

INPUT

This chapter covers the following topics:

- Function
- Input Modes
- Entering Data in Response to an INPUT Statement
- SB - Selection Box
- Error Correction
- Split-Screen Feature
- System Variables with the INPUT Statement

The syntax is described separately. See:

- *INPUT Syntax 1 - Dynamic Screen Layout Specification*
- *INPUT Syntax 2 - Using Predefined Map Layout*

Related Statements: `DEFINE WINDOW` | `REINPUT` | `SET WINDOW`

Belongs to Function Group: *Screen Generation for Interactive Processing*

Function

The `INPUT` statement is used in interactive mode to create a formatted screen or map for data entry.

It may also be used in conjunction with the Natural stack (see the `STACK` statement); and on mainframe computers, it may also be used to provide user data for programs being executed in batch mode.

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

Input Modes

The `INPUT` statement may be used in screen, forms, or keyword/delimiter mode. Screen mode is generally used with video terminals/screens. Forms mode may be used with TTY terminals. Delimiter mode is used with TTY terminals, and also in batch mode (on mainframe computers). The default mode is screen mode.

You can change the input mode with the session parameter `IM`.

Screen Mode

In screen mode, execution of the `INPUT` statement results in the display of a screen according to the fields and positioning notation specified. The message line of the screen is used by Natural for error messages. The position of the message line (top or bottom of screen) may be controlled by the terminal command `%M`. The terminal user may position to specific fields using the various tabulation keys.

As Natural allows for screen window processing, the layout of the logical screen map may be larger (theoretically 250 characters per line and 250 lines, but limited by the internal screen buffer) than the physical screen size.

The windowing terminal command `%W` may be used to modify logical and physical window position and size (see the terminal command `%W` for details of window handling).

For input fields (`AD=A` or `AD=M`) that are not fully displayed on the physical screen, the following rules apply:

- Input fields whose beginning is not inside the window are always made protected.
- Input fields which begin inside and end outside the window are only made protected if the values they contain cannot be displayed completely in the window. Please note that in this case it is decisive whether the value length, not the field length, exceeds the window size. Filler characters (as specified with the profile parameter `FC` or session parameter `AD`) do not count as part of the value.
- Before an input field thus protected can be accessed and processed, the window size must be adjusted so as to fully display the field or value respectively (see the terminal command `%W`).

Non-Screen Modes

The `INPUT` statement may be used for an operation on line-oriented devices or for the processing of batch input from sequential files.

The same map layouts as defined for screen mode operation can also be processed in non-screen mode.

Forms mode and keyword/delimiter mode are also available to process the input either by simulating the screen layout in line mode or by just processing the data without any map layout.

See also:

- *Using the INPUT Statement in Non-Screen Modes*
- *Processing Data from the Natural Stack*

Entering Data in Response to an INPUT Statement

Data for an alphanumeric field must be entered left-justified. Any character, including a blank, is meaningful. The data are assigned one character per byte to the internal field. Data entered for an alphanumeric field are not validated.

Lower and upper case translation are controlled by the terminal commands `%L` and `%U` as well as the attributes `AD=T` and `AD=W`.

Data for a numeric field may be placed anywhere in the input field. Leading and/or trailing blanks, leading zeros, a leading sign and one decimal point are permitted. Natural adjusts the value according to the internal definition of the field. If SG=OFF is specified, Natural does not assume or allocate a position for a sign position. Data for a field defined with format P must be entered in decimal form. Natural will convert decimal to packed wherever necessary. A field containing all blanks is interpreted as a zero value. Data for a numeric field are validated by Natural to ensure that the value consists only of leading and/or trailing blanks, an optional leading sign, an optional decimal point, and numeric characters. If no decimal point is entered, it is assumed to be to the right of the value entered.

Data for a binary field must be entered for all positions (two characters per byte). Only valid hexadecimal characters (0 - 9, A - F) may be used. A blank (H'20' in ASCII or H'40' in EBCDIC respectively) is valid and is converted to binary zeros. Data for a binary field are validated by Natural for hexadecimal characters.

Data for format L fields may be entered as blank (false) or non-blank (true).

Data for format F, D, and T are entered according to the rules stated for F, D, and T constants.

Numeric Edit Mask Free Mode

Within a field element, you may format the representation of the field content with an edit mask. The edit mask is used for two purposes:

- to build the layout for displaying the field on the screen;
- when a string has been modified and ENTER has been pressed, to extract the field data from the string entered.

The advantage of improving the format of the field data displayed with additional insert characters may actually be a disadvantage, because a new data value entered has to perfectly match the format of the edit mask.

Example:

```
SET GLOBALS ID=; DC=,
RESET N (N7,3)
INPUT N (AD=M EM=Z'.'ZZZ'.'ZZZ,999EUR)
END
```

Output value	is displayed as:	Input value	must be entered as:	leads to an input error if entered as:
0	,000EUR	1	1,000EUR	1 1EUR 01,000EUR
1234	1.234,000EUR	1234567	1.234.567,000EUR	1234567 1.234.567 1.234.567EUR
0,123	,123EUR	1,234	1,234EUR	1,234

Another option for entering numeric fields with the edit mask is to use an alternative INPUT mode, which is called the edit mask free mode. When activated (either at session startup with the profile parameter EMFM or in a running Natural session via the terminal command %FM+), all or some of the edit mask insert characters may be left out from input.

However, when a contiguous string of insertion characters appears in the edit mask (like EUR in the example below), you may only supply or leave out the string completely. The number of optional or mandatory digits (edit-mask character Z and 9) to be supplied is not affected.

Example with Edit Mask Free Mode activated:

```
SET GLOBALS ID=; DC=,
SET CONTROL 'FM+' /* activate numeric Edit Mask Free Mode
RESET N (N7,3)
INPUT N (AD=M EM=Z'.'ZZZ'.'ZZZ,999EUR)
END
```

Input value	can be entered as:	leads to an error if entered as:
1	1 1,0 001 1,00EUR 0.001 1,EUR	1EUR
1234567	1234567 1.234.567 1234.567 1234567,0 1.234.567,0 1.234.567,EUR 1.234.567,0EUR 1.234.567,000EUR	1.234.567EUR
1,234	1,234 1,234EUR 001,234 0.001,234EUR 00001,234EUR	1,234EU

Note:

The edit mask free mode applies only for INPUT, but is ignored in a MOVE EDITED statement.

SB - Selection Box

Selection boxes in an INPUT statement are available on mainframe computers only. On Windows, selection boxes may be defined in the map editor only. On UNIX and OpenVMS, selection boxes cannot be defined and are ignored, if they are imported from a Windows or mainframe environment.

Selection boxes can be attached to input fields. They are a comfortable alternative to help routines attached to fields, since you can code a selection box direct in your program. You do not need an extra program as with help routines.

For more information, see the session parameter SB in the *Parameter Reference*.

Error Correction

If the value entered in an input field does not correspond to the format or edit mask of the field, Natural displays an error message (without terminating the program execution) and positions the cursor in the field in error. The user may then enter a valid value, whereupon processing continues.

Split-Screen Feature

In general, each INPUT statement generates a new page (or terminal screen) of output. Any INPUT statement which is specified within an AT END OF PAGE statement will not produce a new screen. This feature allows for the creation of a split screen where the upper portion of the screen may be used to display multiple lines and the lower portion can be used to create an input map for communication. The profile parameter PS (page size) should be used, either in a SET GLOBALS or FORMAT statement, to set the logical page size to ensure that the input map is built on the same physical screen.

The first INPUT line will be placed after the last displayed line. If the NO ERASE option is used, the first INPUT line will be placed at the top of the page.

System Variables with the INPUT Statement

For information on relevant system variables, see the section *Input/Output Related System Variables* in the *System Variables* documentation.