

Editor Commands

This section describes all editor commands provided in addition to the editor commands used for scrolling in the source (see *Editor Commands for Scrolling*) and summarizes commonly used editor command options. Some frequently used commands can be issued using PF keys as indicated in this section.

You enter an editor command in the command line of the editor screen.

You can enter several editor commands in the same input operation if you separate them with a semicolon (;).

Depending on the configuration of your installation, editor commands can be entered in lower case. In this section, however, all commands are shown in upper case to distinguish them as commands. For explanations of the syntax symbols used in this section, refer to *System Command Syntax* in the *System Commands* documentation.

- ADVANCE
- AORDER
- AUTOSAVE
- BNDS
- CANCEL
- CAPS
- CENTER
- CHANGE
- COLS
- CWINDOW
- DELETE
- DWINDOW
- DX, DY, DX-Y
- EMPTY
- EX, EY, EX-Y
- EXCLUDE
- EXIT
- FIND

- FLIP
- HEX
- HOME
- INCLUDE
- JLEFT
- JRIGHT
- JUSTIFY
- LABEL
- LC
- LIMIT
- LOCATE
- LOG
- MASK
- MWINDOW
- NEXT
- ORDER
- POINT
- POWER
- PROF
- PROFILE
- PROTECT
- RCHANGE
- RESET
- RFIND
- SET TYPE
- SHIFT
- SORT

- SPLIT
 - SWAP
 - TABS
 - UC
 - UNDO
 - WINDOW
 - X
 - XSWAP
 - Y
 - Common Command Options
-

ADVANCE

ADVANCE	[<u>ON</u> OFF PAGE]
---------	------------------------------

This command is used to specify whether the cursor moves to the next line automatically after a line update.

ON	The cursor moves to the next line after an update.
OFF	The cursor does not move to the next line after an update.
PAGE	The line containing the cursor is placed at the top of the editing area after an update.

The ADVANCE command issued without a parameter has the same effect as ADVANCE ON.

AORDER

AORDER	[<u>ON</u> OFF]
--------	----------------------

This command is used to specify whether text is to be automatically justified within the set boundaries.

The AORDER command issued without a parameter has the same effect as AORDER ON. The base setting can be changed by editing your profile.

AUTOSAVE

$\left\{ \begin{array}{l} \text{AUTOSAVE} \\ \text{ASAVE} \end{array} \right\} \left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right]$

This command is used to specify whether the editor executes an automatic SAVE command when you issue the EXIT command.

The AUTOSAVE command issued without a parameter has the same effect as AUTOSAVE ON.

BNDS

$\text{BNDS} \left[\begin{array}{l} n \ m \\ n \end{array} \right]$
--

This command is used to restrict the effect of certain commands to a specific range of columns.

These boundaries apply to the editor commands FIND, CHANGE, CENTER, ORDER, JLEFT and JRIGHT, and their corresponding line commands (if available) such as TC, TO, LJ and RJ.

<i>n</i>	The number of the column at which the left boundary is to be placed.
<i>m</i>	The number of the column at which the right boundary is to be placed.

If *n* and *m* are omitted, the boundaries are set at the first and last column of the editing area.

You can issue the BNDS line command to see the current boundary settings.

CANCEL

<u>C</u> ANCEL

Alternative PF key: PF12

This command cancels all changes made since you last saved the source (see also *Saving and Cataloging Sources*) and leaves the editor. Depending on your editor profile settings, you are prompted to save your changes or leave without saving (see also *Exit Function*).

CAPS

$\text{CAPS} \left[\begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{PGM} \end{array} \right]$

This command is used to switch upper-case translation on or off. This command only applies to lines which are created or modified after the command is issued.

ON	Text in line is translated to upper case.
OFF	Text in line is not translated; that is, it remains as entered.
PGM	Text in line is translated to upper case (except for comments, which remain as entered).

The CAPS command issued without a parameter has the same effect as CAPS ON.

CENTER

CENTER	$\left\{ \begin{array}{l} \text{ALL} \\ n \\ n \ m \end{array} \right\}$
--------	--

This command is used to center text.

ALL	Centers the text of all lines.
n	Centers the text from line <i>n</i> to the last line.
n m	Centers the text from line <i>n</i> to line <i>m</i> .

The CENTER command applies only within the horizontal boundaries as set with the BNDS editor command.

For centering, you can also use the line commands TC and TCC.

CHANGE

$\left\{ \begin{array}{l} \text{CHANGE} \\ \text{CHG} \end{array} \right\}$	$\left[\begin{array}{l} * \\ [T]'string1' \\ C'string1' \\ X'string1' \\ P'string1' \end{array} \right]$	$\left\{ \begin{array}{l} * \\ [X]'string2' \end{array} \right\}$			
	$\left[\begin{array}{l} .X \\ .X \ .Y \end{array} \right]$	$\left[\begin{array}{l} n \\ n \\ m \end{array} \right]$	$\left[\begin{array}{l} \text{ALL} \\ \text{NEXT} \\ \text{PREV} \\ \text{FIRST} \\ \text{LAST} \end{array} \right]$	$\left[\begin{array}{l} \text{CHARS} \\ \text{WORD} \\ \text{PREFIX} \\ \text{SUFFIX} \end{array} \right]$	$\left[\begin{array}{l} \text{NX} \\ X \end{array} \right]$

This command is used to replace a character string (*string1*) by another character string (*string2*).

If you want an apostrophe to be part of *string1* or *string2*, you must write it as two apostrophes.

You can specify the string to be replaced (*string1*) as described in the following section.

T' <i>string1</i> '	Replaces <i>string1</i> irrespective of whether it occurs in lower case or upper case. This is the default.
' <i>string1</i> '	Same as T' <i>string1</i> '.
C' <i>string1</i> '	Replaces <i>string1</i> only if it occurs exactly as specified.
X' <i>string1</i> '	Replaces the string that corresponds to the specified hexadecimal character string <i>string1</i> . Replace it by the hexadecimal string <i>string2</i> .
P' <i>string1</i> '	Replaces <i>string1</i> which includes the following wildcard characters: = any character § alphabetic character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Uses the character string specified in a previous command, for example, FIND, CHANGE or EXCLUDE.
.X	See <i>Line Specifications</i> for an explanation.
.X .Y	
n n m	See <i>Column Specifications</i> for an explanation.
ALL NEXT PREV FIRST LAST	See <i>Direction of Operation</i> for an explanation.
CHARS WORD PREFIX SUFFIX	See <i>Special Occurrences</i> for an explanation.
NX X	See <i>Displayed or Non-Displayed Lines</i> for an explanation.

This section covers the following topics:

- Using CHANGE Together with Other Commands
- Examples of the CHANGE Command

Using CHANGE Together with Other Commands

You can use the command RCHANGE to repeat the execution of a CHANGE command.

To search the entire source code for a character string and then decide occurrence by occurrence whether to replace it by another character string, you can use a combination of the commands FIND and CHANGE and the PF keys assigned to the commands RFIND and RCHANGE as described in *Finding and Replacing Text*.

Examples of the CHANGE Command

Example 1:

```
CHG 'LOW' 'HIGH'
```

This command replaces the first occurrence of LOW by HIGH, regardless of upper or lower case.

Example 2:

```
CHG C'OPS' 'SPF' .X .Y 28 32 ALL
```

This command changes OPS (exactly as entered here) into SPF; it changes all occurrences in the block of lines labeled with .X and .Y and between columns 28 and 32.

Example 3:

```
CHG C'NAME' 'APPL' .X .Y ALL PREFIX NX
```

This command changes all occurrences of a string that begins with NAME (exactly as entered here) into APPL in all displayed lines in the block of lines labeled with .X and .Y.

Example 4:

```
CHG * 'NEW'
```

This command replaces the next occurrence of the string specified in the last CHANGE command by the string NEW.

You can specify that only lines which contain a specified character *string* are to be deleted as described in the following section.

T' <i>string'</i>	Deletes lines that contain the <i>string</i> irrespective of lower case or upper case. This is the default.
' <i>string'</i>	Same as T' <i>string'</i> .
C' <i>string'</i>	Deletes lines that contain the <i>string</i> exactly as specified.
X' <i>string'</i>	Deletes lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string'</i>	Deletes lines that contain the <i>string</i> which includes the following wildcard characters: = any character § alphabetic character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Uses the search string specified in a previous command, for example, FIND, CHANGE or EXCLUDE.
.X	See <i>Line Specifications</i> for an explanation.
.X .Y	
n n m	See <i>Column Specifications</i> for an explanation.
ALL NEXT PREV FIRST LAST	See <i>Direction of Operation</i> for an explanation.
CHARS WORD PREFIX SUFFIX	See <i>Special Occurrences</i> for an explanation.

NX X	See <i>Displayed or Non-Displayed Lines</i> for an explanation.
-----------------------	---

If you enter the DELETE command without any parameters, the current line is deleted.

You can also use the line command D, Dn or DD to delete lines.

Example 1:

<code>DEL C'NAME' 1 20 ALL PREFIX NX</code>

This command deletes all lines that contain the string NAME (in upper case exactly as entered here) as a prefix to a word in all lines not excluded from display if NAME occurs between columns 1 and 20.

Example 2:

<code>DEL C'Abc' .X .Y 10 30 ALL</code>

This command deletes all lines that contain the string Abc (exactly as entered here) between columns 10 and 30 within the block of lines labeled with .X and .Y

DWINDOW

DWINDOW

This command is used to delete the last defined data window.

DX, DY, DX-Y

DX
DY
DX-Y

These commands are used to delete marked lines.

- The DX command deletes the line marked with the .X label.
- The DY command deletes the line marked with the .Y label.
- The DX-Y command deletes all lines between the .X and .Y labels.

EMPTY

```
EMPTY [ ON ]
      [ OFF ]
```

This command controls the deletion of blank lines.

OFF	Blank lines are not deleted.
ON	Blank lines are deleted.

The EMPTY command issued without a parameter has the same effect as EMPTY ON.

EX, EY, EX-Y

```
EX
EY
EX-Y
```

These commands are used to delete lines in a source.

- The EX command deletes *all lines* preceding the line marked with the .X label.
- The EY command deletes *all lines* following the line marked with the .Y label.
- The EX-Y command deletes *all lines* preceding the .X label and following the .Y label.

EXCLUDE

```
EXCLUDE [ *
         [T]'string'
         'string'
         C'string'
         X'string'
         P'string'
         ] [ .X
           [ .X .Y ] [ n
                 [ n m ] [ ALL
                       NEXT
                       PREV
                       FIRST
                       LAST
                       ] [ CHARS
                         WORD
                         PREFIX
                         SUFFIX
                         ]
```

This command is used to exclude lines from being displayed.

You can specify that only lines which contain a specified character *string* are to be excluded from display as described in the following section.

T'<i>string</i>'	Excludes lines that contain the <i>string</i> irrespective of lower case or upper case. This is the default.
'<i>string</i>'	Same as T' <i>string</i> '.
C'<i>string</i>'	Excludes lines that contain the <i>string</i> exactly as specified.
X'<i>string</i>'	Excludes lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P'<i>string</i>'	Excludes lines that contain the <i>string</i> which includes the following wildcard characters: = any character § alphabetic character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Uses the search string specified in a previous command, for example, EXCLUDE, FIND or CHANGE.
.X	See <i>Line Specifications</i> for an explanation.
.X .Y	
n n m	See <i>Column Specifications</i> for an explanation.
ALL NEXT PREV FIRST LAST	See <i>Direction of Operation</i> for an explanation.
CHARS WORD PREFIX SUFFIX	See <i>Special Occurrences</i> for an explanation.

If you enter the EXCLUDE command without any parameters, the current line is excluded from display.

You can use the INCLUDE editor command to re-display excluded lines.

Example 1:

```
EXCLUDE .X .Y
```

This command excludes lines from the line labeled with .X to the line labeled with .Y.

Example 2:

```
EXCLUDE C'NAME' ALL PREFIX
```

This command excludes from display all lines which contain strings that begin with NAME (in upper case as entered here).

EXIT

```
EXIT
```

Alternative PF key: PF3

This command is used to leave the editor. If any changes have been made since you last saved the source (see also *Saving and Cataloging Sources*), you are prompted to save your changes or leave without saving, depending on your editor profile settings (see also *Exit Function*).

Note:

If AUTOSAVE is set to ON, you will not be prompted before exiting the session; your changes will then be saved automatically.

FIND

```
FIND [ * ] [ .X ] [ n ] [ ALL ] [ CHARS ] [ NX ]
      [ [T]'string' ] [ .X .Y ] [ n m ] [ NEXT ] [ WORD ] [ X ]
      [ 'string' ] [ ] [ ] [ PREV ] [ PREFIX ]
      [ C'string' ] [ ] [ ] [ FIRST ] [ SUFFIX ]
      [ X'string' ] [ ] [ ] [ LAST ]
      [ P'string' ] [ ] [ ]
```

This command is used to search for a specific character *string*. The cursor is placed at the beginning of the first *string* found. If the line containing the *string* was excluded from display, it is displayed when found.

If you want an apostrophe to be part of the *string*, you must write it as two apostrophes.

You can specify the *string* as described in the following section.

T' <i>string'</i>	Searches for the <i>string</i> irrespective of lower case or upper case. This is the default.
' <i>string'</i>	Same as T' <i>string'</i> .
C' <i>string'</i>	Searches for the <i>string</i> exactly as specified.
X' <i>string'</i>	Searches for the string that corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string'</i>	Searches for a <i>string</i> which includes the following wildcard characters: = any character § alphabetic character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Searches for the <i>string</i> specified in the previous command, for example, FIND, DELETE or EXCLUDE.
.X	See <i>Line Specifications</i> for an explanation.
.X .Y	
n n m	See <i>Column Specifications</i> for an explanation.
ALL NEXT PREV FIRST LAST	See <i>Direction of Operation</i> for an explanation.
CHARS WORD PREFIX SUFFIX	See <i>Special Occurrences</i> for an explanation.
NX X	See <i>Displayed or Non-Displayed Lines</i> for an explanation.

Example 1:

```
F C'NAME' .X .Y ALL PREFIX X
```

This command searches for any occurrence of NAME exactly as entered here as a prefix of a word in any excluded line within the block of lines labeled with .X and .Y.

Example 2:

```
F C'HILITE' X PREV
```

This command searches for the previous occurrence of HILITE exactly as entered here in any excluded line.

You can use the RFOUND command to repeat the execution of a FIND command.

Example 3:

```
F P'RCV#' .X .Z 20 30
```

This command searches for any 4-character string that begins with RCV and whose fourth character is numeric. It searches within the block of lines labeled with .X and .Z and between columns 20 to 30.

Example 4:

```
F X'6C' SUFFIX NX
```

This command searches for the character with hexadecimal representation 6C. Only those occurrences of the character that are at the end of word are found. The search is valid for non-excluded lines only.

Example 5:

```
F '''w'
```

This command searches for the following character string: 'w

Example 6:

```
F 'r''w'
```


If you enter the `INCLUDE` command without any parameters, it includes the first line of an excluded block of lines.

JLEFT

JLEFT	$\left\{ \begin{array}{l} \text{ALL} \\ n \\ n \ m \end{array} \right\}$
-------	--

This command is used to align text left-justified.

ALL	Aligns the text of all lines.
<i>n</i>	Aligns the text from line <i>n</i> to the last line.
<i>n m</i>	Aligns the text from line <i>n</i> to line <i>m</i> .

The `JLEFT` command applies only within the horizontal boundaries as set with the `BND`s command.

For left justification, you can also use the line commands `LJ` and `LJJ`.

See also the `JRIGHT` command.

Example:

<code>BND</code> s 10;JLEFT 15 20

The text between column 10 and the rightmost column of your screen in lines 15 to 20 is left-aligned to column 10.

JRIGHT

JRIGHT	$\left\{ \begin{array}{l} \text{ALL} \\ n \\ n \ m \end{array} \right\}$
--------	--

This command is used to align text right-justified.

ALL	Aligns the text of all lines.
<i>n</i>	Aligns the text from line <i>n</i> to the last line.
<i>n m</i>	Aligns the text from line <i>n</i> to line <i>m</i> .

The `JRIGHT` command applies only within the horizontal boundaries as set with the `BNDS` command.

For right justification, you can also use the line commands `RJ` and `RJJ`.

See also the `JLEFT` command.

Example 1:

```
BNDS 4 40;JRIGHT 6 18
```

The text between columns 4 to 40 in lines 6 to 18 is right-aligned to column 40.

Example 2:

```
BNDS 10;JRIGHT 15
```

The text to the right of column 10 from line 15 to the last line is right-aligned to the rightmost column of your editing screen.

JUSTIFY

```
JUSTIFY { LEFT
         RIGHT
         BOTH }
```

This command is used to set the justification mode for the line commands `TO` and `TOO`.

`TO` and `TOO` are used to join source lines with subsequent lines. Both commands apply only within the horizontal boundaries as set with the `BNDS` command.

LEFT	Aligns text to the left boundary.
RIGHT	Aligns text to the right boundary.
BOTH	Aligns text to both boundaries.

Example:

You set the horizontal boundaries to columns 10 and 60 and activate left justification with the following command:

```
BNDS 10 60;JUSTIFY LEFT
```

When you then mark a line with a TO line command (or a block of lines with two TOO line commands), the text between columns 10 and 60 in the marked line(s) is left-aligned to column 10.

LABEL

```
LABEL .label
```

This command is used to mark the current line (that is, the line which is currently at the top of the editing area) with the specified *label*.

The *label* is a string of 1 to 4 alphanumerical characters.

Example:

Use the following command to label the current line with .X:

```
LABEL .X
```

You can also mark a block of lines with two labels. For example, to mark a block with labels .X and .Y, you first mark the current line (assuming it is the first line of the block to be marked) with .X as shown in the example above; then you scroll until the last line of the block is the current line; then you issue the LABEL .Y command to mark that line with .Y.

You can also use the .label line command to mark a line with a label.

LC

```
LC [ * ] [ [T]'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ ALL ] [ CHARS ] [ NX ]
  [ [T]'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ NEXT ] [ WORD ] [ X ]
  [ 'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ PREV ] [ PREFIX ] [ ]
  [ C'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ FIRST ] [ SUFFIX ] [ ]
  [ X'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ LAST ] [ ] [ ]
  [ P'string' ] [ [ .X ] [ .X .Y ] ] [ [ n ] [ n m ] ] [ ] [ ] [ ]
```

This command is used to change one or more lines to lower case.

You can specify that only lines which contain a specified character *string* are to be changed to lower case. If you want an apostrophe to be part of the *string*, you must write it as two apostrophes.

You can specify the *string* as described in the following section.

T'<i>string</i>'	Changes lines which contain the <i>string</i> irrespective of lower case or upper case. This is the default.
'<i>string</i>'	Same as T' <i>string</i> '.

C' <i>string</i>	Changes lines which contain the <i>string</i> exactly as specified.
X' <i>string</i>	Changes lines which contain the string that corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string</i>	Changes lines which contains a <i>string</i> that includes the following wildcard characters: = any character § alphabetic character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Changes lines which contain the <i>string</i> specified in a previous command, for example, LC, DELETE or EXCLUDE.
.X	See <i>Line Specifications</i> for an explanation.
.X .Y	
n n m	See <i>Column Specifications</i> for an explanation.
ALL NEXT PREV FIRST LAST	See <i>Direction of Operation</i> for an explanation.
CHARS WORD PREFIX SUFFIX	See <i>Special Occurrences</i> for an explanation.
NX X	See <i>Displayed or Non-Displayed Lines</i> for an explanation.

If you enter the LC command without any parameters, the current line is changed to lower case.

Example:

```
LC C'NAME' .X .Y ALL PREFIX NX
```

This command changes to lower case all displayed lines within the block of lines labeled with .X and .Y if they contain the string NAME (in upper case as entered here) as prefix to a word.

LIMIT

```
LIMIT [n]
```

With this command, you specify the maximum number of lines to be searched with a FIND or RFIND command. The parameter *n* is the number of lines to be searched.

LOCATE

```
[LOCATE] { 0
            n
            .label }
```

This command is used to scroll a specific line to the top of the editing area (that is, make it the current line).

The command provides the following options:

0	Makes the first line of the source code current.
<i>n</i>	Makes line <i>n</i> current.
<i>.label</i>	Makes the line labeled with <i>.label</i> current.

Examples:

```
LOC 32
```

Places line number 32 at the top of the editing area.

```
32
```

Same as above.

```
LOC .X
```

Places the line labeled with .X at the top of the editing area.

LOG

```
LOG [ ON ]
     [ OFF ]
```

This command activates or deactivates the internal log file.

The log file is a history of all modifications made in the editor since session begin. When the log file is active, each time you press ENTER, the changes made since the previous ENTER are recorded in the log file. When using the UNDO command, you can consecutively back out changes made since the beginning of the editor session.

Important:

All entries in the log file are cleared when you clear the source work area or read in the source of another Natural object, or when you terminate the program editor session.

MASK

```
MASK [ ON ]
      [ OFF ]
```

This command activates or deactivates the mask function. When the mask function is active, each time you insert a line in the editor, a predefined line of text is entered instead of a blank line. The mask line is defined using the MASK line command, described in the following paragraph. The mask function is useful when you must write several lines of code which are identical or very similar.

▶ To define and use a mask line

1. Enter the MASK line command in any source line and press ENTER.

A blank line indicated by =mask> appears above the line in which you entered the command.

2. In the blank line, type in the text you want to define as a mask line and press ENTER.

The mask line is now available for the current source until you update the mask with a new mask line or until you deactivate the mask function.

3. Enter the MASK ON editor command.

The mask function is activated. The defined mask line now appears in all lines added through a line insert operation.

4. Enter an insert line command, for example: I2

Two new lines are inserted into the source with the text of the mask line. The text of a mask line appears in all lines added with an insert command.

5. Modify the text in the new lines. If you do not modify the text, any inserted line is deleted the next time you press ENTER.

The MASK OFF command deactivates the mask function but does not delete the contents of the mask line.

MWINDOW

MWINDOW	$\left[\begin{array}{l} n \\ n \ m \end{array} \right]$
---------	--

This command is used to move a data window according to the command parameters.

<i>n</i>	The number of the line in which the data window is to be inserted.
<i>m</i>	The number of the column in which the data window is to be inserted.

See also *Copying and Moving Text with a Data Window*.

NEXT

<u>N</u> EXT	$\left[\begin{array}{l} * \\ object-name \end{array} \right]$
--------------	--

This command is used to display the *next* parallel editing session, assuming two or more editing sessions are running concurrently and if the profile parameter EDTRB (see the *Parameter Reference* documentation) is set. The following command parameters are optional:

*	Displays a list of all concurrently running sessions for selection.
<i>object-name</i>	Calls directly by name a concurrently running editing session.

ORDER

ORDER	$\left\{ \begin{array}{l} ALL \\ n \\ n \ m \end{array} \right\}$
-------	---

This command is used to join source lines.

ALL	Joins all lines.
<i>n</i>	Joins the lines from line <i>n</i> to the last line.
<i>n m</i>	Joins lines from line <i>n</i> to line <i>m</i> .

The ORDER command applies only within the horizontal boundaries as set with the BNDS command.

Within the set boundaries, the lines are concatenated and are filled to the greatest possible extent; words that do not fit into one line are automatically placed in the next line.

To join source lines, you can also use the line commands TF, TO and TOO.

POINT

POINT

This command places the line marked by the line command NZ at the top of the editing area.

POWER

POWER

This command switches the editor to insert mode. You are presented with a blank screen into which you can enter one or more lines of text. After entry, press ENTER and the text is inserted into the first line of the editing area.

PROF

PROF [<i>n</i>]

This command displays your editor profile at the top of the editor screen.

With *n* you specify additional lines to be displayed. Possible values for *n* are:

6	Displays your editor profile and all tab positions (as specified by the TABS command).
7	Displays same as 6, plus the mask line (as specified by the MASK command).
8	Displays same as 7, plus boundaries (as specified by the BNDS command).
9	Displays same as 8, plus column numbers (as specified by the COLS command).

PROFILE

`PROFILE`

This command invokes the editor profile facility. It enables you to modify your editor defaults for current and future sessions. The editor profile facility is described in more detail in section *Modifying Profile Settings for Permanent Use*.

PROTECT

`PROTECT [INS
ON
OFF]`

This command is used to protect the prefix area.

INS	Protects the prefix area of lines added using the insert line command.
ON	Activates protection.
OFF	Deactivates protection.

RCHANGE

`RCHANGE`

This command repeats the last CHANGE command.

RESET

`RESET`

This command resets all pending editor and line commands and deletes all line labels.

RFIND

`RFIND`

Alternative PF key: PF5

This command repeats the last FIND command.

SET TYPE

SET TYPE	{	CLASS	}
		4	}
		<u>COPYCODE</u>	
	{	FUNCTION	}
		7	}
		<u>HELPROUTINE</u>	
		<u>PROGRAM</u>	
	{	SUBPROGRAM	}
		N	}
		<u>SUBROUTINE</u>	
	{	<u>TEXT</u>	}

This command changes the type of the object currently in the source work area.

SHIFT

<u>S</u> HIFT [<i>n</i>]	[RIGHT]
		LEFT]

This command shifts a block of lines between the .X and .Y labels to the right or left by *n* columns (or up to the last non-blank character). The default shift is five columns to the right.

<i>n</i>	The number of columns the lines are to be shifted (default value is 5).
RIGHT	Shifts block of lines to the right (default).
LEFT	Shifts block of lines to the left.

SORT

SORT [<i>n m</i>]	[.X]	[<u>A</u>]
		.X .Y]		<u>D</u>]

The SORT command sorts lines in the editor in ascending or descending ^a order. If you enter SORT without any parameters, the command sorts all text in the object in ascending order.

<i>n m</i>	Sorts from column <i>n</i> to column <i>m</i> .
.X	Sorts from the line labeled with .X to the end of the source.
.X .Y	Sorts from the line labeled with .X to the line labeled with .Y (where .X and .Y represent any string of up to four characters).
A	Sorts text in ascending order (A to Z).
D	Sorts text in descending order (Z to A).

SPLIT

```

SPLIT { PROGRAM object-name [library-name]
      VIEW   object-name [SHORT]
      END

```

This command sets split-screen mode and displays the source of another object on the editor screen.

PROGRAM	Displays a program, subprogram, subroutine, help routine, data area (global, local, parameter), copycode, text, map, class or function.
VIEW	Displays a view (DDM, as defined in Predict or SYSDDM). If SHORT is specified, the view is listed in short form (that is, only the Adabas short names and corresponding Natural field names are displayed) without any field headers or field edit mask information.
END	Terminates split-screen mode.

With PROGRAM or VIEW, an asterisk (*) can be used for *object-name* to display a list of all available objects. If the an asterisk (*) is preceded by one or more characters, only those objects whose names begin with these characters are displayed.

For an example of using SPLIT, see *To display and copy definitions with SPLIT*.

SWAP

```

SWAP

```

The SWAP command toggles between two objects in split-screen mode (see the relevant section). During this operation, the cursor switches from one object to the other.

TABS

TABS	[<i>tab-character</i>]
	OFF
	[<u>LEFT</u>] [<i>tab-character</i>] [<i>column...</i>]
	[<u>RIGHT</u>]
	[<u>DECIMAL</u>]

This command is used to control tabulator settings.

You can enable or disable logical or physical tabulation using the command `TABS ON` or `TABS OFF`. Tabulation is also enabled by any command that changes a tabulation setting.

For example, the following command enables logical tabulation with the ampersand sign (&) as the logical tabulation character:

```
TABS &
```

You set tab positions using the `TABS` command. For example, the following command sets tabs in columns 10, 20 and 30:

```
TABS 10 20 30
```

You can enter text and automatically move it to a specific tab position by preceding it with a logical tabulation character. One tabulation character moves the text to the next tab position, two tabulation characters move the text to the second tab position, and so on.

To display the current `TABS` command settings, issue the `PROF` command.

To display the current tab positions, issue the `TABS` line command.

Apart from tab positions, you can specify the following parameters with the `TABS` command:

<u>LEFT</u>	Places the text left-justified at the tab position.
<u>RIGHT</u>	Places the text right-justified at the tab position.
<u>DECIMAL</u>	Places the text so that the decimal point in the text is at the tab position.

Multiple tab characters are possible to tabulate text in a specific column: issue the `TABS` line command and type over each asterisk (*) marking the tab positions with another special character. Any input preceded by any of these special characters is tabulated in the corresponding column. You can type an L (for `LEFT`), an R (for `RIGHT`) or a D (for `DECIMAL`) after each tabulation character to specify placement of the text for the tab position.

In the following examples of tabulation, the ampersand (&) is assumed to be the tabulation character; the COLS line command has been issued to display column positions.

- Example 1 - Tab Positions
- Example 2 - TABS RIGHT
- Example 3 - TABS DECIMAL
- Example 4 - Mixed Justification
- Example 5 - Multiple Tab Symbols
- Example 6 - Using a Blank as Tabulation Symbol

Example 1 - Tab Positions

The command:

```
TABS 10 20 40 LEFT
```

activates logical tabs with tabulation columns 10, 20, and 40 with left justification. After you press ENTER, the input text line

```
&abc &def &ghi
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
          abc      def              ghi
```

Example 2 - TABS RIGHT

The command

```
TABS RIGHT
```

activates logical tabs with right justification. After you press ENTER, the input text line

```
&abc &def &ghi
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
          abc      def              ghi
```

Example 3 - TABS DECIMAL

The command

```
TABS DECIMAL
```

activates logical tabs with justification of the decimal point in the tab position. After you press ENTER, the input text line

```
&15.27$ &16.3 EUR &13 IS
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
          15.27$   16.3 EUR                   13 IS
```

Example 4 - Mixed Justification

Issue the following command:

```
TABS 10 20 30 40 50
```

Then issue the TABS line command. This displays the current tab positions as follows:

```
=tabs>          *          *          *          *          *
```

Type an L, R or D next to each tab position as required (unmarked tab positions assume the value of the last TAB command):

```
=tabs>          *R          *D          *D          *D          *L
```

After you press ENTER, the input text line

```
&start &0.01 &0.02 &0.03 &end
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
          start      0.01    0.02    0.03    end
```

Example 5 - Multiple Tab Symbols

Replace the asterisks in the =tabs> line by other special characters and specify left justification for each one as follows:

```
=tabs>          ]L          &L          #L          $L          =L
```


UNDO

```
UNDO [ ALL ]
      [ n ]
```

Alternative PF key: PF11

If the log file is active (see the LOG command), the UNDO command backs out all changes made since the last time you pressed ENTER. Repeated use of the UNDO command backs out consecutive changes in reverse order. You can thus back out all changes one by one until you restore the member to its original status at session begin.

Important:

All entries in the log file are cleared when you clear the source work area or read in the source of another Natural object, or when you terminate the program editor session.

You can specify the following parameters with the UNDO command:

ALL	Backs out all modifications made in the current editor session.
n	Backs out the last <i>n</i> modifications.

WINDOW

```
WINDOW { line1 line2
        { line1 line2 column1
        { line1 line2 column1 column2 }
```

This command is used to define a data window to be copied or moved. The starting line and column and the end line and column of the window are specified in the command parameters. At least *line1* and *line2* are required.

<i>line1 line2</i>	Defines a window starting at column 1 of <i>line1</i> and ending in the last column of <i>line2</i> .
<i>line1 line2 column1</i>	Defines a window starting at <i>column1</i> of <i>line1</i> and ending at the last column of <i>line2</i> .
<i>line1 line2 column1 column2</i>	Defines a window starting at <i>column1</i> of <i>line1</i> and ending at <i>column2</i> of <i>line2</i> .

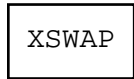
See also *Copying and Moving Text with a Data Window*.

X



This command places the line marked by the line command `.X` at the top of the editing area.

XSWAP



This command is used to exchange displayed lines with excluded lines. Lines are excluded using the `EXCLUDE` command.

Y



This command places the line marked by the line command `.Y` at the top of the editing area.

Common Command Options

There are some options which are available with several editor commands. These options are described in the following section.

- Line Specifications
- Column Specifications
- Displayed or Non-Displayed Lines
- Direction of Operation
- Special Occurrences

Line Specifications

The following options can be used to restrict the effect of an editor command to a line or a block of lines labeled with the `.X` and `.Y` line commands:

<code>.X</code>	The editor command affects only the line labeled with <code>.X</code> . Exception: <code>SORT</code> editor command.
<code>.X</code> <code>.Y</code>	The editor command affects only the block of lines from the line labeled with <code>.X</code> to the line labeled with <code>.Y</code> . If you use this option, you must also supply the parameter <code>ALL</code> with the command.

Note:

.X and .Y can also be any label of 1 to 4 alphabetic characters (see also the LABEL editor command and the .label line command).

Column Specifications

The following options can be used to restrict the effect of an editor command to a certain range of columns. The column numbers refer to the actual source-code columns; the line numbers preceding the source code are not counted. So, if you specify column 1 with a command, this may physically be the 8th column of your screen, but it is in fact the 1st column of the source code you are editing.

n	The command affects only lines in which the specified string begins in column <i>n</i> (that is, the first character of the string must be in column <i>n</i>).
n m	The command affects only lines in which the specified string occurs anywhere between columns <i>n</i> and <i>m</i> .

Displayed or Non-Displayed Lines

The following options can be used to specify that only excluded or only included lines are to be affected by an editor command:

NX	The command affects only non-excluded lines; that is, lines which are currently being displayed.
X	The command affects only excluded lines; that is, lines which are currently <i>not</i> being displayed as specified by the EXCLUDE command. An excluded line remains excluded from display if an editor command function is performed on it.

Direction of Operation

The following options can be used to specify the direction in which an editor command is to operate:

NEXT	The command affects the next line (starting from the cursor position) in which the specified <i>string</i> occurs. This is the default setting.
PREV	The command affects the line that contains the previous occurrence of the specified <i>string</i> .
FIRST	The command affects the first line in which the specified <i>string</i> occurs.
LAST	The command affects the last line in which the specified <i>string</i> occurs.
ALL	The command affects all lines in which the specified <i>string</i> occurs.

Special Occurrences

The following options can be used to specify whether only special occurrences of the specified *string* are to be affected by an editor command:

CHARS	The command affects any line in which the specified <i>string</i> occurs. This is the default setting.
WORD	The command affects only those lines in which the specified <i>string</i> forms a word.
PREFIX	The command affects only those lines in which the specified <i>string</i> is the beginning of a word.
SUFFIX	The command affects only those lines in which the specified <i>string</i> is the end of a word.