

Program Editor

You use the program editor to write and maintain Natural programs, subprograms, subroutines, classes, copycode, help routines and text elements. You can open multiple editing sessions, making it possible to copy or move content from one object to another. The title bar of each editing session displays the name and type of the object. If the object is new and has not yet been saved, it is automatically given the title "Untitled".

The following topics are covered below

- Modifying Program Contents
- Finding Source Contents
- Editing/Listing Referenced Natural Objects
- Splitting the Editor Window
- Expanding/Collapsing Object Listings
- Recording/Replaying Keystrokes
- Using Context-Sensitive Help
- Setting Editor Options

See also:

- Program Editor Accelerators
 - Renumbering of Source-Code Line Number References
-

Modifying Program Contents

- Selecting Text
- Copying Text
- Cutting Text
- Pasting Text
- Deleting Text
- Undoing/Redoing Text Changes
- Renumbering a Program

Selecting Text

To select a single word using the mouse

- Double-click the word.

To select a range of text using the mouse

1. Point to the first character to be selected.
2. Drag the cursor to the last character you want to select.
3. Release the mouse button.

To select text using the keyboard

1. With the arrow keys, move the selection cursor to the first character to be selected.
2. Press and hold down **SHIFT** and use the arrow keys to select the text.

▶ **To select the entire editor contents**

- From the **Edit** menu, choose **Select all**.

▶ **To cancel a mouse selection**

- Click anywhere in the document.

▶ **To cancel a keyboard selection**

- Press an arrow key.

Copying Text

Text can be copied within the same object or between two different objects.

▶ **To copy text**

1. Select the text using the instructions provided in Selecting Text.
2. From the **Edit** menu, choose **Copy**.
Or click the **Copy** toolbar button.
Or press **CTRL+C**.

The text is copied to the clipboard and can now be pasted within the same object or another object. For instructions on pasting text, see Pasting Text.

Cutting Text

The cut function can be used to delete text from an object or to move text within/between objects. When text is cut, it is taken from the object and placed on the clipboard. It remains there until the next cut or copy operation is performed, at which time it is irretrievably discarded from the clipboard to make way for the next cut/copied text.

It is possible to revoke a cutting operation after it has been performed. See Undoing/Redoing Text Changes.

▶ **To cut text**

1. Select the text using the instructions provided in Selecting Text.

From the **Edit** menu, choose **Cut**.

Or click the **Cut** toolbar button.

Or press **CTRL+X**.

The text is cut to the clipboard and can now be pasted within the same object or another object. For instructions on pasting text, see Pasting Text.

Pasting Text

The paste function is used to place text at a specific position within an editor after it has been copied or cut to the clipboard from another position within the same object or another object. A text which has been copied or cut to the clipboard can be pasted repeatedly without recopying it.

It is possible to revoke a paste operation after it has been performed. See Undoing/Redoing Text Changes.

To paste text

1. Copy or cut a portion of text as described in Copying Text and Cutting Text.
2. Select with the I-beam pointer the position in the text where the text is to be inserted.
3. From the **Edit** menu, choose **Paste**.
Or click the **Paste** toolbar button.
Or press **CTRL+V**.
The text is pasted to the object.
4. To paste the same text again, repeat Steps 2 and 3.

Deleting Text

When text is deleted, it is cut from the object but is **not** placed on the clipboard. The only way to recover deleted text is by undoing the deletion. See Undoing/Redoing Text Changes. Note that when text is deleted in the program editor, no warning is provided. This is intentional.

To delete text

1. Select the text using the instructions provided in Selecting Text.
2. From the **Edit** menu, choose **Delete**.
Or click the **Delete** toolbar button.
Or press **DEL**.

The text is deleted.

Undoing/Redoing Text Changes

The text operations you perform in the program editor can be revoked by applying the undo function or reinstated (after being undone) by applying the redo function. Typical operations which can be undone/redone are character input, character deletion, text deletion, text pasting, search/replace, or any action which modifies editor contents. The number of operations which can be undone/redone is determined by the value specified in the preferences for the program editor and is limited by the memory allocated (for more information, see Setting Editor Options).

The commands **SAVE**, **CLOSE** and **CLEAR** cause the undo/redo buffer to be cleared of its contents. A successful **STOW** also clears the buffer of its contents.

Undo / redo operations restore line numbering to its status before the operation.

To undo a text operation

- From the **Edit** menu, choose **Undo**.
Or press **CTRL+Z**.
Or click the **Undo** toolbar button.
The text is restored to its condition before the previous text operation or redo operation.

To redo a text operation which has been undone using an undo operation

- From the **Edit** menu, choose **Redo**.
Or press **CTRL+Y**.
Or click the **Redo** toolbar button.
The text is restored to its condition before the previous undo operation.

Renumbering a Program

As you add lines to a program, Natural numbers the added lines in increments of one. You never have to worry about two lines being assigned the same number. Natural recognizes such conditions and renumbers the program accordingly.

You can renumber a program at any time during an editing session. Every line of the object is renumbered, beginning with 0010 at the first line and increasing by increments of ten for each line.

To renumber a program

- From the **Edit** menu, choose **Renumber**.
Or, in the command line, type "Renumber" and press **ENTER**.
The program is renumbered.

Finding Source Contents

Searching for Source Text

In large objects, it is often difficult to locate a section of source code. Using the search function, you can flexibly search for any character string in source listings. If it should be necessary to replace a frequently occurring text string with another, you can use the combined search and replace function.

Searching for Source Text

To search for a text string in the active program window

1. From the **Edit** menu, choose **Find**.
Or click the **Find** toolbar button.
Or press **CTRL+F**.
The "Find" dialog box appears.
2. In the "Find" text box, enter the string to be searched for.
3. If you want the search to be case sensitive, select the "Case Sensitive" check box.
The search will then only find the string exactly as it appears in the "Find" text box, otherwise both upper case and lower case occurrences of the string will be found.
4. If you want the search string to be found as a whole word only and not as part of other words, select the "Whole Words Only" text box.
If this box is left unselected, all occurrences of the string will be found.
5. If the search is being performed on an object listing, then the "Exclude collapsed blocks" check box is displayed. Select the check box to exclude collapsed blocks of source code from the search; leave blank to search the entire listing.
For more information on expanding and collapsing object listings, see Expanding/Collapsing Object Listings.
6. In the "Direction" group frame, click the search direction up or down to specify whether the search will be conducted from the cursor position to the end of the object or from the cursor position to the beginning of the object. The default is "Down".
7. Choose **OK**.
If no instance of the text searched for is found, a corresponding message is displayed.
If an instance of the search string is found, it is displayed and selected.

To search for additional instances of the search string in the object

- From the **Edit** menu, choose **Find Next**.
Or press **F3**.
Alternatively, the text to find can be entered in the find combo box at the **Edit** toolbar.

Searching for and Replacing Source Text

To search for and replace a text string in the active program window

1. From the **Edit** menu, choose **Replace**.
Or click the **Replace Text** toolbar button.
Or press **CTRL+H**.
The "Replace" dialog box appears.
2. In the "Find" text box, enter the string to be searched for.
3. In the "Replace With" text box, enter the replacement string.
4. If you want the search to be case sensitive, select the "Case Sensitive" check box.
The search will then only find the string exactly as it appears in the "Search For" text box, otherwise both upper case and lower case occurrences of the string will be found.
5. If the search string to be found as a whole word only and not as part of other words, select the "Whole Words Only" text box.
If this box is left unselected, all occurrences of the string will be found.
6. If you want to confirm each change, leave the "Confirm Each Change" check box selected. If you deselect it, all instances of the search string will be converted to the replacement string without prompting you for confirmation.
7. In the "Direction" group frame, choose the search direction up or down to specify whether the search and replace will be conducted from the cursor position to the end of the object or from the cursor position to the beginning of the object. The default is "Down".
8. Choose **Replace**.
If no instance of the text searched for is found, a corresponding message is displayed.
If an instance of the search string is found, and the "Confirm Each Change" box is marked as in Step 6, a message box with the following pushbuttons is displayed:

Replace Replace the search string with the replacement string and continue searching.

Skip Skip to the next instance of the search string.

Cancel Interrupt the search/replace operation.

Repeat Replace

If for any reason you interrupt the search/replace operation, you can resume it at any time using the Replace Next function.

1. To do so, from the **Edit** menu, choose **Replace Next**.
2. Or press **CTRL+F3**.

Searching for a Line Number

If you know which line you are searching for, you can use the "Go To" command to display a specific line number. In the function, you can specify whether you want to go to the numbered line or the physical line in the editor, depending on whether the line number option is switched on or off.

If you use the "Go To" command in a source listing, and the line specified is within a collapsed block, the block is expanded to show the desired line. For more information on expanding and collapsing object listings, see Expanding/Collapsing Object Listings.

To go to a specific line

1. From the **Edit** menu, choose **Go To**.
Or click the **Go To** toolbar button.
Or press **CTRL+G**.
The "Go To" dialog box appears.
2. If the line number option is turned on (program line numbers are displayed), select the **Numbered line** radio button. If the line number option is turned off (program line numbers are not displayed), select **Physical line**.
The physical line number is the default.
3. Enter the line number to be found in the "Line Number" text box.
4. Choose **Go To**.
The editor scrolls to the specified line and the cursor is placed at the beginning of the line.

Editing/Listing Referenced Natural Objects

While you are working in the program editor, you can open or list other Natural objects which are referenced in the program code, assuming these objects exist. If, for example, you are editing a program that calls a subprogram, you can open the subprogram, adapt it to the program, and return to the program.

▶ To edit a referenced Natural object in your program

1. Click or double-click on the object.
2. From the **Program** menu, choose **Open Object**.
Or press **CTRL+O**.
The editor is opened.

▶ To list the source code of a referenced Natural object in your program

1. Click or double-click on the object.
2. From the **Program** menu, choose **List Object**.
The source code of the object is listed.

Splitting the Editor Window

Jumping between Split Screens

You can split the program editor window vertically or horizontally to view and modify two different parts of the object simultaneously. This feature saves you the trouble of printing a program to view two different sections simultaneously. Changes made in one section are made simultaneously in the other section.

▶ To split the screen into two sections vertically or horizontally

1. From the **View** menu, choose **Vertical split (Horizontal split)**.
A vertical (horizontal) line appears in the middle of the window.
2. Split the screen into two sections by moving the mouse vertically (horizontally) to the position you want and clicking the left mouse button.
The screen is split into two sections, each displaying the same information. You can now scroll each section individually and edit both sections as if they were part of the same object (as indeed they are).

▶ To exit split-editor mode and return to a single screen

- From the **View** menu, choose **Unsplit**.
Two editor sections are transformed into one.

Jumping between Split Screens

▶ To jump between vertical/horizontal split screens

- Place the mouse pointer in the screen section of your choice and click.
Or press **F6**.
The cursor moves from one screen section to the other.

Expanding/Collapsing Object Listings

To improve the readability and maintainability of objects with complex program structures, you can display logical blocks of source code in expanded or collapsed form. Moreover, each individual logical block of code in an object can be expanded or collapsed, as required. Expandable/collapsible structures include, for example, DEFINE DATA blocks, REPEAT blocks, IF THEN ELSE blocks and READ blocks.

This feature applies only to source code listings of Natural objects other than DDMs which are saved in structured mode.

In the source code:

- square icons with minus signs (-) indicate the beginning of a collapsible block of code.
- square icons with plus signs (+) indicate the beginning of an expandable block of code.

The following topics are covered below:

- Making Listings Expandable/Collapsible
- Collapsing and Expanding Program Structures
- Additional Information

Making Listings Expandable/Collapsible

Object listings are not displayed by default in expandable/collapsible format. You must specify in the program editor that objects are to be listed in expandable/collapsible format. You can specify this on either session level or object level. Session-level settings are valid for all new listings in a Natural session; temporary settings are valid only for the active listing; they are lost once a listing is closed. If the listing is reopened, the session-level settings are used.

Expand/Collapse: Session Level

To set expandable/collapsible listings on a session-wide basis

1. From the **Object** menu, choose **New** and then, from the cascading menu, **Program**.
2. From the **Tools** menu, choose **Options**.
Or press **Alt+ENTER**.
In the "Options" dialog box select the tab "Program Editor"
3. Select the "Expand/Collapse" check box.
The "Open collapsed" check box is activated.
4. Select the "Open collapsed" check box if listings are to be initially displayed in collapsed format. Leave empty to display the listings initially in expanded (normal) format.
5. Choose **OK**.

Collapsing and Expanding Program Structures

When a block of object code is collapsed, all of the lines of source code between the block begin statement and block end statement are hidden from view, including any other blocks if they are part of the chosen block. Hidden blocks retain their collapsed or expanded status.

To collapse a program structure

- Double-click on the "-" icon marking the collapsible block of program code.
Or place the cursor in the line containing the "-" icon and, from the **View** menu, choose **Collapse Block**.
The block of code is collapsed.
When a block of object code is expanded, all of the lines of source code between the block begin statement and block end statement are brought into view, including any other blocks if they are part of the chosen block.
Previously hidden blocks retain the collapsed or expanded status they had before they were hidden.

To expand a program structure

- Double-click on the "+" icon marking the collapsible block of program code.
Or place the cursor in the line containing the "+" icon and, from the **View** menu, choose **Expand Block**.
The block of code is expanded.

Additional Information

When searching a listed object, it is possible to exclude collapsed blocks from the search. For more information, see [Searching for Source Text](#).

When you are searching for a line number in a listed object which is part of a collapsed block, the block is expanded to display the line. For more information, see [Searching for a Line Number](#).

When you are splitting the editor window vertically or horizontally, one section can be displayed in collapsed format while the other is displayed in expanded format. From the **Options** menu, just choose **Expand/Collapse** for one of the window sections.

Recording/Replaying Keystrokes

You use the record/replay function to record and replay keystroke sequences which you want to repeat in the program editor. This is similar to a tape recorder with the functions record, stop, and play.

As the name implies, the keystroke recorder records and replays only input made directly from the keyboard, and not movements or selections made with the mouse. Thus, it is only possible to record within a single program editor window; recording in multiple program editor sessions is not possible.

To record a keystroke sequence

1. Place the cursor at the position in the editor where you want to begin recording.
2. From the **Tools** menu, choose **Start Recording**.
Or press **Ctrl+Shift+R**.
3. Enter the key sequence you want to record.
Only keyboard input is recorded. Mouse movements and operations are ignored.

To stop recording

- From the **Tools** menu, choose **Stop Recording**.
Or press **Ctrl+Shift+S**.
Recording ends.

To replay a recorded keystroke sequence

1. Place the cursor at the position in the editor where you want to begin replaying the recording.
2. From the **Tools** menu, choose **Replay Recording**.
Or press **Ctrl+Shift+P**.
The recording is replayed.

Using Context-Sensitive Help

- Syntax Help
- Syntax Coloring

Syntax Help

Within the Natural program editor, context-sensitive help is available for the following Natural syntax elements:

- Statements
- System variables
- System functions
- Parameters (for example, the AD parameter)



To call syntax help

- Place the cursor on a keyword within a syntax element for which you require help and press **F1**.
Or double-click on a keyword within the syntax element and press **F1**.

Help for Statements

When you place the cursor on a keyword that forms part of a statement (for example, under the keyword TOP in the statement AT TOP OF PAGE), the help system tries to identify the statement that contains the keyword.

- If the result is unambiguous (that is, one complete statement is identified), help information is displayed for that particular statement.
- If a keyword is found which is used in several statements (for example, AT), a menu appears offering you a choice of statements containing the keyword you specified.
- If no complete statement can be found (for example, if you have only typed in the word DEFINE and pressed **F1** before typing out the full statement) or if the keyword is used in more than one statement, a dialog box appears offering you a choice of all statements beginning with or containing the keyword you specified.

Syntax Default Colors

Coloring is used in the program editor to mark various elements of syntax for better readability.

The default color assignments for Natural are:

- Blue - Natural keywords
- Red - Comments
- Green - Text constants, system functions, system variables
- Black - User-defined variables and remaining syntax elements

You can modify the color assignments. See Syntax Coloring.

Setting Editor Options

You can set preferences for various editor options. These settings are taken as default values each time you start the Program Editor.

To do so

- From the **Tools** menu, select **Options** and then in the options dialog select the tab "Program Editor".

Status Bar

Display the status line at the top of the editor window. For information on the status line contents, see Status Bar Information.

Line Numbers

Show the line numbers in the program editor.

Syntax coloring

Display color-coding for program syntax elements.

On how to modify color-coding, see Syntax Coloring.

Vertical scroll bar

Display the vertical scroll bar for the editor window.

Horizontal scroll bar

Display the horizontal scroll bar for the editor window.

Tabs

Specify the column numbers for tabulator stops in the program editor. You can add or modify existing tabulator stops in the text box.

Expand/Collapse

Enable logical blocks of code in object listings to be switched between a collapsed form and an expanded form. This is valid, for example, for DEFINE DATA blocks, REPEAT blocks, and IF THEN ELSE blocks. This option applies only to objects saved in structured mode. For more information on the expand/collapse functions, see Expanding/Collapsing Object Listings.

Open collapsed

Input to this check box is allowed only if the "Expand/Collapse" check box is marked. Initially display logical blocks of code in collapsed form on listing a program.

Max. number of actions

Specify the number of text operations which can be recalled/reinstated using the undo/redo functions. Zero signifies the maximum limit dependent only on the specified maximum memory size (see below). For more information on the undo/redo functions, see Undoing / Redoing Text Changes.

Max. memory size

Specify the amount of buffer space available for storing text operations made in the program editor. Zero signifies the maximum space available.

Alarm

Produce a "beep" sound when an invalid key or key combination is pressed.

Insert Alignment

Align the cursor with the first non-blank character of the previous line of text when you press **ENTER** or **RETURN**. If there are no non-blank characters before the inserted line, the cursor is aligned with column 1.

Renumber Before Save

Renumber the lines in a program and update line number references before every save.

Syntax Coloring

Coloring is used in the program editor to mark various elements of syntax for better readability.

You can define your own colors.

To do so

- From the "Program Editor Preferences" dialog, choose "Colors".
The "Color Definition" window is displayed.
You can define the colors of text type, as for example "Keywords", "System variables", and of the editor window.

To define the color of the text type

1. Select the text type in the "Text Type" drop-down list box or click on it in the sample area.
In the "Foreground" drop-down list box, select a color. (System is the color defined in Windows.)
2. In the "Background" drop-down list box, select a background color for the text. (System is the color defined in Windows.)
3. Choose **OK**.

The default color assignments for Natural are:

- Blue - Natural keywords
- Red - Comments
- Green - Text constants, system functions, system variables
- Black - User-defined variables and remaining syntax elements

To define the color of the Edit Window

1. Click anywhere in the sample area where there is no text.
"Edit Window" is displayed in the "Text Type" drop-down list box.
2. In the "Background" drop-down list box, select a color. (System is the color defined in Windows.)
3. Choose **OK**.

Font Definition

You can define your preferred font in the "Windows Font" definition window.

To do so

- From the "Program Editor Preferences" dialog, choose "Font".
The Windows "Font definition" window is displayed.
For further information, please refer to the Windows documentation.

Note:

Only monospaced fonts are available.

Status Bar Information

The status line appears at the top of the window where the program is edited. It displays the following information:

Line: *x* of *y*

x: The current cursor line position.

y: The total number of lines in the object.

Col

The current cursor column position.

Size

The total number of characters in the source code.

Structured or Report

The programming mode (structured or report) of the active object.

You can change mode to structured with the system command "GLOBALS SM=ON" or to report "GLOBALS SM=OFF" or by selecting "Session Parameters" from the **Options** menu and then choosing **Miscellaneous** from the cascading menu.

Modified

Indicates that the object has been modified since the last save. If "Modified" is not displayed, then the object has not been modified since the last save.

INS or OVR

INS: Editor is in insert mode. Input does not overwrite existing text.

OVR: Editor is in overwrite mode. Input overwrites existing text.