

# %W - Window Processing

**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.

This chapter covers the following topics:

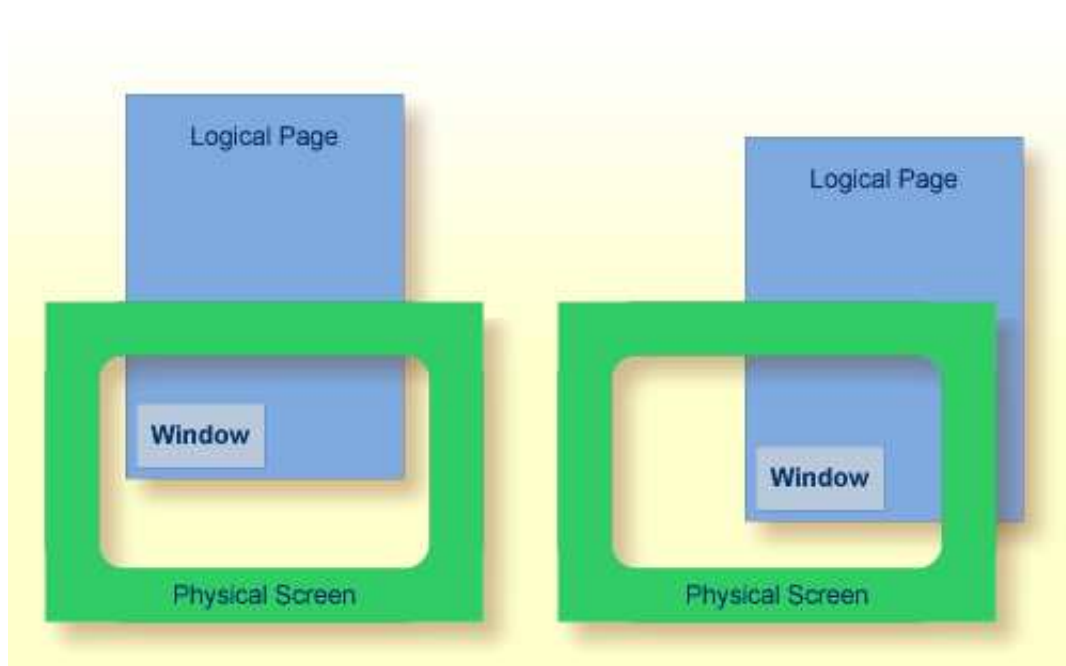
- Window Size and Position on the Physical Screen
- Window Position on a Logical Page
- `%WA` and `%WZ` - Save Screen Image before Window

---

## 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 <i>lll</i> / <i>ccc</i>	The top left corner of the window will be positioned to line number <i>lll</i> , column number <i>ccc</i> (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.
%WB0	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 <i>nn</i>	The line size (horizontal extension) of the window (including frame, if specified) will be set to <i>nn</i> .  If <i>nn</i> 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).
%WC <i>nn</i>	The column size (vertical extension) of the window (including frame, if specified) will be set to <i>nn</i> .  If <i>nn</i> 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.

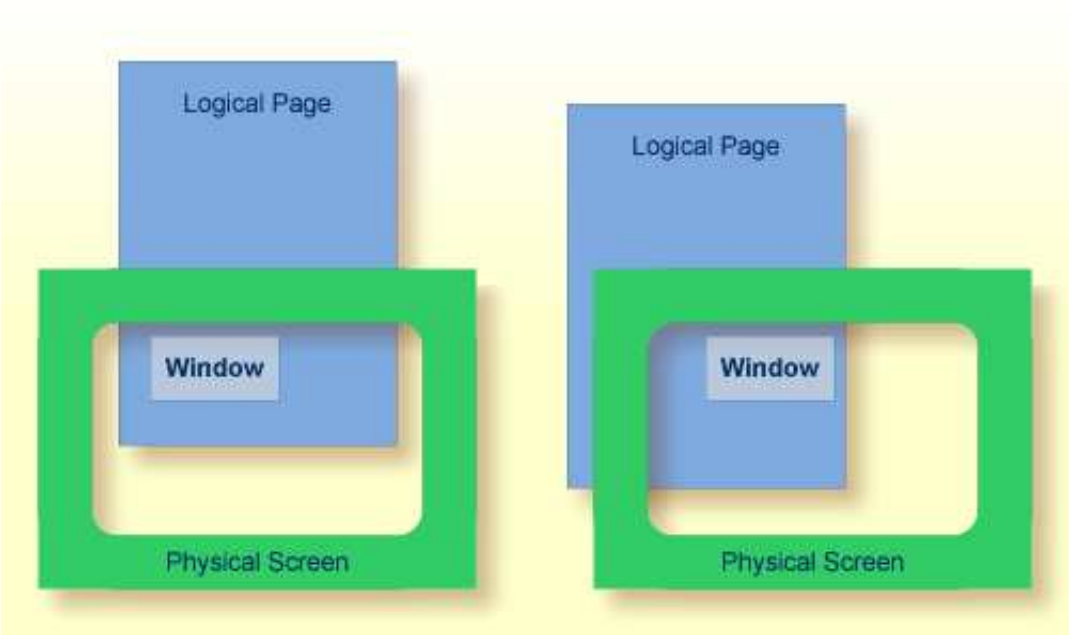
Any incorrect/impossible size or positioning commands will either be ignored or adjusted to the physical possibilities.

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 13 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.
%Wlll,ccc	The position of the logical page determined by line number <i>lll</i> , column number <i>ccc</i> 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 <i>n</i> positions ( $0 \leq n \leq$ 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 <i>n</i> positions ( $0 \leq n \leq$ 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 <i>n</i> lines ( $0 \leq n \leq$ logical page size). (*)
%W-	Shift window up. The number of lines shifted is equal to the number of lines in the window.
%W--	Shift window to top of page.
%W-n	Shift window up <i>n</i> lines ( $0 \leq n \leq$ logical page size).
%WH	By default, the position of the window on the logical page is reset to <i>top left corner</i> 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.

When you specify multiple parameters with the %W command, note that the maximum number of characters after the percent sign (%) is 24; any further characters will be ignored.

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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>
%WZ	With %WZ, you deactivate a previously entered %WA command.