

CALL LOOP

Structured Mode Syntax

```
CALL LOOP operand1 [operand2] ...40
    statement ...
END-LOOP
```

Reporting Mode Syntax

```
CALL LOOP operand1 [operand2] ...40
    statement ...
[LOOP]
```

This chapter covers the following topics:

- Function
- Restriction
- Syntax Description
- Example

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

Related Statements: CALL | CALL FILE | CALLNAT | DEFINE SUBROUTINE | ESCAPE | FETCH | PERFORM

Belongs to Function Group: *Invoking Programs and Routines*

Function

The CALL LOOP statement is used to generate a processing loop that contains a call to a non-Natural program.

Unlike the CALL statement, the CALL LOOP statement results in a processing loop which is used to repeatedly call the non-Natural program. See the CALL statement for a detailed description of the CALL processing.

Restriction

The statements AT BREAK, AT START OF DATA and AT END OF DATA must not be used within a CALL LOOP processing loop.

Syntax Description

Operand Definition Table:

Operand	Possible Structure		Possible Formats												Referencing Permitted	Dynamic Definition		
<i>operand1</i>	C	S				A										yes	no	
<i>operand2</i>	C	S	A	G		A	U	N	P	I	F	B	D	T	L	C	yes	yes

Syntax Element Description:

<i>operand1</i>	The name of the non-Natural program to be called can be specified as a constant or - if different programs are to be called dependent on program logic - as an alphanumeric variable of length 1 to 8. A program name must be placed left-justified in the variable.
<i>operand2</i>	The CALL LOOP statement can have a maximum of 40 parameters. The parameter list is constructed as described for the CALL statement. Fields used in the parameter list may be initially defined in the CALL LOOP statement itself or may have been previously defined.
<i>statement ...</i>	The CALL LOOP statement initiates a processing loop which must be terminated with an ESCAPE statement.
END-LOOP	The Natural reserved word END-LOOP must be used to close the processing loop.

Example

```

DEFINE DATA LOCAL
1 PARAMETER1 (A10)
END-DEFINE
CALL LOOP 'ABC' PARAMETER1
  IF PARAMETER1 = 'END'
    ESCAPE BOTTOM
  END-IF
END-LOOP
END

```