Natural

Terminal Commands

Version 6.3.13 for Windows

October 2012

Natural

This document applies to Natural Version 6.3.13 for Windows.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: NATWIN-NNATTCOM-6313-20121005

Table of Contents

Preface

This documentation describes the Natural terminal commands. It is organized under the following headings:

Introduction to Terminal Commands	What are terminal commands? How to issue terminal commands and how to use them in a program.
Terminal Commands Grouped by Function	Provides an overview of the terminal commands ordered by functional groups.
Terminal Command Key Assignments	How to assign a frequently used terminal command to a function key. Information on CLEAR key.
Terminal Commands in Alphabetical Order	Descriptions of the terminal commands in alphabetical order.

1 Introduction to Terminal Commands

Purpose of Terminal Commands

A complete functional overview of Natural terminal commands is given in the section *Terminal Commands Grouped by Function*.

Changing the Terminal Command Control Character

You can define another special character as control character; this is done with the session parameter `CF`.

When the control character is changed, all terminal commands which have been assigned to function keys will be adjusted accordingly.

Issuing Terminal Commands

Terminal commands can be used in a Natural runtime environment. The following rules apply:

- You can enter the control character as first character in any unprotected field or in any position, if the screen contains only protected fields.
- As soon as you enter the control character, a window is displayed in which you can enter a terminal command.
- Terminal commands which have been entered incorrectly are ignored, but you will not receive a corresponding error message.
- If you have entered data in unprotected fields before the terminal command window is displayed, the data will not be processed.

Using Terminal Commands in Programs

Terminal commands may also be issued from within a program by using the `SET CONTROL` statement. When a terminal command is specified with a `SET CONTROL` statement, the control character is omitted.

Terms Used in the Terminal Command Descriptions

In the descriptions of several terminal commands, the terms “screen” and “window” are used with the following meanings:

Term	Meaning
Screen	Depending on the operating system under which Natural is running, “screen” refers either to the entire terminal screen as such, or to the operating-system window in which the Natural session is running, or to the Natural main output window. However, for the sake of convenience, the term “screen” is used in all these instances.
Window	Always refers to the Natural window (as explained with the terminal command `%W`).

2 Terminal Commands Grouped by Function

The following tables provide an overview of the terminal commands grouped by functions.

Case Translation

Terminal Command	Function
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.

Copy/Clear

Terminal Command	Function
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.

Language, Messages, Error Processing

Terminal Command	Function
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.

Screen and Window Processing

Terminal Command	Function
%K	Simulate PF- and PA-keys.
%K*nn*, %KP*n*	Simulate PF- and PA-keys.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%T and %T*ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.

Terminal Command	Function
%W	Natural window handling.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.

Colors

Terminal Command	Function
%=	Assign colors to fields.

INPUT Statement

Terminal Command	Function
%FM	Activate/deactivate edit mask free mode.
%R	Repeat INPUT statement.

Statistics

Terminal Command	Function
%X	Control of statistics line/infoline.
%<TECH	Display technical information.

Miscellaneous

Terminal Command	Function
%H	Produce hardcopy output.
%J	Invoke helproutine.
%Q	Suppress next input.
%V	Control of print mode.
%% and %.	Interrupt current Natural operation.

Key Assignments

Key(s)	Function
CLEAR	Interrupt current Natural operation.

3 Terminal Command Key Assignments

Assigning Terminal Commands to Function Keys

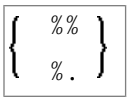
For enhanced operating convenience, you can assign a frequently used terminal command to a function key.

- Within a program, you can assign terminal commands to function keys by using the statement `SET KEY`.

CLEAR Key - Interrupt Current Operation

Pressing the CLEAR key has the same effect as the terminal command `%%`.

4 %% and %. - Interrupt Current Operation



These terminal commands can be used to interrupt the current operation.

Note: The terminal commands `%%` and `%.` will be ignored if the profile parameter `ESCAPE` is set to `OFF`.

%% in Online Mode

If you enter `%%` in any field on the screen, the currently active Natural program will be terminated immediately and Natural will return to command input mode. If you enter `%%` in command input mode, the Natural session will terminated (equivalent to the Natural system command `FIN`).

`%%` has the following effects:

- The contents of the Natural stack will be deleted.
- Any logical database transaction currently being processed is backed out.
- The source program currently in the work area of the editor will not be affected.

%% in Batch Mode

In batch mode, `%%` may be used to set restart points in the input files and thus ensure the synchronisation of the input files in the case of an error.

Influence of Profile Parameter CC

Command	Function
`CC=ON`	If the Natural profile parameter `CC` is set and an error occurs during the compilation/execution of a Natural program in batch mode, the input data stream for the `SYNIN` and `OBJIN` input files will be flushed until a line containing `%%` in the first two positions is encountered (if no `%%` is encountered, it will be flushed until the end-of-file is reached). In addition, the contents of the Natural stack will be deleted. If more data are available in the input stream, Natural resumes processing with the line after `%%`.
`CC=OFF`	Any `%%` in the input data will be ignored.

%. in Online Mode

Online, `%.` is the same as `%%`, except that the Natural stack is not deleted.

%. in Batch Mode

In batch mode, `%.` causes reading of input values for the current `INPUT` statement to be terminated.

5 %* - Inhibit Character Display

```
%*
```

This command may be used when entering sensitive data (for example, passwords). `%*` causes all fields on the current screen to be non-displayable.

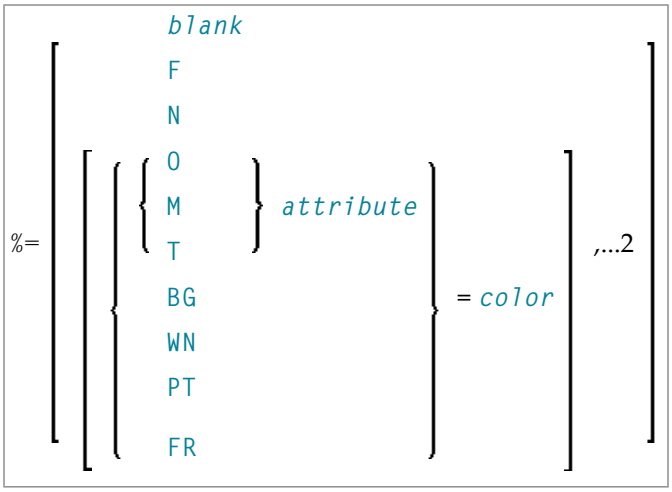
When used with the `SET CONTROL` statement, `%*` causes all fields on the next screen to be non-displayable.

6 %<TECH - Display Technical Information

```
%<TECH
```

This terminal command corresponds to the system command `TECH`.

7 %= - Assigning Colors to Fields



With this command, you can assign colors to field attributes for programs that were originally not written for color support. The command causes all fields/text defined with the specified attributes to be displayed in the specified color.

If predefined color assignments are not suitable for your terminal type, you can use this command to override the original assignments with new ones.

You can also use this command within Natural editors, for example, to define color assignments dynamically during map creation.

Command	Function
General settings:	
blank	Clear color translate table.
F	Newly defined colors are to override colors assigned by program.
N	Color attributes assigned by program are not to be modified.
Field types:	

Command	Function
O	Output fields (AD=O). For detailed information on the session parameter AD, see *Field Input/Output Characteristics* in the *Parameter Reference* documentaion.
M	Input fields; that is, input-only fields (AD=A) and modifiable fields (AD=M).
T	Text constants.
Possible field attributes:	
B	Blinking
C	Italic
D	Default
I	Intensified
U	Underlined
V	Reverse video
A color can also be assigned to the following parts of a screen:	
BG	Background
WN	Foreground (that is, fields for which no color is defined)
PT	Default page title
FR	Frame of a window
Possible colors:	
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

Example:

```
%=TI=RE,OB=YE
```

This example assigns color red to intensified text fields and color yellow to all blinking output fields.

8 %C - Copying Contents of Page Buffer

%C

This terminal command is used to copy the contents of the page buffer to the next available lines in the Natural source work area.

The page currently displayed by Natural will be copied into the Natural source work area. The page content will be written to the next free location in the source work area, where it can be modified using the Natural program editor.

To clear the source work area before copying the page, the %Z terminal command may be used.

Notes:

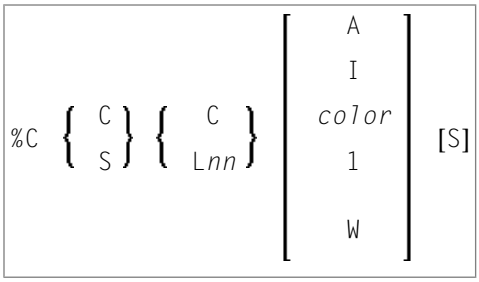
1. %C should not be used in an editor session. Modifications made to the source area outside of editor content are not recognized by the editor.
2. The page buffer (the logical output from Natural) is not necessarily the same as the screen buffer which is displayed on the screen.
3. The program editor will show the new lines at once, if it has the input focus when %C is executed. If the program editor does not have the input focus, the %C changes will not be displayed; the changed source work area has to be re-loaded with the EDIT command.

Example:

```
DEFINE DATA LOCAL
1 I (I2)
END-DEFINE
FOR I = 1 TO 10
  WRITE I
  SET CONTROL 'C'
```

```
END-FOR
END
```

9 %CS and %CC - Copying Data to Stack or *COM



1 and W cannot be specified with L*nn*.

With this terminal command, you can copy parts of a screen into the Natural stack (%CS) or into the system variable *COM (%CC). The protected data from a specific screen line are copied field by field (except with option A; see below).

The second letter in the command determines where the data are copied to:

- %CC...
  Copies the data into the system variable *COM.
- %CS...
  Copies the data into the Natural stack. The data are placed on top of the stack as input data (as with a STACK TOP DATA statement).

The third letter in the command determines the line from which the data are copied:

- %CCC **and**%CSC
  Copies all protected data from the line in which the cursor is positioned, beginning from the field in which the cursor is positioned.

- %CCL *nn***and**%CSL*nn*
  Copies all protected data from line number *nn*.

Moreover, you have the following options:

- %C...A
  Copies all of a line, that is, not only the protected data, but also the modifiable fields; the line is not copied field by field, but as a whole (including field attributes).
- %C...I
  Copies only the intensified fields from a line.
- %C...*color*
  Copies only the fields of that color from a line.
- %C...C1
  Copies only one field, namely the field in which the cursor is positioned (regardless of its attributes). (%C...L *nn*1 is not possible.)
- %C...CW
  Copies only the word (as delimited by blanks or special characters within a field) over which the cursor is positioned. (%C...L*nn*W is not possible.)
- %C...S
  Causes Natural to "stay" on the screen from which the data are copied, when the command is executed. This allows you to copy several different data from a screen, before you process the data.

When you enter the command directly as %C... (or assign it to a PF-key), it applies to the *physical screen* within the active window.

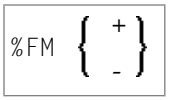
10 %E= - Activate/Deactivate Error Processing

With the terminal command `%E=OFF` any error transaction and `ON ERROR` processing is switched off, with `%E=ON` error transaction and `ON ERROR` processing is switched on again.

Command	Function
`%E=OFF`	Switches any error transaction (as identified by the system variable `*ERROR-TA` or defined in Natural Security) and `ON ERROR` processing off; any error that occurs will then be handled by normal Natural error processing. This may be used to locate an error in your applications' error processing if the structure of the application with various error-handling procedures on different levels makes it impossible for you to find out where exactly an error originally occurred.
`%E=ON`	Switches error transaction and `ON ERROR` processing on again.

For further information, see *Using an Error Transaction Program* in the *Programming Guide*

11 %FM - Numeric Edit Mask Free Mode



This command is used to activate/deactivate edit mask free mode, a special capability to allow literals to be omitted during input into a field with a numeric edit mask.

Command	Function
%FM-	With this command, you switch edit mask free mode off.
%FM+	With this command, you switch edit mask free mode on.

The default setting at session startup is provided with profile parameter EMFM.

See the INPUT statement (in the *Statements* documentation) for additional information on the edit mask free mode.

12 %H - Hardcopy Output

These terminal commands are used to produce hardcopy output from Natural reports on a printer or special destination, such as the source area.

By default, a `%H` command is effective for the current logical output (that is, the current window without message line, function-key lines and statistics line/infoline).

You have the following options:

Command	Function
`%H`	This command activates the hardcopy output. The hardcopy output is deactivated when either an `INPUT` statement with modifiable fields is executed, or the end of the program is reached. All pages, which are processed when the hardcopy output is active, will be forwarded to the printing destination. These pages are not displayed on the screen. If `%H` is entered in an input field, only the current page will be printed. For the definition of the output destination, a window is displayed listing all available printers and special destinations (e.g. SOURCE). From this list, you select the destination on which the hardcopy is to be printed.
`%H`*`destination`*	Same as `%H`, but the printing destination is specified with destination. The *`destination`* can be 1 to 8 characters long.

Command	Function
%H!	With SET CONTROL 'H!', the current page (that is, the page which was output before the SET CONTROL statement was executed) is printed. With %H!, the page which is currently displayed on the screen, is printed. For the definition of the output destination, a window is displayed listing all available printers and special destinations (e.g. SOURCE). From this list, you select the destination on which the hardcopy is to be printed.
%H!*destination*	Same as %H!, but the printing destination is specified with destination. The destination can be 1 to 8 characters long.
%H=	This command activates/deactivates (toggle switch) the logging of all following pages. Logging means that all pages, which are displayed on the screen are forwarded to the printing destination as well. This feature may be used, for example, to log a sequence of output for administrative, debugging or educational purposes. For the definition of the output destination, a window is displayed listing all available printers and special destinations (e.g. SOURCE). From this list you select the destination on which the hardcopy is to be printed.
%H=*destination*	Same as %H=, but the printing destination is specified with destination. The destination can be 1 to 8 characters long.
%H#*destination*	Same as %H*destination*.
%H-	Deactivates the hardcopy output immediately. **Note:** When a SET CONTROL 'H-' statement is executed, data which were already written to the page buffer but have not been output yet, will not be routed to the printer. To also print these data, you have to code an EJECT statement before the SET CONTROL 'H-' statement.

Note: The EJECT statement does not affect the %H command. The %H command always causes Natural to advance pages.

13 %I= - Specify Icon for Output Window

```
%I=icon-file-name.ICO
```

This terminal command is used with a `SET CONTROL` statement to specify an icon that is to be used instead of the Natural icon for the output window. This icon will be shown in the control menu of the output menu in the Windows task bar.

For further information, see *Using Your Own Icon for the Output Window* in the *Operations* documentation.

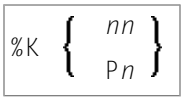
14 %J - Invoke Helproutine

```
%Jhelproutine
```

This terminal command can be used to invoke an interactive helproutine.

If `%J` is used when a function invoked by a system command is active, Natural will search for the specified helproutine in the active library of the system command or in a library that has been defined as a steplib for the system command.

15 %K and %KP - Simulate PF- and PA-Key



These terminal commands can be used to simulate the terminal function (PF, ENTER) and program attention (PA) keys.

Command	Function
%K*nn*	Simulates the terminal function key numbered *nn* (PF1 to PF99). This permits PF-keys 13-24 to be assigned to PF-keys 1-12, or the activation of PF-keys not available on the keyboard used. This terminal command also makes function keys available in batch mode.
%K0	Simulates the ENTER key.
%KP*n*	Simulates the program attention key numbered *n* (PA1 to PA3) (see %K*nn*).

16 %L - Disable Lower- to Upper-Case Translation

```
%L
```

This command prevents that lower-case characters are translated to upper-case by Natural.

`%L` influences the interactive input which is entered, for example, with a Natural `INPUT` statement. It does not, however, influence the input from the stack.

See also the terminal command `%U`.

17 %L= - Set Language Code

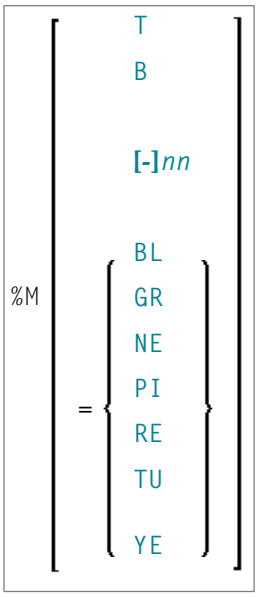
%L=*nn*

With the terminal command %L=*nn*, you can set the language code *nn* to be used by Natural.

For a list of possible language codes, see the system variable *LANGUAGE.

To set the language code already at session start, specify the profile parameter ULANG.

18 %M - Control of Message Line



With this terminal command, you can control the position, the protection mode and the color of the Natural message line.

This terminal command is ignored in batch mode.

Message Line Positioning

Command	Function
%MT	Causes the message line to be the top line of the screen.
%MB	Causes the message line to be the bottom line of the screen.
%M	Causes the current message line position to be switched from the top line to the bottom line of the screen (or vice versa), or from line *nn* to the bottom line.
%M*nn*	Causes the message line to be placed on line *nn* of the screen.
%M-*nn*	Causes the message line to be placed on the *nn*th line from the bottom of the screen. If the line number *nn* or -*nn* is not within the current screen, the message line will not be displayed.

Message Line Color

Command	Function
%M=*color-code*	When a color screen is used, this terminal command causes the message line to be displayed in the specified color (for an explanation of color codes, see the session parameter CD).

See also *Control of the Message Line - Terminal Command %M* in the *Programming Guide*.

19 %N - Activate Non-Conversational Mode

```
%N
```

This terminal command is used with a `SET CONTROL` statement and causes the next logical output screen to be displayed without requiring any user response for processing to continue; that is, after the screen has been displayed, processing will continue immediately without waiting for any user input.

This command may be used to send messages about the progress of program execution to the user.

20 %Q - Suppress Next Input

```
%Q
```

In interactive processing, the `%Q` command is ignored.

`SET CONTROL 'Q'` causes the next `INPUT` statement *not* to be processed. This may be used, for example, if at the end of a helproutine the processing is to continue without the user having to press ENTER upon return from the help to the map.

21 %QS - Simultaneous Output of Multiple Screens

```
%QS
```

With this command, you can display multiple screens simultaneously.

`%QS` causes the next screen I/O not to be executed. The corresponding output screen is kept internally until the following I/O, when it is displayed together with the next screen. Therefore `%QS` only makes sense if the second output screen is a window, that is, if it does not entirely overlay the first screen that was suppressed with `%QS`.

Example: You can suppress the output of a screen A with `%QS`; the next screen B is a window which partially overlays screen A (perhaps a help window for one of the fields on screen A); with the next screen I/O, the window B and the "underlying" screen A are displayed simultaneously.

A `%QS` command only applies to the subsequent screen.

Note: As `%QS` reduces the number of screen I/Os, it also improves performance.

22 %R - Repeat INPUT Statement

```
%R
```

This command causes `INPUT` statement repetition and the output screen to be rebuilt. All output data generated from the beginning of the `INPUT` statement will be reproduced.

23 %T - Position Cursor to Top of Active Window

```
%T
```

This command positions the cursor at the top lefthand corner of the active window on the next screen output.

This command only works if there is an input field (session parameter `AD=A` or `AD=M`) at the top lefthand corner.

24 %Tll/cc - Position Cursor to Line ll, Column cc

`%Tll/cc`

This command positions the cursor at line *ll*, column *cc* on the next screen output.

The line and column positions are counted beginning with 1 within the current logical page. For this reason, the message line, the function-key lines, and the statistics line/infoline (if active) are not taken into account for the cursor position; that is, `%T1/1` always positions to the beginning of the topmost data line of the page (if visible).

You have to execute the terminal command `%T+` before you can place the cursor at any desired position on the screen.

`%Tll/cc` only works if there is an input field (session parameter `AD=A` or `AD=M`) at the specified position.

25 %T* - Position Cursor Outside Window

```
%T*
```

Normally, when a window is active and the window contains no input fields (`AD=A` or `AD=M`), the cursor is placed in the top left corner of the window.

This terminal command causes the cursor to be placed in a `*COM` system variable outside the window when the active window contains no input fields.

Command	Function
`%T*`	Switches between cursor placement in system variable `*COM` outside the window and standard cursor placement within the window.

`%T*` only applies to the next `INPUT` statement, and it must be issued *before* the `INPUT` statement.

26 %U - Translate Lower to Upper Case

```
%U
```

This command causes Natural to translate lower-case characters to upper-case for alphanumeric input data.

Upper-case translation is in effect by default.

`%U` influences the interactive input, which is entered, for example, with a Natural `INPUT` statement. It does not, however, influence the input from the stack.

Upper-case translation is not performed if `AD=W` has been specified for the input field (see the section *Field Upper/Lower Case Characteristics* in the description of the session parameter `AD`).

See also the terminal command `%L`.

27 %V - Control of Print Mode

Syntax Description:

Command	Function
%VON	Sets screen direction to right-to-left.
%VOFF	Sets screen direction to left-to-right.
%V	Toggles between left-to-right to right-to-left screen direction. For detailed information on how to use the terminal command %V, see *Bidirectional Language Support* in the *Unicode and Code Page Support* documentation.

For further information on print modes, see the session parameter PM.

28 %W - Window Processing

```
%W
```

Note: You are strongly recommended to use the `DEFINE WINDOW` statement instead of the `%W` command.

A Natural window is that segment of a logical page, built by a Natural program, which is displayed on the terminal screen.

The `%W` command controls the processing of this window.

The command must always be specified with parameters for the various functions as described hereafter. Multiple parameters may be specified with one `%W` command; they must be specified consecutively without any delimiter characters.

There is always a window present, although you may not be aware of its existence: unless specified differently (with a `%W` command or `DEFINE WINDOW` statement), the size of the window is identical to the physical size of your terminal screen.

See also the `DEFINE WINDOW` statement in the *Statements* documentation for information on window processing.

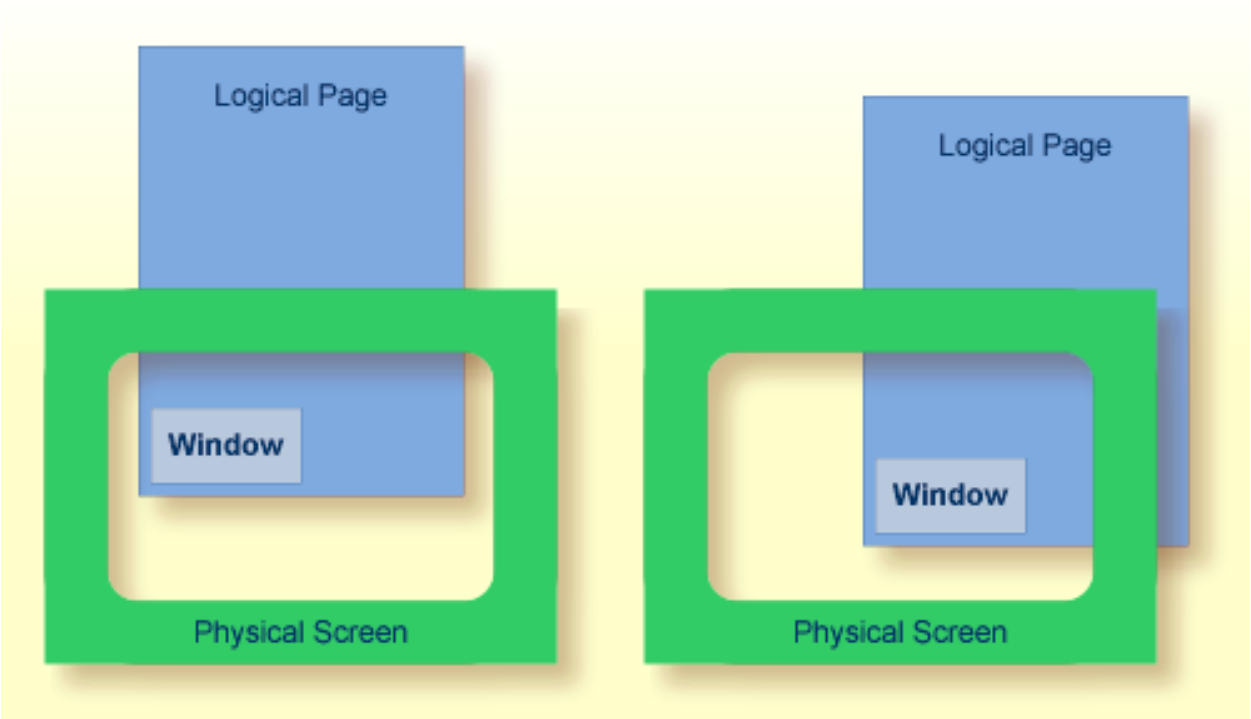
There are two types of window commands:

- commands to control the size and position of the window on the physical screen;
- commands to control the position of the window on the logical page created by the program.

Window Size and Position on the Physical Screen

The following window commands control the size and position of the window on the physical screen.

When you change the position of the window on the physical screen, the position of the window on the logical page will remain unchanged:



For information on possible window sizes, see the `DEFINE WINDOW` statement.

Command	Function
`%WB`	The window size (excluding frame) will be set to physical screen size. If a frame is defined, it will not be visible.
`%WB`*`lll`*`/`*`ccc`	The top left corner of the window will be positioned to line number *`lll`*, column number *`ccc`* (lines and columns are counted on the physical screen). The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
`%WBO`	The window will be positioned to the top left corner of the screen. The window size will remain unchanged.
`%W#`	The top left corner of the window will be positioned to the cursor position. The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
`%W?`	The bottom right corner of the window will be set to the cursor position. The top left corner of the window will remain unchanged, and the size of the window will be adjusted accordingly.
`%WL`*`nn`*	The line size (horizontal extension) of the window (including frame, if specified) will be set to *`nn`*. If *`nn`* is omitted or specified larger than would fit on the screen, the line size will be set to the maximum possible (that is, to the right edge of the screen).

Command	Function
%WC*nn*	The column size (vertical extension) of the window (including frame, if specified) will be set to *nn*. If *nn* is omitted or specified larger than would fit on the screen, the column size will be set to the maximum possible (that is, to the bottom of the screen).

Column size and line size specifications refer to the overall physical size of the window (including frame, if specified), not to the size of what is logically visible inside the window.

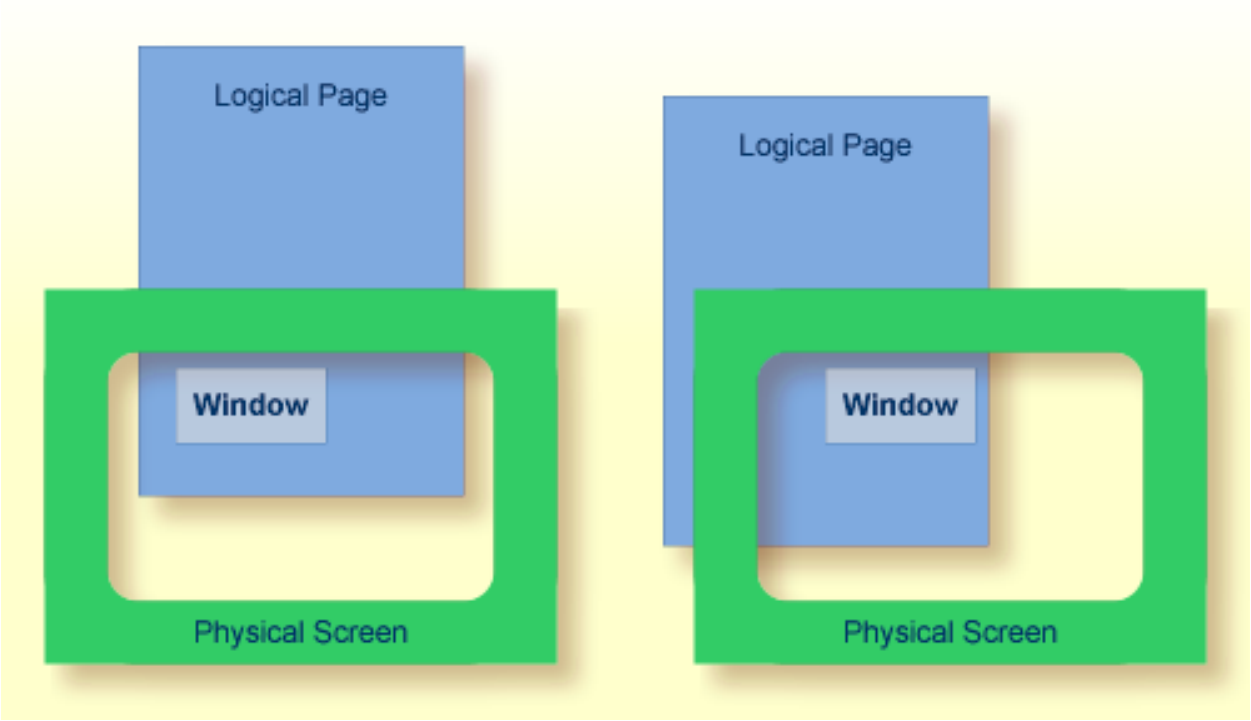
Any incorrect/impossible size or positioning commands will either be ignored or adjusted to the physical possibilities.

Command	Function
%WF	Switch on framing. The boundaries of the window will be indicated by a frame. If the window size is smaller than 4 lines by 12 columns, the frame will not be visible.
%WM	Switch off framing. The boundaries of the window will not be indicated by a frame. Switching off the frame does not change the size of the window (only the size of the page segment visible inside the window).
%WO	Suppress display of PF-key lines, message line and statistics line. This command only applies if the screen is a "real" window (that is, smaller than the physical screen). To cancel the effect of %WO, you issue %WO again (or %WD).
%WP	By default, the PF-key lines, the message line and the statistics line are displayed within a window. To display them on the screen outside the window, use %WP. To cancel the effect of %WP, you use %WD.
%WD	Cancels the effects of %WF, %WO and %WP (and of the TITLE option of the DEFINE WINDOW statement).
%WX	If there is a *COM field outside the window, this field is normally not write-protected. With %WX you can make it write-protected.
%WY	Cancels the effect of %WX.

Window Position on a Logical Page

The following window commands control the positioning of the window on the current logical page, that is, the current report/map produced by the Natural program for display. This logical page may be larger in size than the physical screen.

When you change the position of the window on the logical page, the size and position of the window on the physical screen will remain unchanged. In other words, the window is not moved over the page, but the page is moved "underneath" the window:



Unless specified differently by one of the following commands, the window will be placed at the top left corner of the logical page.

Command	Function
%W*	The position marked on the page by the cursor will be shifted to the top left corner of the window.
%W*lll*,*ccc*	The position of the logical page determined by line number *lll*, column number *ccc* will be shifted to the top left corner of the window. Lines and columns are counted on the logical page.
%W<	Shift window left. The number of positions shifted is equal to the line size (horizontal extension) of the window.
%W<<	Shift window to leftmost position of page.
%W<*n*	Shift window left *n* positions (0 =< *n* =< logical line size).
%W>	Shift window right. The number of positions shifted is equal to the line size (horizontal extension) of the window.
%W>>	Shift window to rightmost position of page.
%W>*n*	Shift window right *n* positions (0 =< *n* =< logical line size).
%W+	Shift window down. The number of lines shifted is equal to the number of lines in the window. (*)
%W++	Shift window to bottom of page. (*)
%W+*n*	Shift window down *n* lines (0 =< *n* =< logical page size). (*)
%W-	Shift window up. The number of lines shifted is equal to the number of lines in the window.

Command	Function
%W--	Shift window to top of page.
%W-*n*	Shift window up *n* lines (0 =< *n* =< logical page size).
%WH	By default, the position of the window on the logical page is reset to *top left corner* after a screen I/O. %WH prevents the window position from being reset by the next I/O, that is, the set window position will be retained. %WH only applies to the next I/O.
%WS	Switch on the STAY option; that is, control will "stay" on the current page until the end of the page. If a page is not yet completely shown in vertical direction, a string of VVVV will appear in the message line. The window will be scrolled downward with every ENTER until the end of the logical page is reached. The next ENTER will cause control to be returned to the program. (This does not apply to pages created by an INPUT statement with input fields (session parameter AD=A or AD=M).)
%WN	Switch off the STAY option. When you press ENTER, control is returned to the program.

* The window can be shifted to the last non-blank line of the page at most.

Notes:

1. If you wish to use one of the above commands within a program to shift the window, assign the command to a function key (with a SET KEY statement).
2. If you wish to specify it with a SET CONTROL statement, this statement must be followed by a REINPUT statement (that is, it must be placed between the REINPUT statement and corresponding INPUT statement); otherwise Natural will not be able to uniquely identify the window to which the command is to be applied (and will ignore it).
3. As a rule, however, no SET CONTROL 'W' statement should be placed between an INPUT statement with WINDOW='*window-name*' option and the corresponding REINPUT statement.

Examples of Command Combinations

The various parameters to be specified with the %W command may also be combined with one another; for example:

%W<<--	Position window to top left corner of page.
%W>>++	Position window to bottom right corner of page.
%W++-	Display the next to last window segment of the page.
%W+3>6	Position window 3 lines down and 6 positions right on the page.
%W10+>	Position window to line 10 of the page, then 1 window down and 1 window right.

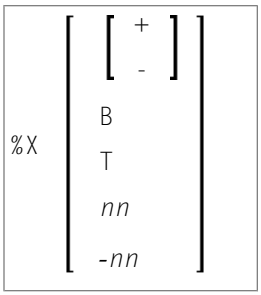
%WL40C10++-3	Define the window with a line size of 40 positions and a page size of 10 lines, move that window to the bottom of the page, then move it up 3 lines on the page.
%WL30C10B3/15--<<	Define the window with a line size of 30 positions, page size of 10 lines, position the window at line 3 column 15 on the physical screen and move this window to the top left corner of the page.
%WFS	Generate a frame around the window and set the STAY option on.

The parameters are evaluated in the sequence in which they are specified, so that different sequences of the same parameters may lead to different results.

%WA and %WZ - Save Screen Image before Window

Command	Function
%WA	With %WA, you activate a "save screen image before window" feature. When this feature is activated and a window is to be opened, all active screen data that will be overlaid by the window are saved. When the same window is moved, the saved screen image is reconstructed before the window is built up at the new location on the screen. In addition, it is possible to rebuild the saved images of multiple dependent windows whenever the calling window becomes active again. When the current INPUT statement uses a window, the screen image is stored before the window is output. Whenever the same INPUT statement is repeated, the current or all subsequent stored screen images are recovered and are written back to the screen. This feature makes it possible, for example, to use windows in a PC-like manner. For a given window, any number of dependent windows can be written to the screen. All these windows will disappear from the screen when the main input window is re-executed. The buffer contents (screen images) are deleted whenever Natural performs a full-screen I/O, when Natural returns to command mode (NEXT), after a LOGON command, or after the CLEAR key has been pressed.
%WZ	With %WZ, you deactivate a previously entered %WA command.

29 %X - Control of Infoline



This terminal command controls the display of the Natural **infoline**.

Command	Function
%X+	Switches the display of the infoline on.
%X-	Switches the display of the infoline off.
%X	Switches the display of the infoline on and off (toggle switch).
%XB	Displays the infoline in the bottom line of the screen.
%XT	Displays the infoline in the top line of the screen.
%X*nn*	Displays the infoline on line *nn* of the screen. If the line number *nn* is not within the current screen, the infoline will not be displayed.
%X-*nn*	Displays the infoline on the *nn*th line from the bottom of the screen. If the line number -*nn* is not within the current screen, the infoline will not be displayed.

Infoline

Data can be written to the infoline by specifying the output destination `INFOLINE` with the `DEFINE PRINTER` statement. Only a single line can be written to the infoline. The infoline can be used to have status information displayed, for example, for debugging purposes; it can also be used as separator line (as defined by SAA standards).

30 %Z - Clear Source Area

```
%Z
```

This command clears the contents of the Natural source area.

It can only be used with the `SET CONTROL` statement.