are displaying the **Debug Main Menu** and want to invoke the exit function, choose PF3 (Exit) or enter the execution control command `EXIT`, the debugger returns either to the calling program (that is, to the interrupted Natural object which is then continued) or to a command prompt, if the debugger has been invoked with the direct command `TEST`, or to the corresponding input field if it has been invoked by the terminal command `%<TEST`. However, if a breakpoint or watchpoint is currently active, the next command of this breakpoint or watchpoint is executed.

If you are not in the **Debug Main Menu** and enter the direct command `EXIT` or choose PF3 (Exit), you leave the current function and return to the previous step of your debugging session.

GO

When you enter the direct command `GO` (or choose PF14), the debugger returns control to the execution of the interrupted Natural object. If a breakpoint or watchpoint was active at the time the Natural object was interrupted, the remaining commands of this break or watchpoint are *not* executed.

NEXT

When you enter the direct command `NEXT` (or choose PF13), the next command specified for a breakpoint or watchpoint is executed. If no further command has been specified, program execution continues.

RUN

When you enter the direct command `RUN`, test mode is switched off and program execution continues, without investigating any further breakpoints and watchpoint.

STEP

When you enter the direct command `STEP`, an interrupted Natural object is continued for *n* executable statement. The default value for *n* is 1.

STEP SKIPSUBLEVEL

When you enter the direct command `STEP SKIPSUBLEVEL` upon a statement which invokes another object (for example, `CALLNAT`), processing is continued with the next executable statement in the current object instead of the first executed statement in the invoked object).

If this command is applied to a statement that does not invoke another object, the debugger reacts as if the command `STEP` had been entered.

STEP SKIPSUBLEVEL *n*

With the command `STEP SKIPSUBLEVEL`, you can specify a superior level number *n*. Step mode then continues within the next object at the specified level. For example: If you enter `STEP SKIPSUBLEVEL 2` in an object at level 4, you continue step mode in the object at level 2.

Object level information can be obtained with the command `OBJCHAIN` as described in the section *Navigation and Information Commands*.

STOP

When you enter the direct command `STOP`, both the debugger and any interrupted Natural object are terminated.

Note:
This command can be disallowed by Natural Security as described in *Components of an Environment Profile* in the *Natural Security* documentation.