

HE - Helproutine

With this session parameter, you assign a helproutine or a help map to a field.

Possible settings		See <i>HE Parameter Syntax</i> below.	
Default setting	none		
Specification within session	yes	Applicable Statements:	INPUT
		Applicable Command:	none

Helproutines can be created with the Natural program editor, help maps with the Natural map editor.

The helproutine or help map may then be invoked during processing of an INPUT statement or a map by choosing either of the following methods:

- In the field for which to invoke the help request, enter the help character (question mark (?) by default) and press ENTER.
- Place the cursor in the field for which to invoke the help request and press the PF key defined as help function key with the SET KEY statement.

The following topics are covered below:

- HE Parameter Syntax
- Execution of Helproutines
- Examples

HE Parameter Syntax

The syntax of this parameter is:

HE=operand1	[,	{	operand2	}]	...20
			=				
			nX				

Operand Definition Table:

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

Syntax Description:

<i>operand1</i>	<p><i>operand1</i> is the name of the helproutine or help map to be invoked. The name may be a 1 to 8 character alphanumeric constant or user-defined variable. If a variable is used, it must have been previously defined. The case of the specified name is not translated. The name may contain an ampersand (&); at execution time, this character will be replaced by the one-character code corresponding to the current value of the Natural system variable *LANGUAGE. This feature allows the use of multi-lingual helproutines or help maps.</p> <p>For additional information on using <i>operand1</i> within a map, see the HE helproutine option described in <i>Extended Field Editing in Map Editor</i> in the <i>Editors</i> documentation.</p>
<i>operand2</i>	<p>You may specify 1 to 20 parameters (<i>operand2</i>) which are passed to the helproutine or help map. They may be specified as constants or as user-defined variables which contain the values of the parameters.</p> <p>For additional information on using <i>operand2</i> within a map, see the HE helproutine option described in <i>Extended Field Editing in Map Editor</i> in the <i>Editors</i> documentation.</p>
=	<p>The equals sign (=) is used to pass an object or a field name to a helproutine or help map:</p> <ul style="list-style-type: none"> ● If the equals sign is entered in the HE= specification at statement level, the name of the object (as contained in the system variable *PROGRAM) being executed is passed to the helproutine or help map. In Example 3, the object name passed is PROGRAM1. ● If the equals sign is entered in the HE= specification at field level, the name of the field is passed to the helproutine or help map. In Example 3, the field name passed is #PARM1. <p>If the equals sign is used as a parameter, the corresponding parameter in the helproutine or help map must be specified with format/length A65.</p>
<i>nX</i>	<p>The notation <i>nX</i> can be used to specify parameters to be omitted, that is, for which no values are to be passed. The corresponding receiving parameters in the called helproutine's DEFINE DATA PARAMETER statement must be defined as OPTIONAL.</p>

Notes:

1. The operands must be separated either by the input delimiter character (as specified with the session parameter ID) or by a comma. However, a comma must not be used for this purpose if the comma is defined as decimal character (with the session parameter DC).
2. If parameters are specified, the helproutine must begin with a DEFINE DATA PARAMETER statement which defines fields that correspond with the parameters in format and length.
3. The value of the field for which a helproutine is specified may be referenced within the helproutine. This is done by specifying a field in the DEFINE DATA PARAMETER statement which corresponds in format and length with the original field. In the block of fields defined within the DEFINE DATA PARAMETER statement, this field must always be defined behind the parameters, if present.
4. If the field for which a helproutine is specified is an array element, its indices may be referenced by the helproutine. To do so, you specify index parameters with format I (integer), N (numeric unpacked), or P (packed numeric) at the end of the DEFINE DATA PARAMETER statement. You

may specify up to three index parameters according to array dimensions.

Execution of Help routines

If a help routine or help map is requested - by entering a question mark (?) in the field, or by pressing the help key (as defined with a SET KEY statement), or via a REINPUT USING HELP statement - all other data that may have been entered into fields are not assigned to the program variables until all help requests have been processed.

Note:

Only one help request per INPUT statement is possible; that is, if help is requested for more than one field (for example, by entering question marks in multiple fields), only the first help request will be executed.

Examples

Example 1:

```
/* MAIN PROGRAM
DEFINE DATA
1 #A(A20/1:3)
END-DEFINE
...
SET KEY PF1=HELP
...
INPUT #A (2) (HE='HELPA',=)
...
END
```

Example 2:

```
/* HELP-ROUTINE 'HELPA'
DEFINE DATA PARAMETER
1 #VARNAME (A65)
1 #PARM1 (A20)
1 #VARINDEX (I2)
END-DEFINE
...
```

Example 3:

```
* Program 'PROGRAM1'
*
DEFINE DATA LOCAL
1 #PARM1 (A65) INIT <'valueparm1'>
END-DEFINE
SET KEY PF1 = HELP
FORMAT KD=ON
*
INPUT (AD=M HE='HELP1',=)
  'Enter ? for name of executed object:'
  / #PARM1
*
INPUT (AD=M)
  'Enter ? for field name:'
  / #PARM1 (HE='HELP1',=)
*
END
```

Parameter Data Area in Example Helproutine HELP1:

```
* Helproutine 'HELP1'  
*  
DEFINE DATA PARAMETER  
1 #FLD1 (A65)  
END-DEFINE  
...
```