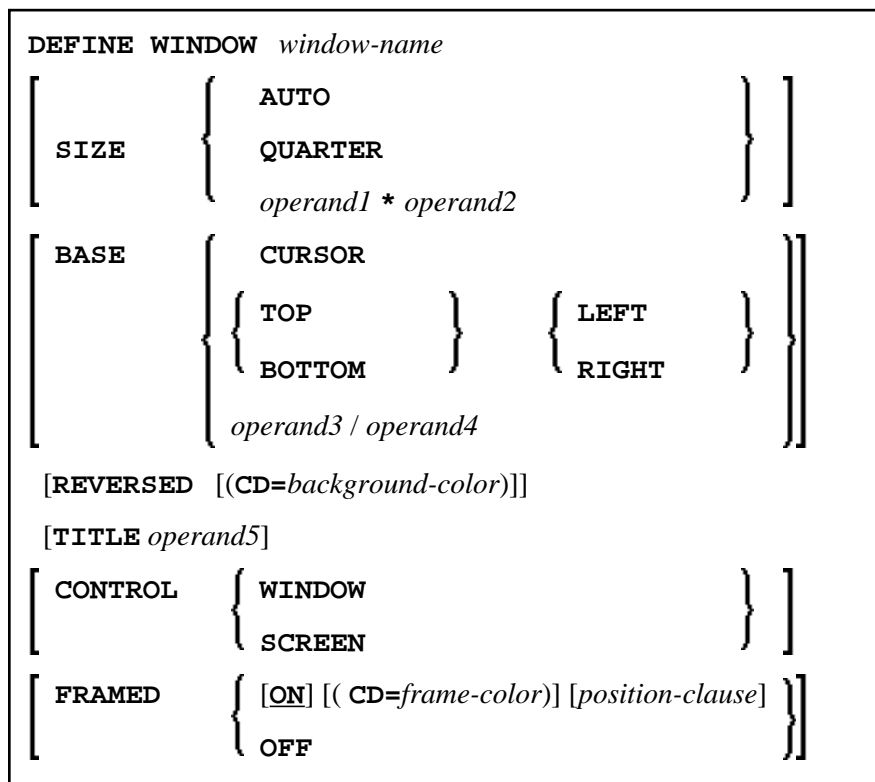


DEFINE WINDOW



This chapter covers the following topics:

- Function
- Syntax Description
- Protection of Input Fields in a Window
- Invoking Different Windows
- Example

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

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

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

Function

The `DEFINE WINDOW` statement is used to specify the size, position and attributes of a window.

A window is that segment of a logical page, built by a program, which is displayed on the terminal screen. There is always a window present, although you may not be aware of its existence: unless specified differently, the size of the window is identical to the physical size of your terminal screen.

SIZE	<p>With the <code>SIZE</code> clause, you specify the size of the window.</p> <p>Note: On mainframe computers, Natural requires additional columns for so-called attribute bytes to be able to display data on the screen (on other platforms, such attribute bytes are not needed). Consequently, on mainframe computers the screen area overlaid by a window is wider, and the size of the page segment visible inside a window is smaller than on other platforms.</p> <p>Example: Assume a window whose size is defined as <code>SIZE 5 * 15</code> (that is, with a width of 15 columns):</p> <ul style="list-style-type: none"> ● On mainframe computers, the screen area overlaid by the window is 16 columns; the size of what is visible inside the window is 14 columns without frame, and 10 columns with frame respectively. ● On other platforms, the screen area overlaid by the window is 15 columns; the size of what is visible inside the window is 15 columns without frame, and 13 columns with frame respectively.
SIZE AUTO	<p>The size of the window is determined automatically by Natural at runtime. The size is determined by the data generated into the window as follows:</p> <ul style="list-style-type: none"> ● The number of window lines will be the number of <code>INPUT</code> lines generated (plus possibly the PF-key lines, message line, and infoline/statistics line). ● The number of window columns is determined by the longest <code>INPUT</code> line: Natural scans, starting from the ends of the lines, for the rightmost significant byte in a line. This may cause an input-only or modifiable field (<code>AD=A</code> or <code>AD=M</code>) to be truncated; to avoid this, you either put a single-character text string after such a field or explicitly set the window size with the following: <p style="text-align: center;"><code>SIZE operand1 * operand2</code></p> <p>If you omit the <code>SIZE</code> clause, <code>SIZE AUTO</code> applies by default.</p> <p>Note: The title is not part of the window data. Therefore, if the window size has been determined as described above <i>and</i> the title is longer than the window, it will be truncated.</p>
SIZE QUARTER	<p>The size of the window will be one quarter of the physical screen.</p>

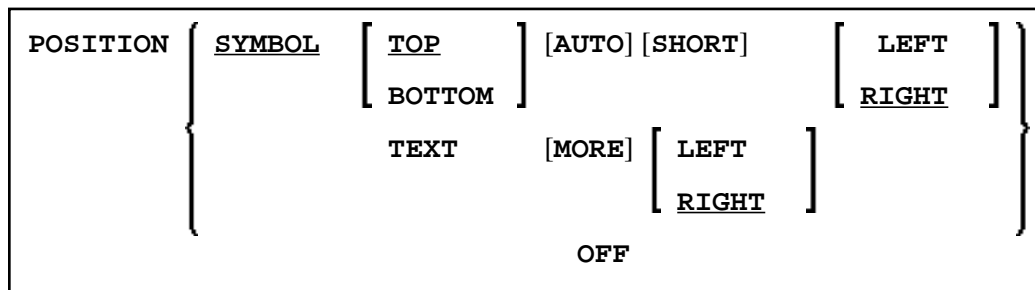
<p>SIZE <i>operand1 * operand2</i></p>	<p>The size of the window will be <i>n</i> lines by <i>n</i> columns. The number of lines is determined by <i>operand1</i>, the number of columns by <i>operand2</i>. Neither of the two operands must contain decimal digits.</p> <p>If the window is <code>FRAMED</code>, the specified size will be inclusive of the frame.</p> <p>The minimum possible window size is:</p> <ul style="list-style-type: none"> ● without frame: 2 lines by 10 columns, ● with frame: 4 lines by 13 columns. <p>The maximum possible window size is the size of the physical screen.</p>
<p>BASE</p>	<p>With the <code>BASE</code> clause, you determine the position of the window on the physical screen. If you omit the <code>BASE</code> clause, <code>BASE CURSOR</code> applies by default.</p>
<p>BASE CURSOR</p>	<p>Places the top left corner of the window at the current cursor position. The cursor position is the physical position of the cursor on the screen. If the size of the window makes it impossible to place the window at the cursor position, Natural automatically places the window as close as possible to the desired position.</p>
<p>BASE TOP/BOTTOM LEFT/RIGHT</p>	<p>Places the window at the top-left, bottom-left, top-right, or bottom-right corner respectively of the physical screen.</p>
<p>BASE <i>operand3/operand4</i></p>	<p>This places the top left corner of the window at the specified line/column of the physical screen. The line number is determined by <i>operand3</i>, the column number by <i>operand4</i>. Neither of the two operands must contain decimal digits.</p> <p>If the size of the window makes it impossible to place the window at the specified position, you will get an error message.</p>
<p>REVERSED</p>	<p><code>REVERSED</code> will cause the window to be displayed in reverse video (if the screen used supports this feature; if it does not, <code>REVERSED</code> will be ignored).</p>
<p>REVERSED <i>CD= background-color</i></p>	<p>This will cause the window to be displayed in reverse video and the background of the window in the specified color (if the screen used supports these features; if it does not, the respective specification will be ignored).</p> <p>For information on valid color codes, see the session parameter <code>CD</code> in the <i>Parameter Reference</i>.</p>

TITLE <i>operand5</i>	<p>With the TITLE clause, you may specify a heading for the window. The specified title (<i>operand5</i>) will be displayed centered in the top frame-line of the window. The title can be specified either as a text constant (in apostrophes) or as the content of a user-defined variable. If the title is longer than the window, it will be truncated. The title is only displayed if the window is FRAMED; if FRAMED OFF is specified for the window, the TITLE clause will be ignored.</p> <p>Note: If the title contains trailing blanks, these will be removed. If the first character of the title is a blank, one blank will automatically be appended to the title.</p>
CONTROL	<p>With the CONTROL clause, you determine whether the PF-key lines, the message line and the statistics line are displayed in the window or on the full physical screen.</p>
CONTROL WINDOW	<p>CONTROL WINDOW causes the lines to be displayed inside the window. If you omit the CONTROL clause, CONTROL WINDOW applies by default.</p>
CONTROL SCREEN	<p>CONTROL SCREEN causes the lines to be displayed on the full physical screen outside the window.</p>
FRAMED	<p>By default, i.e. if you omit the FRAMED clause, the window is framed.</p> <p>The top and bottom frame lines are cursor-sensitive: where applicable, you can page forward, backward, left or right within the window by simply placing the cursor over the appropriate symbol (<, -, +, or >; see <i>position-clause</i> below) and then pressing ENTER. If no symbols are displayed, you can page backward and forward within the window by placing the cursor in the top frame line (for backward positioning) or bottom frame line (for forward positioning) and then pressing ENTER.</p> <p>Note: If the window size is smaller than 4 lines by 12 (or 13 on mainframe computers) columns, the frame will not be visible.</p>
FRAMED OFF	<p>If you specify FRAMED OFF, the framing and everything attached to the frame (window title and position information) will be switched off.</p>

<p>FRAMED (CD=<i>frame-color</i>)</p>	<p>This causes the frame of the window to be displayed in the specified color (if the screen used is a color screen; if it is not, the color specification will be ignored).</p> <p>For information on valid color codes, see the session parameter CD (in the <i>Parameter Reference</i>).</p> <p>Note: In Natural for Windows, this specification is ignored.</p>
<p><i>position-clause</i></p>	<p>The POSITION clause is only evaluated on mainframe computers: on all other platforms it is ignored. For details, refer to <i>Position Clause</i> below.</p>

POSITION Clause

The POSITION clause is only evaluated on mainframe computers: on all other platforms it is ignored.



The POSITION clause causes information on the position of the window on the logical page to be displayed in the frame of the window. This applies only if the logical page is larger than the window; if it is not, the POSITION clause will be ignored. The position information indicates in which directions the logical page extends above, below, to the left and to the right of the current window.

If the POSITION clause is omitted, POSITION SYMBOL TOP RIGHT applies by default.

POSITION SYMBOL	Causes the position information to be displayed in form of symbols: <code>More: < - + ></code> . The information is displayed in the top and/or bottom frame line.
TOP/BOTTOM	Determines whether the position information is displayed in the top or bottom frame line.
AUTO	Is only applicable if the logical page is fully visible in the window as far as its horizontal size is concerned, that is, if only a minus sign character (-) and/or a plus sign character (+) are to be displayed. In this case, <code>AUTO</code> automatically switches from the symbols to the words <code>Top</code> , <code>Bottom</code> and <code>More</code> respectively.
SHORT	Causes the word <code>More:</code> before the symbols <code>< - + ></code> to be suppressed.
LEFT/RIGHT	Determines whether the position information is displayed in the left or right part of the frame line.
POSITION TEXT	Causes the position information to be displayed in text form. The information is displayed in the top and/or bottom frame line with the words <code>More</code> , <code>Top</code> and <code>Bottom</code> . The text is language-dependent and may also be displayed in another language if the language code is set accordingly.
POSITION TEXT MORE	Suppresses the words <code>Top</code> and <code>Bottom</code> and only displays the word <code>More</code> where applicable, i.e., in the top or bottom frame line or both.
LEFT/RIGHT	Determines whether the position information is displayed in the left or right part of the top frame line.
POSITION OFF	Causes the position information to be suppressed; no position information will be displayed.

Protection of Input Fields in a Window

The following rules apply to input fields (`AD=A` or `AD=M`) which are not entirely within the window:

- 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`) do not count as part of the value.

If you wish to access input fields thus protected, you have to adjust the window size accordingly so that the beginning of the field/end of the value is within the window.

Invoking Different Windows

A `DEFINE WINDOW` statement must not be placed within a logical condition statement block. To invoke different windows depending on a condition, use different `SET WINDOW` statements (or `INPUT` statements with a `WINDOW` clause respectively) in a condition.

Example

```

** Example 'DWDEX1': DEFINE WINDOW
*****
DEFINE DATA LOCAL
01 #I (P3)
END-DEFINE
*
SET KEY PF1='%W<<' PF2='%W>>' PF4='%W--' PF5='%W++'
*
DEFINE WINDOW WIND1
    SIZE QUARTER
    BASE TOP RIGHT
    FRAMED ON POSITION SYMBOL AUTO
*
SET WINDOW 'WIND1'
FOR #I = 1 TO 10
    WRITE 25X #I 'THIS IS SOME LONG TEXT' #I
END-FOR
*
END
    
```

Output of Program DWDEX1:

```

> r                                     +-----More:      + >+
All      ....+....1....+....2....+....3.. !           1           !
0010 ** Example 'DWDEX1': DEFINE WIND !           1 THIS !
0020 ***** !           2 THIS !
0030 DEFINE DATA LOCAL !           3 THIS !
0040 01 #I (P3) !           4 THIS !
0050 END-DEFINE !           5 THIS !
0060 * !           6 THIS !
0070 SET KEY PF1='%W<<' PF2='%W>>' PF !           7 THIS !
0080 * ! MORE !
0090 DEFINE WINDOW WIND1 +-----+
0100     SIZE QUARTER
0110     BASE TOP RIGHT
0120     FRAMED ON POSITION SYMBOL AUTO
0130 *
0140 SET WINDOW 'WIND1'
0150 FOR #I = 1 TO 10
0160     WRITE 25X #I 'THIS IS SOME LONG TEXT' #I
0170 END-FOR
0180 *
0190 END
0200
    ....+....1....+....2....+....3....+....4....+....5....+... S 19   L 1
    
```