

Program Editor

The program editor is used to edit Natural objects of the following types: program, subprogram, subroutine, help routine, copycode, class and text. The program editor is based on the Software AG Editor, thus it provides many of the same functions. This section describes all of the functions provided by the program editor.

- Editor Basics
 - Invoking the Editor
 - Leaving the Editor
 - Editor Screen
 - Editor Profile
 - Editor Profile Commands
 - Modifying Profile Settings
 - Editor Buffer-Pool Settings
 - Using Cut and Paste
- Commands
 - Scrolling Commands
 - Split-Screen Commands
 - Editor Commands
 - Command Syntax
 - Common Command Options
 - Line Commands

See also Renumbering of Source-Code Line Number References in the Natural Reference Documentation.

Editor Basics

Invoking the Editor

You invoke the program editor with the EDIT command. This command is described in the section System Commands.

Leaving the Editor

To leave the editor, you issue one of the following commands: EXIT, CANCEL, or a period (.). Before leaving the editor, you are prompted to enable you to save any changes you have made.

Editor Screen

When you invoke the program editor, an edit screen similar to the following appears:

```

>> Columns 001 072 << Program PROG01 Lines 45 User SAG
COMMAND ===> Mode Struct Lib SAGLIB
000010 DEFINE DATA LOCAL
000020     1 FEL (A070)
000030     1 CULFEL (A010/1:400)
000040     1 #SCSTR (A20)
    
```

If you specified no object name and the source work area is empty, an empty screen appears. Otherwise a program appears as displayed above. The left-most six columns contain line numbers for the program.

The editor contains the following output fields:

Field Name	Description
Columns	Columns currently displayed. You can enter text beyond the 72nd column, but to ensure that data can be handled by other platforms you are advised to use only Columns 1-72.
Program	Name of the program currently in working storage. If no name was specified when the editor was invoked, then this field is empty.
Lines	Total number of lines currently used by the editor (data lines and information lines).
Mode	Program mode: structured or reporting.
User	ID of the current user.
Lib	Current library name.

Editor Profile

Each user has an editor profile with parameters which can be set according to individual needs. The first time you invoke the editor, it uses the default values determined by your administrator. There are two ways for you to modify your editor profile settings:

- on a temporary basis at the session level from the command line (example: CAPS OFF/ON). Settings modified in this way are valid for the remainder of the editing session or until you change them again.
- on a permanent basis at the system level from the editor profile utility. Settings modified in this way are valid for each new Natural session. They are activated when you leave Natural and start it again. These settings can be overridden at the session level by temporary settings (see above).

Displaying Profile Settings

To display the current editor profile settings, issue the PROF command. The following lines appear at the top of the editor screen:

```
>> -----Columns 001 072 << Program          Lines 850   User SAG
COMMAND ==>                               Mode  Struct Lib  PROF
***** ***** top of data *****
=prof> date: 10/26/2001 10:03:12 user: SAG   init size: 844 size: 845
=prof> var   - 250,..recovery off (100 0)...autosave off... empty line off
=prof> mask off.caps off.hex off  nulls on .autoren std off..auto order
=prof> log off .mso off.fix off .escape on + . tabs off
=prof> advance on .protect off.limit on 5000
```

The individual items of the editor profile are described in the following table.

Item	Description	Command
date	The current date and time.	Non-modifiable
user	The current logon user.	Non-modifiable
init size / size	Number of lines in program when editor was invoked. / Current number of lines in the program, excluding information lines (for example profile lines and message lines).	Non-modifiable
var	Specifies current line length.	Non-modifiable
autosave	Activates/deactivates automatic save when EXIT command is issued.	AUTOSAVE
empty line	Specifies if lines containing only space characters are to be deleted automatically.	EMPTY ON/OFF
mask	Activates/deactivates the mask line function.	MASK ON/OFF
caps	Specifies whether data are to be translated into upper case.	CAPS ON/OFF/PGM
hex	Specifies whether data are to be displayed in hexadecimal format.	HEX ON/OFF
autoren	Function not yet available.	
auto order	Automatically justifies text within defined boundaries.	AORDER ON/OFF
log	Enables/disables log file. When enabled, UNDO command can be used to backout last changes.	LOG ON/OFF
mso	Enables/disables multi-session operations such as copy.	MSO ON/OFF
fix	Specifies whether fixed number of columns are displayed and how many columns are to be fixed.	FIX ON/OFF <i>n</i>
tabs	Activates/deactivates tabulation.	TABS ON/OFF
advance	Specifies whether the cursor moves to the next line automatically after a line update.	ADVANCE ON/OFF/PAGE
protect	Specifies protection of line numbers.	PROTECT ON/OFF/INS
limit	Specifies the maximum number of lines to be searched by a FIND or RFIND command.	LIMIT <i>n</i>

Editor Profile Commands

The following direct commands can be used in the Editor Profile instead of the corresponding PF keys. Direct commands have to be entered in the command line at the bottom of the editor profile screen.

Command	PF Key	Function
CANCEL	PF12	Cancels the current function and redisplay the screen from which it was invoked. Any modifications made to the profile have no effect for the current session.
EXIT	PF3	Invokes the exit function prompt window.
FLIP	None	Toggles between PF 1-12 and PF 13-24.
READ	PF6	Reads the profile parameters for the User ID currently contained in the Profile Name field. Any modifications made so far, but not yet saved, are overwritten (valid only for the Editor Profile main menu).
SAVE	PF5	Saves all currently valid profile parameters both for the current session and on the database. However, it does not leave the current function (valid only for the Editor Profile main menu).

Modifying Profile Settings

This section describes how to modify your editor profile settings so that the changes take effect not only for the current session (as described in section Displaying Profile Settings), but also for all future edit sessions. To modify your settings, issue the PROFILE command. A screen appears with the following options:

Option	Function
Save	Saves current profile.
Modify	Invokes input screen for modifying editor default settings for PF keys, commands, buffer pool, FIND command parameters.
Read	Reads editor profile of another user which you specify in Profile Name field. The settings from the external profile can then be saved under your profile name.
Tech Info	Displays information for your edit session: library name, program name, object type, Natural mode (report, structured), operating system, steplib (see the section Displaying Technical Information).

Modifying Editor Default Settings

If you select "Modify" from the main menu, a screen is displayed with the following options:

Option	Function
PA/PF Keys	Modify PF-key assignments.
Commands	Modify command defaults.
FIND Parameters	Modify FIND command parameter settings.
General Defaults	Modify prompt window on leaving editor, number of lines displayed before error line.

Modifying PF-Key Assignments

If you select "PA/PF Keys" from the main menu, a screen is displayed with the current PA/PF-key assignments. You can modify the assignments at any time during an edit session. These assignments are then valid for all future sessions or until you modify them again. The default PF-key settings for the editor are:

PF Key	Command	Function
PF1	HELP	Invokes Natural online help facility.
PF2	SAVE	Saves program in edit area.
PF3	EXIT	Exits editor.
PF4	RUN	Checks and runs program in edit area.
PF5	RFIND	Repeats last FIND command.
PF6	STOW	Saves and catalogs program in edit area.
PF7	UP, -	Scrolls upwards.
PF8	DOWN, +	Scrolls downwards.
PF9	CHECK	Checks program.
PF10	HOME	Places cursor in command line.
PF11	UNDO	Backs out last change made to editor.
PF12	CANCEL	Exits editor (after prompting you to save changes).
PF13	PROFILE	Invokes editor profile utility.
PF14	RESET	Resets pending line commands/deletes line labels.
PF15	SWAP	Toggles cursor between upper/lower split screens.
PF16	LAST	Performs most recent command.
PF17	RCHANGE	Repeats last CHANGE command.
PF18	FLIP	Toggles PF-key display between 1-12 and 13-24.
PF19	TOP, --	Scrolls to beginning of program.
PF20	BOTTOM, ++	Scrolls to end of program.

To modify a PF-key assignment, simply overtype the name of an existing command (editor command, system command or program name) with another command (maximum five characters).

Modifying Command Defaults

In this function, you can enable/disable selected editor commands and specify default characters for editor commands. The input fields contained in the screen are explained in the table below. For more information on a specific field, enter a "?" in the field and press ENTER.

Field Name	Description
aorder	Enables/disables autoorder function.
autosave	Enables/disables autosave function.
caps	Specifies whether data are to be translated into upper case.
cols	Specifies whether line is to be displayed showing columns positions.
decimal character	Specifies character used to mark decimal position in numbers used in tabs. (see example TABS DECIMAL).
empty	Specifies if lines containing only space characters are to be deleted automatically.
escape	Specifies whether escape character is used to precede line commands.
escape character	Specifies which character is to be used to precede line commands.
fix	Specifies whether fixed number of columns are displayed beginning with column 1. The number of fixed columns is specified in "fixlength" below.
fixlength	Specifies number of columns starting with column 1 to remain in display when scrolling right.
hex	Specifies whether data are to be displayed in hexadecimal format.
justification	Enables/disables justification or specifies justification type.
limit	Specifies the maximum number of lines to be searched by a FIND or RFIND command.
log file	Enables/disables log file. When enabled, UNDO command can be used to backout last change.
mask line	Enables/disables the mask line function.
message line	Enables/disables message output.
mso	Enables/disables multiple-session operations, such as copies between split-screen sessions.
nulls	Specifies whether the end of each source-code line is to be filled with null characters.
recovery	Function not available.
scroll mode	Specifies how scrolling is to be performed (pagewise, half-pagewise, to cursor)
tabs	Enables/disables tabulation.
tabulator character	Specifies logical tabulation character used to automatically move input to a specific tab position.

Modifying FIND Command Defaults

If you select "FIND parameters" from the main menu, a screen is displayed with the current parameter settings. These settings are used whenever you issue the FIND command without parameters from the command line of the program editor. You have the opportunity to modify these values in a window before the search is performed.

The following parameters can be modified:

Parameter	Specifies
FIND string	The string to be searched.
Search from/to column	The beginning and end columns of the range of column numbers to be searched.
Search from/to label	The start label and end label of the range of line numbers to be searched.

Modifying General Defaults

If you select "General Defaults" from the main menu, a screen is displayed with the current defaults. You can modify these settings as required. A short description of each setting follows:

Setting	Specifies
Prompt Window for EXIT Function	Whether the prompt window is to appear when leaving the program editor using the EXIT function.
Prompt Window for CANCEL Function	Whether the prompt window is to appear when leaving the program editor using the CANCEL function.
Lines before error line	The number of lines to be displayed at the top of the page preceding the line containing a syntax error. If you specify zero, then the line containing the syntax error is placed in the first line of the edit area.

Displaying Technical Information

If you select "Technical Information" from the main menu, a screen is displayed with information about the object currently being edited and the computing environment. The following items are provided:

- user name;
- library name;
- program name;
- object type;
- programming mode;
- operating system;
- steplib.

For further information, see the system command TECH.

Editor Buffer-Pool Settings

Under certain circumstances it may be necessary to increase the size of the buffer pool above the default setting of 400kb. See the profile parameter description of EDTBPSIZE.

Multiple Editor Sessions

When you open a new editing session from the program editor, the old session is stored to the buffer pool if the profile parameter EDTRB is set. When you leave the new session with CANCEL or EXIT, the old session is freed from the buffer pool. The NEXT command can be used to jump from one active session to another.

If the EDTRB parameter is switched OFF, the Ring Buffer is not used, and the new session replaces the present session. If the AUTOSAVE parameter is switched on, the contents of the corresponding session are saved to FUSER.

If you intend to use the LOG command, we recommend to increase the size of the buffer pool to 1 MB. Changes are generally stored in the buffer-pool editor.

Using Cut and Paste

Before pasting text to the program editor, the following commands must be executed:

1. Execute PROTECT ON to protect the prefix area.
2. Execute ESCAPE ON '.' to enable the escape character.
3. Execute .I(n) in the data area to create some empty lines.
4. Now paste the text to the program editor.

5. Use line commands '(n' (Shift left) to delete some copied line numbers

Note:

To copy some text from one program to another, you can also use the Split-Screen command. Mark your lines with the line commands 'CC .. CC' in the SPLIT session and copy it with the line command 'A' to the edit session. After doing this, execute 'SPLIT OFF'.

Commands

The program editor provides two types of commands for editing:

- Editor commands
- Line commands

Note:

Depending on the configuration of your installation, editor commands and line commands may be entered in lower case. In this section, however, all commands are shown in upper case to distinguish them as commands.

Scrolling Commands

Command	Function
BOTTOM or ++	Scrolls to the end of the object being edited.
TOP or --	Scrolls to the beginning of the object being edited.
DOWN	Scrolls forwards by the amount specified by the scroll mode.
DOWN <i>n</i>	Scrolls forwards by <i>n</i> lines.
+ <i>n</i>	Scrolls forwards by <i>n</i> lines.
UP	Scrolls backwards by the amount specified by the scroll mode.
UP <i>n</i>	Scrolls backwards by <i>n</i> lines.
- <i>n</i>	Scrolls backwards by <i>n</i> lines.
LEFT	Scrolls to the left by the amount specified by the scroll mode.
LEFT <i>n</i>	Scrolls to the left by <i>n</i> columns.
RIGHT	Scrolls to the right by the amount specified by the scroll mode.
RIGHT <i>n</i>	Scrolls to the right by <i>n</i> columns.
FIX <i>n</i>	Specifies the number of columns <i>n</i> , starting with column 1, to remain in display when scrolling to the right.

Split-Screen Commands

In split-screen mode, you can simultaneously edit one Natural object while viewing another. The following items can be viewed in split-screen mode:

- views;
- Natural objects, such as data areas, programs, subprograms, subroutines, help routines, copycodes, texts, maps, classes.

Note that it is not possible to edit the object in the display (bottom) section. It is, however, possible to mark lines in this section and copy them into the edit section (using the line commands C and CC).

The following figure shows the program editor in split-screen mode with the source code of a program in the editing section (upper half) and a local data area in the display section (lower half):

```

>> -----Columns 001 072 << Program NEIPBNXT Lines 187 User SAG
Command ==>> Lib SYSLIB
001720 IF (RC NE 0)
001730 then
001740 reset CMD_LINE_2
001750 IF (s_prog ne ' ')
001760 then
001770 assign CMD_LINE_PGM = S_PROG
001780 assign CMD_LINE_LIB = S_LIB
001790 END-IF
001800 END-IF
>> -----Columns 001 072 << Local NPFLCONS Lines 158 User SAG
Command ==>> Lib SYSLIB
***** ***** top of data *****
000001 DEFINE DATA LOCAL
000002 1 NPF_F_READ(A1)
000003 CONST
000004 <'R'>
000005 1 NPF_F_WRITE(A1)
000006 CONST
000007 <'W'>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Save Exit Run Rfind Stow - + Check Home Undo Canc
    
```

Note that, because it is not possible to make data modifications to the display section, not all editor commands are available. PF keys are reserved for the edit section at the top of the screen, thus commands meant for the display section must be issued using the command line.

Note:

The SWAP command can be used to move the cursor swiftly between the command lines of the display and edit sessions.

The following commands can be used to display and position an object in split-screen mode. All commands begin with an "S" or with SPLIT to indicate the working mode.

Command	Function
SPLIT DATA <i>name</i> [<i>library</i>]	Display data area (global, local, parameter).
SPLIT PROGRAM <i>name</i> [<i>lib</i>]	Display program, subprogram, subroutine, help routine, copy code, text, map.
SPLIT VIEW <i>name</i> [SHORT]	Display 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 header or field edit mask information.
SPLIT END	Terminate split-screen mode.

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

Editor Commands

Editor commands are issued from the command line. Some frequently used commands can be issued using PF keys.

Command Syntax

CURSOR	Terms in upper-case bold letters are used for commands.
<u> </u>	Full underlining indicates a default value; partial underlining indicates an abbreviated form.
<i>string</i>	Terms in lower-case italic letters are used for parts of syntax or commands whose values have to be supplied by the user.
[]	Elements contained within square brackets are optional. If the square brackets contain several lines stacked one above the other, each line is an optional alternative. You may choose at most one of the alternatives.
{ }	If the braces contain several lines stacked one above the other, each line is an alternative. You must choose exactly one of the alternatives.
...4	Terms preceding the ellipsis may be repeated. A numeric constant after an ellipsis limits the number of times the term may be repeated.

Common Command Options

There are some options which are available with several editor commands. These options are described below for all commands in which they can be specified.

Line Specifications

With these options, you can restrict the effect of a command to a certain range of lines:

.X	The command affects only the lines from the line labelled ".X" to the last line.
.X.Y	The command affects only the lines from the line labelled ".X" to the line labelled ".Y".

X and Y can also be any label of 1 to 4 alphabetical characters (see LABEL command).

Column Specifications

With these options, you can restrict the effect of a command to a certain range of columns. These 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.

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

Displayed or Non-displayed Lines

With one of the following options, you can specify that only excluded or only included lines are to be affected by a 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

With these options, you can specify the direction in which a command is to operate:

NEXT	The command affects the next line (starting from the cursor position) in which the specified <i>string</i> occurs.
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

With these options, you can specify whether only special occurrences of the specified *string* are to be affected by a command:

CHARS	The command affects any line in which the specified <i>string</i> occurs.
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.

ADVANCE

ADVANCE	<table border="1"> <tr> <td>ON</td> </tr> <tr> <td>OFF</td> </tr> <tr> <td>PAGE</td> </tr> </table>	ON	OFF	PAGE
ON				
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 edit area after an update.

If an unqualified ADVANCE command is issued, it is interpreted as ADVANCE ON.

AORDER

AORDER	<table border="1"> <tr> <td>ON</td> </tr> <tr> <td>OFF</td> </tr> </table>	ON	OFF
ON			
OFF			

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

If an unqualified AORDER command is issued, it is interpreted 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.

If an unqualified AUTOSAVE command is issued, it is interpreted as an AUTOSAVE ON command.

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 (TC, TO, LJ, RJ, etc.).

<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 edit area.

To see the current boundary settings, issue the BNDS line command.

BOTTOM

BOTTOM

This command is used to scroll to the end of the object being edited.

CANCEL

CANCEL

This command cancels all changes made after the last SAVE or STOW command and leaves the editor.

CAPS

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

This command is used to switch upper-case translation on and off. The command applies only to new or modified lines.

ON	Line is translated to upper case.
OFF	Line is not translated; that is, it remains as entered.
PGM	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 $\left\{ \begin{array}{c} \text{ALL} \\ n \\ n\ m \end{array} \right\}$

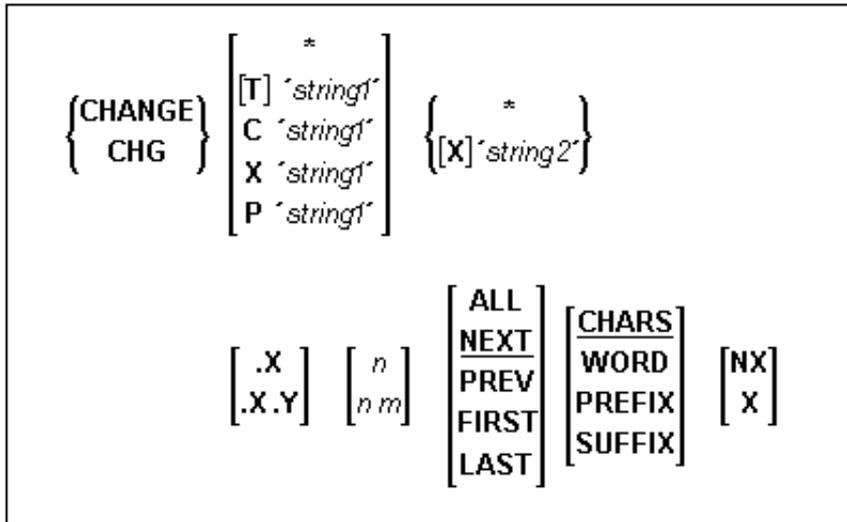
This command is used to center source code.

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

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

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

CHANGE



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

You can specify the string to be replaced (*string1*) as follows:

T ' <i>string1</i> '	Delete lines that contain the <i>string</i> irrespective of lower case or upper case. This is the default.
' <i>string1</i> '	Same as T ' <i>string</i> '.
C ' <i>string1</i> '	Delete lines that contain the <i>string</i> exactly as specified.
X ' <i>string1</i> '	Delete lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P ' <i>string1</i> '	Delete lines that contain the <i>string</i> which includes the following wildcard characters: = any character § alphabetical character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Use the replacement string specified in a previous command (for example, FIND, CHANGE, EXCLUDE).

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

All other options of the CHANGE command are described in the section Common Command Options.

Using CHANGE Together with Other Commands

To repeat the execution of a CHANGE command, you use the command RCHANGE.

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, CHANGE, RFIND and RCHANGE:

First, you search for the string:

FIND *'string'*

When the string has been found, you can decide whether to:

- replace it:
CHANGE *'string'* *'new-string'*
- or search for the next occurrence of the string by repeating the FIND command:
RFIND

When the next occurrence of the string has been found, you can again decide whether to:

- replace it by repeating the CHANGE command:
RCHANGE *'*
- or search for the next occurrence of the string by repeating the FIND command:
RFIND

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 labelled by ".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 prefix "NAME" (exactly as entered here) into "APPL" in all displayed lines in the block labelled by ".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".

Example 5:**CHG** **'OLD'** *****

This command replaces the next occurrence of the string "OLD" by the same new string as specified in the last CHANGE command.

COLS

COLS <table border="1"> <tr> <td>ON</td> </tr> <tr> <td>OFF</td> </tr> </table>	ON	OFF
ON		
OFF		

This command displays a line at the top of the edit area showing column positions.

To display the column positions, you can also use the line commands COLS.

COPY

```
COPY { *object-name } [ssss [nnnn] ]
```

This command copies an external object to the editor and inserts the object after the line marked with the A line command or before the line marked with the B line command (see line commands). The optional parameter *ssss* can be used to specify the line of the external object at which the include operation is to begin. The optional parameter *nnn* can be used to specify the number of lines to be copied.

CURSOR

```
CURSOR
```

This command returns the cursor to the command field when you next press ENTER.

CWINDOW

```
CWINDOW [ n ]  
          [ n m ]
```

This command is used to copy 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.

DELETE

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

This command is used to delete lines.

You can specify that only lines which contain a specified character *string* are to be deleted. You have the following options:

T' <i>string</i>	Delete 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>	Delete lines that contain the <i>string</i> exactly as specified.
X' <i>string</i>	Delete lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string</i>	Delete lines that contain the <i>string</i> which includes the following wildcard characters: = any character § alphabetical character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Use the search string specified in a previous command (for example, FIND, CHANGE, EXCLUDE).

All other options of the DELETE command are described in the section Common Command Options.

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

Example 1:

DEL C'NAME' 1 20 ALL PREFIX NX

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:

DEL C'Abc' .X.Y 10 30 ALL

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

To delete lines, you can also use the line commands D, D*n* and DD.

DOWN

DOWN [<i>n</i>]

This command is used to scroll downwards in the source code.

The parameter *n* specifies the number of lines to be scrolled downwards. If *n* is omitted, the scroll amount is determined by the scroll mode.

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 in the program editor.

- 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 in the editor.

OFF	Empty lines are not deleted.
ON	Empty lines are deleted.

If you enter EMPTY without any parameter, it is interpreted as EMPTY OFF.

ESCAPE

```
ESCAPE [ON]
       [OFF] [character']
```

This command activates/deactivates the escape character to precede line commands entered in the first column of the source code.

The parameter *character* is the special character to be used. The default escape character is the period (.).

If you issue the ESCAPE command without any parameter, it is interpreted as ESCAPE ON.

EX, EY, EX-Y

EX
EY
EX-Y

These commands are used to delete lines in a program.

- 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

{ <u>EXCLUDE</u> }	*	[T	'	string	']	[.X	.]]	[n]	[ALL]	[CHARS]
	string		n	NEXT	WORD															
	string		m	PREV	PREFIX															
	C		'	string	'	FIRST			SUFFIX											
	X		'	string	'	LAST														
P	'	string	'																	

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. You have the following options:

T 'string'	Delete lines that contain the <i>string</i> irrespective of lower case or upper case. This is the default.
'string'	Same as T 'string'.
C 'string'	Delete lines that contain the <i>string</i> exactly as specified.
X 'string'	Delete lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P 'string'	Delete lines that contain the <i>string</i> which includes the following wildcard characters: = - any character § - alphabetical character # - numeric character \$ - special character ^ - non-blank character - - non-numeric character < - lower-case letter > - upper-case letter
*	Use the search string specified in a previous command (for example, FIND, CHANGE, EXCLUDE).

All other options of the EXCLUDE command are described in the section Common Command Options.

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

Example 1:

EXCLUDE .X .Y

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

Example 2:

EXCLUDE C'NAME' ALL PREFIX

This command excludes from display all lines which contain "NAME" (in upper case as entered here) as a prefix to a word.

To re-display excluded lines, you use the editor command INCLUDE.

EXIT

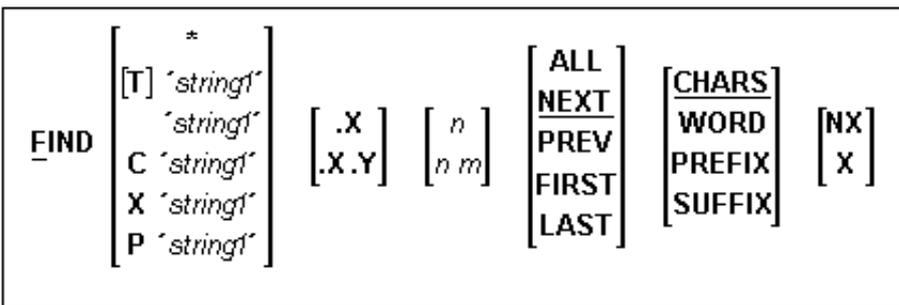


This command is used to leave the editor. If any changes have been made since the last SAVE or STOW, then you are prompted to save your changes or exit without saving.

Note:

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

FIND



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

You can specify the *string* as follows:

T' <i>string</i>	Search 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>	Search for the <i>string</i> exactly as specified.
X' <i>string</i>	Search for the string that corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string</i>	Search for a <i>string</i> which includes the following wildcard characters: = - any character § - alphabetical character # - numeric character \$ - special character ^ - non-blank character - - non-numeric character < - lower-case letter > - upper-case letter
*	Search for the <i>string</i> specified in the previous FIND command.

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

All other options of the FIND command are described in the section Common Command Options.

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 delineated by ".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.

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 delineated by ".X" and ".Z" and between columns 20 to 30.

Example 4:

F X'6C' SUFFIX NX

This command searches for the character that is hexadecimally represented as "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'

This command searches for the following character string: **r'w**

Example 7:**F '''**

This command searches for an apostrophe.

The FIND command differs from the LOCATE command in the following ways:

- The FIND command is more effective for text searches while the LOCATE command is used primarily to find line numbers or line labels.
- The LOCATE command finds only text in upper case beginning in column one of the editor. In addition, in order to find a string, the data in the editor must be in alphabetical order.
- When a line is located with the LOCATE command, the cursor is placed in the prefix area and the line is placed at the top of the editor; with the FIND command, the cursor is placed on the string searched and the line is not necessarily placed at the top of the editor.

To repeat the execution of a FIND command, use the command RFIND.

FIX

FIX [n]

This command is used to specify the number of columns *n*, starting with column 1, to remain in display when scrolling to the right.

FLIP

FLIP

This command is used to toggle the PF key display between PF1-12 and PF13-24.

HEX

HEX [ON] [OFF]

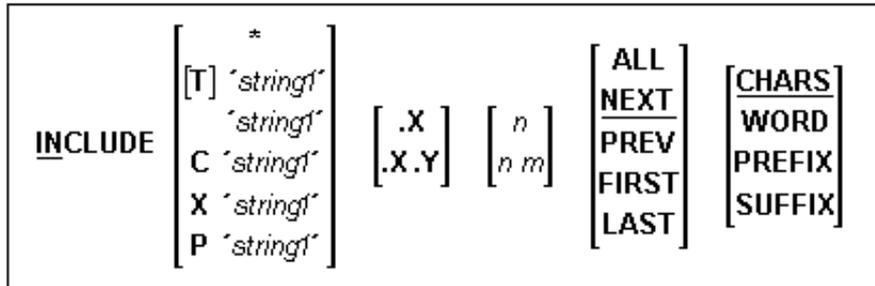
This command is used to switch hexadecimal display mode on or off.

HOME

HOME

This command returns the cursor to the command field at the next ENTER.

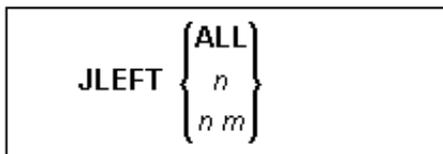
INCLUDE



This command is used to re-display lines that were excluded from display by an EXCLUDE command. The command takes the same parameters as the EXCLUDE command.

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

JLEFT



This command is used to align source code left-justified.

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

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

Example:

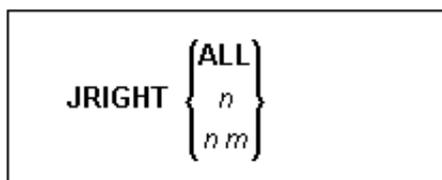
BNDS 10;JLEFT 15 20

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

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

See also the editor command JRIGHT.

JRIGHT



This command is used to align source code right-justified.

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

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

Example 1:

BNDS 4 40;JRIGHT 6 18

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

Example 2:

BNDS 10;JRIGHT 15

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

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

See also the editor command **JLEFT**.

JUSTIFY

JUSTIFY	<table border="0"> <tr> <td rowspan="3" style="font-size: 2em; vertical-align: middle;">}</td> <td>LEFT</td> </tr> <tr> <td>RIGHT</td> </tr> <tr> <td>BOTH</td> </tr> </table>	}	LEFT	RIGHT	BOTH
}	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-code lines with subsequent lines. Both commands apply only within the horizontal boundaries as set with the editor command **BNDS**.

LEFT	The source code is aligned to the left boundary.
RIGHT	The source code is aligned to the right boundary.
BOTH	The source code is aligned to both boundaries.

Example:

With these commands, you set the horizontal boundaries to columns 10 and 60, and activate left-justification:

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 source code 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 edit area) with the specified *.label*.

The *label* is a string of 1 to 5 alphanumerical characters.

Example:

To label the current line with ".X", you enter the command:

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 command "LABEL .Y" to mark that line with ".Y".

To mark a line with a label, you can also use the line command ".label".

LAST

```
LAST
```

This command recalls the editor command last issued. This command is placed in the command line and can be modified as required.

LC

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

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. You have the following options:

T' <i>string</i>	Change 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>	Change lines which contain the <i>string</i> exactly as specified.
X' <i>string</i>	Change lines which contain the string that corresponds to the specified hexadecimal character <i>string</i> .
P' <i>string</i>	Change lines which contains a <i>string</i> that includes the following wildcard characters: = any character § alphabetical character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case letter > upper-case letter
*	Change lines which contain the <i>string</i> used in the previous command in which a string was specified.

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

All other options of the LC command are described in the section Common Command Options.

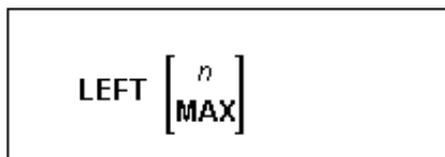
If you enter the LC command without any parameters, the current line is changed to lower case.

Example:

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

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

LEFT

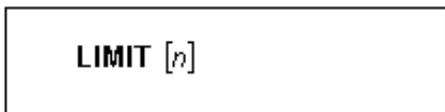


This command scrolls the source code to the left

<i>n</i>	Scrolls <i>n</i> number of columns to the left.
MAX	Scrolls the maximum amount to the left.

If *n* or **MAX** is omitted, the scrolling amount is determined by the scroll mode.

LIMIT



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

[<u>LOCATE</u>]	{	0	}
		<i>n</i>	
		<i>.label</i>	

This command is used to scroll a specific line to the top of the edit 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 labelled <i>.label</i> current.

Examples:

LOC 32 Places line number 32 at the top of the edit area.

32 Same as above.

LOC .X Places the line labelled ".X" at the top of the edit area.

LOG

LOG	[ON]
		OFF	

This command activates or deactivates the 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. Using the UNDO command you can consecutively back out changes made since the beginning of the edit 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 a mask line, type "mask" over any line number in the editor and press ENTER. An empty line appears in which you can type your mask. This mask is active until you update the mask with a new mask line or until you deactivate the mask function.

When the mask function is activated using MASK ON, the mask line appears in all lines added with a line insertion operation. Note, however, that any inserted line is deleted at the next press of ENTER if nothing is added to it.

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

MSO

```
MSO [ON]
    [OFF]
```

This command is used to specify whether multiple session operations are possible or not. A multiple session operation is an operation in which data are exchanged between two editing sessions, for example in copying text from one program to another in split-screen mode.

MWINDOW

```
MWINDOW [ n ]
         [ n m ]
```

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.

NEXT

```
NEXT [ * ]
     [ object-name ]
```

This command is used to display the *next* parallel editing session, assuming two or more editing sessions are running concurrently. 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.

NULLS

```
NULLS [ON]
      [OFF]
```

This command is used to determine if the source-code lines are to be filled with null characters.

ON	The end of each line is filled with null characters.
OFF	Lines are not filled with null characters.

ORDER

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

This command is used to join source-code 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 editor command BNDS.

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-code lines, you can also use the line commands TF, TO and TOO.

OWINDOW

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

This command is used to overlay a data window according to the command parameters. The moved/copied lines are merged with this window, that is, blank characters in the window are overlaid.

<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.

POINT

POINT

This command places the line marked with line command NZ at the top of the data area.

POWER

```
POWER
```

This command switches the editor to text-entry 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 edit area.

PROF

```
PROF [n]
```

This command displays your editor profile at the top of the edit 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 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 your editor profile utility. It enables you to modify your editor defaults for current and future sessions. The editor profile utility is described in more detail in section Modify Profile Settings.

PROTECT

```
PROTECT [INS]
          [ON]
          [OFF]
```

This command is used to protect the prefix area (line numbers). To enter line commands with the prefix area protected, type the line command in column 1 of the edit area preceded by the escape character.

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 line commands and deletes all line labels.

RFIND

RFIND

This command repeats the last FIND command.

RIGHT

RIGHT $\left[\begin{matrix} n \\ \text{MAX} \end{matrix} \right]$

<i>n</i>	Scrolls <i>n</i> number of columns to the right.
MAX	Scrolls the maximum amount to the right.

If *n* or MAX is omitted, the scrolling amount is determined by the scroll mode.

SET TYPE

SET TYPE $\left\{ \begin{matrix} \text{PROGRAM} \\ \text{SUBROUTINE} \\ \text{SUBPROGRAM} \\ \text{HELPROUTINE} \\ \text{COPYCODE} \\ \text{TEXT} \end{matrix} \right\}$

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

SHIFT

```
SHIFT [n] [RIGHT]
                [LEFT]
```

This command shifts a block of lines between the .X and .Y labels to the right or left by *n* columns (or until 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:5).
RIGHT	Shifts block of lines to the right (default).
LEFT	Shifts block of lines to the left.

SORT

```
SORT [n m] [.X] [A]
                [.X.Y] [D]
```

The SORT command sorts lines in the editor in ascending or descending alphabetical order. An unqualified SORT command sorts all data in the object in ascending order.

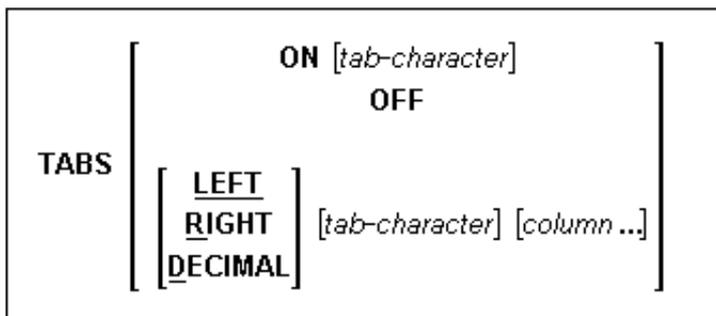
<i>n m</i>	Sorts from column <i>n</i> to column <i>m</i> .
.X	Sorts from line labelled .X to end of object.
.X.Y	Sorts from line labelled .X to line labelled .Y (where .X and .Y are any string of up to four characters).
A	Sorts data in ascending order (A to Z).
D	Sorts data in descending order (Z to A).

SWAP

```
SWAP
```

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

TABS



This command is used to control tabulator settings.

You can enable or disable logical or physical tabulation using the command TABS ON/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 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 data and automatically move them to a specific tab position by preceding them with a logical tabulation character. One tabulation character moves the data to the next tab position, two tabulation characters move the data to the second tab position, etc.

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

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

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

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

To tabulate data in a specific column, multiple tab characters are possible: 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(EFT), R(IGHT) or D(ECIMAL) after each tabulation character to specify placement of data for the tab position.

Examples of the TABS Command

The following examples assume the ampersand (&) to be the tabulation character:

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 DM &13 IS
```

is displayed as follows:

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

Example 4 - Mixed Justification:

Issue the 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:

Type over the asterisks in the =tabs line with other special characters and specify left justification for each one as follows:

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

After you press ENTER, the input text line

```
=first$second#third&fourth]fifth
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
                                                first
                                         second
                                   third
                             fourth
                         fifth
```

Example 6 - Using a Blank as Tabulation Symbol:

Issue the command

TABS ' '

which activates tabulation with one blank as tabulation character. This means that words separated by one blank are tabulated. After you press ENTER, the input text line

```
this is a blank tabulation
```

is displayed as follows:

```
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6
          this    is      a      blank  tabulation
```

TOP

```
TOP
```

This command is used to scroll to the beginning of the object being edited.

UC

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

The UC command converts one or more lines to upper case. It applies the same parameters as the LC command. If you enter the UC command without parameters, it changes the current line to upper case.

UNDO

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

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.

You can specify the following parameters with the UNDO command:

ALL	All modifications made in the current edit session are backed out.
<i>n</i>	The last <i>n</i> modifications are backed out.

UP

```
UP [n]
```

This command scrolls upwards in the source code.

The parameter *n* specifies the number of lines to be scrolled upwards. If *n* is omitted, the scroll amount is determined by the scroll mode.

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> .

X

```
X
```

This command places the line marked with line command *.X* at the top of the data area.

XSWAP

```
XSWAP
```

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

Y

```
Y
```

This command places the line marked with line command *.Y* at the top of the data area.

Line Commands

You can enter a line command on any data line by typing over the line number on the left of your edit screen. A line command always applies to the line in which you enter it (or to a block of lines marked by multiple line commands).

When the insert mode is active, it is not possible to enter line commands in the line number field. Toggle the insert mode to non-insert mode.

Line commands can also be entered in command line at the top of the editor screen. In this case, the command must be preceded by a colon (:) and applies to the line marked by the cursor.

Command	Explanation
)	Moves this line right by two columns.
) <i>n</i>	Moves this line right by <i>n</i> columns, irrespective of any other data in the line: you may lose data in the moved line.
)) <i>n</i>	Marks first line of a block to be moved right by <i>n</i> columns. A second)) <i>n</i> is required to mark the last line of the block. The block is moved regardless of any other data in the block: you may lose data in the moved block.
(Moves this line left by two columns.
(<i>n</i>	Moves this line left by <i>n</i> columns regardless of any other data (you may lose data in the moved lines).
((<i>n</i>	Marks first line of a block to be moved left by <i>n</i> columns. A second ((<i>n</i> is required to mark the last line of the block.
<	Moves data in this line left by two columns.
>	Moves data in this line right by two columns.
> <i>n</i>	Moves data in this line right by <i>n</i> columns (or up to last non-blank character: no data are lost).
>> <i>n</i>	Marks first line in a block to be moved to the right by <i>n</i> columns (or until last non-blank character). A second >> is required to mark the last line of the block.
< <i>n</i>	Moves data in this line left by <i>n</i> columns (or until first non-blank character).
<< <i>n</i>	Marks first line in a block to be moved to the left by <i>n</i> columns (or until first non-blank character). A second << is required to mark the last line of the block.
A	Marks the target line for a move (M, <i>Mn</i> , MM) or copy (C, <i>Cn</i> , CC) line command. The moved/copied line(s) are inserted <i>after</i> this line.
B	Marks the target line for a move (M, <i>Mn</i> , MM) or copy (C, <i>Cn</i> , CC) line command. The moved/copied line(s) are inserted <i>before</i> this line.
BNDS	Displays the boundary positions in this line.
C	Copies this line to the position indicated by an A, B or O line command.
C<i>n</i>	Copies the present line and the next <i>n-1</i> lines to the position indicated by an A, B or O line command.
CC	Marks the first line of a block of lines to be copied. A second CC command is required to mark the last line of the block to be copied. The lines are copied to the position indicated by an A, B or O line command.
CX	Copies the line labelled .X. Inserts data after this line.
CY	Copies the line labelled .Y. Inserts data after this line.
CX-Y	Copies the block of lines from the line labelled .X to the line labelled .Y. Inserts data after this line.
COLS	Displays the column positions in this line.
D	Deletes this line.
D<i>n</i>	Deletes the present line and the next <i>n-1</i> lines.

Command	Explanation
DD	Marks the first line of a block to be deleted. A second DD command is required to mark the last line of the block to be deleted. The deletion is performed after second the DD has been entered.
DX	Deletes the line labelled .X.
DY	Deletes the line labelled .Y.
DX-Y	Deletes the block of lines from the line labelled .X to the line labelled .Y.
F	Includes the first excluded line.
Fn	Includes the first <i>n</i> excluded lines.
I	Inserts one line. The editor switches to insert mode. This means if you type data or enter a blank on the new line and press ENTER, a new line is automatically inserted and the cursor placed in it. If you enter no new data in an inserted line and press ENTER, the editor leaves insert mode and the blank line is deleted (see also the editor command EMPTY). You can also fill an inserted line with a predefined content (see the editor command MASK).
In	Inserts <i>n</i> lines. You may type data in the new lines. When you press ENTER, unused lines are deleted but one blank line remains with the cursor in it (editor stays in insert mode).
.I(obj,ssss,nnnn)	Inserts any object contained in the current library into the edit area. This command is entered in the editor area, not in the prefix area. The period preceding the "I" is the escape character. .I(*) invokes a selection list of objects in the current library. The "ssss" entry indicates the line at which the include operation is to begin. For example, setting "ssss" to 20 causes the insertion to begin with the 20th line in the program. The "nnnn" entry indicates the number of lines to be inserted. If the object is a Natural map, an INPUT USING MAP statement with all defined variables is automatically included in the current line. Note: Only stowed data areas can be included into the source area.
J	Joins next line with this one. You can specify how many of the characters of the following line are to be joined by placing the cursor at the point in the line where it is to be separated and press ENTER. To join the entire line, place the cursor outside the line to be joined. This command is identical to the TJ command.
Ln	Includes the last <i>n</i> excluded lines.
LC	Changes this line to lower case.
LCn	Changes the present line and the next <i>n-1</i> lines to lower case.
LCC	Marks the first line of a block to be changed to lower case. A second LCC is required to mark the last line in the block.
LJ	Justifies the data within the set boundaries in this line with the left boundary.
LJJ	Marks the first line of a block of data within the set boundaries to be justified to the left. A second LJJ command is required to mark the last line of the block to be justified. The justification is performed after the second LJJ command has been issued.
M	Moves this line to the position indicated by an A, B or O line command.
Mn	Moves the present line and the next <i>n-1</i> lines to the position indicated by an A, B or O line command.
MM	Marks the first line of the block to be moved. A second MM command is required to mark the last line of the block to be moved. The lines are moved to the position indicated by an A, B or O line command.

Command	Explanation
MASK	Inserts a blank line in the editor into which you can create a mask. This line is inserted whenever the insert (<i>In</i>) line command is used to create one or more new lines (see also the editor command MASK).
MX	Moves the line labelled .X. Inserts it after this line.
MY	Moves the line labelled .Y. Inserts it after this line.
MX-Y	Moves the block of lines from the line labelled .X to the line labelled .Y. Inserts it after this line.
N	Modifications made in this line do not take effect when ENTER is pressed.
NZ	The line marked with this command is placed at the top of the editor area when a POINT editor command is issued.
O	Marks this line as target line for a move (M, <i>Mn</i> , MM) or copy (C, <i>Cn</i> , CC) line command. The moved/copied line(s) are merged with this line, that is, blank characters in the line are overlaid.
On	Marks the present line and the next <i>n-1</i> lines as target lines for a move (M, <i>Mn</i> , MM) or copy (C, <i>Cn</i> , CC) line command. The moved/copied lines are merged with these lines, that is, blank characters in the lines are overlaid.
OO	Marks the first line of a block of target lines for a move (M, <i>Mn</i> , MM) or copy (C, <i>Cn</i> , CC) line command. A second OO command is required to mark the last line of the block of target lines. The moved/copied line(s) are merged with these lines, that is, blank characters in the lines are overlaid.
R	Repeats this line once.
Rn	Repeats this line <i>n</i> times.
RR	Marks the first line of a block to be repeated. A second RR command is required to mark the last line of the block to be repeated. The repeat operation is performed after the second RR has been entered.
RRn	Repeats the block of lines <i>n</i> times.
RJ	Justifies the data within the set boundaries in this line with the right boundary.
RJJ	Marks the first line of a block of data within the set boundaries to be justified to the right. A second RJJ command is required to mark the last line of the block to be justified. The justification is performed after the second RJJ has been issued.
S	Splits this line into two lines beginning at the cursor position. Type in the command, move the cursor to the position where the line is to be split, and press ENTER.
T	Scrolls the data to make the marked line the top line.
TABS	Displays the tab positions in this line.
TC	Centers the data within the set boundaries in this line.
TCC	Marks the first line of a block of data within the set boundaries to be centered. A second TCC command is required to mark the last line of the block of the centered. The centering is performed after the second TCC command has been issued.
TE	Switches editor to text enter mode (blank screen to end of screen).
TF	Joins this line with the following lines until the next blank line. The bounds settings can be used to restrict the columns affected (see BNDS command).
TFn	This line command may be entered with a numerical value specifying the right boundary, e.g. the line command TF50 orders data with column 50.
TI	Inverts sequence of all characters in the current line and within the set boundaries.

Command	Explanation
TII	Marks the first line of a block of text to be inverted within set boundaries. Requires a second TII to mark the last line of the block.
TJ	Joins next line with this one. Same as the join (J) command.
TO	Joins this line with the next one.
TOO	Marks the first line of a block of data within the set boundaries to be joined. A second TOO command is required to mark the last line of the block to be joined. The function is performed after the second TOO has been issued.
TS	Splits this line into two lines at the cursor position; an empty line is also automatically inserted, but deleted if unused (identical to S line command).
UC	Changes this line to upper case.
UCn	Changes the present line and the next <i>n-1</i> lines to upper case.
UCC	Marks the first line of a block to be changed to upper case. A second UCC is required to mark the last line of the block.
W	Opens window with one line.
Wn	Opens window with <i>n</i> lines.
WC	Copies the data window. The cursor position marks the column at which this line is to be split to insert the copied data.
WCn	Splits this line in column <i>n</i> , and copies the data between the two parts of the line.
WE	Marks end of data window. Works in the same way as WS. If the window is to start and end in the same line, overwrite the the WS command with the WE command. The editor acknowledges the set window with message WW in the line command field.
WM	Moves the data window. Works in the same way as WC, but the original data are deleted after the copy operation.
WMn	Splits this line in column <i>n</i> , and moves the data between the two parts of the line.
WS	Marks start of data window. The cursor position marks the column from which data are read. If the cursor is not in the line for which the command is entered, column 1 is taken.
WSn	Data window starts in column <i>n</i> of this line.
X	Excludes this line.
Xn	Excludes the following <i>n</i> lines.
XX	Marks the first line of the block to be excluded. A second XX is required to mark the second line of the block.
.label	Marks this line with ".label". The <i>label</i> may be any string of 1 to 4 alphabetical characters. See also the editor command LABEL. For example: . X names this line . X. and .Y names this line .Y.