Burroughs

User's Guide

A Series Editor

(Relative to the Mark 3.6 System Software Release)

Distribution Code SE

Priced Item
Printed in U.S.A.

1169976

Burroughs cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the Class specified as "2" (System Software), the Type specified as "1" (F.T.R.), and the Product specified as the seven-digit form number of the manual (for example, "1169976"). The FCF should be sent to the following address:

Burroughs Corporation
PA&S/Orange County
19 Morgan
Irvine, CA 92718

CONTENTS

1 **INTRODUCTION** 1

STRUCTURE OF THIS DOCUMENT 2

RELATED DOCUMENTS. 3

EDITOR CAPABILITIES. 4

ENTERING THE EDITOR. 7

THE EDITOR DISPLAY 10

USE OF THE TERMINAL. 16

2 **GENERAL INFORMATION.** 19

BASIC SYNTAX 19

GROUP AND DESTINATION SPECIFICATIONS 26

FILEKINDS AND RECORD FORMATS 29

PATCH MODE 32

RECOVERY FILES 34

EXTERNAL FILES 35

PENDING COMMANDS 36

3 **EDITOR COMMANDS.** 39

DISPLAY-CHANGING COMMANDS. 41

Offer-line Operation 42

Insert-line Operation. 43

Move-Window Command. 45

GO Command 47

EDITING COMMANDS 53

Edit-line Operation. 54

Split-line Operation 55

JOIN Command 57

DELETE Command 58

MOVE and COPY Commands 59

SHIFT command. 61

PARAGRAPH Command. 63

CENTER Command 65

NUMBER Command 66

RESEQUENCE Command 68

CHANGE Command 70

INSERT Command 71

MERGE Command. 73

RENEW Command. 75

RESTORING COMMANDS 77

UNDO Command 78

REFRESH Command. 79

STATE-CHANGING COMMANDS. 81

OPTIONS Command. 82

FIND Command 92

LABEL Command. 94

CANCEL Command 95

MARK Command 97

COLUMN Command 99

TERMINAL Command 100

EXITING COMMANDS 103
END Command. 104
FORGET Command 105
RECESS Command 106
CROSS-REFERENCING COMMANDS 107
Xref Options 109
REFERENCE Command. 111
DECLARATION and FORWARD Commands 115
ALL Command. 116
RANGE Command. 117
INTERSECTION and UNION Commands. 118
RECENT Command 120
WHERE Command. 121
SPECIAL-PURPOSE COMMANDS 123
TEACH Command. 124
LIST Command 126
PRINT Command. 127
LOAD Command 131
SAVE Command 133
WHAT Command 134
SEQCHECK Command 135

4 CONTROL KEY SYNONYMS FOR EDITOR COMMANDS 137

5 INSTALLING THE EDITOR. 139

UNDERSTANDING RAILROAD DIAGRAMS 141

GLOSSARY. 151

INDEX 157

1 INTRODUCTION

The Editor is a utility program designed to help you interactively create and modify program and data files. It is run on a TD830, MT983, MT987, ET1100, or SR100 terminal through, and in conjunction with, CANDE. Although some features of the Editor are similar to some features of CANDE, editing a workfile with the Editor is entirely different from editing with CANDE.

CANDE is primarily line-oriented, each record being referenced by a sequence number indicating its position in the file. The Editor is screen-oriented, allowing you to view records in context and to "point" to a location to be edited. The terminal's screen is a "window" on your workfile and can be moved forward and backward in the file by using simple commands. In most cases, sequence numbers are not needed and can be ignored.

While the Editor can be used to great advantage on any program, text, or data file in basic CANDE format, it is especially valuable for modifying large program symbolics, for three reasons:

1. The Editor supports "patching" by displaying your workfile (the patch) in its merged context while writing only changed records into the workfile.

2. The Editor can load interactive cross-reference files for your use during the session.

3. The Editor supports editing of very large files (up to 1,048,575 lines).

The Editor has an associated "textbook" that describes the commands and facilities available through the program. This information is accessed through the Editor TEACH command. If you are not familiar with the program and its TEACH mechanism, it is recommended that you enter "]TEACH TEACH" for a description of the TEACH command. Other basic commands that you might consider reading about are GO, UNDO, the edit-line operation, and the insert-line operation. The section entitled "Structure of This Document" may lead you to other information of interest to a new Editor user.

STRUCTURE OF THIS DOCUMENT

Sections

1 INTRODUCTION

Orients new users to the Editor. Topics discussed include the Editor's capabilities and commands (an overview), the syntax notation used, entering the Editor from CANDE, the format of the screen in edit-mode, and the important aspects of the terminal for using the Editor.

2 GENERAL INFORMATION

Information fundamental to the operation of the Editor and background information about different commands are described.

3 EDITOR COMMANDS

The syntax and semantics of each Editor command are described. The commands are grouped by general function, as follows: DISPLAY-CHANGING COMMANDS, EDITING COMMANDS, RESTORING COMMANDS, STATE-CHANGING COMMANDS, EXITING COMMANDS, CROSS-REFERENCING COMMANDS, and SPECIAL-PURPOSE COMMANDS. These general functions are described briefly at the beginning of the section.

4 CONTROL KEY SYNONYMS FOR EDITOR COMMANDS

Control key synonyms for Editor commands are described.

5 INSTALLING THE EDITOR

The installation of the Editor and its files is descibed.

In addition, an explanation of railroad diagrams and a glossary appear at the end of this manual.

RELATED DOCUMENTS

The following document contains related information:

Document	Form No.
I/O Subsystem Reference Manual	1169984
System Software Utilities Reference Manual	1170024

EDITOR CAPABILITIES

Note that when this user's guide refers to a key such as the XMT key or the SPCFY key, it is referring to that key on your terminal that functions as the XMT key or SPCFY key, respectively. Some terminals may not label the keys as XMT and SPCFY. Refer to the manual for your terminal to identify the keys that perform the same function as the described keys.

This section is an overview of the commands recognized by the Editor and the capabilities they provide. If you are running the Editor and are in teach mode, you can display further details on any of these commands by moving the cursor to where the command name appears in the text and pressing the Specify (SPCFY) key.

When you first enter the Editor, it will display the first page (23 lines) of your workfile. Several commands are provided for moving the display window within the file. The move-window command allows you to specify an amount (a number of lines or pages) that the window is to be moved. The GO command moves the Editor's window either to a specific location (such as a sequence number) or the next occurrence of some condition. Commands used in conjunction with the GO command to define locations or other criteria for moving the Editor's window include the FIND command, which establishes a find target for the GO FIND command; the REFERENCE command, which establishes a cross-reference (xref) for the GO REF command; and the LABEL command, which associates an alphanumeric label with a particular line for the GO <label> command.

The Editor normally offers one line of the file for editing by placing the line between forms characters and putting the terminal in forms mode. To cause the Editor to offer a different line for editing, use the offer-line operation (move the cursor to the line you wish to edit and press SPCFY). To insert a new line between two existing lines, use the insert-line operation (press SPCFY on the first character of the offered line).

Several commands enable you to edit the contents of the workfile. The simplest is the edit-line operation, in which you type the new text on the offered line and then press Transmit (XMT). To split a line into two lines (perhaps to make room for additional text on the first line), use the split-line operation. To join two lines, use the JOIN command.

The DELETE command deletes lines. The MOVE command moves lines from one part of the file to another. The COPY command is a form of MOVE that leaves a copy of the records in both places. The INSERT command inserts records from another file. The MERGE command merges records from another file according to sequence numbers and by acting on certain compiler control options (such as VOIDT).

The Editor includes a SHIFT command for shifting the text on the line, a PARAGRAPH command for formatting words in paragraph form, and a CENTER command for centering the text on a line.

Sequence numbers can be assigned automatically with the NUMBER command, or by user specification with the RESEQUENCE command.

The Editor provides several options that allow tailoring of its operation to a particular user or application. The OPTIONS command allows many of these options to be displayed, changed, and preserved from one editing session to the next. The LINES option controls the number of lines offered for editing (by default, only one line is offered at a time). Margin specifications for the PARAGRAPH and CENTER commands can be established, as well as specifications for the SHIFT command. The MAXIMUM and MINIMUM options establish parameters for the automatic sequence-numbering facility. The MARK command establishes a value to be written into the mark field of all changed records and records specifically designated by the CHANGE command.

The following commands restore the workfile or the display in case the results of a command (or an external event) are undesirable: UNDO undoes the action of the previous command; REFRESH rewrites the display; RENEW restores records from the original workfile.

The Editor is capable of loading interactive cross-reference (xref) files generated for ALGOL, COBOL74, DCALGOL, FORTRAN, FORTRAN77, and NEWP programs. The information derived from these files can be displayed using the REFERENCE, ALL, INTERSECTION, and UNION commands. The information is used by the DECLARATION, FORWARD, and GO REFERENCE commands to move the window to the declaration, forward declaration, or next reference of the target identifier. These files are also used for providing environment information for the DISPLAYENVIRONMENT option, the WHERE command, and the GO ENVIRONMENT command. Xreffiles are loaded automatically when they can be found using the workfile or symbol file name. To load xreffiles under a different name or for a different file, use the LOAD command. The OPTIONS command (described above) is also used to establish default specifications for certain xref displays.

Other Editor commands include LIST, which lists an external file, and PRINT, which prints the workfile or the teach textbook.

The Editor will acknowledge the receipt of each transmission by returning the display to a "ready" condition. No input, once acknowledged, will ever be lost by the Editor, except in the unlikely event of an irrecoverable peripheral failure. The Editor's recovery file mechanism provides full recovery.

ENTERING THE EDITOR

The Editor is a system utility that is executed through CANDE using the CANDE "UTILITY" command. Before you can run the Editor, you must have a workfile. Either GET or MAKE a workfile in CANDE, then enter "U EDITOR" to execute the Editor (assuming that the code file for the Editor program is named OBJECT/EDITOR). CANDE will pass your workfile to the Editor for use during the Editor session. When you end the Editor session, the Editor passes your workfile back to CANDE and it becomes your CANDE workfile. Any changes you made during the Editor session are reflected in the file the Editor passes back to CANDE. In order to preserve these changes permanently, you must do a CANDE "SAVE".

The following example illustrates the steps required to execute the Editor and enter TEACH mode; it also explains the details of the program's initialization sequence and possible error situations:

1. Create a CANDE workfile using either the MAKE or GET CANDE command. For example,

   ```
   MAKE A/B ALGOL
   ```

2. Execute the Editor with the CANDE "UTILITY" command

   ```
   U EDITOR
   ```

3. The first time the Editor is run under a usercode from a particular CANDE station, it will ask the following question:

   ```
   Is this terminal a TD830, an MT983, an MT987, an ET1100,
   or an SR100?
   ```

   The appropriate responses are TD830, MT983, MT987, ET1100, SR100, and NO. TD is allowed as an abbreviation for TD830, MT for MT983, MT987 for MT987, ET for ET1100, and SR for SR100. As long as an EDITOR/OPTIONS file is present, the Editor will not repeat this question for a particular usercode and CANDE station. Responding NO indicates that the terminal is not one of these terminals. Since only these terminal types are supported, the Editor will terminate without doing any further work if NO is entered.

4. During initialization, the Editor will display a screen that identifies the Editor and describes the current workfile. Once initialization is complete, the Editor displays the first 23 lines of the workfile and offers a line for editing. For an empty workfile, the screen will be blank except for the following characters displayed on lines 1 and 2 of the screen ("|" indicates the cursor position and "}" and "{" represent the left and right forms-characters, respectively. On the screen, the forms-characters look like small right- and left-pointing triangles):

```
[   ~    ]
 >>>> }|                                                  {
```

   The characters "[~   ]" on the first line indicate that the first line on the page currently has no sequence number.

   To enter TEACH mode and display information about the TEACH command, enter "]TEACH TEACH" and press XMT. The leading right-bracket ("]") character tells the Editor that this input is a command and not new text for the file. For an empty workfile, the screen will look like this just before XMT is pressed:

```
[   ~    ]
 >>>> }]TEACH TEACH|                                      {
```

5. The screen displays information from the textbook describing how to use the TEACH command to learn about the Editor. Press SPCFY when the cursor is in the form at the upper left-hand corner of the screen to display subsequent pages. In TEACH Mode, the Teach menu that appears on the left-hand side of the screen can be used to return to Edit mode, return to the TEACH index page, and go to the next or previous topic.

You must have a CANDE workfile before the Editor can be executed. If there is no current workfile, the Editor terminates with the message:

```
    You do not have a work file.
```

The workfile must also be named. If the workfile is not named, the Editor displays the following message before terminating:

```
    Your work file is not named.
```

Introduction

The CANDE "UTILITY" command allows a file name to be passed to the Editor as a parameter in the form "U EDITOR <filename>". If a file name is passed to the Editor, the Editor executes in "patch mode", which is described in the General Information section (refer to "Patch Mode").

See also

Patch Mode. 32

THE EDITOR DISPLAY

The Editor uses several page formats to display various kinds of information. The most frequently used form is the edit-mode display, which is used whenever the workfile is being displayed for editing. The edit-mode screen format is described in this section; other screen formats are described under the commands that display them.

The following diagram represents a sample edit-mode screen, reduced for illustration from the normal 25 lines of 80 characters each to only 15 lines of 72 characters each. The example shown below is relevant to a "default" FILEKIND file, such as ALGOLSYMBOL or SEQDATA. The displays for other FILEKINDs will vary a little.

The Editor displays many kinds of information on the edit-mode screen, in various locations. For purposes of illustration, the diagram below includes most of these informational items, resulting in an example that is somewhat more complicated than the simplest possible screen, but representative of what might reasonably appear. The "|" character represents the cursor, "}" represents the left forms-character, and "{" represents the right forms-character.

```
[10042220] in SPEX of ED                             10042420<33.182.177
042220*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
042240*%%                                                            %%
042260*%%        GLOBALS FOR MARGINS AND SHIFT SPECIFICATIONS        %%
042280*%%                                                            %%
042300*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
042320
REALS #  REAL MARGINS                            % MARGIN INFO
042360#      ,MARGINSAVE
   ~  *      ,SEED                               % FOR RANDOM
042400       ;
042420}| BOOLEAN BMARGINS=MARGINS                                      {
042430         ,BMARGINSAVE=MARGINSAVE
042440>        ;
[10042440] MOVE FL  Find='%%%%':changes  Xref=MARGINS
```

Three types of lines are displayed in edit mode: the command line (the top line), the status line (the bottom line), and file-display lines (the middle lines). The format for each type of line is described in the following paragraphs.

The Command Line

The top line of the edit-mode display is called the "command line." Text entered on this line is automatically interpreted as an Editor command, even if the "]" command character is omitted. When the command line is not the offered line, it displays other information.

```
[10042220] in SPEX of ED                              10042420<33.182.177
```

The line above is typical of what is usually displayed while editing the file. On the left side, "[10042220]" is the sequence number of the first record on the page or, if the first record is not numbered, of the next numbered line toward the beginning of the file. If there is no numbered line, the characters "[~   ]" are used.

On the right side, three separate items are displayed, all relating to the offered line:

> "10042420" is the sequence number of the offered line;
>
> "<" is the character that would be displayed in the flag field of that line if it weren't pre-empted by the left forms-character;
>
> "33.182.177" is the current contents of the mark field of the offered record.

The environment of the offered line is "in SPEX of ED", displayed as a result of the DISPLAYENVIRONMENT option; if DISPLAYENVIRONMENT is OFF, this information does not appear. If the COLHEADING option is ON, the environment and offered line information is pre-empted by the column ruler. (Refer to the OPTIONS command for more information on DISPLAYENVIRONMENT and COLHEADING.)

The command line is also used to display information relating to the last command performed. For example, following a SHIFT command that changed three records, the message "3 records were SHIFTED" is displayed. Following a FIND command specifying "$" as the find target, the message "The FIND target will be '$'" is displayed. These messages pre-empt all other displays in the center of the command line, but persist only until the next command.

Error and warning messages also pre-empt other command line displays. If an error is detected, the error message is displayed in reverse video on the right side of the command line. The command in error is displayed in the center of the command line, with the misunderstood portion of the command appearing in bright video. If you move the cursor to the command line and press SPCFY to offer the line, the Editor will continue to display the command in error so that you can correct it and retransmit it, if you desire.

File-Display Lines

Aside from the command and status lines, all lines on the screen display records from the file (or, if there are fewer than 23 records in the file, as many records as there are).

```
042220*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
042240*%%                                                             %%
042260*%%       GLOBALS FOR MARGINS AND SHIFT SPECIFICATIONS          %%
042280*%%                                                             %%
042300*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
042320
REALS #  REAL MARGINS                                  % MARGIN INFO
042360#      ,MARGINSAVE
   ~  *      ,SEED                                     % FOR RANDOM
042400       ;
042420}| BOOLEAN BMARGINS=MARGINS                                      {
042430           ,BMARGINSAVE=MARGINSAVE
042440>          ;
```

The first six columns of each line display either the sequence number of the record or other identification information. The displays that may appear in this area are listed below, in order of highest to lowest priority:

<label id> If a label has been assigned to the record (via the LABEL command), the first six characters of the <label id> are displayed ("REALS", in the example above).

+/- <integer> If the RELATIVE option is ON, relative line numbers are displayed (refer to the OPTIONS command).

<sequence number> If the line is numbered, the last six digits of its sequence number are displayed.

>>>> If the line is an insert record (that is, a record that is offered by the insert-line operation but is not yet in the file), the characters " >>> " are displayed.

~ If the line is unnumbered, a tilde (~) is displayed.

The seventh column is called the "flag field." This column contains a character indicating the status of that line, listed below in order of highest to lowest priority:

} (Left forms-character) This is an offered line.

@ The destination for a pending MOVE or COPY command is immediately after this line.

< The record has been marked as the first line of a group for a pending command (refer to <group> and to Pending Commands).

> The record has been marked as the last line of a group for a pending command.

= The record has been marked as the first and last line of a group for a pending command.

* The text field of the record has been changed or the record is new.

The record has been renumbered.

blank The record is unchanged.

In the example above, line 10042420 is the offered line (}), lines 10042340 and 10042360 have been renumbered (#), and lines 10042220 through 10042300 and the unnumbered line after 10042360 have been changed or inserted (*). Line 10042420 (the offered line) has been marked as the first line of the group for the pending MOVE command (described below), and line 10042440 has been marked as the last line of the group.

The major portion of the display for each record is the text field, which comprises the next 72 columns (for an ALGOL file). The last column is reserved for the right forms-character, which is displayed only for an offered line.

The Status Line

The bottom line of the display is the "status line". This line can never be offered, as the cursor cannot be moved to it. On MT983 terminals, the leftmost 16 columns are reserved by the terminal for its own use. On all supported terminals, the rightmost seven columns are reserved for the terminal page-number display. The Editor displays session status information in the remaining portion of the status line.

```
[10042440] MOVE FL  Find='%%%%':changes  Xref=MARGINS
```

The left side of the status line contains the sequence number of the last line on the page or the next numbered line toward the end of the file ("[10042440]" in the example). Immediately following the sequence number is the current MARK value, if any (not shown in the example). If a command is pending, the command name and already-specified information is displayed next; in the example, "MOVE FL" indicates that there is a pending MOVE command for which the first (F) and last (L) lines have been specified (refer to Pending Commands). Following any pending command information, the Editor displays the current FIND target and options, if any ("Find='%%%%':changes" in the example) and the current XREF target, if any ("Xref=MARGINS" in the example).

If the COLFOOTING option is ON, all of these displays are pre-empted by the column ruler (refer to the OPTIONS command).

See also

OPTIONS Command 82
Pending Commands. 36

USE OF THE TERMINAL

The Editor is designed to run on TD830, MT983, MT987, ET1100, and SR100 terminals. These are screen devices, with 80 characters per line and 24 lines per page. The Editor uses many of the specialized features of these terminals, such as forms mode, bright video, and reverse video, as well as some of the less visible features. In general, it is not necessary for the Editor user to understand all aspects of the terminal (for information about configuring terminals for use with the Editor, refer to "Installing the Editor"). However, you do need to understand some function keys and options.

These terminals have at least five function keys, labeled CTRL, SPCFY, LOCAL, RCV, and XMT. The LOCAL and RCV keys are not used for the Editor. The CTRL key may be used to perform certain terminal operations (such as "CTRL T" to put the terminal in upper/lowercase mode and "CTRL Y" to put the terminal in uppercase-only mode); it may also be used to enter the command synonyms listed in the "Control Key Synonyms for Editor Commands" section.

XMT

The Transmit (XMT) key transmits data characters entered in the unprotected form of the Editor display to the system; by default, the "form" (the portion of the display that appears between one or more pairs of left and right forms-characters) is one line. The specific functioning of XMT depends on the terminal configuration with respect to full-line-transmit mode. When the terminal is in full-line-transmit mode, pressing XMT causes the contents of the entire form to be sent to the system. When the terminal is not in full-line-transmit mode, pressing XMT causes all characters between the HOME position (the upper, leftmost character position of the form) and the current cursor position to be transmitted to the system; if the cursor is in the HOME position when XMT is pressed, the entire form is transmitted. The setting of the full-line-transmit option can be changed by using the TERMINAL command.

Another terminal configuration option, line-at-a-time transmission, also affects the functioning of the XMT key. Because the results obtained with the Editor run on a terminal in line-at-a-time transmission mode can be inconsistent with the intended operation of the Editor, it is recommended that the Editor be run only on terminals without this option enabled.

SPCFY

The Specify (SPCFY) key is a special transmit key that sends the cursor location to the system. No other information is transferred, so any characters that were entered before SPCFY was pressed are ignored. SPCFY is used in many commands that involve using the cursor to "point to" or select an item on the display; for example, in order to offer a line for editing, you move the cursor to the line you wish to edit and then press SPCFY to inform the Editor of the line you selected. The following list summarizes the uses of SPCFY (the use of SPCFY is described in the section on each command):

- In the <offer-line operation>, to indicate the line to be offered.
- In the <split-line operation>, to indicate where the line is to be split.
- In the <insert-line operation>, to indicate between which two lines the inserted line is to be placed and, once in insert mode, to exit insert mode.
- In the LIST, LABEL, REFERENCE, ALL, INTERSECTION, and UNION commands, to proceed to the next page of the display.
- In the TEACH command, to indicate a topic about which information is to be displayed, to indicate which menu function is desired (edit mode, index page, previous topic, or next topic), and to request the next page of the teach information.
- In the REFERENCE, ALL, INTERSECTION, and UNION commands, to indicate a sequence number to which the window is to be moved.
- In the LABEL command, to indicate a label to which the window is be moved.
- In the ALL and RECENT commands, to indicate which identifier is to become the cross-reference target.

It is not recommended that you use ETX, RS, US, GS, SCROLL, ROLL, MOVE, FORMS, ALIGN, line insert/delete, or any control or escape sequences not specifically defined for use with the Editor; these may cause undefined results. The character insert and character delete keys, the clear-to-end-of-line key, the cursor control keys, the skip/tab key, and the clear/home key are all safe to use in the Editor. The Return (RET) key is useful in multi-line mode (refer to the LINES option in the OPTIONS command); note that RET does not transmit data to the system.

If the terminal does not seem to be transmitting the correct information to the system, as a last resort you might try powering off the terminal, powering it back on, and entering "|HOME|]REFR" (the HOME key, followed by a REFRESH command).

When displaying files, the Editor translates all nongraphic characters to DEL (rubout) characters to protect the display from datacomm control characters. The characters in the file are left intact, unless a displayed line is retransmitted as an edited line, in which case the DEL characters in that line are put into the workfile.

See also

Control Key Synonyms for Editor Commands. 137
Installing the Editor 139
OPTIONS Command 82
REFRESH Command 79
TERMINAL Command. 100

2 GENERAL INFORMATION

This section contains fundamental information that pertains to the operation of the various Editor commands but that would be burdensome if repeated under each command to which the information applies. Once understood, most of this information will become second nature to you and will become of interest again only when some command gives you an unexpected result.

BASIC SYNTAX

The constructs defined in this section are used in the syntax of one or more Editor commands. The constructs are presented alphabetically, for quick reference.

<column>

An <integer> that represents a column number within the text field of the record. The valid range of values for <column> depends on the FILEKIND of the file.

See also
FILEKINDs and Record Formats. 29

<delimiter>

Any <special character> other than colon (:) or DEL.

<digit>

Any one of the decimal digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

<filename>

```
                             |<-- / --|
                             |        |
     --------------------------<name>------------------------------------|
        |                  |            |              |
        |- * --------------|            |- ON --<name>-|
        |                  |
        |- ( --<name>-- ) -|
```

    <name>

```
       |<------------------|
       |                   |
     ----/17\---<letter>-------------------------------------------------|
               |          |
               |-<digit>--|
               |          |
               |- _ ------|
               |          |
               |- - ------|
```

The Editor recognizes the basic set of A Series and B 5000/B 6000/B 7000 Series systems file names, including usercodes (in parentheses) and family names (following ON). A <filename> may contain up to 14 <name>s.

<graphic>

Any <letter>, <digit>, or <special character>.

<hex digit>

Any one of the hexadecimal digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

<hex number>

```
       |<-------------|
       |              |
    ----<hex digit>-----------------------------------------------------|
```

<id>

```
--<letter>--------------------------------------------------------|
           |                  |
           | |<-------------| |
           | |              | |
           |-----<letter>-----|
               |          |
               |-<digit>--|
               |          |
               |- _ ------|
```

Identifiers may contain letters, digits, and underscores. If the workfile is of FILEKIND COBOLSYMBOL or COBOL74SYMBOL, identifiers may also contain hyphens (-).

<integer>

```
  |<--------|
  |         |
----<digit>-------------------------------------------------------|
```

<label id>

```
--<id>------------------------------------------------------------|
```

A <label id> is an <id> that can be associated with a record in the file (refer to the LABEL Command). <label id>s are case insensitive; that is, uppercase and lowercase letters are synonymous.

See also
LABEL Command 94

<letter>

Any one of the alphabetic characters from A through Z or a through z, inclusive.

<mark>

```
                   |<----------------|
                   |                 |
 --<delimiter>---/10\-<graphic>---<delimiter>-----------------------|
```

A <mark> is a value to be placed in the mark field of certain edited records. The conditions under which the <mark> is written into the mark field are described under the MARK command. The two <delimiter>s specified must be identical, and no <graphic> used may be identical to the <delimiter> used. The <mark> value is case sensitive (that is, any <letter>s in the string of <graphic>s are preserved as they were entered with respect to upper and lowercase).

The size of the mark field depends on the FILEKIND of the file (refer to FILEKINDs and Record Formats). If the specified <mark> is too large for the mark field, the <mark> is truncated on the right-hand side. If the specified <mark> is not as large as the mark field, the field will be blank-filled on the right-hand side.

See also
FILEKINDs and Record Formats. 29
MARK Command. 97

<qualified id>

```
 --<id>-------------------------------------------------------------|
       |           | |                 |
       |- .Value -| | |<-----------| |
                     | |             | |
                     |--- OF --<id>---|
```

A <qualified id> is a name that the Editor recognizes as a program identifier from semantic information derived from cross-reference files (xreffiles) (refer to Cross-Referencing Commands). The use of <qualified id>s requires that xreffiles be loaded, either automatically or by request (refer to the LOAD command).

The <qualified id> construct is used in several Editor commands to refer to a particular program identifier. Because some languages allow the same identifier to be used for different purposes in different program environments, an <id> may require "qualification" with its environment name (or names) to distinguish it from other identifiers of the same name.

A <qualified id> that consists of simply "<id>" refers to the program identifier of that name that is valid in the environment of the offered line. If the current environment has no program identifier of that name, the Editor displays a message to that effect and uses the first identifier of that name in the xreffiles.

If the ".VALUE" syntax is used, the <id> is assumed to be the procedure identifier of a typed procedure in NEWP. The identifier being referenced is "<procedure-id>.VALUE", the value of the typed procedure.

The "OF <id>" syntax qualifies the <id>; that is, it designates a particular instance of that program identifier by specifying the environment in which it was declared. All <id>s that follow the keyword "OF" must be procedure, module, alternative module; or (for COBOL74 only) when duplicate names are involved, an id that precedes the reference id in its record hierarchy.

Successive "OF" clauses are interpreted in order of most local environment to most global environment. Thus, the <qualified id> "I OF GET_NUMBER OF SCANNER" refers to a program identifier called "I" declared within the local environment of the GET_NUMBER procedure, which itself appears within the more global environment of the SCANNER procedure. Note that "I OF SCANNER" is not synonymous with "I OF GET_NUMBER OF SCANNER".

For some FILEKINDs, <qualified id>s are case sensitive (that is, an identifier entered in lowercase is different from the same identifier entered in uppercase); refer to "FILEKINDs and Record Formats".

The Editor allows the keywords "IN" and "AT" in place of "OF" in <qualified id>s. The semantics of these constructs are identical to those described for the INTERACTIVEXREF program in the "System Software Utilities Reference Manual."

See also

Cross-Referencing Commands. 107
FILEKINDs and Record Formats. 29
LOAD Command.. 131

<range list>

```
  |<-/4\------------------------ , -----------------------|
  |                                                       |
------<qualified id>-----------------------------------------------|
     |                                                  |
     |-<sequence number>--------------------------------|
     |                   | |                            |
     |- HERE ------------| |- TO -----<sequence number>-|
                           |      | |                   |
                           |- - --| |- END -------------|
                                    |                   |
                                    |- HERE ------------|
```

The <range list> construct is used in various Editor commands to designate one or more areas of the file that are to be involved in the action of the command. For example, a <range list> can be specified in a PRINT command to designate the portion of the workfile to be printed.

If the <qualified id> form of <range list> is used, cross-reference files (xreffiles) must have been loaded previously, either automatically or by request (refer to the LOAD command). The <qualified id> must be a procedure identifier. The information in the xreffiles converts the <qualified id> to a sequence range, specifically, the range of lines that the procedure comprises (that is, the range from the "DECLARED @" to the "ENDS @" sequence numbers).

The remaining <range list> options specify a range of sequence numbers. The END option refers to the sequence number of the last record in the file. The HERE option refers to the sequence number of the offered line or, when entered on the command line, to a sequence number between the top line displayed and the line preceding it.

If more than one range specification is used (for example, two sequence ranges or a sequence range and a <qualified id>), the specified areas of the file must not overlap. If a specified <range list> applies to an external file (for example, in an INSERT command), the external file must have the same record format as the file being edited.

See also

FILEKINDs and Record Formats. 29
LOAD Command. 131

<sequence number>

An <integer> that represents a value for the sequence field of a record. The maximum number of <digit>s that a <sequence number> may comprise is determined by the FILEKIND of the workfile.

See also

FILEKINDs and Record Formats. 29

<special character>

Any one of the following characters:

!	exclamation point
#	number
%	percent
'	apostrophe
)	close (right) parenthesis
-	minus (hyphen)
~	tilde
_	underscore
@	at
[	open (left) bracket
;	semicolon
:	colon
]	close (right) bracket
,	comma
.	period
/	slash
|	split bar
"	quote
$	dollar
&	ampersand
(	open (left) parenthesis
■	DEL (rubout)
=	equal
\	backslash
^	circumflex (carat)
`	grave accent
{	open (left) brace
+	plus
*	asterisk
}	close (right) brace
<	less than
>	greater than
?	question mark

GROUP AND DESTINATION SPECIFICATIONS

<group>

```
    ----<integer>-----------------------------------------------------|
      |           |                             |
      |           |-<relative line>-------------|
      |                                         |
      | |<------------------------------------| |
      | |                                     | |
      |-----/1\- First -------------------------|
          |             | |                 |
          |-/1\- Last --| |-<relative line>-|
```

<relative line>

```
    ---- + ---<integer>-----------------------------------------------|
      |     |
      |- - -|
```

A <group> specification designates the group of contiguous lines upon which the command being entered is to operate. The group can be defined either by specifying a number of lines (using the "<integer> ..." form) or by designating the first and last lines of the group (using the "First/Last ..." form).

In the "number of lines" form, the <integer> is interpreted as the number of lines in the group.

If the <relative line> specification is omitted, the first line of the group is the offered line. If the <relative line> specification is present, it designates which line, relative to the offered line, is the first line of the group. The <integer> specifies an offset from the offered line. The "+" and "-" indicate after the offered line (toward the end of the file) and before the offered line (toward the beginning of the file), respectively. For example, a <relative line> specification of "-3" indicates that the first line of the group is the line three lines before the offered line. The RELATIVE option, which displays a relative line number for each displayed line, is helpful for specifying relative line numbers (refer to the OPTIONS Command).

The following example of a DELETE command illustrates the use of a <group> of "4 + 2", specifying the group of four lines beginning with the second line down from the offered line and including the three lines in the file after that line:

```
BEFORE                    AFTER
--------------------      ----------------------------------
[00000100]                [00000100]  4 records were DELETED
000100}]delete 4+2        000100 one hundred
000200 two hundred        000200 two hundred
000300 three hundred      000700}seven hundred
000400 four hundred
000500 five hundred
000600 six hundred
000700 seven hundred
```

The First and Last options are used to designate the first and last lines of the group. Both can be specified in the same command or they can be specified one at a time. If they are specified independently, the operation is "pending" during the time that only one has been entered (refer to Pending Commands).

The First option by itself indicates that the offered line is the first line of the group. The Last option by itself indicates that the offered line is the last line of the group. Thus, entering First on the offered line, then specifying on the line three lines down from the first line, and then entering Last on that line would designate a group of four lines, consisting of the original offered line and the three lines that follow it in the file.

The First and Last options can also be followed by a <relative line> specification, indicating that the first or last line is offset from the offered line. For example, "F-1" establishes the line that immediately precedes the offered line as the first line of the group; "L+2" establishes the line two lines after the offered line as the last line of the group, as shown in this example of a DELETE command:

```
BEFORE                    AFTER
--------------------      ----------------------------------

[00000100]                [00000100]  4 records were DELETED
000100 one hundred        000100 one hundred
000200 two hundred        000200 two hundred
000300 three hundred      000700}seven hundred
000400}]DEL F-1 L+2
000500 five hundred
000600 six hundred
000700 seven hundred
```

<destination>

```
 ---- Before -----------------------------------------------------------|
    |          | |                  |
    |- After --| |-<relative line>-|
```

A <destination> specification designates the location at which lines are to be inserted. A destination is required for any MOVE, COPY, or INSERT command and can be specified for a RENEW command.

The destination is indicated by specifying either BEFORE or AFTER the offered line or, if a <relative line> specification appears, the specified relative line. For example, "B" designates a destination of "immediately before the offered line". "AFTER+2" designates a destination of "immediately after the line that is two lines after the offered line".

RESTRICTIONS: If a <group> is specified using the FIRST and LAST keywords, the line marked FIRST must precede (that is, must be closer to the beginning of the file than) the line marked LAST. The destination must not be specified within the <group> being acted on. Insert lines cannot be designated as a <group> or <destination> boundary. When specifying a <group> (such as "4") or a <group> boundary (such as "FIRST"), the <group> or boundary being specified must be within the display window.

See also
    OPTIONS Command 82
    Pending Commands. 36

FILEKINDS AND RECORD FORMATS

The Editor can display and edit most types of workfiles that CANDE will allow you to MAKE or GET. The Editor establishes several of its operating parameters based on the FILEKIND of the file being edited. Also, some operations may be invalid with files of certain FILEKINDs or with files of nonmatching FILEKINDs (such as inserting or merging records from a file of one FILEKIND into a file of a different FILEKIND).

The operating parameters that the Editor bases on the FILEKIND of the file being edited include record format (text, sequence, and mark field lengths), whether or not error files can be loaded, whether to recognize and generate VOIDT or DELETE as the editing compiler control option, and whether to initialize the terminal to uppercase and lowercase mode or uppercase-only mode. The settings of these parameters for various FILEKINDs are listed below.

Editor commands that change the text, sequence, and mark fields of the workfile are described in general terms, referring only to "the mark field" or "the leftmost column of the text field." In all cases, the action of the command is determined by the location and width of these fields.

SEQDATA, TEXTDATA

Record Format:	Text Field	1-72 (72 characters)
	Sequence Field	73-80 (8 characters)
	Mark Field	81-90 (10 characters)
Errorfiles:	NO	
VOIDT or DELETE:	VOIDT for SEQDATA, DELETE for TEXTDATA	
Uppercase Only:	NO	
Other:	The default <trainid> is EBCDIC96. <qualified id>s are case sensitive.	

ALGOLSYMBOL, DCALGOLSYMBOL, FORTRANSYMBOL, NEWPSYMBOL

Record Format:	Text Field	1-72 (72 characters)
	Sequence Field	73-80 (8 characters)
	Mark Field	81-90 (10 characters)
Errorfiles:	YES, except NEWP	
VOIDT or DELETE:	VOIDT	
Uppercase Only:	YES	

COBOLSYMBOL, COBOL74SYMBOL

Record Format:	Sequence Field	1-6	(6 characters)
	Text Field	7-72	(66 characters)
	Mark Field	73-80	(8 characters)
Errorfiles:	YES		
VOIDT or DELETE:	VOIDT for COBOL, DELETE for COBOL74		
Uppercase Only:	YES		
Other:	Hyphens (-) are considered nondelimiters and may appear in <id>s.		

JOBSYMBOL

Record Format:	Text Field	1-72	(72 characters)
	Mark Field	73-82	(10 characters)
	Sequence Field	83-90	(8 characters)
Errorfiles:	NO		
VOIDT or DELETE:	VOIDT		
Uppercase Only:	YES		

Note: In CANDE, JOBSYMBOL files are assumed to have text in columns 1-80, and columns 81-82 are ignored. A CANDE JOBSYMBOL file that contains program text in columns 73-80 could be difficult to edit with the Editor.

RPGSYMBOL

Record Format:	Sequence Field	1-5	(5 characters)
	Text Field	6-75	(70 characters)
	Mark Field	76-85	(10 characters)
Errorfiles:	NO		
VOIDT or DELETE:	DELETE		
Uppercase Only:	YES		
Other:	The COLHEADING and COLFOOTING options are ON by default (refer to the OPTIONS Command).		

NDLIISYMBOL

Record Format:	Text Field	1-72	(72 characters)
	Sequence Field	73-80	(8 characters)
	Mark Field	81-90	(10 characters)
Errorfiles:	NO		
VOIDT or DELETE:	DELETE		
Uppercase Only:	YES		

PASCALSYMBOL

Record Format:	Text Field	1-72	(72 characters)
	Sequence Field	73-80	(8 characters)
	Mark Field	81-90	(10 characters)
Errorfiles:	YES		
VOIDT or DELETE:	DELETE		
Uppercase Only:	NO		
Other:	The default <trainid> is EBCDIC96.		

DATA, all others

Record Format: (assumed)*	Text Field	1-72	(72 characters)
	Sequence Field	73-80	(8 characters)
	Mark Field	81-90	(10 characters)
Errorfiles:	NO		
VOIDT or DELETE:	VOIDT		
Uppercase Only:	NO		
Other:	The default <trainid> is EBCDIC96. <qualified id>s are case sensitive.		

* The Editor assumes that no data exists in columns 73-80, and it will overwrite any data in those columns with sequence numbers.

PATCH MODE

The Editor may be run with just a source file or with both a source file and a symbol file. When the Editor is executed from CANDE with a command of the form "UTILITY EDITOR" (with no further information), the CANDE workfile is passed to the Editor as a source file and there is no symbol file. In this case, the Editor runs in "merge mode," so that the changes made to the file during the Editor session are merged into the source file when an END command is performed.

When the Editor is executed from CANDE with a command of the form "UTILITY EDITOR <filename>", the specified <filename> is passed to the Editor. The Editor interprets this <filename> as the name of a symbol file against which the file being edited (the source file) is a patch. When the Editor has both a source file and a symbol file, it is considered to be in "patch mode." If the source file and the symbol file are not of the same FILEKIND, the Editor will display the message "The FILEKINDs of the WORKSOURCE and the symbol file are different." At this point, you have the option of pressing SPCFY, in which case the session will proceed in spite of the incompatibility, or pressing XMT, in which case the session will terminate, leaving the recovery file intact.

In patch mode, the Editor displays the patch file in context; that is, the records in the patch file are displayed as they would appear after having been merged into the symbol file through SYSTEM/PATCH, a compiler, or the Editor MERGE command. When the Editor is exited, changed records are written to the new workfile in patch form (that is, with editing compiler control records, as described below).

The Editor recognizes and generates two kinds of compiler control records: "null" records ("$" followed by all blanks) and "editing" records (SET and POP VOIDT or DELETE, depending on the FILEKIND). The Editor generates these records when an END or SAVE command is entered. A null compiler control record is generated in the patch file when a single line has been deleted. A pair of editing control records is generated when a group of lines has been deleted. The generation of these compiler control records is complementary to their recognition in the MERGE command.

When the Editor applies a patch, either when initializing in patch mode or when processing a MERGE command, null and editing compiler control records are applied and then are deleted. When the workfile is written (for example, for an END operation), the appropriate null and editing records are regenerated. Thus, the exact sequence numbers at which these records appear may change. Also, adjacent null and editing records may be combined.

If a patch is loaded against a symbol file other than the file against which it was created, compiler control records may be lost or incorrectly resequenced. The Editor will detect this condition and will display the following message: "WARNING: SOME COMPILER CONTROL RECORDS IN YOUR PATCH MAY BE DISCARDED!". At this time, you can press SPCFY, which continues the session in spite of the incompatibility, or press XMT, which terminates the session, leaving the recovery file intact.

See also

END Command 104
FILEKINDs and Record Formats. 29
FORGET Command. 105
MERGE Command 73

RECOVERY FILES

During initialization of a session, the Editor creates a recovery file called "EDITOR/RECOVERY/<filename>", where <filename> is the name of the workfile, under your usercode on your primary family. All changes made to the workfile during the session are stored in this file, as is information about the state of the session. The existence of this file allows the Editor to recover from abnormal terminations (such as a Halt/Load or abnormal program termination) with minimal information loss.

To recover changes from an abnormally-terminated session, use the RECOVER, MAKE, or GET command to obtain the same CANDE workfile that you were editing during the aborted session. Then re-execute the Editor by entering "U EDITOR" (without the <filename> parameter). The Editor will detect that a file called "EDITOR/RECOVERY/<filename>" already exists and will restore the previous state by loading the information from this file.

You can also preserve the recovery file deliberately by using the RECESS and END:RECESS commands. Re-entering the Editor with a saved recovery file is much faster than re-entering the Editor after a normal END command.

Because the information in the recovery file is stored as changes to the workfile, it is important that the workfile not be changed outside the Editor while a recovery file for that workfile exists. Changing the workfile in CANDE, for example, invalidates the recovery file and results in loss of all stored changes. If the Editor detects that the workfile has been changed, it will display the message "The SYMBOL file has been updated since this RECOVERY file was created." At this point, you can press SPCFY, which continues the session in spite of the incompatibility, or press XMT, which terminates the session, leaving the recovery file intact.

See also

END Command 104
RECESS Command. 106

EXTERNAL FILES

The Editor has one file that it uses to access files for the INSERT, MERGE, and LIST commands; this file is called "the external file." If a <filename> is specified in one of these commands, the TITLE of the external file is set accordingly, and the file is accessed. A subsequent INSERT, MERGE, or LIST command can access the same file by specifying "=" instead of a <filename>. Thus, the following two commands, if entered in sequence, will access the same file:

```
    LIST (UC)TEXTFILE
    INSERT = 500-900 AFTER
```

If an attempt is made to use the "=" syntax when the external file has not yet been assigned a <filename>, the Editor will display the error message "No external file".

See also

INSERT Command. 71
LIST Command. 126
MERGE Command 73

PENDING COMMANDS

The DELETE, MOVE, COPY, SHIFT, PARAGRAPH, CENTER, NUMBER, RESEQUENCE, CHANGE, and RENEW commands all require the specification of a <group>, the group of lines on which the command is to operate. In addition, the MOVE and COPY commands require the specification of a <destination>, a location to which the source lines are to be moved or copied.

As described under the <group> construct, there are two ways of specifying a group of lines: by specifying a number of lines (as in "MOVE 3") or by designating the first and last lines of the group. Because the first and last lines can be designated in separate commands, as can the destination, there may be a period of time during which all of the information has not yet been specified. A partially-specified command is considered "pending," waiting for further input.

When a command is pending, the Editor displays the command name on the command line and the status line, followed by a series of letters indicating which pieces of information have already been supplied. "F" indicates that the FIRST line has been marked, "L" the LAST line, and "D" the destination. For all commands except MOVE and COPY, the destination is not required and the "D" is never displayed.

Only one command can be pending at a time. If a command of one type is pending, a command of another pending type cannot be entered, except for a fully-specified DELETE or RENEW command. For example, if a COPY is pending, waiting for the destination to be specified, a DELETE command of the form "DELETE 3" would be allowed, but a DELETE command of the form "DELETE FIRST" would not be. Any command that does not allow a <group> (and so cannot be left pending) is allowed when there is a pending command.

A pending command can be canceled by using the CANCEL command. For example, if the Editor is displaying "MOVE FL", indicating that the first and last lines of the group have been specified but the destination is still unspecified, a "CANCEL MOVE" command would discard the first and last specifications and remove the "MOVE FL" display.

See also

CANCEL Command. 95
CENTER Command. 65
DELETE Command. 58
MOVE and COPY Commands. 59
NUMBER Command. 66
PARAGRAPH Command 63
RENEW Command 75
RESEQUENCE Command. 68
SHIFT Command 61

3 EDITOR COMMANDS

Editor commands may be entered at any time in edit mode (refer to the TEACH command for a discussion of what may be entered in teach mode). These commands are used for moving the display window to a different part of the file, editing the text of the file, entering teach mode, displaying compiler cross-reference information, and many other tasks.

Editor commands are entered on the offered line in edit mode. Commands are distinguished from data being entered for the workfile by the "]" (right bracket) character as the first nonblank character of the input. If the offered line is the command line, the "]" is not required.

Almost all commands are case insensitive; that is, commands are identified by the sequence of characters entered, regardless of whether the characters are in uppercase, lowercase, or a mixture of both. For example, the JOIN command may be entered as "JOIN", "join", "Join", or any other upper/lowercase combination of those letters. In a few cases, command text may be case sensitive, such as in FIND targets, MARK values, and REFERENCE (xref) targets. Exceptions to the case-insensitivity rule are described under the individual commands to which the exception applies.

Most commands (and other Editor keywords) can be abbreviated by supplying one or more of the initial characters of the keyword and omitting the remainder. For example, the JOIN command can be entered as simply "J" (or "j"). The minimum number of characters that are required for the command to be recognized varies from 1 to 6, depending on the command. The syntax diagram for each command indicates the minimum required number of characters by displaying the command keyword in a mixture of uppercase and lowercase characters. Uppercase characters are required, and lowercase characters are optional. In the diagram for the JOIN command, the keyword appears as "Join", indicating that only the first character is required to identify the command. This capitalization convention is used only in syntax diagrams and on the teach file "Index Page"; in all other contexts, the capitalization of keywords is irrelevant.

If an invalid command is entered, the Editor displays the command on the command line with the item in error shown in bright video. This command can then be fixed and retransmitted if necessary.

A few frequently-used commands are entered by pressing SPCFY instead of by entering a "]" followed by a command word. The Specify operation and other special commands are described as "operations" to distinguish them from the alphabetic commands (for example, the "insert-line operation" as opposed to the "INSERT command"). The Specify operations require that the cursor be positioned in a specific location prior to pressing SPCFY; these requirements are described for each operation individually.

Some commands have control key synonyms (refer to the section "Control Key Synonyms for Editor Commands").

In the following subsections, the commands are grouped according to their respective functions:

DISPLAY-CHANGING COMMANDS	Change the edit-mode display but do not affect the workfile.
EDITING COMMANDS	Change the text, sequence, and mark fields of the records in the workfile.
RESTORING COMMANDS	Restore all or part of the workfile or display to a previous state.
STATE-CHANGING COMMANDS	Change and display various session options and variables.
EXITING COMMANDS	Terminate the Editor session.
CROSS-REFERENCING COMMANDS	Make use of interactive cross-reference files.
SPECIAL-PURPOSE COMMANDS	Provide additional, nonediting functions.

See also
Control Key Synonyms for Editor Commands. 137
TEACH Command 124
Use of the Terminal 16

DISPLAY-CHANGING COMMANDS

Display-changing commands cause the Editor to change the format or content of the information it displays on the edit-mode screen (refer to "The Editor Display" for additional details). These commands have no effect on the contents of the workfile.

COMMAND	ACTION
Offer-line	Moves the offered line within the window.
Insert-line	Offers a new line between existing lines.
Move-window	Moves the display window forward or backward.
GO	Moves the offered line to a specific location or to the next instance of some condition.

See also

The Editor Display. 10

Offer-line Operation

```
-- |SPCFY| ----------------------------------------------------------|
```

The offer-line operation selects another line that is currently displayed on the page to be the offered line. To offer a particular line for editing, move the cursor to the text field of the line you wish to edit and press SPCFY. The Editor will offer the line, and the cursor will be placed in the column where it was when SPCFY was pressed (if the cursor was to the left of the first nonblank character of the line, it will be placed on the first nonblank character). Now, you can edit the line.

NOTE

> Similar Specify operations differ only in the placement of the cursor when SPCFY is pressed. If the cursor is on the text field of an offered line, the operation will be interpreted as either an insert-line or a split-line operation.

See also

Editing Commands. 53
Insert-Line Operation 43
Split-Line Operation. 55
Use of the Terminal 16

Insert-line Operation

```
 -- |SPCFY| -----------------------------------------------------------|
```

The insert-line operation enters "insert mode," in which the Editor offers a new line for editing between two lines currently displayed on the page. One form of this operation inserts a line after any displayed line and the other form inserts a line before the offered line.

To insert a line after an existing line displayed on the page, move the cursor to the sequence number or flag field of that line and then press SPCFY. To insert a line before the offered line, position the cursor in the text field, on or to the left of the first nonblank character, and press SPCFY. In response to either of these operations, the Editor displays an unnumbered line as the inserted, offered line. Normally, the inserted line is blank; however, in duplicate mode, the line is a copy of the line that precedes it (refer to the DUPLICATE option in the OPTIONS command).

To exit insert mode, press SPCFY. If the cursor was on the offered insert line, the line following the insert line will be offered. If the cursor was on any other displayed line, that line will be offered.

NOTE

> Similar SPCFY operations differ only in the placement of the cursor at the time SPCFY is pressed. If the cursor is not in the proper location for an insert-line operation, the operation will be interpreted as either an offer-line or a split-line operation.

In the following example, the "|" character indicates the location of the cursor:

```
BEFORE                    AFTER
--------------------      --------------------

[00000100]                [00000100]
000100 one hundred        000100 one hundred
000200|two hundred        000200 two hundred
000300 three hundred       >>>> }|
000400}four hundred       000300 three hundred
                          000400 four hundred
```

In this example, the offered line was line 400; the user moved the cursor to the flag field of line 200 and pressed SPCFY. The Editor offered an insert line after line 200.

See also
Offer-Line Operation. 42
OPTIONS Command 82
Split-Line Operation. 55
Use of the Terminal 16

Move-Window Command

```
 ---- + ---<integer>--------------------------------------------------|
    |     |             |          |
    |- - -|             |- Pages -|
```

The move-window command moves the Editor's window in the file (refer to the GO command for ways to move the offered line). If "+" is specified, the window is moved forward in the file, toward the higher sequence numbers and the end of the file. If "-" is specified, the window is moved backward in the file, toward the lower sequence numbers and the beginning of the file.

]+<integer> Scrolls the window forward the number of lines specified by <integer>.

Example:]+3 moves the window 3 lines forward in your file.

]-<integer> Scrolls the window backward the number of lines specified by <integer>.

Example:]-3 moves the window 3 lines backward in your file.

If the offered line is scrolled off the page, the cursor is left on the command line.

]+<integer>p Moves the window forward the number of pages specified by <integer>. (A screen page is 23 lines.) The cursor is left on the command line.

Example:]+3p moves the window 3 pages forward in your file.

]-<integer>p Moves the window backward in your file the number of pages specified by <integer>. The cursor is left on the command line.

Example:]-3p moves the window 3 pages backward in your file.

NOTE: The "+ and -<integer>" commands are valid in teach mode. (Refer to the TEACH command.)

See also
GO Command. 47
TEACH Command 124

GO Command

```
----<go-next command>-----------------------------------------------|
   |                       |
   |-<go-location command>-|
```

The GO command is the most important command for moving the offered line from one area of the file to another. The <go-next command> moves the offered line to the next occurrence (forward or backward) of some target. The <go-location command> moves the offered line to the specified location.

Because GO is the most frequently-used Editor command, the syntax allows very concise commands, even to the point of dropping the keyword GO in many cases. Some variants of the GO command require "GO" to distinguish them from other Editor commands (for example, "GO END" versus "END").

The following paragraphs discuss each form of the GO command. To illustrate the brevity of these commands, examples shown do not include any optional keywords or letters.

<go-next command>

```
 ------------ + ----- Alterations -----------------------------------|
   |       | |     | |               |
   |- GO -| |- - -| |- Changedtext -|
                     |               |
                     |- Unnumbered --|
                     |               |
                     |- Find --------|
                     |               |
                     |- Reference ---|
                     |               |
                     |- Environment -|
                     |               |
                     |- ERRor -------|
```

The <go-next command> requires a leading "+" or "-" character to specify the direction in which to search for the next occurrence of the target: "+" specifies forward in the file, toward the higher sequence numbers and the end of the file; "-" specifies backward in the file, toward the lower sequence numbers and the beginning of the file.

+/- ALTERATIONS

"]+A" or "]-A" goes either forward or backward, offering the next line that has changed at all, relative to the original workfile. Such changes result from the application of any <editing command>, including commands that do not affect the text field, such as DELETE, SHIFT, and RESEQUENCE. If one or more lines have been deleted, the line immediately following (for "]+A") or immediately preceding (for "]-A") the deleted area is offered.

+/- CHANGEDTEXT

"]+C" or "]-C" goes either forward or backward, offering the next line with a significantly-changed text or mark field. Such changes result from the application of all <editing command>s except DELETE, MOVE, NUMBER, and RESEQUENCE.

+/- UNNUMBERED

"]+U" or "]-U" goes either forward or backward, offering the next line that is on a boundary between a numbered area and an unnumbered area.

+/- FIND

"]+F" or "]-F" goes either forward or backward to the next occurrence of the find target, leaving the cursor on the first character of the target. If the target is not found, the message "No more FIND occurrences" is displayed. This command is not valid unless a find target has been established (refer to the FIND command).

The search begins with the offered line, unless that line was offered as a result of a previous "]+F" or "]-F" command. During the searching process, the Editor periodically displays on` the command line the sequence number of the line it is searching. If any input is received during the search, the Editor will abort the GO command and will offer the previously offered line.

+/- REFERENCE

"]+R" or "]-R" goes either forward or backward to the next occurrence of the current cross-reference (xref) target, leaving the cursor on the first character of the xref target if it appears literally in the line. If a reference is not found, the message "No more references" is displayed. If the xref target appears on the offered line, that occurrence is considered the "next occurrence" unless the line was offered as a result of a previous "]+R" or "]-R" operation.

This command is not valid unless xreffiles have been loaded, either automatically or by request (refer to the LOAD command). The xref target and all other xref parameters are established through the use of various xref commands (refer to "Cross-Referencing Commands").

+/- ENVIRONMENT

"]+E" or "]-E" offers the line that is either at the end ("]+E") or at the beginning ("]-E") of the most local environment that contains the currently offered line. This command is not valid unless xreffiles have been loaded, either automatically or by request (refer to the LOAD command).

+/- ERROR

"]+ERR" or "]-ERR" goes either forward or backward to the next line flagged in the currently loaded compiler errorfile. This command is not valid unless an errorfile has been loaded, either automatically or by request (refer to the LOAD command).

When this command is entered, the Editor offers the line in error, places the cursor where the error occurs, and displays the error message on the command line.

<go-location command>

```
 -----------<sequence number>------------------------------------------|
   |      |                      |
   |- GO -|                      |
   |                             |
   |- GO --- BEGINning ----------|
          |                      |
          |- ENDing -------------|
          |                      |
          |- First --------------|
          |                      |
          |- Last ---------------|
          |                      |
          |- Destination --------|
          |                      |
          |-----------<label id>-|
          |         |            |
          |- LABel -|            |
          |                      |
          |-<rcw spec>-----------|
```

<sequence number>

"]<sequence number>" centers the requested line in the window and offers it for editing. If a line with the specified sequence number does not exist, the Editor moves the window to the area around the specified number, displays the message "No such line in file", and leaves the cursor on the command line.

BEGINNING

"]GO BEGIN" offers the first line of the file.

ENDING

"]GO END" offers the last line of the file.

FIRST

"]GO F" offers the line that was designated as the first line of the <group> for the currently pending command (refer to "Pending Commands"). This command is not valid unless there is a pending command for which FIRST has already been specified.

LAST

"]GO L" offers the line that was designated as the last line of the <group> for the currently pending command (refer to "Pending Commands"). This command is not valid unless there is a pending command for which LAST has already been specified.

DESTINATION

"]GO D" offers the line that was designated as the destination for the currently-pending command (refer to "Pending Commands"). This command is not valid unless there is a pending command for which a destination has already been specified.

<label id>

"]GO <label id>" offers the line associated with the specified <label id> (<label id>s are associated with lines by using the LABEL command). The keyword LABEL is optional, but is allowed so that <label id>s that duplicate other recognized words in the GO command can be referenced. For example, "]GO END" would reposition the window to the last page of the file, even if "END" had been defined in an earlier LABEL command as a <label id>. The command "]GO LAB END" causes the Editor to unconditionally interpret "END" as a <label id>.

<rcw spec>

"]GO <rcw spec>" offers the line associated (via lineinfo) with the specified RCW. The syntax for <rcw spec> is defined under the WHERE command. This command is not valid unless a code file has been loaded (refer to the LOAD command).

See also

Cross-Referencing Commands.	.107
Editing Commands.	53
FIND Command.	92
LABEL Command	94
LOAD Command.	.131
Pending Commands.	36
WHERE Command	.121

EDITING COMMANDS

Editing commands change the workfile. Each command in this group alters the text field, the sequence field, and/or the mark field of one or more records. The conditions under which the mark field is changed are described in the section on the MARK command. The following is a summary of the editing commands.

COMMAND	ACTION
Edit-line	Places data into the text field of a record.
Split-line	Splits the offered line into two lines.
JOIN	Connects the text of two lines.
DELETE	Deletes a group of lines.
MOVE	Moves a group of lines to a new location.
COPY	Copies a group of lines at a given location.
SHIFT	Shifts the text of a group of lines.
PARAGRAPH	Formats the text of a group of lines into a paragraph.
CENTER	Centers the text of a group of lines.
NUMBER	Invokes the Editor's automatic sequence numbering to number a group of lines.
RESEQUENCE	Assigns user-specified sequence numbers to a group of lines.
CHANGE	Marks a group of lines as changed.
INSERT	Inserts records from an external file.
MERGE	Merges records from an external file.
RENEW	Returns a group of lines to its unedited state.

Each command is detailed in the following pages.

See also

MARK Command. 97

Edit-line Operation

<edit-line operation>

> Any sequence of <graphic>s that is not entered on the command line, does not have a "]" character as the first nonblank character, and does not have the station's datacomm control character in column 1.

The edit-line operation, the most basic of the editing commands, is simply the entering of data that you wish the Editor to place in the text field of the record that is currently the offered line.

In order to be interpreted as data for the text field, the entered data must not be interpretable as an Editor command or a datacomm control record. That is, the input must not be entered on the command line, must not begin with a "]" as the first nonblank character, and must not have the station's datacomm control character ("?" by default) in column 1.

If your terminal has the full-line transmit option enabled (refer to "Use of the Terminal") or if the cursor is in the HOME position when you press XMT, the entire offered line will be transmitted to the system, and the Editor will replace the entire text field of the record with the data it received. If, instead, you transmit a partial line by pressing XMT when the cursor is not in the HOME position and the terminal does not have full-line transmit enabled, only the characters between the HOME position and the cursor will be transmitted. The Editor will replace only that part of the text field of the record with the data it received; the data to the right of the cursor position will remain as it was in the originally-displayed record (even if you had modified it on the display).

The Editor marks the edited record as changed unless the data transmitted is identical to what is currently in the text field of the offered line. In this case, the Editor does not consider the line to have been edited and will not mark the line changed or put the record in the patch file (refer to "Patch Mode").

Following the edit-line operation, the Editor will offer the next line in the file for editing. If the edited line was the last line in the file, the Editor will offer an insert line at the end of the file.

See also

Patch Mode. 32
Use of the Terminal 16

Split-line Operation

```
 -- |SPCFY| -----------------------------------------------------------|
```

The split-line operation splits the offered line into two lines. To perform a split-line operation, place the cursor in the position where you would like the line to be split and then press SPCFY. The characters to the left of the cursor remain on the offered line; the character under the cursor and all characters after it are put on a new line, which is placed in the file following the original offered line. The characters on the new line are aligned with the first nonblank character of the original line. After the operation, the cursor is left in the same position as it was before the operation.

If you press SPCFY when the cursor is on one of the leading blanks of a line, on the first nonblank character of a line, or on an all-blank line, the operation is interpreted as an insert-line operation.

NOTE

> A common mistake is to perform a split-line operation when an offer-line or insert-line operation was desired. If you split a line by mistake, type "]UNDO" to restore the line.

To split the offered line into several pieces, start at the right side of the line, as the cursor remains on the line with the left side portion of the text. If you intend to split several lines, consider using "split mode" (refer to the SPLIT option in the OPTIONS command).

In the following example, the "|" character shows the location of the cursor before and after the split-line operation:

```
    BEFORE                          AFTER
    --------------------            --------------------

    [00000100]                      [00000100]
    000100}a very long|text line    000100}a very long|
    000200 was here                    ~   * text line
                                    000200 was here
```

See also
- Insert-Line Operation 43
- Offer-Line Operation. 42
- OPTIONS Command 82
- UNDO Command. 78

JOIN Command

```
-- Join ------------------------------------------------------------|
```

The JOIN command connects the text on the offered line with the text on the line following it (the "joining line"). The first nonblank character on the joining line, and all characters after it, are connected to the last nonblank character of the offered line. Trailing blanks on the joining line are discarded.

A blank is inserted between the text of the two lines if no other delimiter is present; a delimiter is any character other than a <letter>, <digit>, or "_" (for COBOLSYMBOL and COBOL74SYMBOL files, "-" is also not a delimiter). The joining line is deleted, and the cursor is left on the offered line at the point where the two character strings were connected.

If the nonblank text on the joining line is longer than the trailing blanks on the offered line, the Editor displays the message "Won't fit", offers the joining line, and leaves the cursor on the joining line at the rightmost column that would have fit.

The PARAGRAPH command may be useful if you are doing a series of JOIN commands.

In the following example, the "|" character shows the location of the cursor following the JOIN command:

```
BEFORE                        AFTER
---------------------         ---------------------

[00000100]                    [00000100]
000100}]JOIN                  000100}abc|def
000200       def
```

See also
    PARAGRAPH Command 63

DELETE Command

```
 -- DELete --<group>---------------------------------------------------|
```

The DELETE command deletes the group of records specified by <group>. After the delete operation, the line following the deleted group is offered. If the last line of the file is deleted, the last remaining line of the file is offered.

Example

```
    BEFORE                    AFTER
    --------------------      ----------------------------------

    [00000100]                [00000100]  3 records were DELETED
    000100}]delete 3+2        000100 one hundred
    000200 two hundred        000200 two hundred
    000300 three hundred      000600}six hundred
    000400 four hundred
    000500 five hundred
    000600 six hundred
```

See also
    Group and Destination Specifications. 26

MOVE and COPY Commands

```
---- Move ---------------<destination>--------------------------------|
  |        | |         |               |
  |- Copy -| |-<group>-|               |
             |                         |
             |-<group>-----------------|
```

The MOVE and COPY commands insert records from one part of the file (the source) at another location in the file (the destination). The MOVE command then deletes the records where they originally appeared; the COPY command does not.

The <group> specifies the group of records to be moved or copied. The <destination> specifies between which two lines the moved or copied records are to be inserted.

A MOVE or COPY command that includes a <destination> but no <group> establishes the destination and leaves the command pending, waiting for a group to be specified (refer to "Pending Commands"). A command that includes a <group> with no <destination> establishes part or all of the group specifications and leaves the command pending, waiting for a <destination> to be specified. If both the <group> and the <destination> are specified, the command is performed immediately. After the MOVE or COPY is executed, the first line of the <group> is offered.

For example, "MOVE 1 B-2" causes the offered line to be inserted before the line that is two lines backward from the offered line, as illustrated below:

```
BEFORE                    AFTER
--------------------      -------------------------------

[00000100]                [00000100] One record was MOVED
000100 one hundred        000100 one hundred
000200 two hundred           ~  }four hundred
000300 three hundred      000200 two hundred
000400}]move 1 b-2        000300 three hundred
000500 five hundred       000500 five hundred
```

Copied lines are marked as "changed" for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands. Moved lines are marked as "changed" only for the "GO +/-ALTERATIONS" command.

See also
GO Command. 47
Group and Destination Specifications. 26
Pending Commands. 36

SHIFT command

```
-- SHift --<group>--------------------------------------------------|
                   |                     |
                   |-------<shift specs>-|
                   |     |
                   |- : -|
```

<shift specs>

```
------ RIght -----<integer>-------------------------------------------|
  | |         | |
  | |- > -----| |
  |             |
  |--- LEft ----|
    |        |
    |- < ----|
```

The SHIFT command shifts the group of lines specified by <group>.

The shift operation requires a direction and an amount to shift the text field of each line. The direction may be specified as RIGHT (synonym ">") or LEFT (synonym "<"). The amount is specified as an <integer>, which must be greater than or equal to 0 and less than the width of the text field of the record (refer to "FILEKINDs and Record Formats"). The <shift specs> construct can be included in the command to override the current session shift specifications (refer to the OPTIONS command). The shift specifications appearing in a SHIFT command apply only to that command.

Shifted records are marked as "changed" for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands.

If a line is encountered that cannot be shifted the amount specified without losing nonblank characters, the line is offered for editing, and the SHIFT command is left pending with LAST specified, the direction and amount specified, and FIRST unspecified (refer to "Pending Commands"). The command is left in this pending state even if the <group> was originally given in "number of lines" form, which normally could not be left pending. The command is left pending to allow you to edit the "unshiftable" line and then restart the shift on the following line just by entering "]SHIFT FIRST" (LAST is still pending as the last line of the originally-specified <group>).

Example

```
    BEFORE                         AFTER
    -----------------------        ----------------------------------

    [00000100]                     [00000100]  3 records were SHIFTED
    000100}]shift 3 right 3        000100}    one hundred
    000200 two hundred             000200*    two hundred
    000300 three hundred           000300*    three hundred
```

See also

FILEKINDs and Record Formats. 29
Group and Destination Specifications. 26
OPTIONS Command 82
Pending Commands. 36

PARAGRAPH Command

```
-- PARagraph ---<group>-----------------------------------------------|
              |         | |                         |
              |- * -----| |       |<--------------| |
                          |       |    |      |   | |
                          |       |    |<- , -|   | |
                          |       |               | |
                          |---------<margin spec>---|
                          |     |
                          |- : -|
```

<margin spec>

```
----/1\--- LEFTMargin ----<column>-----------------------------------|
  |     |               |         |
  |     |- LM ----------|         |
  |                     |         |
  |-/1\--- RIGHTMargin -|         |
  |     |               |         |
  |     |- RM ----------|         |
  |                               |
  |-/1\- INDent --------<integer>-|
  |             |     |           |
  |             |- + -|           |
  |             |     |           |
  |             |- - -|           |
  |                               |
  |-/1\--- RIGHTJustify ----------|
        |               | |       |
        |- RJ ----------| |- ON --|
                          |       |
                          |- OFF -|
```

The PARAGRAPH command reformats the text fields of a group of records into a "paragraph" by placing the maximum number of "words" on each line (any string of contiguous nonblank characters is considered a "word").

The <group> construct specifies the group of records to be paragraphed. If "*" is specified, the group is considered to be all records between a preceding and a following blank record. If the offered line is itself blank or is an insert line, the "*" group is considered to be the records between the offered line and the next previous blank record.

Margin specifications determine the format of the paragraph and control the left and right margins (LM and RM, respectively), the amount the first line is indented (INDENT), and whether or not the lines are right-justified (RJ).

The <column> specified for the left and right margins must be within the defined text field range for the FILEKIND of the workfile (for example, <column> must be between 1 and 72 for an ALGOL file). The left margin <column> must be less than the right margin <column>.

The <integer> given for the INDENT specification may be preceded by a "+", indicating indent to the right, or a "-", indicating indent to the left ("exdent"); if the <integer> is not preceded by a "+" or a "-", indent to the right is assumed. When indenting to the right, the specified <integer> must be small enough such that the column in which the first line begins is not beyond the right margin (that is, the relation "LM+INDENT < RM" must be true). When indenting to the left, the specified <integer> must be small enough such that the column in which the first line begins is not less than the first column of the text field.

One or more <margin spec>s can be included in the PARAGRAPH command to override the corresponding session margin specification(s) (refer to the OPTIONS command). Margin specifications used in a PARAGRAPH command apply only to that command.

The records to be paragraphed are deleted and then replaced by the reformatted records. These new records are marked as "changed" for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands.

See also

FILEKINDs and Record Formats. 29
GO Command. 47
Group and Destination Specifications. 26
OPTIONS Command 82

CENTER Command

```
-- CENter --<group>-----------------------------------------------------|
                     |                         |
                     |       |<--------------| |
                     |       |    |      |   | |
                     |       |    |<- , -|   | |
                     |       |               | |
                     |---------<margin spec>---|
                     |     |
                     |- : -|
```

The CENTER command centers the nonblank portion of the text field of each line in the group specified by <group>.

Centering is controlled by the right and left margin specifications (refer to the PARAGRAPH Command). One or more <margin spec>s can be included in the CENTER command to override the corresponding session margin specification(s) (refer to the OPTIONS command). Margin specifications used in a CENTER command apply only to that command.

Example

```
BEFORE                    AFTER
--------------------      -----------------------------------

[00000100]                [00000100]  2 records were CENTERED
000100 SPECIFICATION      000100*        SPECIFICATION
000200   TITLE            000200*            TITLE
000300}]center 2-2        000300}
```

See also
Group and Destination Specifications. 26
OPTIONS Command 82
PARAGRAPH Command 63

NUMBER Command

```
 -- NUMber ---<group>---------------------------------------------------|
              |          |
              |- Region -|
              |          |
              |- ALL ----|
```

The NUMBER command automatically assigns values to the sequence fields of a specified range of records. The algorithm used by the Editor to choose an appropriate sequence base and increment is described under the MAXIMUM and MINIMUM options, as is the warning given when the records cannot be numbered without extending the range (refer to the OPTIONS command). If you would like to explicitly specify the sequence base and increment, use the RESEQUENCE command instead.

The records to be numbered may be specified in any one of the following ways:

<group>

If a <group> is given, the records specified by the <group> syntax are numbered. If only a part of the file is to be numbered, the group of records to be numbered must be bounded by records that already have sequence numbers.

REGION

If REGION is specified, the group of unnumbered records that includes or immediately precedes the offered line is numbered. If the offered line does not appear in or immediately after an unnumbered region, the error message "No unnumbered text here" is displayed.

ALL

If ALL is specified, all unnumbered groups in the file are numbered.

Records that have been renumbered are displayed with a "#" in the flag field, if no other change has been made to the record. These records are considered "changed" for the "GO +/-ALTERATIONS" command, but not for the "GO +/-CHANGEDTEXT" command (unless the text field of the records has been changed by some other command).

See also

- GO Command. 47
- OPTIONS Command 82
- RESEQUENCE Command. 68

RESEQUENCE Command

```
 -- RESequence ---<group>---------------------------------------------->
                 |          |
                 |- Region -|
                 |          |
                 |- ALL ----|

 >---------------------------------------------------------------------|
   |                                            |
   |- : ---------------------- + <integer> -|
        |                    |
        |-<sequence number>-|
```

The RESEQUENCE command assigns or changes the sequence numbers of the specified group of records. The RESEQUENCE command requests that you provide a sequence increment and, optionally, a sequence base, instead of generating these values automatically (as the NUMBER command does).

The group of records to be renumbered may be specified in any one of the following ways:

<group>

If a <group> is given, the records specified by the <group> are numbered. Unlike the NUMBER command, the <group> need not be bounded by numbered lines.

REGION

If REGION is specified, the group of unnumbered records that includes the offered line is numbered. If the offered line does not appear in an unnumbered area, the error message "No unnumbered text here" is displayed.

ALL

If ALL is specified, the entire file is renumbered.

The ":" syntax specifies the sequence base (<sequence number>) and the sequence increment (+ <integer>). If the sequence base is not specified, the sequence number of the record preceding the first record in the group is incremented and used.

If the ":" syntax is omitted, the Editor displays two forms on the command line and places the cursor in the appropriate position for entering the sequence increment. First, enter the increment. Then, if you would also like to enter the sequence base, move the cursor to the second form and enter the base. Then, press XMT.

Unlike the NUMBER command, a RESEQUENCE command never extends the range of records it numbers. If the sequence number and increment are too large to allow the records to fit, the Editor displays the message "Doesn't fit" and requests a new number and/or increment. If you would like to cancel the RESEQUENCE command at this point, enter "CANCEL" in the sequence increment portion of the form.

Records that have been renumbered are displayed with a "#" in the flag field. These records are considered "changed" for the "GO +/-ALTERATIONS" command, but not for the "GO +/-CHANGEDTEXT" command (unless the text field of the records has been changed by some other command).

See also

GO Command. 47
NUMBER Command. 66

CHANGE Command

```
 -- CHange --<group>--------------------------------------------------|
                     |             |
                     |- : -- MARK -|
```

The CHANGE command directs the Editor to mark the group of records specified by <group> as "changed", for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands, even if the text fields of these records have not been changed. In patch mode, these records will be placed in the patch file written when the Editor is exited.

The mark fields of the records in the <group> will not be set to the current mark value unless the MARK option appears. For example, "CHANGE 1" will mark the offered line as changed, but will not change the mark field, whereas "CHANGE 1 : MARK" will both mark the offered line as changed and set the mark field to the current mark value.

See also
GO Command. 47
Group and Destination Specifications. 26
MARK Command. 97
Patch Mode. 32

INSERT Command

```
-- INsert ---<filename>--------------------<destination>--------------|
           |            | |                |              |           |
           |- = --------| |-<range list>-|              |- : MARK -|
```

The INSERT command inserts records from an external file into the workfile at a specified point.

If a <filename> is specified, the records are inserted from the specified file and that file becomes the current external file (refer to "External Files"). If "=" is specified, the records are inserted from the current external file; if there is no current external file, the error message "No external file" is displayed. The inserted file must have the same record format as the file being edited.

The <range list>, if present, specifies the group of records to be inserted from the external file. If the <range list> includes a <qualified id>, the currently-loaded xreffiles are used to determine the sequence range, regardless of whether or not the xreffiles were originally generated for the external file. If no <range list> is specified, the entire external file is inserted.

The <destination> specifies the location at which the records are to be inserted.

The MARK option, if present, specifies that the mark fields of the inserted records are to be set to the current mark value (refer to the MARK command). If MARK is not specified, the records are inserted with the mark field left as it was in the external file.

Inserted lines are considered "changed" lines, for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands.

NOTE

When editing a COBOL or COBOL74 file, a blank must precede any hyphen appearing in either the <destination> syntax or the sequence number form of the <range list> syntax. This avoids possible syntactic ambiguities with COBOL identifiers. For example, "INSERT MYFILE 100-500 A-2", which is a valid command when editing a file that is not COBOL, must be entered as "INSERT MYFILE 100 -500 A -2" when editing a COBOL or COBOL74 file.

Example

```
BEFORE                          AFTER
-------------------------       ----------------------------------

[00000100]                      [00000100] One record was INSERTED
000100}]ins x/y 999 after       000100 one hundred
000200 two hundred                 ~  }line 999 of x/y
000300 three hundred            000200 two hundred
                                000300 three hundred
```

See also

External Files. 35
FILEKINDs and Record Formats. 29
GO Command. 47
Group and Destination Specifications. 26

MERGE Command

```
-- MERge ---<filename>-------------------------------------------------|
           |            | |              | |             |
           |- = --------| |-<range list>-| |- : -- MARK -|
```

The MERGE command merges an external "patch" file into the workfile by sequence number.

In the simplest case, where the external file does not contain any compiler control records ("$-cards"), the MERGE command is identical in operation to the CANDE "RMERGE" command. The records from the external file are interleaved with the existing records in the workfile according to the value of the sequence field of each record. If a record in the workfile has the same sequence number as a record in the external file, the workfile record is overwritten by the record from the external file.

The Editor recognizes two specific compiler control records: "null" records ("$" followed by all blanks) and "editing" records (SET and POP VOIDT or DELETE, depending on the FILEKIND of the file). A null compiler control record in the external (patch) file causes the workfile record of the same sequence number to be deleted. A pair of records containing "SET DELETE" and "POP DELETE" (or VOIDT) causes the workfile records with sequence numbers between and including the sequence numbers of these records to be deleted. All other compiler control records are merged into the workfile as ordinary data records.

This action is similar to the operation of the SYSTEM/PATCH program and is complementary to the Editor's patch generation facility (refer to "Patch Mode"). If a null or editing compiler control record is found that does not actually delete one or more records from the workfile, the following message is displayed: "WARNING: SOME COMPILER CONTROL RECORDS IN YOUR PATCH MAY BE DISCARDED!". If this action is not what you intended, enter an UNDO command, which will return the workfile to its state prior to the MERGE command.

If a <filename> is specified, the records are merged from the specified file and that file becomes the current external file (refer to "External Files"). If "=" is specified, the records are merged from the current external file; if there is no current external file, the error message "No external file" is displayed.

The <range list>, if present, specifies the records to be merged from the external file. If the <range list> includes a <qualified id>, the currently-loaded xreffiles are used to determine the sequence range, regardless of whether or not the xreffiles were originally generated for the external file. If no <range list> is specified, the entire external file is merged.

The MARK specification, if present, specifies that the mark fields of the merged records are to be set to the current mark value (refer to the MARK command). If MARK is not specified, the records are inserted with the mark field left as it was in the external file.

Merged lines are considered "changed" lines for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands.

Example

```
BEFORE                        AFTER
----------------------        ------------------------------------

[00000100]                    [00..]}    File merged into workfile
000100}]merge x/y             000100 original one.
000200 original two.          000200*NEW TWO.
000300 original three.        000250*NEW TWO-FIFTY.
000400 original four.         000300 original three.
                              000400 original four.
```

In this example, the user requested that the entire file X/Y be merged into the workfile. X/Y contained only two records, record 200 and record 250. Record 200 overwrote the original record 200; record 250 fell between line 200 and line 300, where there previously was no record.

See also

External Files.	35
GO Command.	47
MARK Command.	97
Patch Mode.	32

RENEW Command

```
-- RENew ---<group>----------------------------------------------|
           |                |
           |-<destination>-|
           |                |
           |- Vicinity ----|
```

The RENEW command returns an edited area of the workfile to its original, unedited state. The RENEW process consists of deleting the lines in the specified area and then restoring any lines that appear in the symbol file in the same sequence range. Because the RENEW process is based on a sequence range, the lines that determine the boundaries of the renewed area must have sequence numbers.

The area to be renewed is designated by the <group>, <destination>, or VICINITY specification:

<group>

If <group> is specified, the first line of the group, the last line of the group, and all lines in between are renewed. The first and last lines of the group must have sequence numbers.

<destination>

RENEW <destination> recovers deleted lines. If AFTER is specified as the <destination>, the deleted lines between the <destination> and the line before it are recovered. Both the <destination> and the line before it must have sequence numbers.

If BEFORE is specified as the <destination>, the deleted lines between the <destination> and the line after it are recovered. Both the <destination> and the line after it must have sequence numbers.

A relative line can be specified with either BEFORE or AFTER.

VICINITY

If VICINITY is specified, an area around the offered line is renewed; the renewed area begins with the first unchanged line preceding the offered line and ends with the first unchanged line following the offered line. Renumbered lines are considered changed lines for RENEW.

The examples below refer to a file containing the following lines, some of which have been changed or renumbered from the original symbol file:

```
000100 unchanged line
   ~  *changed line
000300*changed line
000400*changed line
000450#renumbered line
000600 unchanged line
```

]RENEW 2	entered on line 300	renews sequence range 300-400
]RENEW BEFORE	entered on line 300	is invalid, because the line preceding line 300 is unnumbered
]RENEW BEFORE	entered on line 400	renews sequence range 301-399
]RENEW AFTER	entered on line 300	renews sequence range 301-399
]RENEW VICINITY	entered on line 300	renews sequence range 100-600

NOTE

Implementation of the RENEW command is heavily dependent on sequence numbers. DATA files do not have sequence numbers. Therefore, this operation is not allowed for DATA and CDATA files.

See also

Group and Destination Specifications. 26

RESTORING COMMANDS

Restoring commands return either the file or the display to a previous state. Refer also to the FORGET command, which causes all changes to the workfile to be discarded, and the RENEW command, which reinstates records from the unedited symbol file.

COMMAND	ACTION
UNDO	Retracts the action of the previous command.
REFRESH	Rewrites all or part of the Editor display.

See also
FORGET Command. 105
RENEW Command 75

UNDO Command

```
 ---- UNdo --------------------------------------------------------------|
    |        |
    |- OOPS -|
```

The UNDO command retracts the last operation, returning the workfile to its state prior to the entering of that operation. This command is useful for undoing many types of mishaps, but is limited in scope to the immediately-previous operation only.

UNDO will retract the effects of all editing commands and of most other commands. It will not retract the establishment of a find or cross-reference target or the creation of an external file (such as a SAVE, PRINT, or OPTIONS:SAVE command).

An UNDO command cannot itself be undone by a second UNDO command.

Example

```
    BEFORE                                  AFTER
    -----------------------------------     --------------------

    [00000100]  3 records were DELETED      [00000100]
    000100 one hundred                      000100 one hundred
    000200 two hundred                      000200 two hundred
    000600}]undo                            000300}three hundred
                                            000400 four hundred
                                            000500 five hundred
                                            000600 six hundred
```

REFRESH Command

```
 -- REFResh -------------------------------------------------------------|
           |        |
           |- Line -|
           |        |
           |- Page -|
```

The REFRESH command is useful for restoring the validity of the information on the page when it has been corrupted by whatever means (for example, by your receiving a message from CANDE). REFRESH LINE causes the Editor to retransmit the command line and the offered line. REFRESH PAGE causes the Editor to retransmit the entire page. If neither LINE nor PAGE is specified, PAGE is assumed.

Whenever you are in doubt about the state of the edit-mode display, enter a REFRESH command.

REFRESH is not considered an operation for purposes of the UNDO command. UNDO can still be used to undo the last change to the file, even if one or more REFRESH commands have been entered in the meantime.

See also
UNDO Command. 78

Editor Commands

STATE-CHANGING COMMANDS

State-changing commands cause the Editor to assign new values, or display existing values, of various session options.

COMMAND	ACTION
OPTIONS	Assigns, displays, and saves a large number of the available session options.
FIND	Establishes a find target for subsequent searches through the file.
LABEL	Establishes and displays labeled locations for moving the offered line.
CANCEL	Cancels pending commands and other session state information.
MARK	Establishes a character string to be placed in the mark field of changed records.
COLUMN	Specifies a column in which the cursor is to be placed on the offered line.
TERMINAL	Assigns and displays terminal options.

OPTIONS Command

```
-- Options ------------------------------------------------------------|
             |                | |              | |              |
             |- REINITialize -| | |<--- , ---| | |- : -- SAve -|
             |                | | |          | |
             |- DEFAULT ------| |---<option>---|
<option>
```

```
----<xref option>------------------------------------------------------|
   |                                        |
   |-<margin spec>--------------------------|
   |                                        |
   |-/1\- SHift --<shift specs>-------------|
   |                                        |
   |-/1\- MINimum ---<integer>--------------|
   |                |                       |
   |-/1\- MAXimum -|                        |
   |                                        |
   |-/1\- LINes --<integer>-----------------|
   |                                        |
   |-/1\- RELative -------------------------|
   |                              | |       |
   |-/1\--- DISPLAYEnvirorment ---| |- ON --|
   |     |                      | | |       |
   |     |- DENV ---------------| | |- OFF -|
   |                              |
   |-/1\- COLHeading -------------|
   |                              |
   |-/1\- COLFooting -------------|
   |                              |
   |-/1\- CAsed ------------------|
   |                              |
   |-/1\- DUPlicate --------------|
   |                              |
   |-/1\- SPLit ------------------|
```

The Editor maintains several state variables that you may change to tailor the Editor displays or other actions to your requirements. Two copies of these state variables (or simply "options") are maintained by the Editor: the "current options," which apply only to the Editor session in progress, and the "saved options," which are stored in the "EDITOR/OPTIONS" file. When beginning a new session, the Editor initializes the current options from the saved options.

The OPTIONS command displays or changes option values. "OPTIONS" by itself displays the current set of options. All other forms of the OPTIONS command change the current and, optionally, the saved set of options. An OPTIONS command that changes option settings does not take effect until the command has been fully processed. If the command is in error, no option settings are changed.

The REINITIALIZE specification sets the current options to the values stored in the "EDITOR/OPTIONS" file. The DEFAULT specification sets the current options to the Editor's default values. The following list defines the Editor's DEFAULT option settings:

<xref option>s

ALIASES	OFF
ASSIGN	OFF
ENVIRONMENT	ON
PRINTER	OFF
TEXT	ON

<margin spec>s

LEFTMARGIN	Leftmost column of the text field
RIGHTMARGIN	Rightmost column of the text field
INDENT	0
RIGHTJUSTIFY	OFF

<shift specs>

RIGHT	2
MAXIMUM	2000
MINIMUM	2
LINES	1
RELATIVE	OFF
DISPLAYENVIRONMENT	OFF
COLHEADING	ON for RPG files, OFF otherwise
COLFOOTING	ON for RPG files, OFF otherwise
CASED	OFF
DUPLICATE	OFF
SPLIT	OFF

The command can include a list of <option>s, separated by commas. Each <option> listed is changed in the current option set. The <option>s are described later in this section.

If SAVE is specified, the current option set, as modified by any <option>s specified, is written to the "EDITOR/OPTIONS" file.

OPTIONS AVAILABLE THROUGH THE OPTIONS COMMAND

<xref option>

<xref option> is described in the section on "Cross-Referencing Commands" (refer to "Xref Options").

See also
Xref Options. 109

<margin spec>

<margin spec> is described in the section on the PARAGRAPH command.

See also
PARAGRAPH Command 63

<shift specs>

<shift specs> are described in the section on the SHIFT command.

See also
SHIFT Command 61

MINIMUM and MAXIMUM

The MAXIMUM option changes the value of the MAXINC variable; the MINIMUM option changes the value of the MININC variable. For example, "]OPTIONS MAX 1000, MIN 10" sets MAXINC to 1000 and MININC to 10. These variables are used by the Editor's automatic sequence-numbering facility, which is invoked by the NUMBER command and by default in the END, SAVE, and PRINT commands. The <integer> specified for MINIMUM must be greater than or equal to zero and less than MAXINC; the <integer> specified for MAXIMUM must be greater than MININC and less than the largest sequence number allowed for the FILEKIND of the file, as determined by the size of the sequence field (for example, the largest sequence number allowed for an ALGOL file is 99999999).

When numbering automatically, the Editor selects the largest sequence increment that allows all records in the group to be numbered within the sequence range bounded by the sequence numbers of the records on either side of the group. The value chosen is the largest multiple of 2, 5, or 10 times a power of 10 that does not exceed the current value of MAXINC. For example, if MAXINC were 2000, the Editor would consider 2000, 1000, 500, 200, 100, 50, 20, 10, 5, 2, and 1, in that order, choosing the largest increment that allows the newly-numbered records to "fit" between the records surrounding the group.

If the largest number that allows the records to fit is less than MININC, the Editor temporarily numbers all records in increments of MININC and continues numbering past the last record of the specified range to keep the sequence numbers in monotonically increasing order. Before considering the operation finished, the Editor notifies you that the numbered range was extended and allows you to either accept (by pressing SPCFY) or reject (by pressing XMT) the numbering of that range. If you accept the extension, the Editor assigns the sequence numbers and finishes the operation; if you reject the extension, the Editor discards the proposed numbers and returns to edit mode with the offered line on the first line of the unnumbered group that required the extension.

The base sequence number is selected, after the appropriate sequence increment is found, by centering the newly numbered records within the range of available sequence numbers between the bounds of the specified group of records.

If the whole file is unnumbered, both the base and the increment will be set to MAXINC.

See also

NUMBER Command. 66

LINES

The LINES option controls "multi-line mode." By default, the edit-mode "form" (the unprotected portion of the edit-mode display) is one line, allowing only one line to be changed and retransmitted by the user. In multi-line mode, several lines can be changed and retransmitted at once.

The <integer> specifies the number of lines the Editor is to offer for editing, inserting, or duplicating whenever it would normally offer one. For example, "]OPTIONS LINES 5" causes the Editor to offer five lines instead of just one. The <integer> must be in the range 1 to 23, inclusive. If the <integer> specified is 23, the entire page (except for the command line and the status line) is offered for editing.

Because the multi-line display operates in a similar manner to the single-line display for most commands and operations, the multi-line display is not described elsewhere. Wherever the phrase "the offered line" appears, it should be interpreted as "the offered line or lines".

The specific action taken for commands that are affected by the multi-line display can be predicted from the following rules:

1. The first line of the form is considered the principal line of the form for operations that specifically involve the offered line. For example, the first line is the only line on which commands can be entered. It is the only line that can be split by a <split-line cperation>. Also, when a new offered line is selected, such as by an <offer-line operation> or a "GO <sequence number>" command, the selected line is offered as the first line of the form. In <group> specifications that include offsets from the offered line, the offset is considered relative to the first line of the form. When the first line of the form is scrolled off the page, the entire form is deleted and the command line becomes the only offered line; however, when the last line of the form is scrolled off the page, only that line is removed (the form becomes <integer>-1 lines).

2. New data can be entered on all lines of the multi-line display. All cursor contrcls (such as up, down, left, right, TAB, RTAB, and RET) can be used to move the cursor within the multi-line form. At any point, the data entered can be transmitted to the system using the XMT key, which is described in the "Use of the Terminal" section. Note that characters that appear after a visible RET character on a line are ignored. The leftmost character of the top line of the form is considered the HOME position.

3. The offered range of lines moves as a group, according to how many lines were transmitted. For example, if three lines were displayed and the data were transmitted from the middle of the second line, the three-line form would be moved forward two lines.

When in duplicate mode, all insert lines are displayed with the text of the line preceding the first line of the group (refer to the DUPLICATE option, in this section).

Example

```
BEFORE                      AFTER
--------------------        --------------------

[00000100]                  [00000100]
000100}]opt lines 3         000100}one hundred
000200 two hundred          000200}two hundred
000300 three hundred        000300}three hundred
000400 four hundred         000400 four hundred
```

See also

Edit-Line Operation	54
GO Command.	47
Insert-Line Operation	43
Offer-Line Operation.	42
Use of the Terminal	16

RELATIVE

The RELATIVE option controls "relative mode". For example, "]OPTIONS RELATIVE" turns relative mode ON. When RELATIVE is OFF, as it is by default, the left side of the edit-mode screen displays the sequence number of each record. In relative mode, the sequence number area displays an offset of each line relative to the offered line. Lines above the offered line are numbered -1, -2, -3, ..., proceeding toward the beginning of the file. Lines below the offered line are numbered +1, +2, +3, ..., proceeding toward the end of the file. This display facilitates calculating line offsets for the <relative line> form of <group> and <destination>. A tilde (~) precedes the relative line number of any record that is unnumbered (that is, that does not have an assigned sequence number).

Example

```
    BEFORE                    AFTER
    --------------------      --------------------

    [00000100]                [00000100]
    000100 one hundred           - 3 one hundred
       ~   unnumbered         ~  - 2 unnumbered
    000300 three hundred         - 1 three hundred
    000400}]opt rel on              }four hundred
    000500 five hundred          + 1 five hundred
    000600 six hundred           + 2 six hundred
    000700 seven hundred         + 3 seven hundred
```

See also
Group and Destination Specifications. 26

DISPLAYENVIRONMENT

The DISPLAYENVIRONMENT option controls "environment mode". For example, "]OPTIONS DENV" turns environment mode ON. When in environment mode, the environment of the offered line is displayed on the command line. If the environment information is too long to fit on the command line, the trailing (outermost) environments are truncated.

The environment display is overwritten when an error message is displayed, when the offered line is the command line, when the Editor is reporting the results of a recent editing command, and when COLHEADING is ON (refer to the COLHEADING option, in this section).

If DISPLAYENVIRONMENT is ON in the permanently-saved EDITOR/OPTIONS file, the Editor automatically attempts to load cross-reference files during initialization of the Editor session. This option has no effect if cross-reference files are not loaded (refer to the LOAD command).

See also
Cross-Referencing Commands. .107
LOAD Command. 131

COLHEADING and COLFOOTING

In some circumstances, it is useful to be able to enter data in specific columns, referenced by column number. The COLHEADING and COLFOOTING options cause the Editor to continuously display a "ruler" indicating the column numbers across the page. For example, "]OPTIONS COLHEADING, COLFOOTING" causes both rulers to be displayed. The column numbers are relative to the text field of the record, not to the Editor's display.

COLHEADING displays the ruler on the command line, eliminating the display of the sequence number and other information normally displayed on the command line. The ruler display is overwritten when an error message is displayed, when the offered line is the command line, and when the Editor is reporting the results of a recent editing command.

COLFOOTING displays the ruler on the status line, eliminating the display of the information that normally appears on the status line.

Note: Column-mode is also useful for fixed-format data entry (refer to the COLUMN command).

Example

```
001200}]OPT COLHEADING
```

```
                  column 24               column 47
                      *                       *
. . . . 1 . . . . 2 . . . . 3 . . . . 4 . . . . 5 . . . . 6 . . . . 7 .
```

The ruler is shown above, approximately as it would appear on the command line. Each multiple of ten columns is indicated by a single digit, followed by a "." for each even-numbered column in between. The Editor displays the ruler in bright video.

See also
COLUMN Command. 99
FILEKINDs and Record Formats. 29

CASED

The CASED option establishes a default value for the FIND command CASED parameter. For example, "]OPTIONS CASED" turns the CASED parameter ON. If the current option value for CASED is ON, FIND commands will be case sensitive, unless CASED is overridden in the FIND command itself. If the current option value is OFF, FIND commands will be case insensitive by default. Case-sensitive and case-insensitive find operations are described under the FIND command.

See also

FIND Command. 92

DUPLICATE

The DUPLICATE option controls "duplicate mode," which is useful for entering a series of new records where each record is very similar in content to the record that immediately precedes it. For example, "]OPTIONS DUPL" turns duplicate mode ON.

Normally, the Editor displays a blank line when offering a new line in insert mode. In duplicate mode, the Editor displays each line offered in insert mode as a copy of the line that precedes it, rather than as a blank line. The duplicate text does not become part of the workfile unless it is transmitted in an edit-line operation.

Example

```
BEFORE                      AFTER
--------------------        --------------------

[00000100]                  [00000100]
000100 SIGN-UP SHEET        000100 SIGN-UP SHEET
000200 1.___________        000200 1.___________
 >>>> }]opt dup on           >>>> }1.___________
000300                      000300
```

In this example, the offered line was an insert line before line 300, where the user entered "]opt dup on". Once in duplicate mode, instead of providing a blank insert line, the Editor duplicated the contents of line 200. At this point, the user would most likely change the "1" on the offered insert line to a "2" and then press HOME, XMT. If so, the Editor offers another insert line, this time with the contents duplicated from the previously-inserted line (that is, "2.__________").

This process continues until the user enters "]opt dup off" or leaves insert mode. If insert mode is re-entered while still in duplicate mode, the text for the new insert line is duplicated from the line immediately preceding it; for example, if insert mode is entered after line 100, "SIGN-UP SHEET" would be displayed in the text field of the insert line.

See also

Edit-Line Operation 54
Insert-line Operation 43

SPLIT

The SPLIT option controls "split mode." For example, "]OPTIONS SPLIT" turns split mode ON. When in split mode, pressing SPCFY in the text field of a displayed line causes the line to be split where the cursor is positioned, even if that line is not the offered line. If you plan to split only one or two lines, consider using the split-line operation.

The normal SPCFY key operations are affected as follows:

<offer-line operation>

The offer-line operation is not available in split mode.

<split-line operation>

This operation works in split mode.

<insert-line operation>

The "insert-after" form of this operation works normally in split mode (that is, pressing SPCFY on the sequence or flag field of a line changes split mode to insert mode after that line). The "insert-before" form is not available in split mode (pressing SPCFY anywhere on the offered line will split the line).

See also

Insert-Line Operation 43
Offer-Line Operation. 42
Split-Line Operation. 55

FIND Command

```
 -- FIND --<find target>------------------------------------------------|
                          |                                 |
                          |     |<----------- , ----------| |
                          |     |                         | |
                          |- : -----/1\- Changed -----------|
                                  |                     |
                                  |-/1\- CAsed ---------|
                                                |       |
                                                |- ON --|
                                                |       |
                                                |- OFF -|
```

<find target>

```
 ----<id>--------------------------------------------------------------|
   |                                       |
   |-<integer>-----------------------------|
   |                                       |
   |             |<-----------|            |
   |             |            |            |
   |-<delimiter>---<graphic>---<delimiter>-|
```

The FIND command establishes a find target: the token or string of characters to be searched for when a subsequent "GO +/-FIND" command is entered to initiate the search. The FIND command is also used to specify other search parameters.

Depending on how the find target is specified, the Editor performs either a "token search" or a "literal search"; these types of searches are very similar to the searches performed for the CANDE "FIND" command. The token search considers the target as a language token; that is, in order for the target to be found, it must appear in the text as an isolated token, delimited by blanks or special characters. The literal search considers the target as a string of characters; in order for the target to be found, it need only appear as a substring in some record of the file. A literal search will find "SET" within the identifier "RESETTOGGLE", whereas a token search will not.

When an <id> is given as the <find target>, a token search is performed for the specified identifier.

When an <integer> is given as the <find target>, a token search is performed, with one modification: If a token is found that differs from the <find target> only in that it includes leading zeros, that token is found. Thus, a search with a <find target> of 3 will find the token 003 in the file.

When a string of <graphic>s within <delimiter>s is given as the <find target>, a literal search for the target is performed. The two <delimiter>s must match, and no <graphic> used may be the same as the <delimiter> used.

The CHANGED option searches only changed records, which is usually much faster than searching all records.

The CASED option specifies whether or not the search is to be "case sensitive" and applies to both literal and token searches. A case-insensitive search will find all occurrences of the target that contain the proper letters in the proper positions, without regard to whether the letters are in uppercase or lowercase. A case-sensitive search will find only those occurrences of the target that are identical to the <find target> with respect to the case of the letters. "CASED ON", or just "CASED", specifies a case-sensitive search. "CASED OFF" specifies a case-insensitive search. If CASED appears in a FIND command, the setting applies only to that single command. If CASED is not specified in the command, it defaults to the current value of the CASED option (refer to the OPTIONS command).

The current find target and search parameters can be discarded by entering a CANCEL command.

See also

CANCEL Command. 95
GO Command. 47
OPTIONS Command 82

LABEL Command

```
-- LABel -----------------------------------------------------------|
          |             |
          |-<label id>-|
```

The LABEL command associates a label with the currently offered line, allowing the location to be reached later through a "GO <label id>" command.

"LABEL <label id>" associates the specified <label id> with the offered line. If the label is already in use, its former line association is discarded. The LABEL command is invalid if the offered line is an insert line.

If a labeled line appears on the edit-mode page, the label is displayed in place of the sequence number for the line. Moving a labeled line also moves the label. Copying a labeled line has no effect on the label (the label remains with the original source). Deleting or renewing a labeled line discards the label.

The LABEL command by itself causes the Editor to display a list of all currently assigned labels and their locations. Moving the cursor to the name or location of a label in this display and then pressing SPCFY causes the Editor to offer the line associated with that label (this action is identical to the "GO <label id>" command).

A maximum of 12 labels can be defined at one time. The "CANCEL LABEL" command can be used to discard labels.

See also
CANCEL Command. 95
GO Command. 47
Insert-line Operation 43

CANCEL Command

```
-- CANcel ------------------------------------------------------------|
            |                       |
            |- Move ----------------|
            |                       |
            |- Copy ----------------|
            |                       |
            |- DELete --------------|
            |                       |
            |- RENew ---------------|
            |                       |
            |- NUMber --------------|
            |                       |
            |- RESequence ----------|
            |                       |
            |- PARagraph -----------|
            |                       |
            |- CENter --------------|
            |                       |
            |- SHift ---------------|
            |                       |
            |- CHange --------------|
            |                       |
            |- LABels --------------|
            |          |            |
            |          |-<label id>-|
            |                       |
            |- Find ----------------|
            |                       |
            |--- Xref --------------|
            | |           |         |
            | |- REFerence -|       |
            |                       |
            |- MARK ----------------|
```

The CANCEL command is used to back out a pending command that you do not wish to continue with, to discard labels, to reset the current find or cross-reference target to null, or to reset the current mark value to null.

A pending command is a command for which not all of the required location designations (first, last, and/or destination) have been provided (refer to "Pending Commands"). Occasionally, when a command is pending, you may decide not to proceed with the command.

The CANCEL MOVE, COPY, DELETE, RENEW, NUMBER, RESEQUENCE, PARAGRAPH, CENTER, SHIFT, and CHANGE commands discard the pending first, last, and/or destination specifications for the corresponding command. CANCEL by itself cancels whatever command is pending.

The CANCEL LABELS command discards one or all label settings. If a specific <label id> is given, only that label is discarded; all other labels remain as they are currently specified. The CANCEL LABELS command by itself discards all currently-defined labels. (Refer to the LABEL command.)

The CANCEL FIND command resets the find target to null (no specified target). (Refer to the FIND command.)

The CANCEL XREF command resets the xref target to null (no specified target). CANCEL REFERENCE is a synonym for CANCEL XREF. (Refer to "Cross-Referencing Commands.")

The CANCEL MARK command resets the current mark value to all blanks (refer to the MARK command).

Example

```
BEFORE                    AFTER
--------------------      --------------------

[00000100]  MOVE D        [00000100]
000100@destination        000100 destination
000200                    000200
000300}]cancel move       000300}
```

In this example, the user had marked line 100 as the destination for a MOVE command; "MOVE D" is displayed on the command line, indicating that a MOVE command is pending and the destination (D) has already been designated (line 100, as indicated by the "@" in the flag field). After the CANCEL MOVE command, the MOVE is no longer pending.

See also
Cross-Referencing Commands. 107
FIND Command. 92
LABEL Command 94
MARK Command. 97
Pending Commands. 36

MARK Command

```
 -- MARK --<mark>-------------------------------------------------------|
```

The MARK command changes the current mark value. The current mark value is written into the mark fields of records that have been significantly changed by editing commands, that are newly-created, or that have been explicitly designated to be marked by a CHANGE command with the MARK option specified. By default, the <mark> is all blanks.

The effects of each type of editing command on the mark fields of affected records is described below:

Edit-Line Operation — If the text field has merely been shifted (that is, if the only change to the text is that the number of leading and trailing blanks is changed), the mark field is not changed. Any other change to the text field is considered significant and will cause the mark field to be set to the current mark value. All newly-created (insert) lines are marked.

Split-Line Operation — The mark fields of both new records are set to the current mark value.

JOIN — The mark field of the resulting record is set to the current mark value.

DELETE — The mark value is not applicable, because the records are deleted.

MOVE — The mark fields of moved records are preserved.

COPY — The mark fields of the records inserted at the destination by a COPY command are set to the current mark value.

SHIFT — The mark fields of shifted records are preserved.

PARAGRAPH — The mark fields of the new paragraphed records are set to the current mark value.

CENTER — The mark fields of centered records are preserved.

NUMBER — The mark fields of numbered records are preserved.

RESEQUENCE — The mark fields of renumbered records are preserved.

CHANGE — The mark fields of changed records are preserved, unless the MARK option is specified.

INSERT The mark fields of inserted records are preserved from the external file, unless the MARK option is specified.

MERGE The mark fields of merged records are preserved from the external file, unless the MARK option is specified.

RENEW The mark fields of renewed records are preserved from the symbol file.

The mark value persists throughout the session unless changed by a subsequent MARK command or canceled by a CANCEL MARK command.

See also

CANCEL Command. 95
CHANGE Command. 70
Editing Commands. 53
Insert-line Operation 43

COLUMN Command

```
 -- COLumn ------------------------------------------------------------|
             |          |
             |- ON -----|
             |          |
             |- OFF ----|
             |          |
             |-<column>-|
```

The COLUMN command controls "column mode." When offering a line for editing, the Editor normally places the cursor where it was when SPCFY was pressed, on the first nonblank character of the line, or on the find or cross-reference target, depending on how the line became the offered line. In column mode, the Editor unconditionally places the cursor on the specified column.

The COLUMN command by itself turns column mode ON and sets the target column to the column in which the "]" character was entered (as the first nonblank character of the command). For example, if "]COL" were entered with the "]" character in column 10, the Editor would place the cursor in column 10 whenever offering a line for editing. "COLUMN ON" is a synonym for "COLUMN".

"COLUMN <column>" turns column mode ON and sets the target column to the specified <column>. For example, the command "]COL 65" causes the Editor to always place the cursor in column 65 of the offered line, regardless of where the "]" character of the command was entered. The specified <column> must be in the range defined for the text field of the FILEKIND of the workfile. For example, "]COL 3" is an error if the workfile is a COBOLSYMBOL file.

"COLUMN OFF" turns column mode OFF, returning the Editor to its default behavior.

Whenever column mode is ON, the current column setting is displayed on the command line and on the status line.

See also
    FILEKINDs and Record Formats. 29

TERMINAL Command

```
-- TERMinal -----------------------------------------------------------|
             |                   |
             |- FULL ------------|
             |        |          |
             |        |- ON -----|
             |        |          |
             |        |- OFF ----|
             |                   |
             |- TD830 -----------|
             |          | |      |
             |- MT983 --| |- ON -|
             |          |
             |- MT987 --|
             |          |
             |- ET1100 -|
             |          |
             |- SR100 --|
```

The TERMINAL command sets and displays options relating to terminal characteristics.

The TERMINAL command by itself displays the current option settings.

If an option appears, it is set either ON or OFF as specified. If neither ON nor OFF appears, ON is assumed.

The FULL option controls the terminal's full-line-transmit option (refer to "Use of the Terminal"). Changing this option will change the firmware in the terminal. This setting persists until the next time the terminal is reinitialized, which is usually after the terminal has been powered off and then powered back on. If the terminal is reinitialized during an Editor session, enter the "]TERM FULL ON" or "]TERM FULL OFF" command again to be sure that the setting is correct. The FULL option is not allowed for an SR100 terminal.

The TD830, MT983, MT987, ET1100, and SR100 options designate the type of terminal you are using. These options are mutually exclusive; turning any one ON automatically turns the others OFF. Because the Editor requires that one of these options be ON, it will ask you to provide this information when you first execute the program.

Once you have responded to the question "Is this terminal a TD830, an MT983, an MT987, an ET1100, or an SR100?", the Editor will create or add to the "EDITOR/OPTIONS" file under your usercode and will automatically save the terminal option settings, associating these option settings with your CANDE station. The Editor will not request this information again for the same usercode and CANDE station.

The Editor will close and reopen its remote file and refresh the screen when a TERMINAL command is entered.

For further information on the characteristics of supported terminals, refer to the "Use of the Terminal and Installing the Editor" section.

See also
    Installing the Editor 139
    Use of the Terminal 16

EXITING COMMANDS

The exiting commands terminate the Editor session and specify the disposition of the changes made to the workfile during the session.

COMMAND	ACTION
END	Writes an updated CANDE workfile, (usually) discards the recovery file, and exits.
FORGET	Does not write an updated CANDE workfile, discards the recovery file, and exits.
RECESS	Does not write an updated CANDE workfile, saves the recovery file, and exits.

END Command

```
 ---- END --------------------------------------------------------------|
    |       | |                              |
    |- BYE -| |     |<--------- , ---------| |
              |     |                      | |
              |- : ----/1\- Recess ----------|
                      |                  |
                      |-/1\- Unnumbered -|
```

The END command terminates the current Editor session, returning the workfile to CANDE with all of the changes incorporated from the "EDITOR/RECOVERY" file. The recovery file is then discarded, unless RECESS is specified. RECESS may be specified only in patch mode (refer to "Patch Mode"). BYE is a synonym for END.

The UNNUMBERED option causes the Editor to leave records that are unnumbered in the file unnumbered as it is writing the final workfile. If UNNUMBERED is specified, unnumbered records are left with blanks in the sequence field. If UNNUMBERED is not specified, the Editor will automatically assign sequence numbers to all unnumbered records as they are written to the new file (refer to the MINIMUM and MAXIMUM options under the OPTIONS command for a description of the automatic numbering facility and the warning that is given if the range of records must be extended). UNNUMBERED is valid only in patch mode.

See also

OPTIONS Command 82
Patch Mode. 32

FORGET Command

```
 -- FORGET ------------------------------------------------------------|
```

The FORGET command ends the Editor session without merging the changes from the "EDITOR/RECOVERY" file into the workfile. The recovery file is discarded, and the workfile is returned to CANDE unchanged.

The Editor gives you a chance to reconsider by not purging the recovery file until you have confirmed that this action is what you intended. When you enter FORGET, the Editor displays the message "You are about to throw away the results of your efforts." At this point, you have the option of pressing SPCFY, which discards the recovery file and ends the session, or pressing XMT, which aborts the FORGET command and returns the Editor to edit mode.

RECESS Command

```
 -- RECESS --------------------------------------------------------------|
```

The RECESS command ends the Editor session without merging the changes from the "EDITOR/RECOVERY" file into the workfile; instead, the changes are preserved in the recovery file for use at a later time. Recovery from a RECESSed Editor session is functionally equivalent to recovery from a system failure or a CANDE "?DS" command (refer to "Recovery Files" for additional information).

See also

Recovery Files. 34

CROSS-REFERENCING COMMANDS

Cross-referencing commands, with cross-reference files ("xreffiles") generated by a compiler, aid in developing and modifying programs by providing access to language-oriented information about the identifiers that appear in the workfile. Before you can use any of these commands, xreffiles must have previously been loaded, either automatically or by request (refer to the LOAD command).

The ALGOL, COBOL74, DCALGOL, FORTRAN, FORTRAN77, and NEWP compilers will generate xreffiles suitable for use with the Editor if the compiler control option XREFFILES is set to TRUE. Cross-reference information is generated by a compiler during the compilation of a particular version of a particular program. Thus, xreffiles do not reflect changes made to the file after the compilation was performed.

Cross-referencing commands are very useful when patching large programs. If your program is more than a few thousand lines, you might consider using the xreffiles facility.

COMMAND	ACTION
REFERENCE	Establishes a target and/or displays references to a target identifier.
DECLARATION	Moves the offered line to the declaration of a target identifier.
FORWARD	Moves the offered line to the forward declaration of a target identifier.
ALL	Displays references to all identifiers of a specified name or partial name.
RANGE	Establishes a set of sequence ranges that restricts the scope of other xref commands.
INTERSECTION	Displays references on which two or more target identifiers both appear.
UNION	Displays references on which any one of two or more target identifiers appears.
RECENT	Displays the last few target identifiers.
WHERE	Displays the environment of a given line or RCW.

See also
LOAD Command. 131

Xref Options

<xref option>

```
 ----/1\- Aliases ------------------------------------------------------|
   |                   | |       |
   |-/1\- ASsign ------| |- ON --|
   |                   | |       |
   |-/1\- Environment -| |- OFF -|
   |                   |
   |-/1\- Printer -----|
   |                   |
   |-/1\- Text --------|
```

Xref options allow settings to be specified for options that affect the cross-referencing commands. These options can be used in the OPTIONS command or in cross-referencing commands (they cannot be entered as stand-alone commands). The current session and saved values of these options can be displayed and changed through the OPTIONS command. In some cross-referencing commands, the current values can be overridden by specifying the option directly in the command. In this case, the specified setting applies only to the command in progress.

In the <xref option> syntax, the option is activated if the option name is followed by ON or nothing, and is deactivated if followed by OFF. For example, "T" specifies TEXT ON, while "E OFF" specifies ENVIRONMENT OFF.

ALIASES

In NEWP, an identifier may be address-equated to the location of another identifier; these address-equated identifiers are called "aliases." The ALIASES option causes references to aliases of the specified identifier to be included in the list of references.

ASSIGN

The ASSIGN option limits the displayed list of references to only those lines at which the compiler has indicated that the identifier is being assigned a value.

ENVIRONMENT

The ENVIRONMENT option causes the environment for each reference to be displayed in the list of references.

PRINTER

The PRINTER option causes the reference display to be written to a printer file instead of being displayed on the terminal. If PRINTER is ON for a REFERENCE, INTERSECTION, or UNION command, TEXT and ENVIRONMENT are implied ON for that command unless specifically set OFF in the command itself.

TEXT

The TEXT option causes the text field for each referenced line to be displayed in the list of references. If the Editor encounters a reference for which a record cannot be found in the current effective workfile, it displays the string "*** MOVED OR DELETED ***" in place of the text.

See also

ALL Command 116
INTERSECTION and UNION Commands 118
OPTIONS Command 82
REFERENCE Command 111

REFERENCE Command

```
 -- REFerence ------------------------------------------------------->
                 |                 | |                 |
                 |-<qualified id>-| |-<range list>-|
                 |                 |
                 |- = ------------|

 >-------------------------------------------------------------------|
    |                        |
    |     |<-------- , -------| |
    |     |                   | |
    |- : -----<xref option>-----|
              |               |
              |-/1\- Once ----|
              |               |
              |-/1\- Summary -|
              |               |
              |-/1\- NO ------|
```

The REFERENCE command establishes a cross-reference (xref) target and displays a list of references for the current xref target or a specified <qualified id>.

If a <qualified id> is specified, references to that identifier are displayed. If "=" or nothing is specified, references to the current xref target are displayed (the "=" syntax is provided to avoid confusion when references to the current xref target are to be displayed and a <range list> is explicitly specified).

The <range list> specifies the portion of the file for which you would like references to be displayed. If <range list> is not specified, it defaults to the current RANGE specifications (refer to the RANGE command).

One or more <xref option>s may be specified to override the current option settings (refer to the OPTIONS command). Options specified in a REFERENCE command apply only to that command. All options are valid for the REFERENCE command.

If the ONCE option does not appear, the current xref target is discarded and the specified <qualified id> becomes the new xref target (the old xref target can be retrieved with the RECENT command). If ONCE appears, the specified <qualified id> is used, but the current xref target remains unaltered.

The SUMMARY option causes the Editor to display summary information only. This information includes a description of the identifier and summarized statistics about the references, such as total number of references, number of references in address-equations, and number of changed references.

The NO specification causes the Editor to establish the specified <qualified id> as the current xref target but not to display any information about that identifier.

The Reference Display

The reference list is a display page that includes a summary of the identifier's name, environment, type, declaration location, forward location, and other general information, followed by a list of sequence numbers of records at which the identifier appears. Each listed reference may be preceded by one or more of the following characters:

- * An asterisk indicates that the identifier was assigned a value at that line.
- = An equal sign indicates that the identifier appeared in an address-equation at that line.
- # A number sign indicates that the identifier appeared in a define at that line.
- + A plus sign indicates that the reference displayed was to an alias of the identifier.

Depending on the options specified in the OPTIONS command or in the REFERENCE command itself, the reference display may include additional information, such as the actual text of the line for each reference or the environment of each reference. (The environment is described later in this section.) For NEWP programs that include module declarations, references displayed in reverse video are references to the identifier that appear outside the module in which the identifier is declared.

After displaying the reference list, the Editor waits for input before displaying the next page (if any). When you press SPCFY without moving the cursor, the Editor displays the next page, if there is one. If there is no other page, the Editor returns to edit mode. When you move the cursor to the sequence number of a displayed reference and press SPCFY, the Editor terminates the reference display and offers that line for editing. Any other input returns the Editor to edit mode.

In xreffiles for ALGOL and NEWP, the location at which a LABEL variable occurs is not considered a reference. To move to where the label occurs, enter a REFERENCE command for the label <id> and include the ": SUMMARY" option, then press SPCFY with the cursor on the sequence number after "OCCURS @".

See also

OPTIONS Command	82
RANGE Command	117
RECENT Command	120
Xref Options	109

DECLARATION and FORWARD Commands

```
---- DEClaration ------------------------------------------------------|
  |                | |                 | |             |
  |- FORward -----| |-<qualified id>-| |- : -- Once -|
```

The DECLARATION command moves the offered line to the declaration of the specified <qualified id> or, if null, to the declaration of the current xref target. The FORWARD command moves the offered line to the forward declaration of the specified <qualified id> or xref target; if there is no forward declaration, the offered line is moved to the declaration of the identifier.

If ONCE does not appear, the current xref target is discarded and the specified <qualified id> becomes the new xref target (the old xref target can be retrieved with the RECENT command). If ONCE appears, the specified <qualified id> is used, but the current xref target remains unaltered.

See also
RECENT Command. 120

ALL Command

```
 -- ALL --<id>---------------------------------------------------------|
               |     | |              | |                       |
               |- = -| |-<range list>-| |     |<------ , -----| |
                                        |     |               | |
                                        |- : ---<xref option>---|
```

The ALL command displays information about all unique identifiers called <id> that were declared within the specified range.

If "=" appears, information is displayed about all identifiers that begin with the character string specified by the <id>. For example, the command "]ALL I=" would display information about all identifiers beginning with the letter I.

The <range list> specifies the portion of the file for which you would like the identifiers to be displayed. If <range list> is not specified, it defaults to the current option setting for RANGE (refer to the RANGE command).

One or more <xref option>s may be specified to override the current option settings (refer to the OPTIONS command). Options specified in an ALL command apply only to that command. The only valid option for the ALL command is PRINTER.

After displaying the identifier list, the Editor waits for input before displaying the next page (if any). When you press SPCFY without moving the cursor, the Editor displays the next page, if there is one. If there is no next page, the Editor returns to edit mode. When you press SPCFY with the cursor on a displayed identifier, the Editor assigns the specified identifier as the cross-reference target and then goes to the declaration of that identifier (as if a DECLARATION command had been entered for that identifier). When you press SPCFY with the cursor on a displayed sequence number, the Editor offers that line. Any other input will cause the Editor to return to edit mode.

See also
DECLARATION and FORWARD Commands. 115
OPTIONS Command 82
RANGE Command 117
Xref Options. 109

RANGE Command

```
-- RANGE ----------------------------------------------------------|
          |              |
          |-<range list>-|
          |              |
          |- OFF --------|
```

The RANGE command restricts the cross-reference displays generated by various cross-referencing commands to only a portion of the file.

"RANGE" by itself displays the current range specifications.

"RANGE <range list>" enables the range feature for the REFERENCE, ALL, INTERSECTION, and UNION commands and establishes the areas of the file for which references are to be displayed. RANGE specifications pertain to the current session only.

"RANGE OFF" disables the range feature.

See also

ALL Command 116
INTERSECTION and UNION Commands 118
REFERENCE Command 111

INTERSECTION and UNION Commands

```
                         |<-/3\---- , ----|
                         |                |
---- INTERsection -----<qualified id>---------------------------------->
  |                |                      |              |
  |- UNION --------|                      |-<range list>-|

>-----------------------------------------------------------------------|
  |                         |
  |        |<------ , -----| |
  |        |               | |
  |- : -----<xref option>---|
        |                   |
        |- Summary ---------|
```

The INTERSECTION and UNION commands display lists of references that are combinations of the normal reference lists displayed for the REFERENCE command. The reference list for the INTERSECTION command contains only those lines on which all of the specified <qualified id>s appear (that is, the list is the intersection of the reference lists for each of the specified identifiers). The reference list for the UNION command contains all lines on which any of the specified <qualified id>s appear (that is, the list is the union of the reference lists for each of the specified identifiers).

The <range list> specifies the portion of the file for which you would like the identifiers to be displayed. If <range list> is not specified, it defaults to the current RANGE specifications (refer to the RANGE command).

One or more <xref option>s may be specified to override the current option settings (refer to the OPTIONS command). Options specified in an INTERSECTION or UNION command apply only to that command. All options except the ALIASES option are valid for the INTERSECTION and UNION commands.

The SUMMARY option causes the Editor to display summary information only.

After displaying the reference list, the Editor waits for input before displaying the next page (if any). When you press SPCFY without moving the cursor, the Editor displays the next page, if there is one. If there is no other page, the Editor returns to edit mode. When you move the cursor to the sequence number of a displayed reference and press SPCFY, the Editor terminates the reference display and offers that line for editing. Any other input returns the Editor to edit mode.

See also

OPTIONS Command 82
RANGE Command 117
REFERENCE Command 111
Xref Options. 109

RECENT Command

```
 -- RECent ------------------------------------------------------------|
```

The RECENT command displays a description of the last eight (or fewer) identifiers that were once the cross-reference (xref) target.

After displaying the identifier list, the Editor waits for input. When you press SPCFY without moving the cursor, the Editor returns to edit mode. When you move the cursor to a displayed identifier and then press SPCFY, the Editor assigns the specified identifier as the xref target and then goes to the declaration of that identifier (as if a DECLARATION command had been entered for that identifier). When you move the cursor to a displayed sequence number and press SPCFY, the Editor offers that line (as if a "GO <sequence number>" command had been entered). Any other input returns the Editor to edit mode.

See also

DECLARATION and FORWARD Commands. 115
GO Command. 47
REFERENCE Command 111

WHERE Command

```
 -- WHERE --------------------------------------------------------------|
           |                   |
           |-<sequence number>-|
           |                   |
           |-<rcw spec>--------|
```

<rcw spec>

```
 -- RCW --<sdi>----------<pwi>---------<psi>----------------------------|
                 |     |       |     |
                 |- : -|       |- : -|
```

<sdi>

A <hex number> representing the Segment Dictionary Index of the code address.

<pwi>

A <hex number> representing the Program Word Index of the code address (also known as the Program Index Register (PIR)).

<psi>

A <hex number> representing the Program Syllable Index of the code address (also known as the Program Syllable Register (PSR)).

The WHERE command displays a description of the environment of the specified line.

"WHERE" displays the environment description for the current offered line.

"WHERE <sequence number>" displays a description of the environment that contains the record associated with the specified <sequence number>.

"WHERE <rcw spec>" displays a description of the environment that contains the record associated with the sequence number associated with the specified RCW. The sequence number is derived from the lineinfo in the code file, which must have been loaded previously (refer to the LOAD command).

See also

LOAD Command. 131

SPECIAL-PURPOSE COMMANDS

Special-purpose commands display information, create and access external files, and perform other nonediting functions. These commands do not affect the contents of the workfile.

COMMAND	ACTION
TEACH	Displays the Editor's textbook.
LIST	Displays an external file.
PRINT	Writes all or part of the file to the printer.
LOAD	Loads xref files, error files, and code files for use with subsequent commands.
SAVE	Saves all or part of the file without exiting the Editor.
WHAT	Displays file information and statistics.
SEQCHECK	Verifies that the file has properly ascending sequence numbers.

TEACH Command

```
---- TEACH -----------------------------------------------------------|
  |         |         |
  |         |-<topic>-|
  |                   |
  |- HELP ------------|
```

<topic>

Any keyword or metatoken recognized by the Editor as a teachable subject. Many of these identifiers are displayed on the index page of the teach information.

The TEACH command causes the Editor to enter "teach mode". In teach mode, the Editor textbook is displayed instead of the file being edited. The textbook describes all Editor commands and various aspects of the program's operation.

Depending on the exact TEACH command you enter, the Editor displays some page of the textbook. [Once in teach mode, press SPCFY to move to the next page of the textbook.] "TEACH" by itself displays the textbook's "index page," on which many of the available teachable topics appear. "TEACH <topic>" will display the page associated with the requested topic; if the <topic> is not recognized, the index page is displayed. "HELP", which is a synonym for "TEACH TEACH", will display this description of the TEACH command.

Once in teach mode, there are several ways of moving around within the textbook. Depending on your location in the textbook, more than one of these methods may move you to the same page; for example, if you are on the last page of a topic, you can press SPCFY in the "Next Topic" box or in the form on the top line (meaning "go to next page"). Both methods move you to the same place: the next page. Use whichever method seems most convenient at the time.

To move to the next page, press SPCFY.

To move to a particular topic, enter the topic in the form at the top left-hand corner of the teach display and press XMT. For example, for information on the COPY command, enter "COPY", then press XMT. You can move the cursor to wherever the topic appears on the screen and press SPCFY, instead of entering the topic in the form. For example, if the word COPY appears on the screen, place the cursor anywhere on that word and press SPCFY to move the Editor to the information about the COPY command. If you enter or press SPCFY on an unrecognized topic, the Editor displays "Unknown topic" and remains at the same location.

To move to the index page, enter "INDEX" in the form at the top of the screen, or move the cursor to the "Index Page" box of the SPCFY menu and press SPCFY.

To scroll forward or backward within the current topic, enter "+n" to scroll "n" lines forward or "-n" to scroll "n" lines backward. For example, "-10" will scroll the text backward 10 lines.

NOTE

> The CTRL-key synonyms for the scrolling commands are also valid in teach mode; refer to "CTRL-Key Synonyms for Editor Commands."

Scrolling is limited by the boundaries of the current topic; the Editor will display "End of Topic" on the status line when the last page of the topic is displayed.

To move to the next topic, move the cursor to the "Next Topic" box of the SPCFY menu and press SPCFY.

To move to the previous topic, move the cursor to the "Previous Topic" box of the SPCFY menu and press SPCFY.

To refresh the page, enter one or more blanks in the form and press XMT or enter "CTRL 00" (the CTRL-key synonym for REFRESH).

To return to edit mode (exit teach mode), move the cursor to the "Edit Mode" box of the SPCFY menu and press SPCFY. The Editor will offer the line that was offered prior to entering teach mode.

You can print the entire Editor textbook by entering the "PRINT TEACH" command in edit mode (refer to the PRINT Command).

See also

Control Key Synonyms for Editor Commands. 137
PRINT Command 127

LIST Command

```
 -- List ---<filename>-------------------------------------------->
          |            | |               |
          |- = --------| |--<range list>-|

 >--------------------------------------------------------------------|
   |                             | |                     |
   |- @ --<column>---------------| |- : -- Unnumbered -|
                 |               |
                 |- - --<column>-|
```

The LIST command lists all or part of a disk file on the terminal.

If a <filename> is given, the specified file is listed and that file becomes the current external file (refer to "External Files"). If "=" is given, the current external file is listed; if there is no current external file, the error message "No external file" is displayed.

The <range list>, if present, specifies the range of records to be listed. If no <range list> is specified, the entire external file is listed. If the <qualified id> form of <range list> is used, xreffiles for the file to be listed must have been previously loaded, either automatically or by request (refer to the LOAD command).

If the "@<column>" syntax appears, only the specified column range will be listed.

If the UNNUMBERED option appears, the sequence fields of the listed records will not be displayed.

As much of each record will be listed as will fit on one line; if the text to be displayed (including sequence numbers, if any) is longer than one line, it will be truncated on the right-hand side.

See also
    External Files. 35
    LOAD Command. 131

PRINT Command

```
 -- PRINT -------------------------------------------------------------->
            |           | |              |
            |- Patch ---| |-<range list>-|
            |           |                |
            |- Merged --|                |
            |           |                |
            |- Flagged -|                |
            |           |                |
            |- PAGed ---|                |
            |                            |
            |- TEACH --------------------|

 >----------------------------------------------------------------------|
   |                                              |
   |     |<------------------ , ----------------| |
   |     |                                      | |
   |- : -----/1\- TRAinid ----------<trainid>-----|
            |               |     |           |
            |               |- = -|           |
            |                                 |
            |-/1\- Graphics ------------------|
            |                                 |
            |-/1\- UPpercased ----------------|
            |                                 |
            |-/1\- Unnumbered ----------------|
            |                                 |
            |-/1\- Withsequencenumbers -------|
            |                                 |
            |-/1\- COPies -- = --<integer>----|
            |                                 |
            |-/1\- Environment ---------------|
                                |             |
                                |- ON --------|
                                |             |
                                |- OFF -------|
```

<trainid>   Any one of the mnemonics defined for the TRAINID file attribute in the "I/O Subsystem Reference Manual." Examples: EBCDIC96, ASCII72.

The PRINT command is used to write all or part of the workfile (or all of the TEACH file) to a printer file.

The mode of printing is selected by the PATCH, MERGED, FLAGGED, PAGED, or TEACH keywords:

The PATCH keyword prints all changed records in the workfile in a format similar to the COMPARE output of SYSTEM/PATCH.

The MERGED keyword prints the effective workfile.

The FLAGGED keyword prints the effective workfile and also prints any deleted or modified lines in their original form.

The PAGED keyword prints the effective workfile in paged format. Each page contains 55 lines of text, each line consisting of the text field of the record, followed by two blanks and the mark field. With the exception of the first page (considered page 1), each page is printed with a page number in the upper right-hand corner, beginning with page 2. No page number is printed on page 1.

The TEACH keyword prints the entire teach textbook.

If none of these keywords appear, PATCH is assumed if the Editor is in patch mode, and MERGED is assumed if the Editor is in merge mode (refer to "Patch Mode").

The <range list> specifies the portion of the file to be printed. If a value is not specified, the <range list> value defaults to the entire file. The RANGE specifications established by the RANGE command do not apply to the PRINT command.

TRAINID

The TRAINID option specifies the desired trainid for the printer file. If this option is not specified, the Editor allows the TRAINID attribute to default for all FILEKINDs except those for which the Editor's default is EBCDIC96.

GRAPHICS

The GRAPHICS option improves the mapping between ASCII (terminal) graphics and EBCDIC (printer) graphics. The following shows the characters that differ between ASCII and EBCDIC and the corresponding graphics that are printed with GRAPHICS ON and OFF:

Terminal Character	EBCDIC96 Printer	EBCDIC96 Printer
	with GRAPHICS ON	with GRAPHICS OFF
right brace (})	right brace	exclamation point
split bar (¦)	vertical bar	lozenge
exclamation point (!)	exclamation point	vertical bar
grave accent (')	lozenge	center dot
tilde (~)	cent sign	cent sign
rubout/DEL (■)	square spot	question mark
nongraphics	center dot	question mark

If GRAPHICS is selected and no TRAINID is specified, the TRAINID EBCDIC96 is used. For PRINT TEACH, if no TRAINID is specified, GRAPHICS and TRAINID=EBCDIC96 are used.

UPPERCASED

The UPPERCASED option causes the Editor to translate all lowercase letters to uppercase before printing.

UNNUMBERED

The UNNUMBERED option causes the Editor to leave records that are unnumbered in the workfile unnumbered on the printer output. If UNNUMBERED is not specified, the Editor automatically assigns temporary sequence numbers to all unnumbered records as they are printed (refer to the MAXIMUM and MINIMUM options under the OPTIONS command for a description of the automatic numbering facility). These temporary numbers are discarded after the PRINT command. UNNUMBERED is invalid for PRINT TEACH.

WITHSEQUENCENUMBERS

The WITHSEQUENCENUMBERS option is valid only for PRINT PAGED operations. If WITHSEQUENCENUMBERS is specified, the sequence field of each record will be printed on the far right-hand side of the listing.

COPIES

The COPIES option causes <integer> number of copies to be printed. Each copy is printed starting from the top of the page. If the COPIES option is not specified, one copy is printed.

ENVIRONMENT

For a PRINT PATCH operation (either explicit or default), the ENVIRONMENT option will cause environment boundaries to be displayed around each changed region in the workfile. ENVIRONMENT ON, or just ENVIRONMENT, activates the option; ENVIRONMENT OFF deactivates the option. If ENVIRONMENT does not appear, it defaults to the current value of the ENVIRONMENT <xref option>. If ENVIRONMENT is ON (either explicitly or by default) and if an attempt has not previously been made to load xreffiles, the Editor will attempt to load them (refer to the LOAD command). The ENVIRONMENT option is valid only for PRINT PATCH operations.

See also

FILEKINDs and Record Formats	29
LOAD Command	131
OPTIONS Command	82
Patch Mode	32
Xref Options	109

LOAD Command

```
 -- LOAD --- Codefile ----<filename>-----------------------------------|
          |             |
          |- Errorfile -|
          |             |
          |- Xreffiles -|
```

The LOAD command directs the Editor to interpret certain special files.

CODEFILE

"LOAD CODEFILE <filename>" causes the Editor to search for a file of the specified <filename> and access the lineinfo for use with the "GO <rcw spec>" and "WHERE <rcw spec>" commands. The codefile must have been compiled with the LINEINFO compiler control option set to TRUE.

ERRORFILE

"LOAD ERRORFILE <filename>" causes the Editor to search for a file of the specified <filename> and interpret the file as a compiler error file (a file that contains syntax error messages generated during program compilation). Information from the errorfile is used for the "GO +/- ERROR" command. When the first "GO +/- ERROR" command is entered, the Editor automatically attempts to load a file called "ERRORFILE" if an errorfile has not previously been loaded. If this file is not found, the message "No errorfile loaded" will be displayed.

To generate an errorfile to be loaded by the Editor, compile the program with the ERRLIST compiler control option set to TRUE and with the following file equation associated with the compile task:

```
    COMPILER FILE ERRORFILE(KIND=DISK,PROTECTION=SAVE)
```

(For COBOL74SYMBOL and PASCALSYMBOL, use ERRORS in place of ERRORFILE and ERRORLIST in place of ERRLIST.) If desired, the TITLE attribute can also be changed in the file-equated attribute list. If the title of the file is not ERRORFILE, the file must be loaded explicitly (the Editor will not be able to find the file by default).

XREFFILES

"LOAD XREFFILES <filename>" causes the Editor to search for and load cross-reference files (xreffiles) for the specified <filename>. Normally, the Editor will load xreffiles automatically whenever the first cross-referencing command is entered (that is, the Editor infers a LOAD XREFFILES command, using the title of the workfile or, if in patch mode, the title of the symbol file as the <filename>). If xreffiles are not found automatically or if xreffiles are to be loaded for a file other than the workfile, a LOAD command may be required.

The usercode and family name are stripped off the <filename> and are reinserted in the appropriate locations during the file search operation. Whether performing an automatic load operation using the default <filename> or a user-requested load operation using the <filename> specified in the LOAD command, the Editor performs the following search sequence to find the xreffiles:

1. The Editor searches for a file called "XREFFILES/<filename>/DECS".
2. If this file is not present, then if the first <name> in the <filename> is SYMBOL, the Editor searches for a file with SYSTEM as the first <name> (for example, if the <filename> is SYMBOL/ABC, the Editor searches for XREFFILES/SYSTEM/ABC/DECS).
3. If this file is not present, then the Editor searches for a file called "XREFFILES/OBJECT/<filename>/DECS".
4. If this file is not present, the Editor replaces the first <name> in the <filename> by OBJECT and searches for the file (for example, if the <filename> is SYMBOL/ABC, the Editor searches for XREFFILES/OBJECT/ABC/DECS).
5. If this file is not present, the Editor reports an error.

If the search is successful, the REFS file of the corresponding name is searched for. If the REFS file is not present, an error is reported.

A LOAD XREFFILES command can be entered at any time. However, the command discards the current xref target (and RECENT xref targets).

See also
- Cross-Referencing Commands. 107
- GO Command. 47
- RECENT Command. 120
- WHERE Command 121

SAVE Command

```
-- SAve ------------------------------- AS --<filename>--------------->
         |          | |              |
         |- Patch --| |-<range list>-|
         |          |
         |- Merged -|

>---------------------------------------------------------------------|
  |                   |
  |- : -- Unnumbered -|
```

The SAVE command writes all or part of the current workfile to a new file and continues the Editor session. If PATCH is specified, changed records are written to the new file in patch form. If MERGED is specified, the effective workfile is written to the new file (refer to "Patch Mode"). If neither is specified, PATCH is assumed if the Editor is in patch mode, and MERGED is assumed if the Editor is in merge mode.

The <range list> specifies the portion of the file to be written. If a value for <range list> is not specified, it defaults to the entire file. RANGE specifications established in the RANGE command do not apply to the SAVE command.

The <filename> specifies the name of the file to which the workfile records are to be written. If the file already exists, it will be removed and replaced by the new saved file.

The UNNUMBERED option causes the Editor to leave records that are unnumbered in the workfile unnumbered in the new file (the sequence number fields of these records are assigned all blanks). If UNNUMBERED is not specified, the Editor will automatically assign sequence numbers to all unnumbered records as they are written to the new file (refer to the MAXIMUM and MINIMUM options under the OPTIONS command for a description of the automatic numbering facility).

See also
    OPTIONS Command 82
    Patch Mode. 32

WHAT Command

```
 -- WHAT -------------------------------------------------------------|
```

The WHAT command displays information about the current Editor session, the workfile, and the Editor itself (such as the Editor version).

SEQCHECK Command

```
 -- SEQCHeck ---------------------------------------------------------|
```

The SEQCHECK command causes the Editor to scan the workfile forward (starting from the offered line), stopping on the first unnumbered record or the first record whose sequence field is not greater than the sequence field of the previous record.

As it searches, the Editor displays its current search location on the command line every two or three seconds. Any input received during the search process will terminate the search.

Note that the Editor scans the entire effective workfile (including the unchanged portions of the symbol file, if any). The "GO +/-UNNUMBERED" command is a more efficient way to locate unnumbered records in the changed portions of the file.

See also
    GO Command. 47

4 CONTROL KEY SYNONYMS FOR EDITOR COMMANDS

TD830, MT983, MT987, ET1100, and SR100 terminals are capable of transmitting special control character sequences to the system. Several of these sequences, of the form

```
-- |CTRL| --<digit>--<digit>-- |XMT| --|
```

are defined as synonyms for some Editor commands. These control sequences can be entered at any time. The cursor position is irrelevant; the cursor will not move while you enter the control sequence, nor will the two <digit>s be displayed on the screen.

Any characters that are entered between a transmit operation and a control sequence are lost (that is, if you type 10 characters for entry into the text field of the offered line and then enter "CTRL 88", the 10 characters will be ignored and the "CTRL 88" command will be performed).

The following control key sequences are recognized as Editor command synonyms:

CTRL	COMMAND	CTRL	COMMAND
00	REFRESH PAGE		
09	UNDO		
20	CANCEL MOVE and CANCEL COPY	30	CANCEL DELETE and CANCEL RENEW
21	MOVE FIRST	31	DELETE FIRST
22	MOVE LAST	32	DELETE LAST
23	MOVE 1	33	DELETE 1
		34	RENEW VICINITY
25	COPY FIRST	35	RENEW FIRST
26	COPY LAST	36	RENEW LAST
27	COPY 1	37	RENEW 1
28	MOVE BEFORE or COPY BEFORE	38	RENEW BEFORE
29	MOVE AFTER or COPY AFTER	39	RENEW AFTER

CTRL	COMMAND	CTRL	COMMAND
40	CANCEL NUMBER and CANCEL RESEQUENCE	50	CANCEL PARAGRAPH, CANCEL SHIFT, CANCEL CENTER, and CANCEL CHANGE
41	NUMBER FIRST	51	PARAGRAPH FIRST
42	NUMBER LAST	52	PARAGRAPH LAST
43	NUMBER 1	53	PARAGRAPH 1
44	NUMBER REGION	54	PARAGRAPH *
45	RESEQUENCE FIRST	55	SHIFT FIRST
46	RESEQUENCE LAST	56	SHIFT LAST
47	RESEQUENCE 1	57	SHIFT 1
48	RESEQUENCE REGION		
49	NUMBER ALL	59	CENTER 1
70	GO BEGINNING	80	GO END
71	GO -ALTERATIONS	81	GO +ALTERATIONS
72	GO -CHANGEDTEXT	82	GO +CHANGEDTEXT
73	GO -UNNUMBERED	83	GO +UNNUMBERED
74	GO -ENVIRONMENT	84	GO +ENVIRONMENT
75	GO -ERROR	85	GO +ERROR
76	-6	86	+6
77	-11	87	+11
78	-22	88	+22
79	GO -FIND (if the find target is not null) or GO -REFERENCE (if the find target is null)	89	GO +FIND (if the find target is not null) or GO +REFERENCE (if the find target is null)

90	REFRESH LINE	
91	OPTIONS RELATIVE ON	if RELATIVE is currently OFF
	OPTIONS RELATIVE OFF	if RELATIVE is currently ON
92	CHANGE 1 : MARK	
93	JOIN	
94	OPTIONS DENV ON	if DENV is currently OFF
	OPTIONS DENV OFF	if DENV is currently ON
96	COLUMN ON	if COLUMN is currently OFF; the <column> is set to the column in which the Editor most recently placed the cursor
	COLUMN OFF	if COLUMN is currently ON
97	OPTIONS DUPLICATE ON	if DUPLICATE is currently OFF
	OPTIONS DUPLICATE OFF	if DUPLICATE is currently ON

5 INSTALLING THE EDITOR

The Editor files are available as

```
Symbol file:      SYMBOL/EDITOR
Textbook file:    SYMBOL/EDITOR/TEXTBOOK
Code file:        SYSTEM/EDITOR
```

Because the Editor is meant to be run using the CANDE "UTILITY" command, it is suggested that the Editor code file be copied to disk under some simple name to minimize typing. For example, if the Editor code file is copied to disk under the simple name OBJECT/ED, the Editor could be run using the following CANDE UTILITY command:

```
U ED
```

The Editor will create a file called EDITOR/OPTIONS on the primary pack family of each usercode that runs the program. User-specified options used by the Editor are stored in this file (refer to the OPTIONS command). Removing the EDITOR/OPTIONS file will result in the loss of the saved option information for that user. Other files may also be created and removed by the Editor at various times. These files include the Editor's recovery file (EDITOR/RECOVERY/<workfile name>), which is described under "Recovery Files."

The Editor has been designed to take advantage of the special characteristics of TD830, MT983, MT987, ET1100, and SR100 terminals. The program will function correctly only when run on these terminals and only when they are properly configured for use with the Editor. The following information is intended to assist installation management personnel in configuring terminals for use with the Editor.

The Editor uses the standard NDL request sets and NDLII algorithms for TD terminals released in the example files SYMBOL/SOURCENDL and SYMBOL/SOURCENDLII. Special request sets and algorithms are not required by the Editor.

The Editor assumes that the terminal is configured for 80 characters per line and 24 lines per page. TD830 firmware level 1 is not supported.

During initialization, the Editor unconditionally takes the terminal out of scrolling mode by displaying and retransmitting the characters "?-S" in the upper left-hand corner of the screen. The data communications subsystem responds with the characters ".ok.". The Editor then erases both sets of characters from the screen. During normal program termination, the Editor unconditionally places the terminal in scrolling mode by displaying and retransmitting the characters "?+S". If the Editor terminates abnormally, the terminal will be left out of scrolling mode.

The Editor reads the length of the data communications buffer from the terminal's memory and resizes its output buffers accordingly. For some transmissions to the terminal, the Editor may need to send more characters than the terminal's buffer can hold. In these cases, the Editor will send the output as two transmissions instead of one. For such terminals, a short delay may be noticed between the two transmissions required to fill the screen.

The Editor assumes that the US (right-pointing triangle) and RS (left-pointing triangle) characters are used as the left and right forms mode delimiters, respectively. The terminal is correctly configured if register 0086 contains the value "1F" and register 0087 contains the value "1E". If any other characters are present in these terminal registers, the Editor will not enter forms mode correctly.

The terminal's full-line-transmit option may be enabled and disabled using the TERM command. Because this feature temporarily alters the terminal's firmware and the behavior of the Editor, users should be aware of the setting of this option. Refer to the discussions of the TERMINAL command and "Use of the Terminal" for additional details. If the terminal is in line-at-a-time transmission mode (bit 7 of register 00A0), the Editor will not function correctly.

See also
OPTIONS Command 82
Recovery Files. 34
TERMINAL Command. 100
Use of the Terminal 16

UNDERSTANDING RAILROAD DIAGRAMS

WHAT IS A RAILROAD DIAGRAM?

A railroad diagram is a way of representing the syntax of a command or statement graphically. It shows which items are required or optional, the order in which they should appear, how often you can repeat them, and any required punctuation.

HOW TO READ A RAILROAD DIAGRAM

Normally, you read a railroad diagram from left to right. However, there are some exceptions; in those cases, arrows indicate a right-to-left direction.

If a diagram is too long to fit on one line and must continue on the next, a right arrow (>) appears at the end of the first line and another at the beginning of the next line, like this:

```
----------------------------------------------------------------------->

>-----------
```

The end of a railroad diagram is denoted by a vertical bar (|) or percent sign (%). The vertical bar means the command or statement can be followed by a semicolon and another command or statement. The percent sign means the command or statement must be on a line by itself.

CONSTANTS AND VARIABLES

Consider a hypothetical command for giving instructions to a house painter:

```
-- PAINT ------------- LIVING ROOM ---<color>--|
   ---    |       | |               |
          |- THE -| |- DINING ROOM -|
                    |               |
                    |- BEDROOM -----|
                    |               |
                    |- BATHROOM ----|
                    |               |
                    |- KITCHEN -----|
```

This command tells the painter to paint a designated room in the color you specify.

The example introduces two important features of railroad diagrams:

- Constants
- Variables

Constants

Constants are items that you cannot vary. You must enter a constant as it appears in the diagram, either in full or abbreviated. If you abbreviate a constant, you must enter everything that is underlined in the railroad diagram, optionally followed by one or more of the remaining characters.

You can recognize constants in railroad diagrams by the fact that they are never enclosed in angle brackets.

In the example, the word PAINT is a constant. You could enter PAINT in full or abbreviate it to PAI or PAIN, but not to PA or PAN. If no part of the constant is underlined, you cannot abbreviate it at all.

Variables

Variables are items that you can replace with other data to suit a particular situation; that is, you can vary the information you enter in place of the variable, subject to rules defined for the particular command or statement.

Variables appear in a railroad diagram enclosed in angle brackets (<>).

In the example, <color> is a variable item. If the description of the PAINT command defines the allowable colors as BLUE, GREEN, PINK, and YELLOW, you can enter any one of these in your command.

FOLLOWING THE PATHS OF A RAILROAD DIAGRAM

The paths of a railroad diagram lead you through the diagram from beginning to end. They are represented by horizontal and vertical lines.

A path shows the allowable syntax. Some diagrams have just one path that goes from the beginning to the end of the diagram. Others contain several paths, each covering a part of the diagram. A path shows which items you can include in a command or statement, which you can omit, and the number of times you can include a particular item or group of items.

To follow a path through a railroad diagram, you need to understand the items you may encounter along the way. These items are

- Required items and punctuation
- Optional items
- Loops

A description of each item follows.

Required Items and Punctuation

Required items and punctuation must be entered in the command or statement; you cannot omit them. A required item appears by itself in a path (horizontal line). A required item can be either a constant or a variable. For example, if a railroad diagram indicates

```
 -- PAINT -- BEDROOM --<color>--|
```

the words PAINT and BEDROOM are required constants, and <color> is a required variable. You could correctly enter

```
    PAINT BEDROOM BLUE
```

but not

```
    PAINT BEDROOM
```

because the required item <color> would be missing.

Optional Items

Optional items appear one below another in a vertical list. You can choose any one of the items in the list. If the list also contains an empty path (all dashes), you can omit the item entirely. An optional item can be either a variable or a constant. The PAINT command in the example contains two lists. The first is

```
--------------|
  |       |
  |- THE -|
```

which gives you two options:

- Enter the constant THE
- Omit it (because there is an empty path)

The second list has five optional constants:

```
---- LIVING ROOM ----|
  |                |
  |- DINING ROOM -|
  |                |
  |- BEDROOM -----|
  |                |
  |- BATHROOM ----|
  |                |
  |- KITCHEN -----|
```

You must enter one of the optional items (LIVING ROOM, DINING ROOM, BEDROOM, BATHROOM, or KITCHEN) because there is no empty path in this list.

Loops

A loop is an item or group of items that you can repeat. The number of repetitions allowed is controlled by an item called the bridge.

A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, like this:

```
      |<------- <return character> ---------|
      |                                     |
    -----<bridge>--<content of the loop>-------
```

or

```
      |<-- <bridge> -- <return character> --|
      |                                     |
    --------- <content of the loop> -----------
```

The bridge shows the maximum number of times you can go through the loop. The bridge can precede the contents of the loop, or it can precede the return character on the upper line of the loop to specify the number of times the right-to-left path can be traversed. The bridge is an integer enclosed in sloping lines, / \, for example, /4\. Not all loops have bridges. Those that do not can be repeated any number of times.

The top line is a right-to-left path that contains information about repeating the loop. The return character is the character to use before each repetition of the loop (often, a comma). Not all loops contain a return character; if none is shown, just enter one or more spaces before repeating the loop.

The other lines show the content of the loop (the data you enter each time you go through the loop). This can be any combination of optional items, required items, lists, and even other loops. The content of a loop can range from simple (one item), to very complex (many items, lists, and loops).

Example 1. A Simple Loop

The PAINT command as first shown is of limited usefulness. To tell the painter to do several rooms, you need a separate command for each room. It would be much easier if you could tell him to do several rooms in one command.

You can do that by making the list of rooms into a loop. The command would then look like this:

```
                    |<---------- , ----------|
                    |                        |
-- PAINT -------------/5\--- LIVING ROOM -----<color>--|
           |       |       |               |
           |- THE -|       |- DINING ROOM -|
                           |               |
                           |- BEDROOM -----|
                           |               |
                           |- BATHROOM ----|
                           |               |
                           |- KITCHEN -----|
```

The bridge has a value of 5, so you can go through the loop up to five times, for a total of five rooms. The return character is a comma, which you must enter before repeating the loop content.

You can now enter

```
PAINT THE LIVING ROOM, BEDROOM, KITCHEN YELLOW
```

or

```
PAINT DINING ROOM, BEDROOM, BATHROOM BLUE
```

or

```
PAINT BEDROOM PINK
```

or

```
PAINT BEDROOM, BATHROOM, BEDROOM, BEDROOM BLUE
```

or any other valid combination.

This simple loop makes the PAINT command more versatile, but a significant drawback remains. Although you can include up to five rooms in a command, you cannot specify different colors.

Example 2. A More Complex Loop

If the content of the loop were to include the color, you could specify a different color for each room.

```
                     |<-------------- , -------------|
                     |                               |
 -- PAINT -------------/5\--- LIVING ROOM ---<color>----|
           |       |       |               |
           |- THE -|       |- DINING ROOM -|
                           |               |
                           |- BEDROOM -----|
                           |               |
                           |- BATHROOM ----|
                           |               |
                           |- KITCHEN -----|
```

The content of the loop now consists of the

- List of optional constants that indicate rooms

- Required variable <color>

The bridge value is 5, and the return character is a comma. Given this railroad diagram, some valid PAINT commands would be

PAINT THE BEDROOM PINK

PAINT THE LIVING ROOM BLUE, DINING ROOM GREEN, KITCHEN YELLOW

PAINT BEDROOM GREEN, KITCHEN BLUE

and so on.

Example 3. Another Loop

In some bridges an asterisk follows the number. For example,

```
                     |<-/4*\--------- , ---------|
                     |                           |
 -- PAINT --------------- LIVING ROOM ---<color>----|
           |       |   |               |
           |- THE -|   |- DINING ROOM -|
                       |               |
                       |- BEDROOM -----|
                       |               |
                       |- BATHROOM ----|
                       |               |
                       |- KITCHEN -----|
```

The asterisk means you must take the right-to-left path at least once. You cannot, for example, enter PAINT BEDROOM BLUE; you must tell the painter at least two rooms to paint. The maximum number of rooms to be painted is still five: the first time through the loop with up to four repetitions.

A valid form of the command would be

```
    PAINT BEDROOM BLUE, KITCHEN YELLOW
```

Example 4. Another Use of the Bridge

A bridge can also control the number of times you take an individual path within a loop. For example, another command to the painter might be:

```
 -- WORK -----------------------------|
          |                        |
          | |<-------------------| |
          | |                    | |
          |-----/1\- EVENINGS -----|
              |                |
              |-/1\- WEEKENDS -|
              |                |
              |-/1\- HOLIDAYS -|
```

Each bridge /1\ indicates you can take that path once or not at all. That is, you can enter each of the items EVENINGS, WEEKENDS, and HOLIDAYS once at most. Some valid commands are

```
    WORK EVENINGS WEEKENDS HOLIDAYS

    WORK WEEKENDS

    WORK HOLIDAYS EVENINGS
```

but

```
    WORK EVENINGS EVENINGS
```

is invalid.

A FINAL WORD

To familiarize you with railroad diagrams, this explanation describes the elements of the diagrams and gives a few simplified examples. Some of the actual diagrams you will encounter in a book may be considerably more complex.

However, the principles are the same no matter how complex the diagram. The more you work with railroad notation, the more easily you will understand even the most complex diagrams.

GLOSSARY

column mode

An edit-mode state in which the cursor is always placed in the same column when a line is offered (refer to the COLUMN Command).

command line

The top line of the edit-mode display, used by the Editor for error messages, pending command displays, information about the offered line, and other information (refer to "The Editor Display"). Editor commands may be entered on the command line without the leading "]" character.

CTRL (control)

A function key on the terminal that is used to generate control sequences for transmission to the system. Some control sequences have been defined as synonyms for Editor commands (refer to "Control Key Synonyms for Editor Commands").

cursor

The box-like character that marks the location on the screen where the next entered character will appear.

duplicate mode

An edit-mode state in which the text displayed for an insert line is copied from the line immediately preceding the insert line (refer to the DUPLICATE option in the OPTIONS Command).

edit mode

The state of the Editor in which it will accept Editor commands (as opposed to teach mode). Edit mode is the default mode.

Editor command

Any text entered on the command line or text entered on the offered line preceded by a "]".

find target

The text item to be searched for when a "GO +FIND" or "GO -FIND" command is entered (refer to the FIND command).

flag field

A column on the edit-mode display that appears just to the left of the text field of the displayed records. The flag field displays status information for each record (refer to "The Editor Display").

form

The portion of any Editor display that appears between one or more pairs of left and right forms-characters (refer to "Use of the Terminal").

forms-characters

Forms-characters are used to delimit the unprotected area of the display (the "form," described above). The left forms-character appears as a right-pointing triangular character, displayed at the left side of an Editor-displayed form; this character is also referred to as "US". The right forms-character appears as a left-pointing triangular character, displayed at the right side of an Editor-displayed form; this character is also referred to as "RS".

HOME

The uppermost, leftmost character position of an Editor form (refer to "Use of the Terminal").

insert mode

An edit-mode state of the Editor in which it offers a new line between two existing records for editing.

joining line

The line following the offered line when a JOIN command has been entered.

mark field

A portion of each record defined for use for mark information (refer to "FILEKINDs and Record Formats" and to the MARK command).

merge mode

An Editor state that results from executing the Editor without a symbol file, the absence of which indicates that the changes made during the Editor session are to be merged with the source file when an END or SAVE command is performed (refer to the "Patch Mode" section and to "patch mode" below).

offered line

The edit-mode form (refer to "form" above). The offered line (or lines, if the LINES option has been changed) is the only portion of the edit-mode screen in which data may be entered.

patch mode

An Editor state that results from executing the Editor with an additional parameter specifying the symbol file to which the workfile is to be considered a patch (refer to the "Patch Mode" section and to "merge mode" above).

sequence field

A portion of each record that is defined for use as a location in which to store record sequencing information (refer to "FILEKINDs and Record Formats").

SPCFY (Specify)

A function key on the terminal that transmits the cursor position to the system (refer to "Use of the Terminal").

split mode

An edit-mode state that results in a line being split into two lines each time the SPCFY key is pressed. (Refer to the SPLIT option in the OPTIONS command).

status line

The bottom line of the terminal screen (line 25), used by the Editor to display session-oriented information (refer to "The Editor Display").

teach mode

An Editor mode in which the Editor is displaying information from its teach textbook. Once in teach mode, a specific action is required to return to edit mode (refer to the TEACH command).

text field

A portion of each workfile record that is defined to contain the program text or other symbolic information (refer to "FILEKINDs and Record Formats").

XMT (Transmit)

A function key on the terminal that transmits entered characters to the system (refer to "Use of the Terminal").

xref

An abbreviation for "cross-reference," which refers to a table associating program identifiers with a list of sequence numbers of records on which the identifiers appear. This information is most often generated by a compiler.

xreffiles

Compiler cross-reference information stored in files of a form recognized by the Interactive XREF utility and by the Editor.

xref target

An identifier established as the item for which the cross-referencing commands are to display information (refer to "Cross-Referencing commands").

window

A conceptual aperture through which any screen-sized group of records in the workfile can be viewed. The window is said to move forward in the file if records of higher sequence numbers are being displayed and backward if records of lower sequence numbers are being displayed.

Index

Abbreviations, 39
ALGOLSYMBOL format, 29
ALIASES, XREF option, 109
ALL
   NUMBER command, 66
   RESEQUENCE command, 68
ALL command, 116
   Specify key, 17
ALTERATIONS
   go-next command, 48
Ampersand (@)
   flag field, 14
ASSIGN, XREF option, 109
Asterisk (*)
   flag field, 14
   PARAGRAPH command, 63
   REFERENCE command display, 113

Basic syntax, 19
BEGINNING, go-location command, 50
Blank, flag field, 14
BYE command, See END command

CANCEL command, 95
   pending commands, 36
CANDE UTILITY command, 7
CASED
   FIND command, 93
   OPTIONS command, 90
CDATA files, with use of the RENEW command, 76
CENTER command, 65
CHANGE command, 70
CHANGED, FIND command, 93
CHANGEDTEXT, go-next command, 48
Characters
   special, 25
   terminal, 129
COBOLSYMBOL format, 30
COBOL74SYMBOL format, 30
CODEFILE, LOAD command, 131
COLFOOTING, OPTIONS command, 89
COLHEADING, OPTIONS command, 89
Column, 19
   LIST command, 126
   PARAGRAPH command, 64
COLUMN command, 99
Column mode, 99
Command entering, 39
Command line, 11
Commands
   ALL, 116

Commands (cont.)
CANCEL, 95
CENTER, 65
CHANGE, 70
COLUMN, 99
Control key synonyms, 137
COPY, 59
cross-referencing, 107
DECLARATION, 115
DELETE, 58
display-changing, 41
editing, 53
END, 104
exiting, 103
FIND, 92
FORGET, 105
FORWARD, 115
GO, 47
go-location, 50
Go-next, 48
HELP, 124
INSERT, 71
INTERSECTION, 118
JOIN, 57
LABEL, 94
LIST, 126
LOAD, 131
MARK, 97
MERGE, 73
MOVE, 59
move-window, 45
NUMBER, 66
OPTIONS, 82
PARAGRAPH, 63
pending, 36
PRINT, 127
RANGE, 117
RECENT, 120
RECESS, 106
REFERENCE, 111
REFRESH, 79
RENEW, 75
RESEQUENCE, 68
restoring, 77
SAVE, 133
SEQCHECK, 135
SHIFT, 61
special-purpose, 123
State-changing, 81
TEACH, 124
TERMINAL, 100
UNDO, 78
UNION, 118

Commands (cont.)
  WHAT, 134
  WHERE, 121
  XREF, 107
Compiler control records, 32
  MERGE command, 73
Configuration, 139
Control key synonyms for Editor commands, 137
COPIES, PRINT command, 130
COPY command, 59
Correcting mistakes, 78
Cross-referencing commands, See XREF commands

DATA
  files, with use of the RENEW command, 76
  format, 31
DCALGOLSYMBOL format, 29
DECLARATION command, 115
DEFAULT, OPTIONS command, 83
DELETE command, 58
Delimiter, 19
  FIND command, 93
DENV, See DISPLAYENVIRONMENT, OPTIONS command
Destination, 27
  INSERT command, 71
  RENEW command, 75
DESTINATION, go-location command, 51
Digit, 19
  hex, 20
Display, 10
Display changing commands, 41
DISPLAYENVIRONMENT, OPTIONS command, 88
Duplicate mode, 90
DUPLICATE, OPTIONS command, 90

Edit-line operation, 54
Edit-mode screens, 10
Editing commands, 53
Editing records, 32
  MERGE command, 73
Editor
  display, 13
  ending, 103
  starting, 7
END command, 104
  RECESS and recovery files, 34
Ending Editor, 103
ENDING, go-location command, 50
Entering
  commands, 39
  data, See Edit-line operation

- Entering (cont.)
  - Editor, 7
- ENVIRONMENT
  - go-next command, 49
  - PRINT command, 130
  - XREF option, 110
- Equal sign (=)
  - ALL command, 116
  - flag field, 14
  - INSERT command, 71
  - LIST command, 126
  - MERGE command, 73, 111
  - reference display, 113
- ERROR, go-next command, 49
- ERRORFILE, LOAD command, 131
- Errors, correcting, 78
- ET1100, TERMINAL command, 100
- Exiting commands, 103
- External file, 35

- File
  - external, 35
  - patch, 32
  - recovery, 34
- File-Display lines, 13
- Filekind, 29
- Filename, 20
  - INSERT command, 71
  - LIST command, 126
  - LOAD command, 131
  - MERGE command, 73
  - SAVE command, 133
- Filetype, 29
- FILL command, See PARAGRAPH command
- FIND
  - command, 92
  - go-next command, 49
  - target, 92
- FIRST, 27
  - go-location command, 50
- Flag field, 14
- FLAGGED, PRINT command, 128
- FORGET command, 105
- Formats of records, 29
- FORTRANSYMBOL format, 29
- FORWARD command, 115
- Full-line-transmit, 16
  - TERMINAL command, 100
- FULL, TERMINAL command, 100

- GO command, 47

Index

Go-location command, 50
Go-next command, 48
Graphic, 20
Graphics, FIND command, 93
GRAPHICS, PRINT command, 129
Greater than sign (>)
   flag field, 14
   SHIFT command, 61
Group, 26
   CENTER command, 65
   CHANGE command, 70
   COPY command, 59
   DELETE command, 58
   MOVE command, 59
   NUMBER command, 66
   PARAGRAPH command, 63
   RENEW command, 75
   RESEQUENCE command, 68
   SHIFT command, 61

HELP command, 124
Hex digit, 20
Hex number, 20

Id, 21
   qualified, 22
Identifier, 21
INCLUDE command, See INSERT command
INDENT, PARAGRAPH command, 63
INSERT command, 71
Insert mode, 43
Insert-line operation, 43
   OPTIONS SPLIT command, 91
   Specify key, 17
Installation, 139
Integer, 21
   FIND command, 92
   move-window command, 45
   PARAGRAPH command, 64
   PRINT command, 130
   RESEQUENCE command, 68
INTERSECTION command, 118
   Specify key, 17

JOBSYMBOL format, 30
JOIN command, 57

LABEL command, 94
   Specify key, 17

Label id, 13, 21
  go-location command, 51
  LABEL command, 94
LAST, 27
  go-location command, 51
Left forms-character, See Right brace (})
LEFT, SHIFT command, 61
LEFTMARGIN, PARAGRAPH command, 63
Less than sign (<)
  flag field, 14
  SHIFT command, 61
Letter, 21
Line restoring, 79
Line-at-a-time transmission, 16
  effect on Editor, 140
LINES, OPTIONS command, 86
LIST command, 126
  Specify key, 17
LOAD command, 131

Margin spec, 63
  CENTER command, 65
  OPTIONS command, 83
  PARAGRAPH command, 63
MARGINS, PARAGRAPH comma:1d, 63
Mark, 22
MARK command, 97
MARK option
  CHANGE command, 70
  INSERT command, 71
  MERGE command, 74
MAXIMUM, OPTIONS command, 84
MERGE command, 73
Merge mode, 32
MERGED
  PRINT command, 128
  SAVE command, 133
MINIMUM, OPTIONS command 84
Minus sign (-)
  Editor display, 13
  go-next command, 48
  move-window command, 45
  PARAGRAPH command, 64
Mistakes, correcting, 78
Modes
  column, 99
  duplicate, 90
  insert, 43
  merge, 32
  multi-line, 86
  patch, 32
  scrolling, 140

- Modes (cont.)
  - split, 91
  - teach, 124
  - transmission, 16, 140
- MOVE command, 59
- Move-window command, 45
- MT983, TERMINAL command, 100
- MT987, TERMINAL command, 100
- Multi-line mode, 86

- Name, 20
  - file, 20
- NDLIISYMBOL format, 30
- NEWPSYMBOL format, 29
- NO
  - REFERENCE command, 112
- Null records, 32
  - MERGE command, 73
- NUMBER command, 66
- Number sign (#)
  - flag field, 14
  - REFERENCE command display, 113
- Number, hex, 20

- OFF
  - COLUMN command, 99
  - FIND command, 93
  - PRINT command, 130
  - RANGE command, 117
  - TERMINAL command, 100
  - XREF option, 109
- Offer-line operation, 42
  - Specify key, 17
- Offered line, 11
- ON
  - COLUMN command, 99
  - FIND command, 93
  - OPTIONS command, 88
  - PRINT command, 130
  - TERMINAL command, 100
  - XREF option, 109
- ONCE
  - DECLARATION and FORWARD commands, 115
  - REFERENCE command, 111
- OOPS command, 78
- Operations
  - edit-line, 54
  - insert-line, 43
  - offer-line, 42
  - split-line, 55
- Option, OPTIONS command, 82

OPTIONS command, 82

Page restoring, 79
PAGED, PRINT command, 128
Paging
  move-window command, 45
  Specify key, 17
PARAGRAPH command, 63
PASCALSYMBOL format, 31
PATCH
  PRINT command, 128
  SAVE command, 133
Patch mode, 32
Pending commands, 36
Plus sign (+)
  Editor display, 13
  go-next command, 48
  move-window command, 45
  PARAGRAPH command, 64
  REFERENCE command display, 113
Pound sign, See Number sign (#)
PRINT command, 127
PRINTER, XREF option, 110
PSI (Program Syllable Index)
  WHERE command, 121
PWI (Program Word Index)
  WHERE command, 121

Qualified id, 22
  DECLARATION and FORWARD command, 115
  INTERSECTION and UNION commands, 118
  REFERENCE command, 111

Railroad diagrams, explanation of, 141
RANGE command, 117
Range list, 24
  ALL command, 116
  INSERT command, 71
  INTERSECTION and UNION commands, 118
  LIST command, 126
  MERGE command, 73
  PRINT command, 128
  RANGE Command, 117
  REFERENCE command, 111
  SAVE command, 133
RCW
  go-location command, 51
  WHERE command, 121
RECENT command, 120
  Specify key, 17

Index

RECESS
   command, 106
   END command, 104
   Recovery file, 34
Record formats, 29
Recovery file, 34
   FORGET command, 105
REFERENCE command, 111
   display, 113
   go-next command, 49
   Specify key, 17
REFRESH command, 79
REGION
   NUMBER command, 66
   RESEQUENCE command, 68
REINITIALIZE, OPTIONS command, 83
Relative line, 26
RELATIVE, OPTIONS command, 87
RENEW command, 75
RESEQUENCE command, 68
Restoring commands, 77
Right brace (}), 14
   flag field, 14
   PRINT command, 129
Right bracket (])
   Editor commands, 39
RIGHT, SHIFT command, 61
RIGHTJUSTIFY, PARAGRAPH command, 63
RIGHTMARGIN, PARAGRAPH command, 63
RPGSYMBOL format, 30
Rulers, in OPTIONS command, 89

SAVE command, 133
Scroll-window command, See Move-window command
Scrolling mode, 140
SDI (Segment Dictionary Index)
   WHERE command, 121
SEARCH, See FIND
SEQCHECK command, 135
SEQDATA format, 29
Sequence number, 25
   Editor display, 11
   go-location command, 50
   RESEQUENCE command, 68
   WHERE command, 121
SHIFT command, 61
Shift specs
   OPTIONS command, 83
   SHIFT command, 61
SPCFY key, 17
Special character, 19, 25
Special-purpose commands, 123

Specify key, 17
Split mode, 91
Split-line operation, 55
   Specify key, 17
SPLIT, OPTIONS command, 91
SR100, TERMINAL command, 100
Starting Editor, 7
State-changing commands, 81
Status line, 15
SUMMARY
   INTERSECTION and UNION commands, 118
   REFERENCE command, 111
Synonyms for Editor commands, Control key, 137
Syntax, 19

TD830, TERMINAL command, 100
TEACH command, 124
   PRINT command, 128
   Specify key, 17
Teach mode, 124
Terminal
   character, 129
   identification, 7
   use of, 16
TERMINAL command, 100
Terminating Editor, 103
Text field, 14
TEXT, XREF option, 110
TEXTDATA format, 29
Tilde (~)
   Editor display, 13
   GRAPHICS option, 129
Top line, See Command line
Topic, in TEACH command, 124
TRAINID, 127
   PRINT command, 128
Transmit key, 16

UNDO command, 78
UNION command, 118
   Specify key, 17
UNNUMBERED
   END command, 104
   go-next command, 48
   LIST command, 126
   PRINT command, 129
   SAVE command, 133
UPPERCASED, PRINT command, 129

VICINITY, RENEW command, 75

Index

Warning message
   MERGE command, 73
   patch mode, 33
WHAT command, 134
WHERE command, 121
WITHSEQUENCENUMBERS, PRINT command, 129
WRITE command, See PRINT command

XMT key, 16
XREF commands, 107
XREF options, 109
   ALL command, 116
   INTERSECTION and UNION commands, 118
   REFERENCE command, 111
Xreffiles, 107
XREFFILES, LOAD command, 132

<, See Less than sign (<)

+, See Plus sign (+)

], See Right bracket (])

*, See Asterisk (*)

-, See Minus sign (-)

>, See Greater than sign (>)

#, See Number sign (#)

@, See Ampersand (@)

=, See Equal sign (=)

~, See Tilde (~)

}, See Right brace (})