



NATURAL

Natural
Software AG Editor
Version 3.1.5 for Mainframes



This document applies to Natural Version 3.1.5 for Mainframes and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© March 2002, Software AG
All rights reserved

Software AG and/or all Software AG products are either trademarks or registered trademarks of Software AG. Other products and company names mentioned herein may be the trademarks of their respective owners.

Table of Contents

Getting Started with the Software AG Editor	1
Getting Started with the Software AG Editor	1
Invoking the Software AG Editor	1
Using the Software AG Editor Screen	1
Entering Editor Main Commands	2
Specifying Default Scrolling Amounts	2
Entering Editor Line Commands	2
Entering Data	3
Using Commands	3
A Few Tips	3
Setting Your Editor Profile	3
Using the Software AG Editor	6
Using the Software AG Editor	6
Change a Specific Character String	6
REPEAT FIND and REPEAT CHANGE Commands	8
Center Data	8
Copy, Move, Overlay and Repeat Lines	9
Example of Overlaying Data	10
Copy a Window with Data	12
Line Commands Available	12
Main Commands Available	12
Define Horizontal and Vertical Boundaries	15
Define a Mask Line	16
Display Boundary, Tab and Column Positions	17
Exclude/Include Lines from Display	17
Find a Specific Character String	18
String Definition Operand	18
String-Matching Operand	19
Direction Operand	19
Line-Type Operand	19
Block Operand	19
Columns Operand	20
Examples of FIND Command	20
Insert/Delete Lines	20
POWER Command	21
DELETE Command	21
Justify Data	22
Locate a Specific Line	24
Differences between the LOCATE and FIND Commands	24
Order Data Between Specified Boundaries	24
Scroll Data on the Editor Screen	25
PF Keys	25
Settings for SCROLL Field	25
Main Commands for Scrolling	26
Sort Lines in Alphabetical Order	27
Store Data / Leave the Editor	27
Use the Physical/Logical Tabulator	28
TABS main command	28
TABS Line Command	28
Software AG Editor Main Commands	32
Software AG Editor Main Commands	32
General Information	32
Main Commands Command Syntax	32

Common Command Options	33
Line Specifications	33
Column Specifications	33
Displayed or Nondisplayed Lines	33
Direction of Operation	33
Special Occurrences	34
Main Command Descriptions	34
ADVANCE	34
AORDER	34
AUTOREN	34
AUTOSAVE	35
BNDS	35
BOTTOM	35
CANCEL	35
CAPS	35
CENTER	36
CHANGE	36
CLEAR	38
COLS	38
CURSOR	38
CWINDOW	39
DELETE	39
DOWN	40
DWINDOW	40
EMPTY	40
END	40
ESCAPE	41
EXCLUDE	41
FIND	42
FIX	43
HEX	44
INCLUDE	44
JLEFT	44
JRIGHT	45
JUSTIFY	45
LABEL	46
LC	46
LEFT	47
LIMIT	47
LOCATE	48
LOG	48
MASK	49
MSO	49
MWINDOW	49
NULLS	49
ORDER	50
POWER	50
PROFILE	50
PROTECT	51
RCHANGE	51
RECOVERY	51
RENUMBER	51
RESET	52
RFIND	52
RIGHT	52
SAVE	52

SORT	52
TABS	53
TOP	55
UC	56
UNDO	56
UNREN	56
UP	56
WINDOW	57
XSWAP	57
Software AG Editor Line Commands	58
Software AG Editor Line Commands	58
General Information	58
Entering a Line Command	58
Overview of Line Commands	58

Getting Started with the Software AG Editor

This section covers the following topics:

- Invoking the Software AG Editor
 - Using the Software AG Editor Screen
 - Using Commands
 - A Few Tips
 - Setting Your Editor Profile
-

Invoking the Software AG Editor

The screen from which you enter the Software AG Editor depends on the application you are using and the type of object you wish to edit or display.

For example, if you wish to edit a Natural member in Natural ISPF, you might enter the Software AG Editor from the Natural object entry screen.

The ways of entering the Software AG Editor are described under the appropriate section headings in the documentation for the application you are using.

The commands available to you depend on the application you are using.

Examples of defined objects using the Software AG Editor are given in the appropriate sections of the documentation for the application you are using.

Using the Software AG Editor Screen

If you specify a new object name for editing, an editor screen similar to the following appears:

Entering Data

Enter **data** in the data area to the right of the apostrophes or line numbers.

You can start your edit session by entering data into the Software AG Editor screen and using the edit commands described in the section Software AG Editor Main Commands and Software AG Editor Line Commands.

Using Commands

the Software AG Editor provides two types of commands with which you can control your editing session:

- **Editor commands**
Enter Editor commands in the COMMANDline on the second screen line.
You can enter several commands in the same input operation if you separate them with a semi-colon (;).
- **Line commands**
Line commands refer to the line on which they are entered or to a block of data delineated by line commands.

There are three ways of entering line commands:

- Type over the line number or apostrophes in any line on the left of your screen and press **Enter**.
- Enter the line command in the COMMANDline preceded by a colon (:).
The cursor then marks the line to be addressed. Press **Enter**.
- Enter the line command from the first column in the data area of any line preceded by the escape character.

For detailed information on all commands, see Software AG Editor Main Commands and Software AG Editor Line Commands.

A Few Tips

When using commands, please bear in mind the following:

- Depending on the configuration of your installation, main commands and line commands can be entered in lower case. In the description of commands that follows, all commands appear in upper case to distinguish them as commands;
- Commands are processed in the following order:
 - Data updates
 - Line commands
 - Main commands
- If line numbers appear in the first six left-hand columns of your screen, Column 8 on the screen corresponds to Column 1 of the edit area. This is important when specifying columns in some main commands;
- The command format in the examples show the abbreviated form of each command. These examples are by no means exclusive. For a complete summary of command syntax, see the sections Software AG Editor Main Commands and Software AG Editor Line Commands. These sections give a description of the available main and line commands grouped according to their functional area.

Setting Your 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.

You can modify single settings in your Editor Profile using appropriate commands. The new settings are valid for the remainder of the edit session or until you change them again using the appropriate commands.

To display the current settings of your Editor Profile, issue the PROFILE command.

The following lines appear at the top of the editor screen:

```

EDITNAT: NSPF101 (JOB1JCL) -Program->Struct-Free-30K----- columns 001 072
COMMAND===>                                SCROLL===> CSR
***** ***** top of data *****
=prof> date: 05/06/89 10:24:02 user: MBE      init size:00000 size:00000
=prof> var   - 088,..recovery on (0004 0000)..autosave off... empty line off
=prof> mask off.caps off.hex off          nulls off.autoren off 10..auto order off
=prof> log on 0001.mso on .fix off        .escape off + . .tabs off    ...
=prof> advance off.protect off.limit on
=tabs>          *           *                       *                               *
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
***** ***** bottom of data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up    Down Swap Left Right Curso
    
```

The individual items of the Editor Profile are described in the following table.

Item	Description	Command
advance	Specifies whether the cursor moves to the next line automatically after a line update.	ADVANCE ON/OFF/PAGE
auto order	Automatically justifies text within defined boundaries.	AORDER ON/OFF
autoren	Specifies whether the Editor activates or deactivates the RENUMBER function.	AUTOREN ON/OFF
autosave	Activates/deactivates automatic save when END command is issued.	AUTOSAVE ON/OFF
caps	Specifies whether data are to be translated into upper case.	CAPS ON/OFF/PGM
date	The current date and time.	Nonmodifiable
empty line	Specifies if lines containing only space characters are to be deleted automatically.	EMPTY ON/OFF
escape	Specifies whether escape character is used to precede line commands.	ESCAPE ON/OFF <i>char</i>
fix	Specifies whether fixed number of columns are displayed and how many columns are to be fixed.	FIX ON/OFF <i>n</i>
hex	Specifies whether data are to be displayed in hexadecimal format.	HEX ON/OFF
init size / size	Number of lines in object when Editor was invoked. / Current number of lines in the object, excluding information lines (for example profile lines and message lines).	Nonmodifiable
limit	Specifies the maximum number of lines to be searched by a FIND or RFIND command.	LIMIT <i>n</i>
log	Enables/disables log file. When enabled, UNDO command can be used to backout last changes.	LOG ON/OFF
mask	Activates/deactivates the mask line function.	MASK ON/OFF
mso	Enables/disables multi-session operations such as copy.	MSO ON/OFF
nulls	Specifies whether the end of each source-code line is to be filled with null characters.	NULLS ON/OFF
protect	Specifies protection of line numbers.	PROTECT ON/OFF/INS
recovery	Activates/deactivates recovery function, specifies how often checkpoint save is made.	RECOVERY ON/OFF <i>n</i>
size	See init size.	Nonmodifiable
tabs	Activates/deactivates tabulation.	TABS ON/OFF
var	Specifies current line length.	BNDS <i>n m</i>
user	The current logon user.	Nonmodifiable

Using the Software AG Editor

This section describes the functions that are available for all object types using main commands. Note that the commands available to you depend on the application you are using.

This section covers the following topics:

- Change a Specific Character String
- Center Data
- Copy Lines
- Copy a Window with Data
- Define Horizontal and Vertical Boundaries
- Define a Mask Line
- Delete Lines
- Display Boundary, Tab and Column Positions
- Exclude/Include Lines from Display
- Find a Specific Character String
- Include Lines from Display
- Insert Lines
- Justify Data
- Leave the Editor
- Locate a Specific Line
- Move Lines
- Order Data Between Specified Boundaries
- Overlay Lines
- Repeat Lines
- Scroll Data on the Editor Screen
- Sort Lines in Alphabetical Order
- Store Data
- Use the Physical/Logical Tabulator

Change a Specific Character String

You can find and automatically change a given character string into another character string using the CHANGE main command.

The same operands as for the FIND command can be used with the CHANGE command. For the CHANGE command, however, the ALL directions operand means change **all occurrences of the specified string**.

After the change is performed, the message ==**chg**> appears in the line command field of the changed line.

Examples of CHANGE Command

Command	Explanation
CHG 'LOW' 'HIGH'	Change the first occurrence low into high (upper and lower case ignored).
CHG C'ops' 'SPF' .X .Y 28 32 all	Change ops (exactly as entered here) into SPF ; change all occurrences in the block labelled .X and .Y and between the Columns 28 and 32.
Repeated CHANGE commands:	
CHG * 'NEW'	Change the next occurrence of the string specified in the last CHANGE command to new string NEW .
CHG 'OLD' *	Change string OLD into the same new string as specified in the last CHANGE command.

If single quotation marks are part of the string to be changed, you must use a different separator in the CHANGE command, for example double quotation marks:

CHANGE "'string1'" "'string2'".

The screens below illustrate the second example.

To change all occurrences of the string *ops* to the string *SPF* between lines 140 and 170 and within columns 28 to 32 issue the following command:

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

The screen appears as shown below before the command is issued. The command appears in the COMMAND line on this screen:

```
S*>>EDIT-NAT:NSPF101(JOBLJCL)-Program->Report-Free-29K----- columns 001 072
COMMAND===> CHG C'OPS' 'SPF' .X .Y 28 32 ALL          SCROLL===> CSR
***** ***** top of data *****
=cols> -----1-----2-----3-----4-----5-----6-----7--
000010  RESET #JOBNAME(A8)
000020  RESET #FD(N3) #FL(A8) #FF(N3)
000030  RESET #TD(N3) #TL(A8) #TF(N3)
000040  COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050  SET CONTROL 'WL60C6B005/010F'
000060  INPUT 'ENTER PARAMETERS FOR LIBRARY COPY:'
000070  / 'FROM:  DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080  / 'TO :   DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090  // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100  /*JOBPARM LINES=2000
000110  //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120  // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130  // 'EJ=OFF,IM=D,ID='';',MAINPR=1,INTENS=1')
.X //STEPLIB DD DISP=SHR,DSN=OPS.SYSF.V5.ADALOAD
000150  // DD DISP=SHR,DSN=OPS.SYSF.V101.LOAD
000160  // DD DISP=SHR,DSN=OPS.SYSF.PROD.INST * OPS INSTALL
.Y // DD DISP=SHR,DSN=OPS.SYSF.SOURCE * OPS DOCUMENTS
000180  //DDCARD DD *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help Split End Suspe Rfind Rchan Up Down Swap Left Right Curso
```

When you press **Enter** to invoke the command entered on the COMMAND line of the screen above, the screen appears as shown below:

```

S*>>EDIT-NAT:NSPF101(JOB1JCL)-Program->Report-Free-29K 00004 char 'OPS'
changed
COMMAND===> SCROLL===> CSR
***** ***** top of data *****
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000010  RESET #JOBNAME(A8)
000020  RESET #FD(N3) #FL(A8) #FF(N3)
000030  RESET #TD(N3) #TL(A8) #TF(N3)
000040  COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050  SET CONTROL 'WL60C6B005/010F'
000060  INPUT 'ENTER PARAMETERS FOR LIBRARY COPY:'
000070  /      'FROM:  DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080  /      'TO   :  DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090  // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100  /*JOBPARM LINES=2000
000110  //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120  // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130  // 'EJ=OFF,IM=D,ID='';',MAINPR=1,INTENS=1')
.X      //STEPLIB DD DISP=SHR,DSN=SPF.SYSF.V5.ADALOAD
==chg> //          DD DISP=SHR,DSN=SPF.SYSF.V101.LOAD
==chg> //          DD DISP=SHR,DSN=SPF.SYSF.PROD.INST * OPS INSTALL
.Y      //          DD DISP=SHR,DSN=SPF.SYSF.SOURCE * OPS DOCUMENTS
000180  //DDCARD DD *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up      Down Swap Left Right Curso

```

All occurrences of the string **ops** have been changed to the string **SPF** between lines 140 and 170 and within the Columns 28 to 32.

REPEAT FIND and REPEAT CHANGE Commands

When changing character strings, good use can be made of the **REPEAT FIND** and **REPEAT CHANGE** feature, for instance the sequence:

```

FIND 'abc'
CHANGE 'abc' 'def'
RFIND
RCHANGE

```

allows you to find occurrences of a certain string and optionally change them with relatively small effort.

These commands can also be assigned to PF keys.

Center Data

The Editor also provides commands with which you can center specified data within set boundaries. For example, the sequence:

```

BNDS 5 60;CENTER 5 15

```

centers data in Lines 5 to 15 between Columns 5 and 60.

Note:

Only text already within the boundaries is centered. Text to the left and right of the boundaries is not affected.

Alternatively, you can use certain line commands to perform the centering function:

Line Command	Explanation
TC	Centers this line within the set boundaries.
TCC	Marks the first line in a block of data to be centered. Requires a second TCC line command to delineate the block. The centering is performed after the second TCC command is issued.

Copy, Move, Overlay and Repeat Lines

With the following line commands, you can copy, move or repeat lines or blocks of data. **Move** and **Copy** line commands can only be used in conjunction with an After (**A**), Before (**B**) or an Overlay (**O**) command.

Line Command	Explanation
A	Marks target line for Copy and Move line commands. Data are inserted after this line.
B	Marks target line for Copy and Move line commands. Data are inserted before this line.
C	Copy this line to position marked by an A , B or O line command.
C15	Copy the next 15 lines to position marked by an A , B or O line command.
CC	Marks the first line of the block to be copied. A second CC is required to delineate block. Copying is performed after target has been marked with 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.
M	Move this line to position marked by an A , B or O line command.
M15	Move the next 15 lines to position marked by an A or B line command.
MM	Marks the first line of the block to be moved. A second MM is required to delineate the block. The move is performed after the target has been marked with an A , B or O line command.
MX	Moves the line labelled .X . Inserts data after this line.
MY	Moves the line labelled .Y . Inserts data after this line.
MX-Y	Moves the block of lines from the line labelled .X to the line labelled .Y . Inserts data after this line.
O	Marks target line for Copy or Move line commands. Data are merged with this line. (Only blank characters within boundaries are changed).
O15	Merges first 15 lines of a block defined by a Copy or Move line command with the next 15 lines. (Only blank characters within boundaries of the target lines are changed).
OO	Marks the first line of a block to be merged with a block of data defined by a Copy or Move line command. A second OO command is required to delineate target block. (Only blank characters within boundaries are changed).
R	Repeat this line once.
R15	Repeat this line 15 times.

RR	Marks the first line of a block to be repeated. A second RR is required to delineate the block. The repeat is performed after second RR is entered.
RR15	Repeat block delineated by two RR15 line commands 15 times. The parenthesis line commands , below, move text the full number of columns specified, but only within the set boundaries , therefore, part of the moved text could disappear.
)	Move this line right by one column beginning with left boundary.
)15	Move this line right by 15 columns.
))15	Marks first line of a block to be moved right by 15 columns. A second))15 is required to delineate the block. The move is performed after the second))15 command is entered. Two unqualified) line commands move the block right by 1 column.
(Move this line left by one column.
(15	Move this line left by 15 columns.
((15	Marks first line of a block to be moved left by 15 columns. A second ((15 is required to delineate the block. The move is performed after the second ((15 command is entered. Two unqualified ((line commands move the block left by 1 column. If you use the greater than or less than symbols, below, to move data, the maximum move possible is up to the next non-blank character within the set boundaries .
>	Move this line right by one column.
>15	Move this line right by 15 columns.
>>15	Marks first line of a block to be moved right by 15 columns. A second >> is required to delineate the block. The move is performed after the second >> command is entered. Two unqualified >> line commands move the block right by 1 column.
<	Move this line left by one column.
<15	Move this line left by 15 columns.
<<15	Marks first line of a block to be moved left by 15 columns. A second << is required to delineate the block. The move is performed after the second << command is entered. Two unqualified << line commands move the block left by 1 column.

Example of Overlaying Data

The **O** line command allows you to merge single-column lists into multi-column format (i.e. tabular form). You can use the **O** line command in conjunction with the **M** (Move) or **C** (Copy) line commands.

The following two figures illustrate this function:

```

S*>>EDIT-NAT:NSPF101(JOB1JCL)-Program->Report-Free-30K----- columns 001 072
COMMAND===>                                SCROLL===> CSR
***** ***** top of data *****
000090 //JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 // 'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
oo0140 //STEPLIB DD
000150 //          DD
000160 //          DD
oo0170 //          DD
MM0180                                DISP=SHR,DSN=SPF.SYSF.V5.ADALOAD
000190                                DISP=SHR,DSN=SPF.SYSF.V101.LOAD
000200                                DISP=SHR,DSN=SPF.SYSF.PROD.INST * OPS INSTALL
MM0210                                DISP=SHR,DSN=SPF.SYSF.SOURCE * OPS DOCUMENTS
000220 //DDCARD DD *
000230 ADARUN DA=9,DE=3380,SVC=249
000240 //CMPRINT DD SYSOUT=X
000250 //CMPRT01 DD SYSOUT=X
000260 //CMWKF01 DD DUMMY
000270 //CMSYNIN DD *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up      Down Swap Left  Right Curso

```

In the figure above, lines 180 to 210 are marked with the **MM** (Move) line command. They are to be overlaid on lines 140 to 170, which are marked with the **OO** (Overlay) line command.

This figure shows the result of the line commands displayed in the first figure. Lines 180 to 210 have been overlaid on Lines 140 to 170:

```

S*>>EDIT-NAT:NSPF101(JOB1JCL)-Program->Report-Free-30K----- columns 001 072
COMMAND===>                                SCROLL===> CSR
***** ***** top of data *****
000090 //JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 // 'EJ=OFF,IM=D,ID='';'',MAINPR=1,INTENS=1')
000140 //STEPLIB DD          DISP=SHR,DSN=SPF.SYSF.V5.ADALOAD
000150 //          DD          DISP=SHR,DSN=SPF.SYSF.V101.LOAD
000160 //          DD          DISP=SHR,DSN=SPF.SYSF.PROD.INST * OPS INSTALL
000170 //          DD          DISP=SHR,DSN=SPF.SYSF.SOURCE * OPS DOCUMENTS
000220 //DDCARD DD *
000230 ADARUN DA=9,DE=3380,SVC=249
000240 //CMPRINT DD SYSOUT=X
000250 //CMPRT01 DD SYSOUT=X
000260 //CMWKF01 DD DUMMY
000270 //CMSYNIN DD *
000280 LOGON SYSMAIN2
000290 CMD C C * FM #FL DBID #FD FNR #FF TO #TL DBID #TD FNR #TF REP
000300 FIN
000310 /*
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up      Down Swap Left  Right Curso

```

Copy a Window with Data

You can specify a window with data for move or copy operations. This allows you to copy or move data that does not start or end at the beginning or end of a line. This function can be performed using Editor line commands and/or main commands.

When you define a window, all data on your screen between start and end of the window become part of the window.

Line Commands Available

Line Command	Explanation
WS	Marks start of data window. Cursor position marks the column from which data are read. If the cursor is not in the line for which the command is entered, the window starts in column 1.
WS <i>n</i>	Data window starts in Column n of this 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 WS command with the WE command. The Editor acknowledges the set window with messages WS (WS<i>n</i>) and WE (WE<i>n</i>) in the line command field, or with WW if the start and end of the window are in the same line. Before defining a new window, reset the old window with the RESET command to avoid a command conflict.
WE <i>n</i>	Data window ends in Column n of this line.
WC	Copies the data window. The cursor position marks the column at which this line is to be split to insert the copied data.
WC <i>n</i>	Splits this line in Column n , and copies the data between the two parts of the line.
WM	Moves the data window. Works in the same way as WC , but the original data is deleted after the copy operation.
WM <i>n</i>	Splits this line in Column n , and moves the data between the two parts of the line.

Main Commands Available

Main Command	Explanation
WINDOW	<p>Defines a window. The starting line and column, and the end line column are specified in the command parameters. At least one parameter is required. Examples:</p> <p>WINDOW 5 10 24 13 Defines a window starting in Line 5 / Column 24, and ending in Line 10, Column 13.</p> <p>WINDOW 5 10 24 Defines a window starting in Line 5 / Column 24, and ending in Line 10 / last column.</p> <p>WINDOW 5 10 Defines a window starting in Line 5 / first column, and ending in Line 10 / last column.</p> <p>WINDOW 5 5 Defines a window starting in line 5 / first column, and ending in line 5 / last column.</p>
CWINDOW	<p>Copy a window defined with the WINDOW command. Optional parameters specify the line at which the window is to be inserted. Examples:</p> <p>CWINDOW 5 Copies the window after Line 5.</p> <p>CWINDOW 5 24 Splits Line 5 at Column 24 and copies the window in between the two parts.</p>
DWINDOW	<p>Deletes a window of data defined by the WINDOW command.</p>
MWINDOW	<p>Moves window defined by the WINDOW command. Works like the CWINDOW command, but data in the original window is deleted after the copy operation.</p>

Example:

Consider the following text:

```

EDIT-PDS:MBE.COMN.SOURCE(WINEX) ----- Columns 001 072
COMMAND==>                                SCROLL==> CSR
***** top of data *****
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 Editor line commands and/or main commands.
000007
000008 Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
000010 of the window become part of the window. Available line commands are:
***** bottom of data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up    Down Swap Left  Right Curso
    
```

If you wish to move the whole sentence starting **Note that when...** (Line 8) to follow the first sentence of the displayed text ending **...copy operations** (Line 3).

1. Type the line command **WS** in Line 8, the first line of data to be moved, place the cursor in the required column (**N** of the word **Note**) and press **Enter**. The message **WS55** appears in the prefix area of Line 8, indicating the column number selected:

```

EDIT-PDS:MBE.COMN.SOURCE(WINEX) ----- Block is pending
COMMAND===>                                SCROLL===> CSR
***** ***** top of data *****
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 Editor line commands and/or main commands.
000007
WS55   Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
000010 of the window become part of the window. Available line commands are:
***** ***** bottom of data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up    Down Swap Left Right Curso

```

2. Now type the line command **WE** in Line 10, the last line of data to be moved, move the cursor to the last column to be moved (full stop '.' after **window**) and press **Enter**. The message **WE40** appears in the prefix area of Line 10:

```

EDIT-PDS:MBE.COMN.SOURCE(WINEX) ----- Block is pending
COMMAND===>                                SCROLL===> CSR
***** ***** top of data *****
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations. This
000004 allows you to copy or move data that does not start or end at the
000005 beginning or end of a line. This function can be performed using
000006 Editor line commands and/or main commands.
000007
WS55   Below are some examples of copying windows with data. Note that when
000009 you define a window, all data on your screen between start and end of
WE40   of the window become part of the window. Available line commands are:
***** ***** bottom of data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up    Down Swap Left Right Curso

```

3. Now type the line command **WM** in the line command field of Line 3 (the data will be moved to the following line, Line 4), and move the cursor to the column at which Line 3 is to be split (the blank before the word **This**). Press **Enter**. The following screen illustrates the result:

```

EDIT-PDS:MBE.COMN.SOURCE(WINEX) ----- Columns 001 072
  COMMAND===>                                SCROLL===>   CSR
***** ***** top of data *****
000001 Copy a Window with Data
000002
000003 You can specify a window with data for move or copy operations.
000004 Note that when
000005 you define a window, all data on your screen between start and end of
000006 of the window become part of the window.
000007 This
000008 allows you to copy or move data that does not start or end at the
000009 beginning or end of a line. This function can be performed using
000010 Editor line commands and/or main commands.
000011
000012 Below are some examples of copying windows with data.
000013
                                Available line commands are:
***** ***** bottom of data *****

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up    Down Swap Left Right Curso

```

To achieve the same result using main commands, you must use the following command sequence:

```
WINDOW 8 10 55 40;MWINDOW 3 64
```

Define Horizontal and Vertical Boundaries

The Editor provides commands which allow you to set horizontal and vertical boundaries within which certain functions can be performed.

With the BNDS main command, you can define horizontal boundaries. For example, the command:

```
BNDS 20 50
```

sets horizontal limits at the Columns 20 and 50. Commands which refer to these boundaries include the FIND, CHANGE, CENTER, ORDER, JLEFT, and JRIGHT main commands, as well as their corresponding line commands (TC, TO, LJ, RJ, etc.).

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

The following figure shows the result of the **BNDS 20 50** main command followed by a BNDS line command in Line 1:

```

S*>>EDIT-NAT:NATLIB1(JOB1JCL)-Program->Report-Free-30K----- columns 001 072
COMMAND==> SCROLL==> CSR
***** ***** top of data *****
=bnds> < >
000010 RESET #JOBNAME(A8)
000020 RESET #FD(N3) #FL(A8) #FF(N3)
000030 RESET #TD(N3) #TL(A8) #TF(N3)
000040 COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050 SET CONTROL 'WL60C6B005/010F'
000060 INPUT 'ENTER PARAMETERS FOR LIBRARY COPY:'
000070 / 'FROM: DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080 / 'TO : DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090 // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000130 /*JOBPARM LINES=2000
000140 //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000150 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000160 // 'EJ=OFF,IM=D,ID='';',MAINPR=1,INTENS=1')
000170 //STEPLIB DD DISP=SHR,DSN=OPS.SYSF.V5.ADALOAD
000180 // DD DISP=SHR,DSN=OPS.SYSF.PROD.LOAD
000190 //DDCARD DD *
000200 ADARUN DA=9,DE=3380,SVC=249
000210 //CMPRINT DD SYSOUT=X
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Split End Suspe Rfind Rchan Up Down Swap Left Right Curso

```

With the LABEL main command, you can label the current line (line currently at the top of edit area), or a block of lines. For example, the command:

```
LABEL .X
```

labels the current line **.X**. To delineate a block of lines, you can now scroll to the next required delimiting line and issue a LABEL **.Y** command. The new current line is labelled **.Y**. You can also perform the labelling function with line commands. Simply enter **.X** and **.Y** directly in the line command fields of the required lines. You can use any string to label lines, for example **.START** and **.END**.

For examples of the use of labelled lines, see the sections Find a Specific Character String and Change a Specific Character String.

Define a Mask Line

You can define data that is automatically placed in a line added through a line insertion operation (for example, using the line command **I** or **W**). Such a line is referred to as a **mask line** and is defined by typing the line command MASK in the line command field, pressing **Enter**, and entering the required data in the new line. This mask line is indicated by the message **=mask** in the line number field when you press **Enter**.

Note: You can only have one mask line during an edit session. If you define a new mask line, any existing mask line definition is automatically updated with the new value.

When you have defined a mask line, you must activate the mask feature using the command:

```
MASK ON
```

The defined mask line now appears in all lines added through a line insert operation (but remember that any line added by a line insert operation is deleted if no data or blank is added). This feature is useful if you often type the same or similar line.

To deactivate the mask feature, issue the main command:

```
MASK OFF
```

The defined mask line remains in force for the duration of the edit session. This means that you can reactivate the mask line with the MASK main command.

Display Boundary, Tab and Column Positions

You can display the positions of your boundaries (set with a BNDS main command) and tabs (set by a TABS main command), as well as the edit area column positions on any line by using the appropriate line command as listed below:

Line Command	Explanation
BNDS	Displays the boundary positions on this line.
COLS	Displays the column positions on this line.
TABS	Displays the tab positions on this line.

Exclude/Include Lines from Display

You can exclude specific lines from the display using the EXCLUDE main command. For example, the command:

```
EXC 'ABC' .X .Y ALL
```

excludes all lines with the string **ABC** within the block labelled **.X** and **.Y** from display. An unqualified EXCLUDE command excludes the current line. Each excluded line or block of lines is replaced by a line of dashes and a message informing you how many lines are excluded.

To recall excluded lines to display, use the INCLUDE main command. For example, the command:

```
IN C'abc' ALL
```

includes all excluded lines containing the string **abc** exactly as entered here. An unqualified INCLUDE command recalls the first line in the excluded block.

The EXCLUDE and INCLUDE main commands can be issued with the same string and search operands as described for the FIND command, except that the ALL search direction operand means **exclude/include all lines with the given string**.

Lines can also be excluded or recalled to display using any of the line commands listed below.

Line Command	Explanation
X	Exclude this line from display.
X15	Exclude the next 15 lines from display.
XX	Marks the first line of a block of data to be excluded from display. A second XX line command is required to delineate the block. The exclusion is performed after the second XX is entered.
F	Recall this line to display or recall the first line of the excluded block to display.
F15	Recall the first 15 lines of the excluded block to display.
L15	Recall the last 15 lines of the excluded block to display.

You can issue the main command XSWAP to exchange excluded lines with displayed lines.

Find a Specific Character String

To locate a specific character string, you can use the FIND main command with operands defining the string, the area to be searched and the direction of search. The cursor is placed on the first character of the string. If the line containing the string was excluded from display, it is now included in the display.

The following sections describe the possible command operands.

String Definition Operand

The string operand defines the character string to be located. You can specify any of the following:

Operand	Explanation
*	Find string specified in previous FIND command.
'abc'	Find string abc regardless of whether the string is upper case or lower case.
C'Abc'	Find string exactly as entered here.
P'a(char)c'	Find string whose first character is a and third character is c . (char) stands for a special character acting as a wildcard character with the following meaning: = - any character \$ - alphabetic character # - numeric character \$ - special character * - non-blank character - - non-numeric character < - lower-case character > - upper-case character
T'abc'	Find string abc regardless of whether the string is upper case or lower case.
X'D4A8'	Find string that corresponds to hexadecimal D4A8.

String-Matching Operand

The string-matching operand specifies whether any special occurrence of the string is to be located. The following options are possible:

Operand	Explanation
CHARS	No restrictions (any occurrence of the string).
PREFIX	Only those occurrences which are the prefix of a word.
SUFFIX	Only those occurrences which are the suffix of a word.
WORD	Only those occurrences which form a word.

Default is CHARS.

Direction Operand

The direction operand specifies the direction of the search operation. The following options are possible:

Operand	Explanation
ALL	Any occurrence of the string (search all directions).
FIRST	First occurrence of the string.
LAST	Last occurrence of the string.
NEXT	Next occurrence of the string starting from the cursor position.
PREV	Previous occurrence of the string.

Default is NEXT.

Line-Type Operand

This line-type operand specifies whether excluded or included lines only are to be searched. The following options are possible:

Operand	Explanation
X	Search excluded lines only.
NX	Search non-excluded lines only.

If this operand is omitted, the Editor searches all data for the given string, included and excluded lines. If the string is found in an excluded line, it is returned to display.

Block Operand

If you have labelled lines or a block of lines, you can use the block operand to restrict the search area for the FIND command.

Two examples of the block operand follow:

Operand	Explanation
.X	Search from line labelled .X to end of data.
.X.Y	Search from line labelled .X to line labelled .Y .

where **X** and **Y** can be any alphabetic character or four-character string.

Columns Operand

The column operand allows you to restrict the search for the given string between certain columns. Below are two examples of the columns operand.

Operand	Explanation
20	Locate given string starting in column 20 (the first character of the string must be in column 20).
20 40	Locate given string anywhere between columns 20 to 40.

Examples of FIND Command

Command	Explanation
F C'HILITE' X PREV	Find the previous occurrence of the string HILITE exactly as entered here; search excluded lines only.
F P'RCV#' .X.Z 20 30	Find the string starting RCV with a numeric fourth character within the block .X.Z and between columns 20 to 30.
F X'6C' SUFFIX NX	Find the character corresponding to the hexadecimal 6C in non-excluded lines only. The character must end a word.

If single quotation marks are part of the string to be found, you must use a different separator in the FIND command, for example double quotation marks:

FIND C""string"".

You can repeat a previous FIND command with the **RFIND** main command.

Insert/Delete Lines

The following line commands are available to insert and delete lines on your Editor screen.

Line Command	Explanation
D	Deletes this line.
D15	Deletes the next 15 lines.
DD	Marks the first line of a block to be deleted. A second DD line command is required to delineate the block. The deletion is performed after the second DD is 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 .
I	Inserts one line after this one. 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. If you have defined a mask line and the MASK setting is ON , the mask line is inserted when you issue this command.
I15	Inserts 15 lines after this one. You can 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).
TE	Switches the Editor to text enter mode . This means that beginning with this line the Editor screen is blank (without line numbers) and you can enter data. When you press Enter , any remaining blank lines are deleted, the line numbers are re-displayed and the text is reformatted within the set margins and with the specified justification. See also the following section POWER Command .
W	Opens a window of one line. No new line is inserted if you enter data in the window line and press Enter .
W15	Opens a window of 15 lines. When you press Enter , all unused lines are deleted.

POWER Command

You can also use the POWER main command to switch to **text enter mode**.

When you issue the POWER main command, you are presented with a blank Editor screen (without line numbers) and you can enter data. When you press **ENTER**, any remaining blank lines are deleted, the line numbers are redisplayed and the text is reformatted within the set margins and with the specified justification.

DELETE Command

Line deletion can also be performed with the DELETE main command.

For example, the command:

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

deletes all lines containing the string **Abc** exactly as entered here between columns 10 to 30 within the block delineated by the labels **.X** and **.Y**.

You can specify all operands described for the FIND command above, except that the ALL direction operand specifies deletion of all lines with the given string. An unqualified DELETE command deletes the current line.

Justify Data

A number of main commands and line commands are available to rearrange lines or blocks of data on your screen, depending on the setting of your horizontal boundaries (BNDS main command); see the section Define Horizontal and Vertical Boundaries.

The JLEFT and JRIGHT main commands justify the specified data with the left and right boundaries respectively. For example, the sequence:

```
BNDS 16 80;JLEFT 14 17
```

justifies the data between columns 16 to 80 in lines 14 to 17 with column 16.

The figure below illustrates this example:

```
S*>>EDIT-NAT:NSPF101(JOB1JCL)-Program->Report-Free-29K----- columns 001 072
COMMAND===> BNDS 16 80;JLEFT 14 17                                SCROLL===> CSR
***** ***** top of data *****
=cols>  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
000010  RESET #JOBNAME(A8)
000020  RESET #FD(N3) #FL(A8) #FF(N3)
000030  RESET #TD(N3) #TL(A8) #TF(N3)
000040  COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050  SET CONTROL 'WL60C6B005/010F'
000060  INPUT  'ENTER PARAMETERS FOR LIBRARY COPY:'
000070  /      'FROM:  DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080  /      'TO   :  DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090  // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100  /*JOBPARM LINES=2000
000110  //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120  // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130  //      'EJ=OFF,IM=D,ID='';',MAINPR=1,INTENS=1')
000140  //STEPLIB DD          DISP=SHR,DSN=SPF.SYSF.V5.ADALOAD
000150  //          DD          DISP=SHR,DSN=SPF.SYSF.V101.LOAD
000160  //          DD          DISP=SHR,DSN=SPF.SYSF.PROD.INST  * OPS INSTALL
000170  //          DD          DISP=SHR,DSN=SPF.SYSF.SOURCE    * OPS DOCUMENTS
000180  //DDCARD DD          *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Split End  Suspe Rfind Rchan Up      Down Swap Left Right Curso
```

The following screen shows the result of the command displayed in the command line of the first screen:

```
S*>>EDIT-NAT:NSPF101(JOB1JCL)-Program->Report-Free-29K--- file has been ordered
COMMAND==> BNDS 16 80;JLEFT 14 17 SCROLL==> CSR
***** ***** top of data *****
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6----+----
000010 RESET #JOBNAME(A8)
000020 RESET #FD(N3) #FL(A8) #FF(N3)
000030 RESET #TD(N3) #TL(A8) #TF(N3)
000040 COMPRESS *INIT-USER 'SM' INTO #JOBNAME LEAVING NO SPACE
000050 SET CONTROL 'WL60C6B005/010F'
000060 INPUT 'ENTER PARAMETERS FOR LIBRARY COPY:'
000070 / 'FROM: DBID:' #FD 'FNR:' #FF 'LIB:' #FL
000080 / 'TO : DBID:' #TD 'FNR:' #TF 'LIB:' #TL
000090 // #JOBNAME JOB JWO,MSGCLASS=X,CLASS=G,TIME=1400
000100 /*JOBPARM LINES=2000
000110 //COPY EXEC PGM=NATBAT21,REGION=2000K,TIME=60,
000120 // PARM=('DBID=9,FNR=33,FNAT=(,15),FSIZE=19',
000130 // 'EJ=OFF,IM=D,ID='';',MAINPR=1,INTENS=1')
000140 //STEPLIB DD DISP=SHR,DSN=SPF.SYSF.V5.ADALOAD
000150 // DD DISP=SHR,DSN=SPF.SYSF.V101.LOAD
000160 // DD DISP=SHR,DSN=SPF.SYSF.PROD.INST * OPS INSTALL
000170 // DD DISP=SHR,DSN=SPF.SYSF.SOURCE * OPS DOCUMENTS
000180 //DDCARD DD *
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Split End Suspe Rfind Rchan Up Down Swap Left Right Curso
```

The sequence:

```
BNDS 10;JRIGHT 15
```

justifies the data on the right of Column 10 in lines 15 to the end of the screen with the last column of your Editor screen (Column 88 of your terminal screen).

Alternatively, you can justify lines or blocks of data using any of the line commands listed in the following table:

Line Command	Explanation
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 boundaries (set with the BNDS main command) to be justified to the left. A second LJJ line command is required to delineate the block. Justification is performed after the second LJJ is issued.
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 boundaries (set with the BNDS main command) to be justified to the right. A second RJJ line command is required to delineate the block. Justification is performed after the second RJJ is issued.

You can also justify data to the left boundary, or order data between left and right boundary in conjunction with the JUSTIFY command.

For example, the command sequence:

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

enables justification to the left boundary. Mark a block of data with two TOO line commands (explained below) to reformat the data between Columns 10 and 60, justified to Column 10.

Locate a Specific Line

If you wish to display a specific line at the top of your Editor screen (i.e. make it the current line), use the LOCATE main command with a parameter describing the line you wish to become the current line.

Examples:

- The main command **L 32** makes Line 32 the current line;
- The main command **32** makes Line 32 the current line;
- The main command **L .X** makes the line labelled **.X** the current line;
- The main command **L 'ABC'** makes the first line that starts with the string **ABC** the current line (useful when browsing sorted data such as directory lists).

Differences between the LOCATE and FIND Commands

Please note the following differences between the LOCATE and FIND commands:

- If you issue the LOCATE command with a character string (**L 'ABC'**), the string is only found if it starts in column 1. The FIND command searches the whole data area;
- With the LOCATE command, it is assumed that the data to be searched is sorted in ascending alphabetical order;
- When a line is located with the LOCATE command, the cursor is placed in the prefix area; with the FIND command, the cursor is placed on the found string and the line is not necessarily made the current line.

Order Data Between Specified Boundaries

You can change the indentation of specified lines using the ORDER main command together with a boundary setting. For example, the command sequence:

```
BNDS 3;ORDER 5 20
```

moves lines 5 to 20 right to start in Column 3.

Note:

If the end of any ordered line traverses the right boundary, it is automatically split.

To re-justify the shifted data to the left, use a JLEFT command.

You can also change the indentation of lines or of a block of data using certain line commands. Here, too, if the end of any line traverses the right boundary, it is automatically split.

Line Command	Explanation
TF	Orders data from the line on which it was entered to the end of the paragraph or to the next blank line with the right boundary. This line command can be entered with a numerical value specifying the right boundary, e.g. the line command TF50 orders data with Column 50.
TO	Marks one line to be ordered.
TOO	Marks the first line of a block of data to be ordered. Requires a second TOO line command to delineate the block. Ordering is performed after the second TOO is issued.

Data can be ordered within set boundaries and justified to the left boundary, right boundary or both using the JUSTIFY command. For example, the command:

```
BNDS 6 60;JUSTIFY BOTH
```

activates justification to Columns 5 and 60. To perform the ordering, mark a block of data with two TOO line commands.

The Editor also provides a line command with which you can split a single line into two. Type the line command *S* in the command field of the line you wish to split, move the cursor to the position where the split is to occur and press **Enter**.

Scroll Data on the Editor Screen

PF Keys

Commands for scrolling data are often assigned to the following PF keys:

- **PF7** (main command UP) to scroll toward top of data.
- **PF8** (main command DOWN) to scroll toward bottom of data.
- **PF10** (main command LEFT) to scroll data to the left
- **PF11** (main command RIGHT) to scroll to the right.

Settings for SCROLL Field

In the SCROLL field at the top right of your screen, you can enter scroll settings. These settings are used to set the scroll amount for the PF keys, above, and some are also used with the scrolling main commands on the following page.

Possible settings for the SCROLL field are:

Scroll Setting	Explanation
<number>	Scroll up or down a specified number of lines. Scroll right or left a specified number of columns.
CSR (default)	Scroll down to cursor position if cursor is on a line of text. Cursor line becomes first line of text. When scrolling up, cursor line becomes last line of text. Scroll a page length, if cursor is in COMMAND line. Scroll right or left to cursor position.
DATA	Scroll a page length minus one line. When scrolling down, the bottom line becomes the top line. When scrolling up, the top line becomes the bottom line. When scrolling right, the last column becomes the first column. When scrolling left, the first column becomes the last column.
HALF	Scroll half a page in any direction.
LINE	Scroll up to beginning of line or down to end of line.
MAX	Scroll to top or bottom of data. Scroll to extreme right or left of data.
PAGE	Scroll a page length in any direction.
PARA	Scroll up or down to first character of next paragraph.
SENT	Scroll up to first character of current sentence or down to first character of following sentence. When scrolling up, if cursor is on first character of sentence, scroll to first character of previous sentence.
WORD	Scroll up to first character of next word or down to first character of following word.

Main Commands for Scrolling

Apart from the LOCATE main command which scrolls data to a specified line, several main commands are available for vertical and horizontal scrolling.

The following table shows all possible scrolling commands and their meaning:

Main Command	Explanation
BOTTOM <i>or</i> ++	Scrolls to the end of the object being edited.
TOP <i>or</i> - -	Scrolls to the beginning of the object being edited.
DOWN	Scrolls forward by the amount specified in the SCROLL field.
DOWN <i>n</i>	Scrolls forward by <i>n</i> lines.
+ <i>n</i>	Scrolls forward by <i>n</i> lines.
UP	Scrolls backwards by the amount specified in the SCROLL field.
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 in the SCROLL field.
LEFT <i>n</i>	Scrolls to the left by <i>n</i> columns.
RIGHT	Scrolls to the right by the amount specified in the SCROLL field.
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.

Sort Lines in Alphabetical Order

You can sort data lines in ascending or descending alphabetical order according to sorting criteria. For example, the command:

```
SORT 10 15
```

sorts all lines in the member in ascending order according to the characters beginning in Column 10 and ending in Column 15.

To sort only a block of lines, for example, label the lines where the block is to start and end with **.X** and **.Y** respectively. The command:

```
SORT .X .Y D
```

sorts all lines in the block marked by **.X** and **.Y** in descending order.

To sort a block of lines according to the characters beginning in Column 5 and ending in Column 20, for example, label the lines where the block is to start and end with **.X** and **.Y** respectively. The command:

```
SORT 5 20 .X .Y
```

sorts all lines in the block marked by **.X** and **.Y** in ascending order according to the characters beginning in Column 5 and ending in Column 20.

Store Data / Leave the Editor

You can store data and/or leave the Editor using any of the following main commands:

Main Command	Explanation
CANCEL	Leaves the Editor. Any changes made during this edit session do not take effect.
END	If AUTOSAVE is ON, stores data including any changes, and leaves the Editor. If AUTOSAVE is OFF, the END main command acts as a CANCEL command if no data was modified. If changes were made, a message asks you to issue a SAVE or CANCEL command.
SAVE	Stores the data, including any changes. Editing session continues.

Use the Physical/Logical Tabulator

TABS main command

When you issue the TABS ON main command, the **standard tab positions** set in your Editor profile are turned **on** and **tabs on std** appears in your profile. Issue the TABS OFF main command to turn tabulation **off** again.

Setting Standard Tab Positions

To turn tabulation **on** and set the **standard tab positions** for your profile to Columns 10, 20, 30, 40 and 50, for example, issue the main command **TABS 10 20 30 40 50**.

Setting the Logical Tab Character

To turn tabulation **on** and set the **logical tab character** to %, for example, issue the main command **TABS %**.

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

Setting Justification Parameters

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

Parameter	Explanation
DECIMAL	Orders data with the decimal point in the data at the tab position.
LEFT	Orders data to the left of the tab position.
RIGHT	Orders data to the right of the tab position.

For further information, see the Examples 1, 2 and 3 below.

To display the current logical tab character and shift parameter (excluding tab positions), issue the PROFILE main command.

TABS Line Command

When you issue the TABS line command in any line, the current tab positions set in your Editor Profile are displayed in that line and marked with asterisks * if no logical tab character has been set. This command does not turn tabulation on.

For example, issue the TABS line command to display the positions set with the main command **TABS 10 20 30 40 50**.

This displays the current tab positions as follows:

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

Setting Multiple Logical Tab Characters and Mixed Justification Parameters

To tabulate data in a specific column and with a specific shift, multiple logical tab characters and mixed justification parameters are possible.

To set the multiple logical tab characters, issue the TABS line command and overwrite each asterisk * with a special character. Any data typed in preceded by any of these logical tab characters are tabulated in the corresponding column.

To set the mixed justification parameters, type **L** (Left), **R** (Right) or **D** (Decimal) to the right of each logical tab character for **left**, **right** or **decimal** ordering.

For further information, see the Examples 4 and 5 below.

Examples:

Below are some examples of tabulation (the percentage sign % is assumed to be the tabulation character; the COLS line command has been issued to display column position):

Example 1: Tab Positions

The command:

```
TABS 10 20 40 LEFT
```

activates logical tabs with Tabulation Columns 20, 30 and 40 with left justification. After you press **Enter**, the input text line:

```
000010 %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:

```
000010 %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:

```
000010 %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 **TABS** command):

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

After you press **Enter**, the input text line:

```
000010 %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 Logical Tab Characters

Overtype 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:

```
000010 =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 Logical Tab Character:

Issue the command:

```
TABS ' '
```

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

```
000010 this is a blank tabulation
```

is displayed as follows:

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

Software AG Editor Main Commands

This section covers the following topics:

- General Information
- Main Commands Command Syntax
- Common Command Options
- Main Command Descriptions

General Information

The Software AG Editor provides two types of commands for editing:

- **Main commands** (described below).
- **Line commands** (see the section Software AG Editor Line Commands).

This section gives a short description of each standard **main command** and a complete overview of the command syntax.

Main commands are issued from the command line `COMMAND===>` in the second screen line.

The main commands available to you depend on the application you are using.

Note:

Depending on the configuration of your installation, main 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.

Main Commands Command Syntax

<u>C</u>R SOR	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.
[]	Optional command operands are enclosed within square brackets. If a choice is offered, one or more operands can be specified.
{ }	Braces indicate that a choice of operands must be made. Only one of the operands enclosed within braces can be supplied.
[{ }]	A combination of square brackets and braces indicates that the enclosed operands are optional, but if you wish to specify one, only one can be selected.
...4	Terms preceding the ellipsis may be repeated. A numeric constant after an ellipsis limits the number of times the term may be repeated.
&&	The Editor provides a command redisplay feature: if you precede a command with two ampersands (&&), it remains displayed in the command line and is executed every time you press Enter until you delete the command or overwrite it.

Common Command Options

There are some options which are available with several main 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 the 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 data columns; the line numbers preceding the data 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 data you are editing.

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

Displayed or Nondisplayed 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 not being displayed as specified by the EXCLUDE command. An excluded line remains excluded from display if a main 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.

Main Command Descriptions

ADVANCE

ADVANCE [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. The default setting is ADVANCE ON and can be changed by editing your profile; see Setting Your Editor Profile.

AORDER

AORDER [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; see Setting Your Editor Profile.

AUTOREN

AUTOREN [ON OFF]

For PDS members and sequential data sets only. This command is used to specify whether the Editor automatically activates the RENUMBER function.

If an unqualified AUTOREN command is issued, it is interpreted as an AUTOREN ON command. The base setting can be changed by editing your profile; see Setting Your Editor Profile.

AUTOSAVE

```
AUTOSAVE { ON }
ASAVE { OFF }
```

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

If an unqualified AUTOSAVE command is issued, it is interpreted as an AUTOSAVE ON command. Default setting is AUTOSAVE ON and can be changed by editing your profile; see Setting Your Editor Profile.

BNDS

```
BNDS [ n m ]
      [ n ]
```

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

These boundaries apply to the main 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 { ON }
      { OFF }
      { PGM }
```

This command is used to switch upper-case translation on and off.

ON	The data is translated to upper case.
OFF	The data is not translated; that is, it remains as entered.
PGM	The data 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. The default is CAPS ON. Edit your profile to change this; see Setting Your Editor Profile.

CENTER



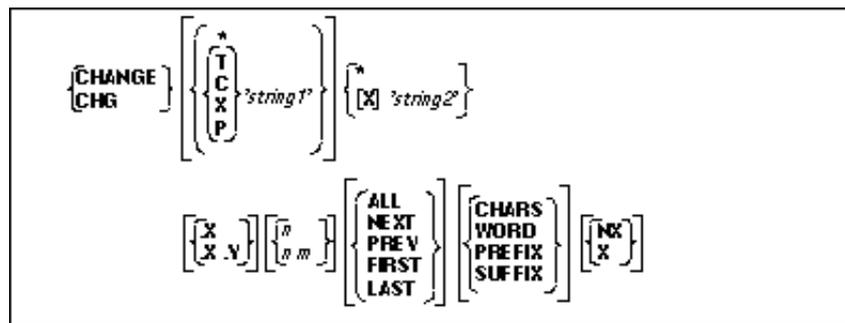
This command is used to center data.

ALL	Centers the data of all lines.
n	Centers the data from line n to the last line.
$n\ m$	Centers the data from line n to line m .

The CENTER command applies only within the horizontal boundaries as set with the main 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>'	Replace <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>'	Replace <i>string1</i> only if it occurs exactly as specified.
X'<i>string1</i>'	Replace 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>'	Replace <i>string1</i> which includes the following wildcard characters: = any character \$ alphabetical character # numeric character \$ special character ^ non-blank character - non-numeric character < lower-case character > upper-case character
*	Use the search 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 data 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.

CLEAR

```
CLEAR
```

This command clears the edit area of content.

Warning:

This command may result in loss of data. Any changes made since the last save of the object in the edit area are lost. To prevent data loss, issue the SAVE command prior to the CLEAR command.

COLS

```
COLS [ ON ]
```

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 command COLS.

CURSOR

```
CURSOR
```

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

CWINDOW

```
CWINDOW [ /n ]
```

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

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

DELETE

```
DELETE [ { *
          T
          C
          X
          P
        } 'string' ] [ [X]
                    [XY] ] [ [n]
                          [nm] ] [ ALL
                                NEXT
                                PREV
                                FIRST
                                LAST ] [ CHARS
                                         WORD
                                         PREFIX
                                         SUFFIX ] [ [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 'string'	Delete lines that contain the <i>string</i> irrespective of whether it is in 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 character > upper-case character
*	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, Dn and DD.

DOWN

```
DOWN [n]
```

This command is used to scroll downwards in the data.

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.

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 ON. The default setting is EMPTY OFF (no suppression) and can be changed by editing your profile; see Setting Your Editor Profile.

END

Stores the data including all changes and leaves the Editor.

The command format is:

```
END
```

If AUTOSAVE is set to OFF and you have changed data, the Editor asks you to issue the SAVE or CANCEL command.

ESCAPE

ESCAPE $\left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right]$ [*character*]

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

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. Default is ESCAPE OFF. Can be changed by editing profile; see Setting Your Editor Profile.

EXCLUDE

$\left\{ \begin{array}{l} \text{EXCLUDE} \\ \text{X} \end{array} \right\}$ $\left[\begin{array}{l} * \\ \text{T} \\ \text{C} \\ \text{X} \\ \text{P} \end{array} \right]$ '*string*' $\left[\begin{array}{l} \text{X} \\ \text{XY} \end{array} \right]$ $\left[\begin{array}{l} \text{N} \\ \text{NM} \end{array} \right]$ $\left[\begin{array}{l} \text{ALL} \\ \text{NEXT} \\ \text{PREV} \\ \text{FIRST} \\ \text{LAST} \end{array} \right]$ $\left[\begin{array}{l} \text{CHARS} \\ \text{WORD} \\ \text{PREFIX} \\ \text{SUFFIX} \end{array} \right]$

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'<i>string</i>'	Exclude lines that contain the <i>string</i> irrespective of whether it is in lower case or upper case. This is the default.
'<i>string</i>'	Same as T'<i>string</i>' .
C'<i>string</i>'	Exclude lines that contain the <i>string</i> exactly as specified.
X'<i>string</i>'	Exclude lines that contain the string which corresponds to the specified hexadecimal character <i>string</i> .
P'<i>string</i>'	Exclude 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 character > upper-case character
*	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:

EX 10

This command excludes line 10 from display.

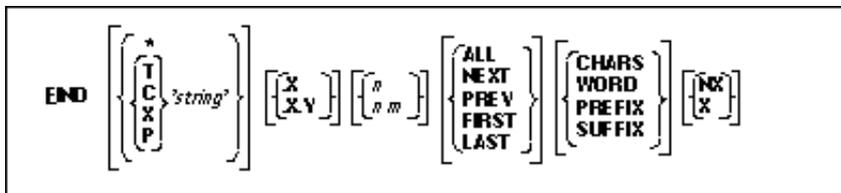
Example 2:

EX C 'NAME' .X ALL PREFIX

This command excludes from display all lines which contain "NAME" (in upper case as entered here) as a prefix to a word, starting from the line labelled with ".X".

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

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 whether it is in 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 character > upper-case character
*	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.

HEX

```
HEX [ON  
OFF]
```

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

The default setting is HEX ON and can be changed by editing your profile; see Setting Your Editor Profile.

INCLUDE

```
INCLUDE [ *  
I  
C  
X  
P ] 'string' [ X  
X.Y ] [ n  
m ] [ ALL  
NEXT  
PREV  
FIRST  
LAST ] [ CHARS  
WORD  
PREFIX  
SUFFIX ]
```

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.

Example:

```
IN C'NAME' .X ALL PREFIX
```

This command recalls all excluded lines with NAME as a prefix to a word exactly as entered here, starting from the line labelled .X.

JLEFT

```
JLEFT { ALL  
n  
n m }
```

This command is used to align data left-justified.

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

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

Example:

```
BNDS 10;JLEFT 15 20
```

The data 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 main command JRIGHT.

JRIGHT



This command is used to align data right-justified.

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

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

Example 1:

BNDS 4 40;JRIGHT 6 18

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

Example 2:

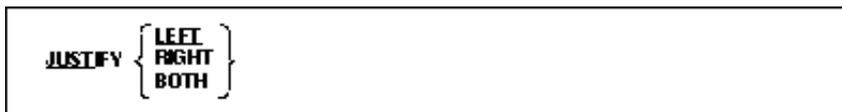
BNDS 10;JRIGHT 15

The data to the right of column 10 in line 15 is right-aligned to the rightmost column of your editing screen.

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

See also the main command JLEFT.

JUSTIFY



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

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

LEFT	The data is aligned to the left boundary.
RIGHT	The data is aligned to the right boundary.
BOTH	The data 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 data between columns 10 and 60 in the marked line(s) are 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 top of the edit area) with the specified *label*.

The *label* may be a string of 1 to 4 alphabetical 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"; see the section Software AG Editor Line Commands.

LC

```
LC [ [ [ * ] [ T ] [ C ] [ X ] [ P ] ] string ] [ [ X ] [ Y ] ] [ [ n ] [ m ] ] [ [ ALL ] [ NEXT ] [ PREV ] [ FIRST ] [ LAST ] ] [ [ CHARS ] [ WORD ] [ PREFIX ] [ SUFFIX ] ] [ [ 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 whether it is in 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 character > upper-case character
*	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

```
LEFT [ n LEFT ]
```

This command scrolls the data to the left.

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

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

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. This setting can be changed by editing your profile; see Setting Your Editor Profile.

LOCATE



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 data current.
<i>n</i>	Makes line <i>n</i> current.
<i>.label</i>	Makes the line labelled <i>.label</i> current.
' <i>string</i> '	Makes the first line that starts with this string current. (The string is only found if it starts in Column 1.)

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.

The LOCATE command differs from the FIND 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 edit area; with the FIND command, the cursor is placed on the string searched and the line is not necessarily placed at the top of the edit area.

LOG



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. Edit your profile to change the setting; see Setting Your Editor Profile.

Important: You must ensure the LOG is activated before starting to edit.

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. 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. The default setting is MASK OFF and can be changed by editing your profile; see Setting Your Editor Profile.

For further information, see Define a 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 [l  
/n]
```

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

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

NULLS

```
NULLS [ON  
OFF]
```

This command is used to determine if the data 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.

The default setting is NULLS ON and can be changed by editing your profile; see Setting Your Editor Profile.

ORDER

ORDER { ALL <i>n</i> <i>n m</i> }

This command is used to join data 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 main 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 data lines, you can also use the line commands TF, TO and TOO.

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.

PROFILE

PROFILE

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).

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.

The default setting is PROTECT ON and can be changed by editing your profile; see Setting Your Editor Profile.

RCHANGE

```
RCHANGE
```

This command repeats the last CHANGE command.

RECOVERY

```
RECOVERY { ON
           OFF } [n]
```

This command is used to activate or deactivate the recovery feature for the current edit session. You can also specify the number of updates to be performed before a checkpoint save is performed.

Using parameter *n*, you specify the number of updated lines after which a checkpoint save is performed.

The default setting is RECOVERY ON and can be changed by editing your profile; see Setting Your Editor Profile.

RENUMBER

```
RENUMBER { ON
          OFF
          n1 n2 n3 }
```

For PDS members and sequential data sets only. Specifies renumbering of the lines in the edit area according to the parameters.

ON	Activates renumbering.
OFF	Deactivates renumbering.
<i>n1</i>	Increment of numbering (default is in your edit profile).
<i>n2</i>	Starting column for the new line number (default: 73).
<i>n3</i>	End column for the new line number (default: 80).

To deactivate line renumbering, see the UNREN 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 [ nRIGHT ]
```

This command scrolls data to the right.

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

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

SAVE

Saves the content of the current Editor session. Syntax and other functionalities depend on the application you are using.

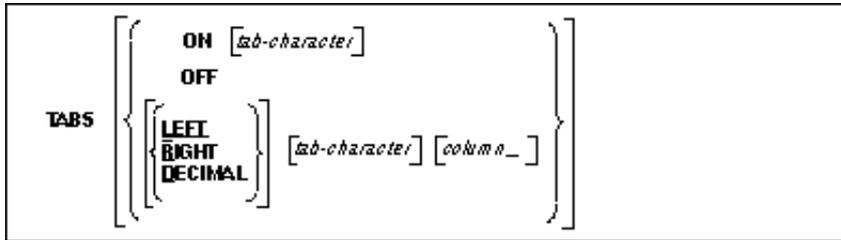
SORT

```
SORT [ n m ] [ XXY ] [ AD ]
```

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).

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 main command PROFILE.

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

The default setting is TABS OFF blank and can be changed by editing your profile; see Setting Your Editor Profile.

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 are 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:

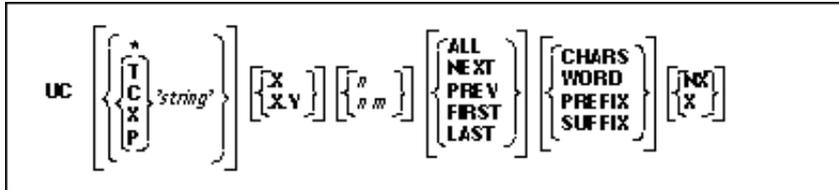
```
=cols> ----+----1----+----2----+----3----+----4----+----5----+----6
          this      is      a      blank  tabulation
```

TOP

```
TOP
```

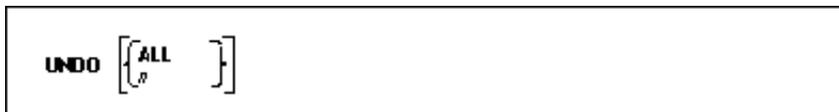
This command is used to scroll to the beginning of the object being edited.

UC



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



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.
n	The last <i>n</i> modifications are backed out.

UNREN



Deactivates the renumbering of lines.

n	Specifies the starting column of the line numbers (default: 73).
m	Specifies the end column of the line numbers (default: 80).

To activate line renumbering, see the RENUMBER command.

UP



This command scrolls upwards in the data.

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 l1 l2 { n
               n m }
```

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 *l1* and *l2* are required.

<i>l1 l2</i>	Defines a window starting at Column 1 of Line <i>l1</i> and ending in the last column of Line <i>l2</i> .
<i>l1 l2 n</i>	Defines a window starting at Column <i>n</i> of Line <i>l1</i> and ending at the last column of Line <i>l2</i> .
<i>l1 l2 n m</i>	Defines a window starting at Column <i>n</i> of Line <i>l1</i> and ending at Column <i>m</i> of Line <i>l2</i> .

Note that all data in the data area within the specified points become part of the window. For an example, see the section Copy a Window with Data.

XSWAP

```
XSWAP
```

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

Software AG Editor Line Commands

This section covers the following topics:

- General Information
 - Entering a Line Command
 - Overview of Line Commands
-

General Information

The Software AG Editor provides two types of commands for editing:

- Main commands (see the section Software AG Editor Main Commands)
- Line commands

This section gives a short description of each standard **line command**.

Entering a Line Command

A line command always applies to the line in which you enter it (or to a block of lines marked by multiple line commands).

- You can enter a line command on any data line by typing over the prefix information on the left of your edit screen.
- You can also enter a line command in the first column of the object itself if you precede it with the Editor escape character. The escape character is generally a period (.) or comma (,).
- Line commands can also be entered in the 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.

Note:

Depending on the configuration of your installation, main 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.

Overview of Line Commands

Line Command	Explanation
)	Moves this line right by one column.
) <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 one column.
(<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 one column.
>	Moves data in this line right by one column.
> <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.

Line Command	Explanation
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 after 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 before 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.
Cn	Copies the next <i>n</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.

Dn	Deletes the next <i>n</i> lines.
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 <i>.X</i> .
DY	Deletes the line labelled <i>.Y</i> .
DX-Y	Deletes the block of lines from the line labelled <i>.X</i> to the line labelled <i>.Y</i> .
F	Includes the first excluded line.
F<i>n</i>	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 main command EMPTY). You can also fill an inserted line with a predefined content (see the main 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).
J	Joins next line with this one. Identical to TJ line command.
Ln	Includes the last <i>n</i> excluded lines.
LC	Changes this line to lower case.
LC<i>n</i>	Changes the following <i>n</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.
M<i>n</i>	Moves the next <i>n</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.
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 main command MASK).
MX	Moves the line labelled <i>.X</i> . Inserts it after this line.
MY	Moves the line labelled <i>.Y</i> . Inserts it after this line.
MX-Y	Moves the block of lines from the line labelled <i>.X</i> to the line labelled <i>.Y</i> . Inserts it after this line.
N	Modifications made in this line do not take effect when Enter is pressed.
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.
O<i>n</i>	Marks the following <i>n</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, Mn, MM) or copy (C, Cn, 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.
TFn	Joins this line with the following lines until the next blank line, ignoring data that is to the right of column <i>n</i> .
TI	Inverts sequence of all characters in the current line and within the set boundaries.
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 (identical to J line 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 following <i>n</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, overtype 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	Marks this line .X
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 label may be any string of 1 to 4 alphabetical characters. See also the main command LABEL.
.Y	Marks this line .Y