

Input/Output Related System Variables

This chapter covers the following topics:

- *CURS-COL
- *CURS-FIELD
- *CURS-LINE
- *CURSOR
- *LINE-COUNT
- *LINESIZE
- *LOG-LS
- *LOG-PS
- *PAGE-NUMBER
- *PAGESIZE
- *PF-KEY
- *PF-NAME
- *WINDOW-LS
- *WINDOW-POS
- *WINDOW-PS

*CURS-COL

Format/length: P3

Content modifiable: Yes (however, a negative value must not be assigned)

This system variable contains the number of the column in which the cursor is currently positioned.

The cursor position is defined within the currently active window, regardless of its physical placement on the screen, starting with position 1/1 from the upper left corner of a logical page.

If the value of *CURS-COL is negative, this indicates that the cursor is outside the active window. If *CURS-COL is negative, *CURS-LINE will also contain a negative value. In this case, the absolute values of both system variables indicate the position of the cursor on the physical screen.

Note:

The message line, function-key lines and infoline/statistics line are not counted as data lines on the screen.

See also *Dialog Design, Column-Sensitive Processing* in the *Programming Guide*.

***CURS-FIELD**

Format/length: I4

Content modifiable: No

This system variable contains the internal identification of the field in which the cursor is currently positioned.

*CURS-FIELD cannot be used by itself, but only in conjunction with the POS function. You may use them to check if the cursor is currently positioned in a specific field and have processing performed depending on that condition. See the POS function for details.

If the cursor is not in a field or if no REINPUT is possible, *CURS-FIELD contains 0.

In Natural for Ajax applications, *CURS-FIELD identifies the operand that represents the value of the control that has the input focus. You may use *CURS-FIELD in conjunction with the POS function to check for the control that has the input focus and perform processing depending on that condition.

Note:

The value of *CURS-FIELD serves only as internal identification of the field and cannot be used for arithmetic operations. If *CURS-FIELD identifies an occurrence of an X-array (an array for which at least one bound in at least one dimension is specified as extensible), the value of *CURS-FIELD may change after the number of occurrences for a dimension of the array has been changed using the EXPAND, RESIZE or REDUCE statements.

See also *Dialog Design, Field-Sensitive Processing* in the *Programming Guide*.

***CURS-LINE**

Format/length: P3

Content modifiable: Yes (however, a negative value or 0 must not be assigned)

This system variable contains the number of the line in which the cursor is currently positioned.

The cursor position is defined within the current active window, regardless of its physical placement on the screen, starting with position 1/1 from the upper left corner of a *logical* page.

Note:

The message line, function-key lines and infoline/statistics line are not counted as data lines on the screen.

*CURS-LINE may also contain one of the following values:

| Value | Cursor Position |
|-------|---|
| 0 | On the top or bottom horizontal frame line of a window. |
| -1 | On the Natural message line. |
| -2 | On the Natural infoline/statistics line. |
| -3 | On the upper function-key (number) line. |
| -4 | On the lower function-key (name) line. |

If the value of *CURS-COL is negative, which indicates that the cursor is outside the active window, *CURS-LINE will also contain a negative value. In this case, the *absolute* values of both system variables indicate the position of the cursor on the *physical* screen.

See also *Dialog Design, Line-Sensitive Processing* in the *Programming Guide*.

*CURSOR

Format/length: N6
Content modifiable: No

This system variable contains the position of the cursor on the input screen at the time the ENTER key or a function key is pressed.

Note:

Instead of *CURSOR, it is recommended that the system variables *CURS-LINE and *CURS-COL be used. *CURSOR only continues to be available for compatibility with previous Natural versions.

*LINE-COUNT

Format/length: P5
Content modifiable: No

This system variable contains the line number of the current line within the current page.

This variable is used by Natural to determine the line number for the next line of the report.

The value of *LINE-COUNT is incremented by 1 for each line to be output. The value is updated during the execution of a WRITE, SKIP, DISPLAY, PRINT or INPUT statement and contains the number of the last line on the page that has been output.

An EJECT or NEWPAGE statement causes *LINE-COUNT to be reset to 1 (except in the case of NEWPAGE WITH TITLE, where the value of *LINE-COUNT depends on the number of lines output as title).

The maximum line number permitted is 250.

If multiple reports are being produced by the program, (*rep*) notation after *LINE-COUNT is used to specify the report identification for which the current line number is being requested.

*LINESIZE

Format/length: N7

Content modifiable: No

This system variable contains the physical line size of the I/O device from which Natural was invoked (if the TP system is able to provide such).

*LOG-LS

Format/length: N3

Content modifiable: No

This system variable contains the line size of the logical page that is output with the primary report.

*LOG-LS is only applicable to the primary report, not to any additional report.

*LOG-PS

Format/length: N3

Content modifiable: No

This system variable contains the page size of the logical page that is output with the primary report.

*LOG-PS is only applicable to the primary report, not to any additional report.

*PAGE-NUMBER

Format/length: P5

Content modifiable: Yes

This system variable contains the current value for page number of an output report.

If multiple reports are being produced by the program, (*rep*) notation after *PAGE-NUMBER is used to specify the report identification for which the current page number is being requested.

This variable is defined by Natural at the time formatting for the report is started. Therefore, the parameter has no meaning until the first FORMAT, WRITE, or DISPLAY statement for any given report has been issued. This variable may be modified by a Natural program.

This variable is used by Natural to determine the page number for the next page of the report. The value is always incremented by 1 for the next page initiated by WRITE, DISPLAY, SKIP or NEWPAGE statements. EJECT does not cause *PAGE-NUMBER to be incremented.

***PAGESIZE**

Format/length: N7

Content modifiable: No

This system variable contains the physical page size of the I/O device from which Natural was invoked (if the TP subsystem is able to provide such).

***PF-KEY**

Format/length: A4

Content modifiable: No

This system variable contains the identification of the key which was pressed last.

*PF-KEY can contain one of the following values:

| Value | Description |
|--------------|--------------------------------|
| PA1 to PA3 | Program Attention keys 1 to 3. |
| PF1 to PF48 | Program Function keys 1 to 48. |
| ENTR | ENTER key. |
| CLR | CLEAR key. |
| PEN | Light pen. |
| PGDN | PAGE DOWN key. |
| PGUP | PAGE UP key. |

*PF-KEY only contains the identification of a key if that key is currently sensitive; otherwise *PF-KEY will contain ENTR.

Note:

When you compare the content of *PF-KEY with a range of values, remember that *PF-KEY contains an alphanumeric value.

See also

- SET KEY statement (for effects on the contents of *PF-KEY).

- *Processing Based on Function-Keys* in the *Programming Guide*.

*PF-NAME

Format/length: A10
Content modifiable: No

This system variable contains the name of the function key that was pressed last, that is, the name as assigned to the key with the NAMED clause of the SET KEY statement.

This allows you to perform processing depending on a specific function name, not a specific key. For example, if you wish to allow users to invoke help by pressing either PF1 or PF13, you assign the name HELP to the keys PF1 and PF13 and make the invoking of help dependent on *PF-NAME= 'HELP': the help will then be invoked no matter whether the user presses PF1 or PF13 to invoke it.

See also *Dialog Design, Processing Based on Function-Key Names* in the *Programming Guide*.

*WINDOW-LS

Format/length: N3
Content modifiable: No

This system variable contains the line size of the logical window (without frame). See also the DEFINE WINDOW statement.

*WINDOW-POS

Format/length: N6
Content modifiable: No

This system variable contains the position which corresponds to the upper left corner of the window. See also the DEFINE WINDOW statement.

The position is counted in characters across multiple lines, beginning with 0 (upper left corner).

*WINDOW-PS

Format/length: N3
Content modifiable: No

This system variable contains the page size of the logical window (without frame). See also the DEFINE WINDOW statement.