

# TERMINATE

**TERMINATE** [*operand1* [*operand2*]]

This chapter covers the following topics:

- Function
- Syntax Description
- Program Receiving Control after Termination
- Example

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

---

## Function

The **TERMINATE** statement is used to terminate a Natural session. A **TERMINATE** statement may be placed anywhere within a Natural program. When a **TERMINATE** statement is executed, no end-of-page or end-loop processing will be performed.

The behaviour of the **TERMINATE** statement matches that of the **STOP** statement. Processing of return values is not supported.

For Natural Remote Procedure Call (RPC): See *Notes on Natural Statements on the Server* in the *Natural Remote Procedure Call (RPC)* documentation.

## Syntax Description

Operand Definition Table:

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition	
<i>operand1</i>	C	S					N	P	I									yes	no
<i>operand2</i>	C	S	A				A	U										yes	yes

Syntax Element Description:

Syntax Element	Description
<i>operand1</i>	<p><i>operand1</i> may be used to pass a return code to the program receiving control when Natural terminates. For example, a return code setting may be passed as exit code to the shell.</p> <p>See also <i>Natural Startup Errors</i> in the <i>Operations</i> documentation.</p> <p>The value supplied for <i>operand1</i> must be in the range 0 - 255.</p>
<i>operand2</i>	<i>operand2</i> may be used to pass additional information to the program which receives control after the termination.

## Program Receiving Control after Termination

After the termination of the Natural session, the program whose name is specified with the profile parameter PROGRAM will receive control.

Natural passes *operand2* and the value of the profile parameter PRGPAR to that program, if they are specified. The program receives these parameters in the usual way as arguments:

```
int main(int argc, char *argv[])
{
  /* Number of arguments passed. */
  printf("Number of arguments: %d\n", argc);
  /* Program name. */
  if ( argc > 0 )
    printf("Program: %s\n", argv[0]);
  /* Value of operand2 of the TERMINATE statement. */
  if ( argc > 1 )
    printf("Operand 2: %s\n", argv[1]);
  /* Value of the profile parameter PRGPAR. */
  if ( argc > 2 )
    printf("PRGPAR: %s\n", argv[2]);
  return 0;
}
```

If the PROGRAM parameter is not set, the UNIX command shell will receive control after the termination.

## Example

```
** Example 'TEREX1': TERMINATE
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 SALARY (1)
*
1 #PNUM      (A8)
1 #PASSWORD (A8)
END-DEFINE
*
INPUT 'ENTER PASSWORD:' #PASSWORD
*
IF #PASSWORD NE 'USERPASS'
```

```
/*
TERMINATE
/*
END-IF
*
INPUT 'ENTER PERSONNEL NUMBER:' #PNUM
*
FIND EMPLOY-VIEW WITH PERSONNEL-ID = #PNUM
  DISPLAY NAME SALARY (1)
END-FIND
*
END
```