B 5000 / B 6000 / B 7000 Series

EDITOR Textbook

Version 820416

1 INTRODUCTION

The Editor is a utility program designed to aid in the interactive creation and modification of program and data files. It is run on a TD830, MT983, 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 may be moved forward and backward in the file through the use of very 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 xref files for your use during the session, and 3) the Editor supports editing of very large files.

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.

1.1 STRUCTURE OF THIS DOCUMENT

The first portion of this document contains information that is designed to help orient new Editor users. 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.

The General Information section contains information that is fundamental to the operation of the Editor. This section serves as background for the description of the various commands.

The major portion of the document describes the syntax and semantics of each Editor command. 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.

A list of CTRL key synonyms for various Editor commands, a glossary of Editor terminology, and instructions for installing the Editor appear at the end of the document.

1.2 EDITOR CAPABILITIES

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 SPCFY.

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 (in terms of lines or pages) that the window is to be moved. The GO command is used to move 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 an xref target 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 (which is just moving the cursor to the line you wish to edit and pressing SPCFY). To insert a new line between two existing lines, use the insert-line operation (one form of which is pressing SPCFY on the first character of the offered line).

There are several commands provided for editing the contents of the workfile. The simplest is the edit-line operation, which consists of typing the new text on the offered line and then pressing 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 is used to delete 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 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, using the NUMBER command, or by user specification, using 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 are used to 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 are used to 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, DCALGOL, NEWP, and FORTRAN 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.

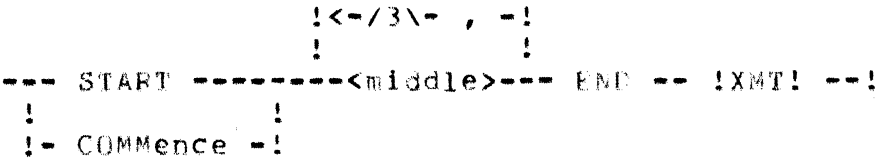
Other Editor commands include LIST, which lists an external file, and PRINT, which prints the workfile or the teach textbook.

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. Full recovery capability is provided by the Editor's recovery file mechanism.

1.3 SYNTAX NOTATION

The syntax of the Editor commands is expressed in "railroad" notation. This notation is explained and used in several Burroughs language manuals, including CANDE, ALGOL, and WFL. The railroad notation used in this document varies only slightly from the notation in those manuals. To avoid confusion, the notation used here is described below in its entirety.

A railroad diagram is a graphic representation of the syntax of a command or other form of input. Any sequence of tokens that can be generated by following the paths shown in the diagram is valid; any sequence of tokens for which a path does not exist is invalid. The following diagram is presented as an example for discussion:



The default direction for traversing a railroad diagram is from left to right; exceptions are noted by the explicit appearance of "<" characters. Beginning from the left of the diagram, the syntax allows a choice between one of two keywords, START and COMMENCE. The last four characters in the keyword COMMENCE are shown in lower case to indicate that they are optional; COMM is the minimum abbreviation, but COMME, COMMEN, COMMENC, and COMMENCE are acceptable alternatives.

The "<middle>" notation represents a metalinguistic variable, the syntax for which would be defined in another diagram if it were an actual language item. Any valid path through the diagram for <middle> is valid within the syntax described above. <middle> appears within a "loop", indicating that multiple instances of the token are allowed. The comma (,) in the loop indicates that a comma must separate each pair of <middle>s.

The "/3\" notation is a "bridge". The integer represents the maximum number of times the loop can be traversed (3, in this example, which has the effect of limiting the number of occurrences of <middle> to 4).

At the end of the example diagram, the keyword END appears as a required token, followed by another required token, "!XMT!". This notation is used to represent the pressing of a special key on the terminal, in this case the XMT (transmit) key.

The shortest valid sequence of tokens that is allowed by the syntax diagram above is the following:

START <middle> END !XMT!

The following is the longest valid sequence of tokens:

COMMENCE <middle>,<middle>,<middle>,<middle> END !XMT!

1.4 ENTERING THE EDITOR

The Editor is a system utility that is executed through CANDE using the CANDE "UTILITY" command. In order to 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 question

   ```
   Is this terminal a TD830 or an MT983 or an SR100?
   ```

   The appropriate responses are TD830, MT983, SR100, and NO. TD is allowed as an abbreviation for TD830, MT for MT983, 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 identifying the Editor and describing the current work file. Once initialization is complete, the Editor displays the first 23 lines of the work file and offers a line for editing. For an empty work file, the screen will be blank except for the following characters displayed on lines one and two 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 the XMT key. The leading right-bracket ("]") character tells the Editor that this input is a command and not new text for the file. For an empty work file, the screen will look like this just before the XMT key 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 the SPCFY key 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 program can be executed. If there is no current workfile, the Editor terminates with the message:

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

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

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

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

1.5 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
```

There are three types of lines 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 is used to display 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 from which to derive the number to be displayed, the characters "[~]" are used.

On the right side, there are three separate items 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 3 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 desired.

File-Display Lines

Aside from the command and status lines, all lines on the screen are used to 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 are used to 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

```
[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
  Pending Commands
  OPTIONS Command

1.6 USE OF THE TERMINAL

The Editor is designed to run on TD830, MT983, 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, there are some function keys and options that should be understood.

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-and-lower-case mode and CTRL Y to put the terminal in upper-case-only mode); it may also be used to enter the command synonyms listed in the "CTRL-Key Synonyms for Editor Commands" section.

XMT

The XMT key is used to transmit 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 the XMT key is dependent on the terminal configuration with respect to full-line-transmit mode. When the terminal is in full-line-transmit mode, pressing the XMT key causes the contents of the entire form to be sent to the system. When the terminal is NOT in full-line-transmit mode, pressing the XMT key causes all characters between the HOME position (the upper-most, left-most 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 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 the SPCFY key 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 the SPCFY key (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 xref 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 RET (return) 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 non-graphic characters to DEL (rubout) characters to protect the display from data comm control characters. The characters in the file are left intact, unless a displayed line is retransmitted as an edited line, in which case the DELs in that line are put into the workfile.

* See also
  REFRESH Command
  OPTIONS Command
  TERMINAL Command
  CTRL-Key Synonyms for Editor Commands
  Installing the Editor

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 doesn't do quite what you expected.

2.1 BASIC SYNTACTIC COMPONENTS

The syntactic items defined in this section are used in the specification of the syntax of one or more Editor commands. The items 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 (refer to FILEKINDs and Record Formats).

* See also
  FILEKINDs and Record Formats

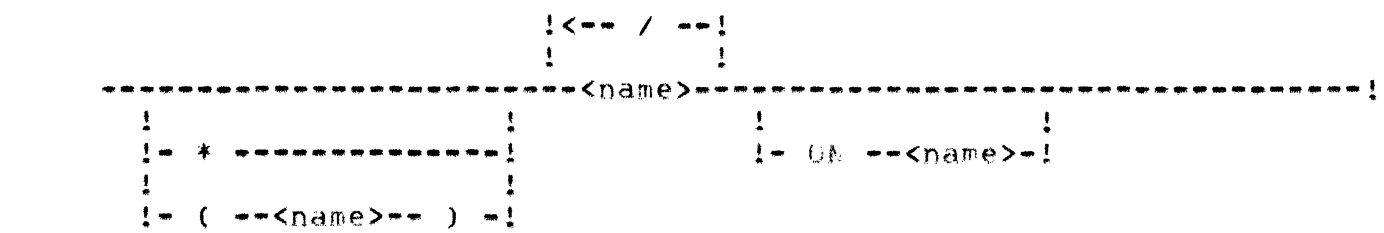
<delimiter>

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

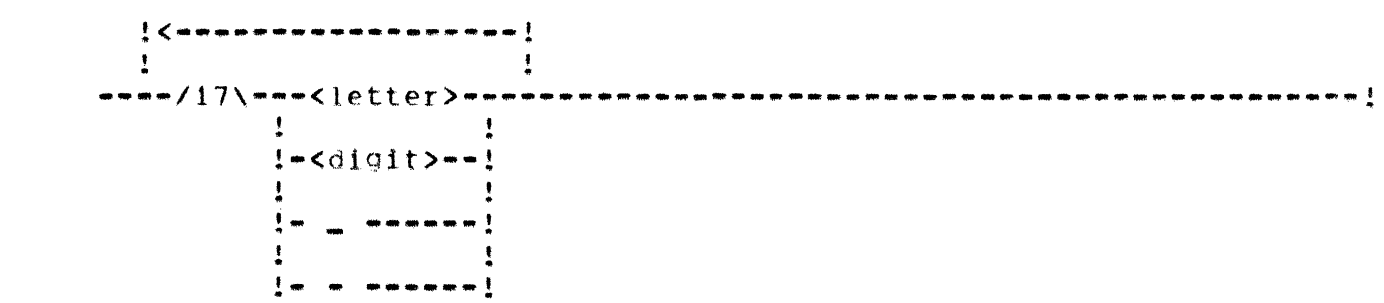
<digit>

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

<filename>



<name>



The Editor recognizes the basic set of Large Systems file names, including usercodes (in parentheses) and family names (following ON). A <filename> may contain up to 14 <name>s.

<graphic>

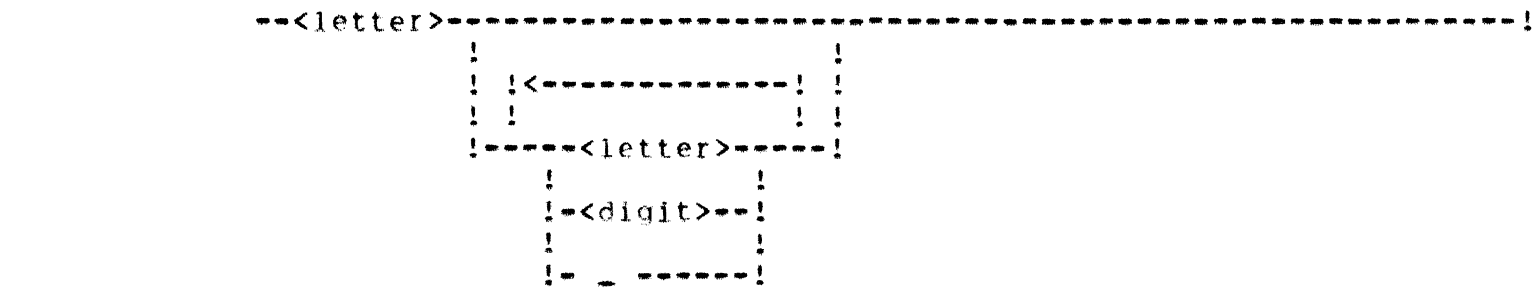
Any <letter>, <digit>, or <special character>.

<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>

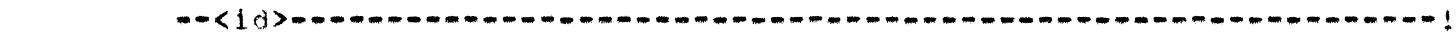
<id>



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

<integer>

<label id>



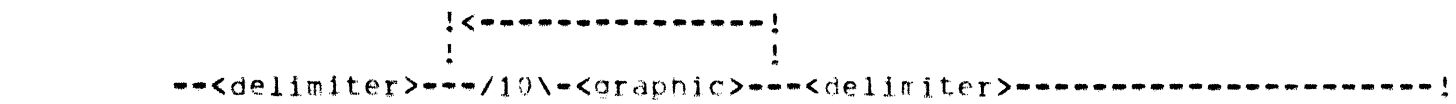
A <label id> is an <id> that can be associated with a record in the file (refer to the LABEL Command). <label id>s are case insensitive; that is, upper case and lower case letters are synonymous.

* See also
  LABEL Command

<letter>

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

<mark>

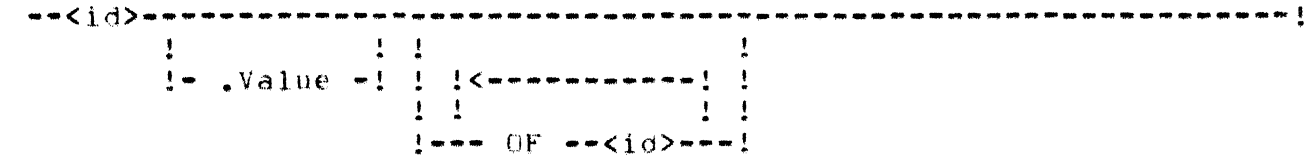


A <mark> is a value to be placed in the mark field of certain edited records. The conditions under which the <mark> is written into the mark field are described under the MARK command. The two <delimiter>s specified must be identical, and no <graphic> used may be identical to the <delimiter> used. The <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 lower case).

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
  MARK Command

<qualified id>



A <qualified id> is a name that the Editor recognizes as a program identifier from semantic information derived from cross-reference files (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).

<qualified id>s are 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 there is no program identifier of that name valid in the current environment, 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.

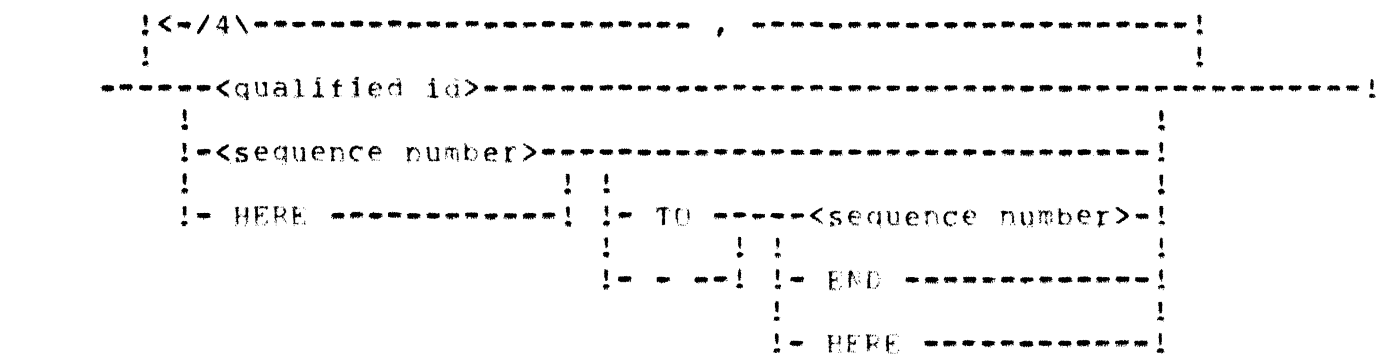
The "OF <id>" syntax qualifies the <id>; that is, it designates a particular instance of that program identifier by specifying the environment in which it was declared. All <id>s that follow the keyword "OF" must be procedure, module, or alternative module identifiers. Successive "OF" clauses are interpreted 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 procedure GET_NUMBER, which itself appears within the more global environment of the procedure SCANNER. 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 lower case is different from the same identifier entered in upper case); 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 B 7000/B 6000 Series System Software Operational Guide, Volume 1 (form number 5011661).

* See also
  FILEKINDs and Record Formats
  Cross-Referencing Commands
  LOAD Command

<range list>



A <range list> 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> may 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, 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 is used to convert 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 forms of <range list> are used to specify a range of sequence numbers. END refers to the sequence number of the last record in the file. HERE 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
  LOAD Command

<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

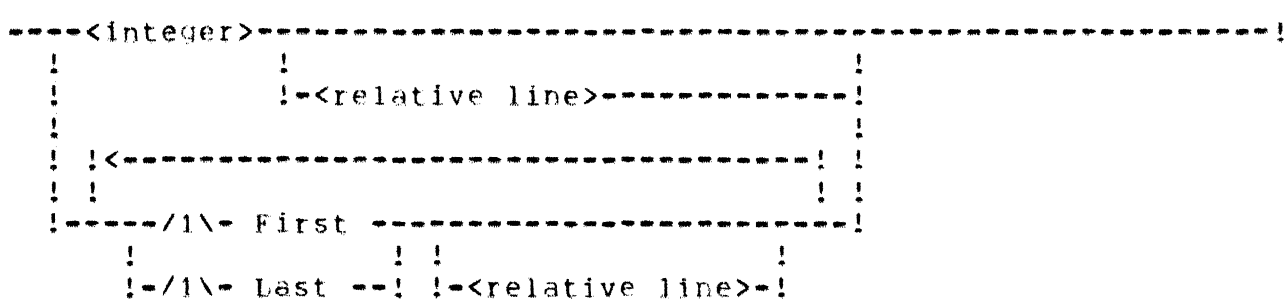
<special character>

Any one of the following characters:

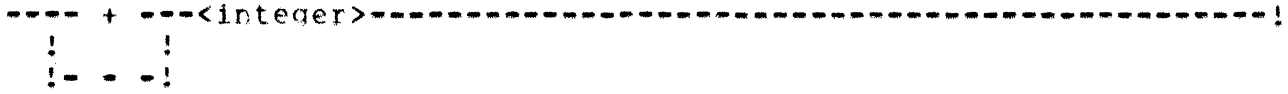
!	exclamation point	"	quote
#	number	$	dollar
%	percent	&	ampersand
'	apostrophe	(	open (left) parenthesis
)	close (right) parenthesis	?	DEL (rubout)
-	minus (hyphen)	=	equal
~	tilde	\	backslash
_	underscore	^	circumflex (carat)
@	at	`	grave accent
[	open (left) bracket	{	open (left) brace
;	semicolon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

2.2 GROUP AND DESTINATION SPECIFICATIONS

<group>



<relative line>



A <group> specification is used to designate the group of contiguous lines upon which the command being entered is to operate. The group may 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 "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
```

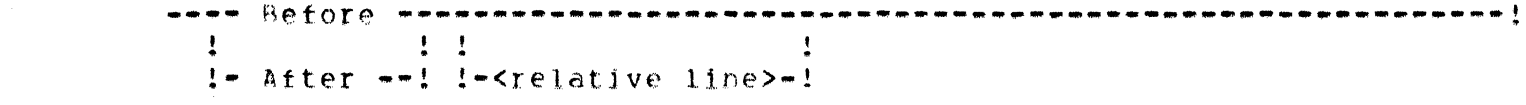
FIRST and LAST are used to designate the first and last lines of the group. Both can be specified in the same command or they can be specified one at a time. If they are specified independently, the operation is "pending" during the time that only one has been entered (refer to Pending Commands).

FIRST by itself indicates that the offered line is the first line of the group. LAST by itself indicates that the offered line is the last line of the group. Thus, entering FIRST on the offered line, then SPCFYing 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.

FIRST and LAST may 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>



A <destination> specification is used to designate the location at which lines are to be inserted. A <destination> is required for any MOVE, COPY, or INSERT command. A <destination> may 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
  Pending Commands
  OPTIONS Command

2.3 FILEKINDS AND RECORD FORMATS

The Editor is capable of displaying and editing 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 upper and lower case mode or upper-case-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 left-most 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	
Upper Case 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	
Upper Case 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		
Upper Case Only:	YES		
Other:	Hyphens (-) are considered non-delimiters 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		
Upper Case 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		
Upper Case Only:	YES		
Other:	The COLHEADING and COLFOOTING options are ON by default (refer to the OPTIONS Command).		

NDLITSYMBOL

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		
Upper Case Only:	YES		

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	
Upper Case Only:	NO	
Other:	The default <trainid> is EBCDIC96.	
	<qualified id>s are case sensitive.	

2.4 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", referring to the fact 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 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.

* See also
  FILEKINDs and Record Formats
  MERGE Command
  END Command
  FORGET Command

2.5 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, RECOVER, MAKE, or GET the same CANDE workfile that you were editing during the aborted session and 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.

The recovery file can also be preserved deliberately, through the use of 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 of the Editor while a recovery file for that workfile exists. Changing the workfile in CANDE, for example, could result in invalidation of the recovery file and 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 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.

* See also
END Command
RECESS Command

2.6 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
  MERGE Command
  LIST Command

2.7 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 <group>, 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
DELETE Command
MOVE and COPY Commands
SHIFT Command
PARAGRAPH Command
CENTER Command
NUMBER Command
RESEQUENCE Command
RENEW Command
CANCEL Command

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 non-blank 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 upper case, lower case, or a mixture of both. For example, the JOIN command may be entered as "JOIN", "join", "Join", or any other upper and lower case 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 upper and lower case characters. Characters appearing in upper case are required, and characters appearing in lower case 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 upper-and-lower-case 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 desired.

A few frequently-used commands are entered by pressing the SPCFY key instead of by entering a "]" followed by a command word. SPCFY 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 SPCFY operations require that the cursor be positioned in a specific location prior to the SPCFY; these requirements are described for each operation individually.

For some commands, there are CTRL-key synonyms (refer to CTRL-key Synonyms for Editor Commands).

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

DISPLAY-CHANGING COMMANDS, which change the edit-mode display but do not affect the workfile;

EDITING COMMANDS, which are used to change the text, sequence, and mark fields of the records in the workfile;

RESTORING COMMANDS, which restore all or part of the workfile or display to a previous state;

STATE-CHANGING COMMANDS, which change and display various session options and variables;

EXITING COMMANDS, which are used to terminate the Editor session;

CROSS-REFERENCING COMMANDS, which make use of interactive xref files;

SPECIAL-PURPOSE COMMANDS, which provide additional, non-editing functions.

* See also
  Use of the Terminal
  TEACH Command
  CTRL-Key Synonyms for Editor Commands

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

* See also
  The Editor Display

3.1.1 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 the SPCFY key. The Editor will offer the line, and the cursor will be placed in the column where it was when the SPCFY was done (if the cursor was to the left of the first non-blank character of the line, it will be placed on the first non-blank character). You may now edit the line.

Note: There are similar SPCFY operations that differ only in the placement of the cursor at the time the SPCFY key is pressed. If the cursor is not on the text field of a non-offered line, the operation will be interpreted as either an insert-line or a split-line operation.

* See also
  Use of the Terminal
  Insert-Line Operation
  Editing Commands
  Split-Line Operation

3.1.2 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. There are two forms of this operation, one that inserts a line after any displayed line and one that 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 the SPCFY key. To insert a line before the offered line, position the cursor in the text field, on or to the left of the first non-blank character, and press the SPCFY key. In response to either of these operations, the Editor will display an unnumbered line as the inserted, offered line. Normally, the inserted line will be blank; however, in duplicate mode, the line will be 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: There are similar SPCFY operations that differ only in the placement of the cursor at the time the SPCFY key is pressed. If the cursor is not in the proper location for an insert-line operation, the SPCFY will be interpreted as either an offer-line or a split-line operation.

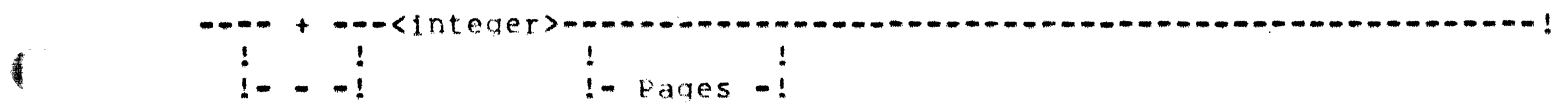
In the following example, the "!" character inoicates the location of the cursor:

```
      (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 the SPCFY key. The Editor offered an insert line after line 200.

* See also
  Use of the Terminal
  Offer-Line Operation
  Split-Line Operation
  OPTIONS Command

3.1.3 Move-Window Command



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.

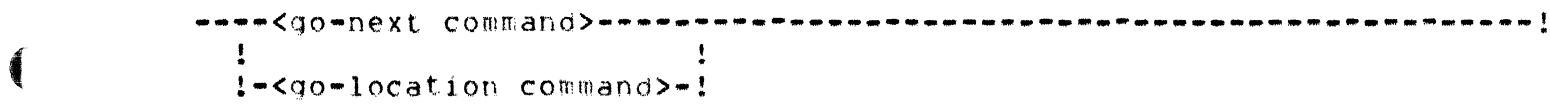
"]+<integer>" or "]-<integer>" scrolls the window forward or backward the number of lines specified by <integer>. If the offered line is scrolled off the page, the cursor is left on the command line.

"]+<integer> P" or "]-<integer> P" moves the window forward or backward the number of screen pages specified by <integer> (a screen page is 23 lines). The cursor is left on the command line.

Note: The "+/-<integer>" commands are valid in teach mode (refer to the TEACH command).

* See also
  GO Command
  TEACH Command

3.1.4 GO Command

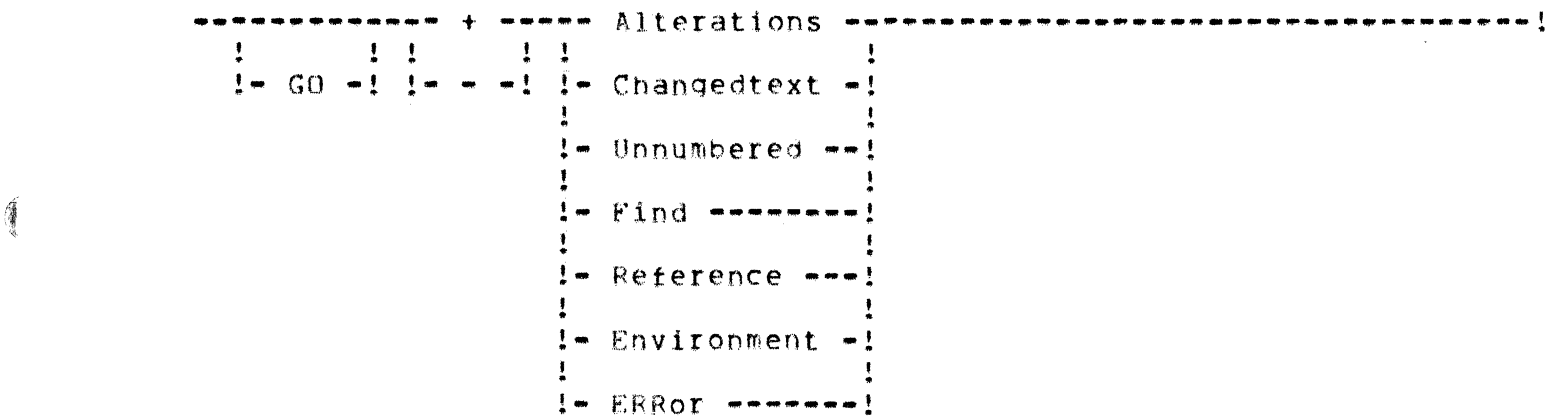


The GO command is the most important command for moving the offered line from one area of the file to another. The <go-next command> moves the offered line to the next occurrence (forward or backward) of some target. The <go-location command> moves the offered line to the specified location.

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" vs. "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>



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 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 cross-referencing commands (refer to Cross-Referencing Commands).

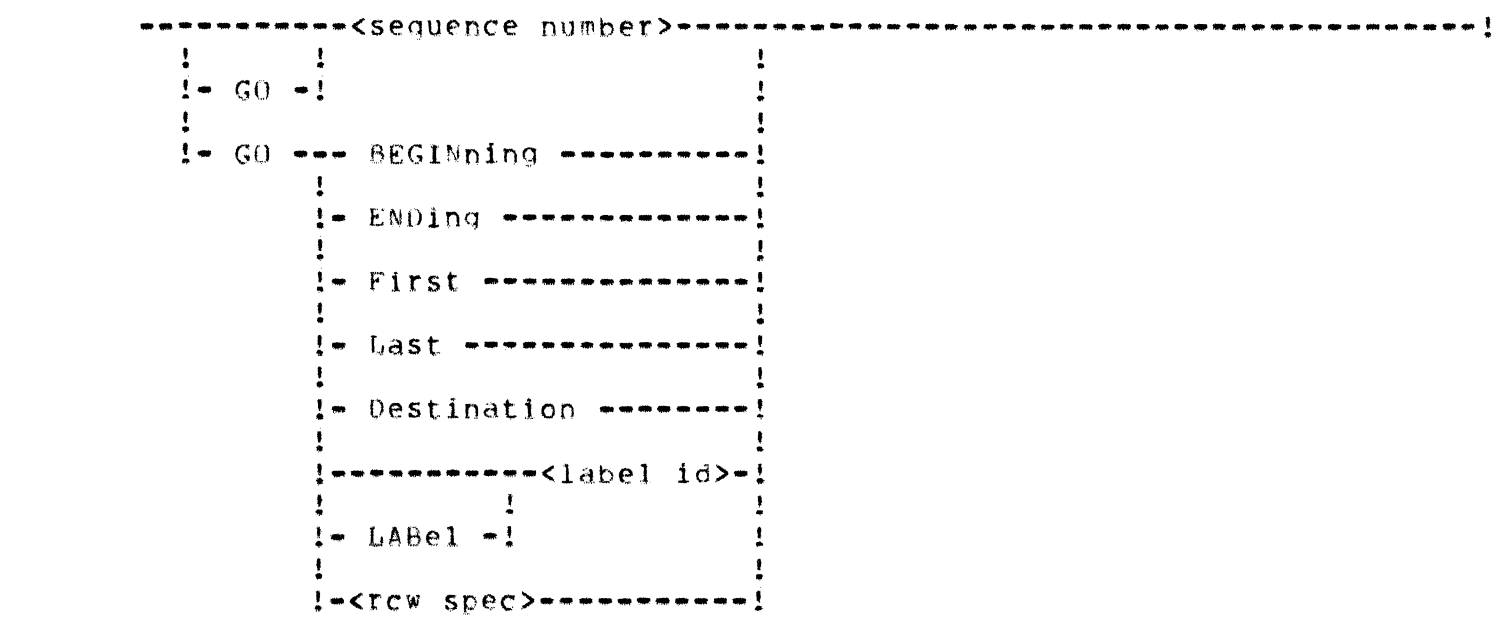
+/- ENVIRONMENT
"]+E" or "]-E" offers the line that is either at the end ("]+E") or at the beginning ("]-E") of the most local environment that contains the currently-offered line. This command is not valid unless xreffiles have been loaded, either automatically or by request (refer to the LOAD command).

+/- 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 will offer the line in error, placing the cursor where the compiler has indicated that the error occurred, and will display the error message on the command line.

<go-location command>



<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 codefile has been loaded (refer to the LOAD command).

* See also
  Pending Commands
  Editing Commands
  FIND Command
  LABEL Command
  Cross-Referencing Commands
  WHERE Command
  LOAD Command

3.2 EDITING COMMANDS

Editing commands are used to actually 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.

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

* See also
MARK Command

3.2.1 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 non-blank character, and does not have the station's data comm 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 data comm control record. That is, the input must not be entered on the command line, must not begin with a "]" as the first non-blank character, and must not have the station's data comm 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 the XMT key, 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
  Use of the Terminal
  Patch Mode

3.2.2 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 non-blank character of the original line. After the operation, the cursor is left in the same position as it was before the operation.

If SPCFY is pressed when the cursor is on one of the leading blanks of a line, on the first non-blank 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 "JUNDO" to restore the line.

Usage Note: 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
Offer-Line Operation
Insert-Line Operation
UNDO Command
OPTIONS Command

3.2.3 JOIN Command

```
-- Join ----------------------------------------------------------!
```

The JOIN command concatenates the text on the offered line with the text on the line following it (the "joining line"). The first non-blank character on the joining line, and all characters after it, are concatenated to the last non-blank 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, "-" also is 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 concatenated.

If the non-blank 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 right-most column that would have fit.

Usage Note: 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

3.2.4 DELETE Command

```
-- DELete --<group>-------------------------------------------------!
```

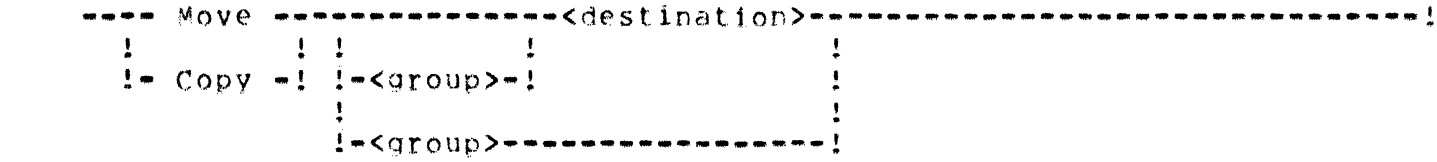
The DELETE command deletes the group of records specified by <group>. After the delete operation, the line following the deleted group is offered. If the last line of the file is deleted, the last remaining line of the file is offered.

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

3.2.5 MOVE and COPY Commands



The MOVE and COPY commands insert records from one part of the file (the source) at another location in the file (the destination). MOVE then deletes the records where they originally appeared; COPY 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.

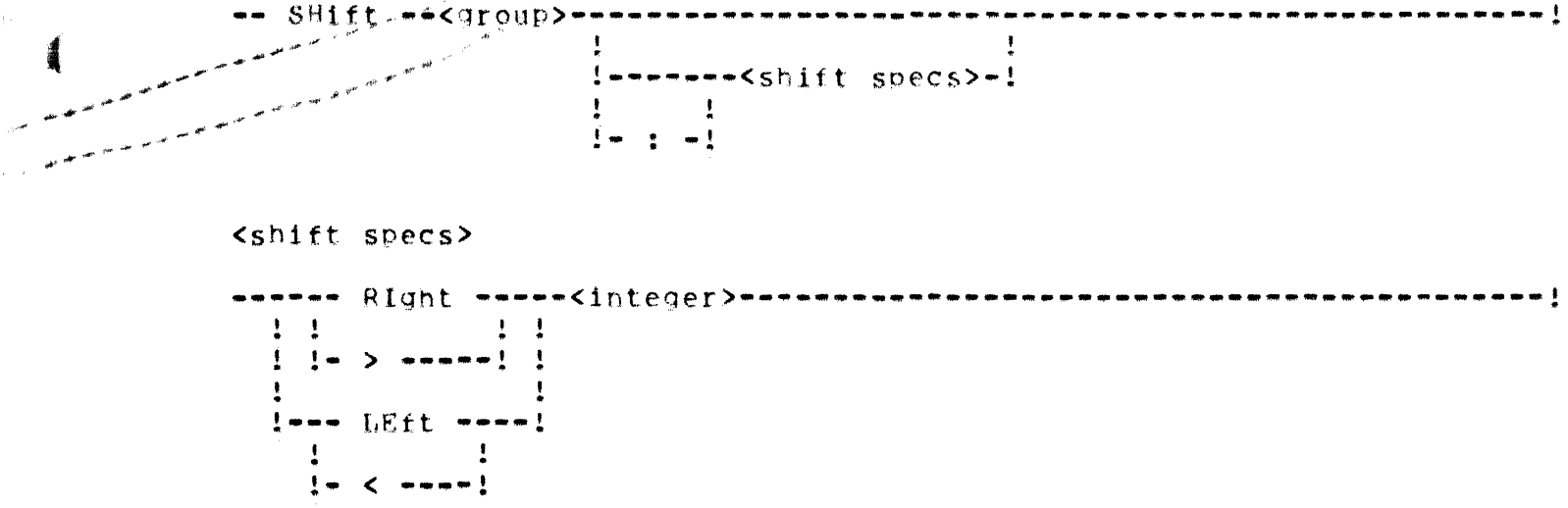
For example, "MOVE 1 B-2" causes the offered line to be inserted before the line that is two lines backward from the offered line, as illustrated below:

```
        (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
  Group and Destination Specifications
  Pending Commands
  GO Command

3.2.6 SHIFT Command



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). <shift specs> may be included in the command to override the current session shift specifications (refer to the OPTIONS command). <shift specs> 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.

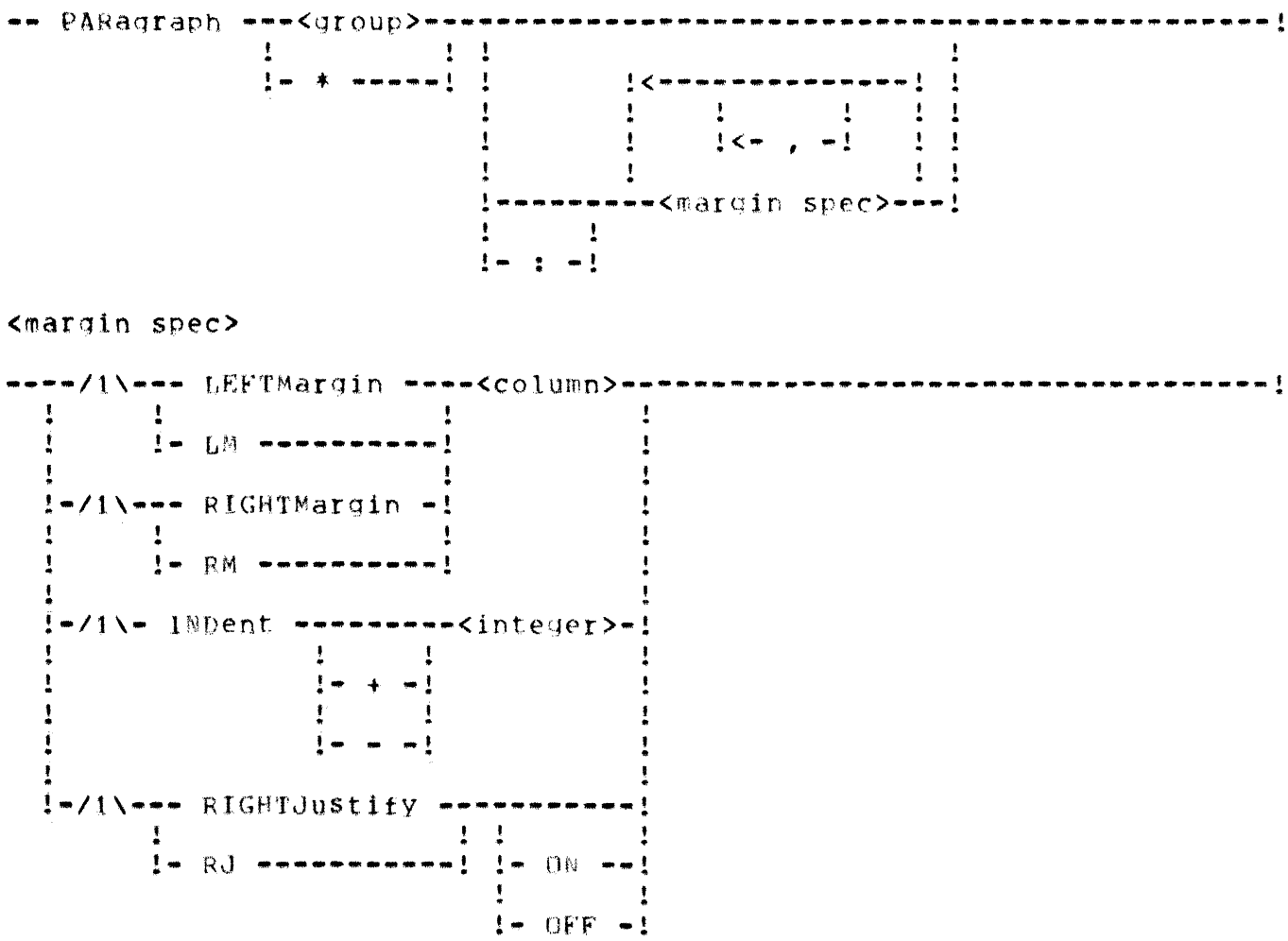
If a line is encountered that cannot be shifted the amount specified without losing non-blank characters, the line is offered for editing, and the SHIFT command is left pending with LAST specified, the direction and amount specified, and FIRST unspecified (refer to Pending Commands). The command is left 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
Group and Destination Specifications
FILEKINDs and Record Formats
Pending Commands
OPTIONS Command

3.2.7 PARAGRAPH Command



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 non-blank characters is considered a "word").

<group> 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.

The format of the paragraph is determined by margin specifications, which 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 may be included in the PARAGRAPH command to override the corresponding session margin specification(s) (refer to the OPTIONS command). <margin spec>s appearing 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
  Group and Destination Specifications
  FILEKINDs and Record Formats
  GO Command
  OPTIONS Command

3.2.8 CENTER Command

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

The CENTER command centers the non-blank 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 may be included in the CENTER command to override the corresponding session margin specification(s) (refer to the OPTIONS command). <margin spec>s appearing in a CENTER command apply only to that command.

Example:

```
          (before)                          (after)
   [000001001]                    [000001001]  2 records were CENTERED
   000100 SPECIFICATION           000100*       SPECIFICATION
   000200   TITLE                 000200*           TITLE
   000300?]center 2-2             000300?
```

* See also
  Group and Destination Specifications
  PARAGRAPH Command
  OPTIONS Command

3.2.9 NUMBER Command

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.

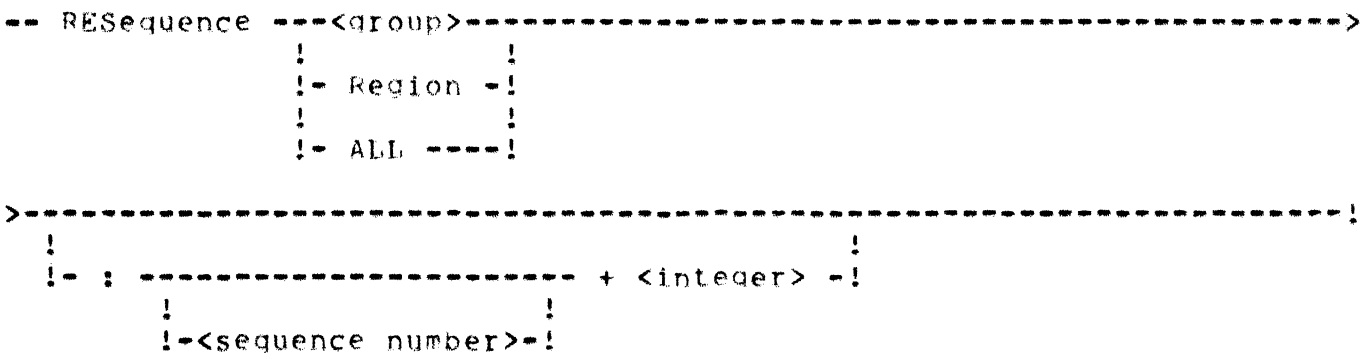
REGION
If REGION is specified, the group of unnumbered records that includes or immediately precedes the offered line is numbered. If the offered line does not appear in or immediately after an unnumbered region, the error message "No unnumbered text here" is displayed.

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
  RESEQUENCE Command
  OPTIONS Command

3.2.10 RESEQUENCE Command



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 for 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 is used to specify 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.

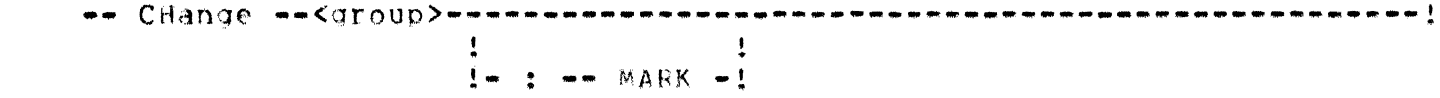
Unlike the NUMBER command, a RESEQUENCE command never extends the range of records it numbers. If the sequence number and increment are too large to allow the records to fit, the Editor displays the message "Doesn't fit" and requests a new number and/or increment. If you would like to cancel the RESEQUENCE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

3.2.11 CHANGE Command



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
  Group and Destination Specifications
  Patch Mode
  GO Command
  MARK Command

3.2.12 INSERT Command

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

If a <filename> is given, the records are inserted from the specified file and that file becomes the current external file (refer to External Files). If "=" is given, 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.

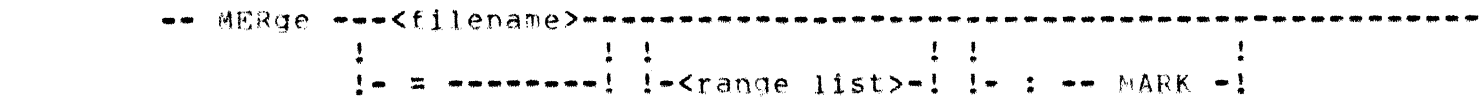
Inserted lines are considered "changed" lines, for purposes of the "GO +/-ALTERATIONS" and "GO +/-CHANGEDTEXT" commands.

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
Group and Destination Specifications
FILEKINDs and Record Formats
External Files
GO Command
MARK Command

3.2.13 MERGE Command



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 given, the records are merged from the specified file and that file becomes the current external file (refer to External Files). If "=" is given, 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 xretfiles 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
  Patch Mode
  External Files
  GO Command
  MARK Command

3.2.14 RENEW Command

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>
If a <destination> is specified, only the area between the lines surrounding the specified <destination> is renewed; the lines surrounding the <destination> are not renewed. Both lines must have sequence numbers.

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
```

* See also
  Group and Destination Specifications

3.3 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 command	Retracts the action of the previous command.
REFRESH command	Rewrites all or part of the Editor display.

* See also
  RENEW Command
  FORGET Command

3.3.1 UNDO Command

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.

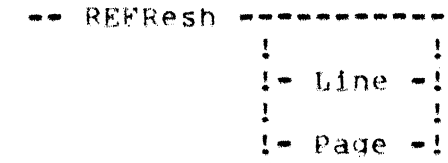
UNDO will retract the effects of all editing commands and of most other commands. It will not retract the establishment of a find or xref target or the creation of an external file (such as a SAVE, PRINT, or OPTIONS:SAVE command).

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
```

3.3.2 REFRESH Command



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.

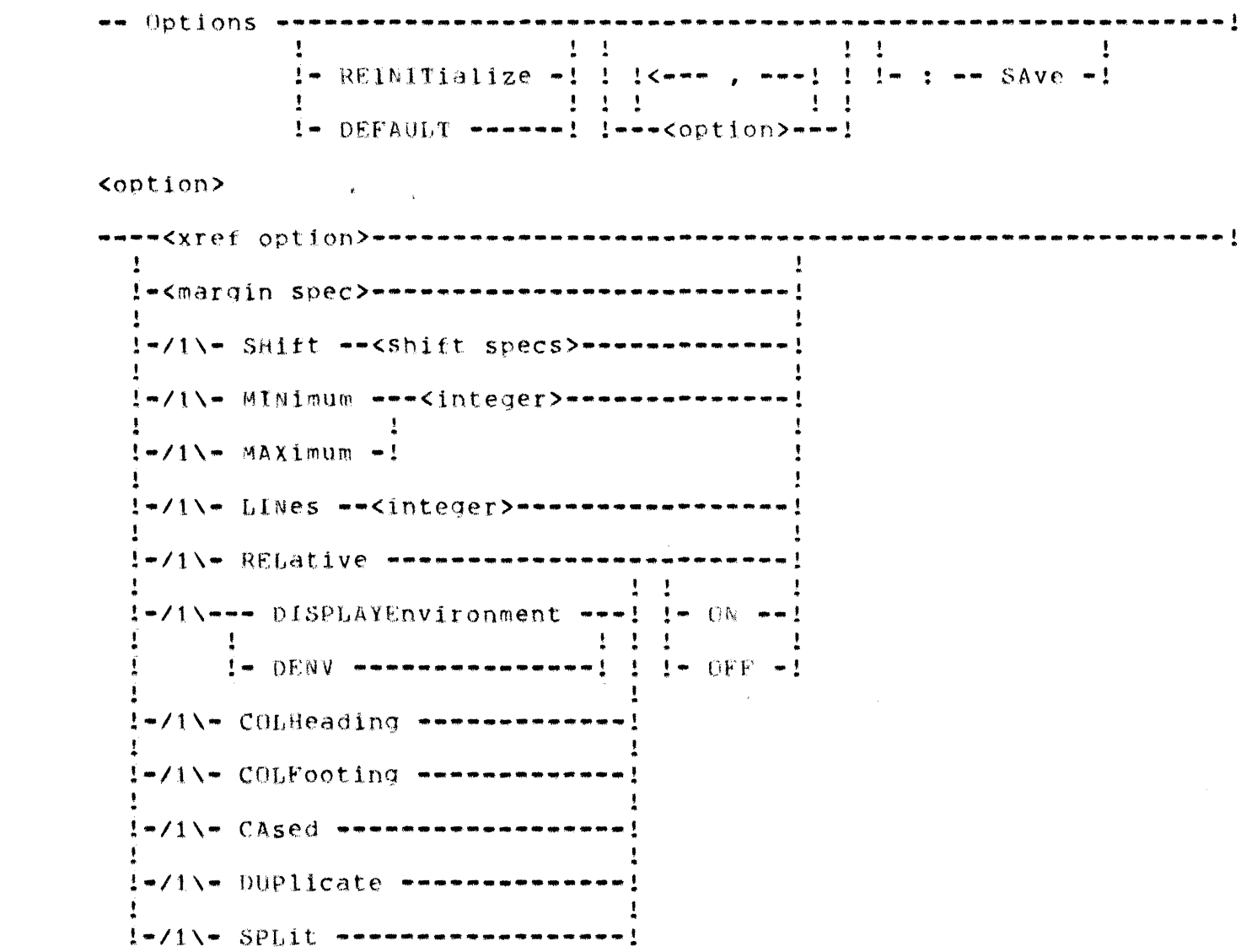
* See also
  UNDO Command

3.4 STATE-CHANGING COMMANDS

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

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

3.4.1 OPTIONS Command



The Editor maintains several state variables that you may change to tailor its displays or other actions to your requirements. There are two copies of these state variables (or simply "options") 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 is used to display or change 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	left-most column of the text field
RIGHTMARGIN	right-most 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 may 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

<xref option>

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

* See also
  Xref Options

<margin spec>

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

* See also
  PARAGRAPH Command

<shift specs>

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

* See also
  SHIFT Command

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, "JOPTIONS 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 will temporarily number all records in increments of MININC and will continue 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 will notify you that the numbered range was extended and will allow you to either accept (by pressing SPCFY) or reject (by pressing XMT) the numbering of that range. If you accept the extension, the Editor will assign the sequence numbers and will finish the operation; if you reject the extension, the Editor will discard the proposed numbers and return 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

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 may 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, "JOPTIONS 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 operation>. 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 may be entered on all lines of the multi-line display. All cursor controls (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 left-most character of the top line of the form is considered the HOME position.

3. The offered range of lines will move as a group, according to how many lines were transmitted. For example, if the number of lines displayed was 3 and the data were transmitted from the middle of the second line, the 3-line form would be moved forward 2 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
  Use of the Terminal
  Offer-Line Operation
  Insert-Line Operation
  GO Command
  Edit-Line Operation

RELATIVE

The RELATIVE option controls "relative mode". For example, "JOPTIONS RELATIVE" turns relative mode ON. When RELATIVE is OFF, as it is by default, the left side of the edit-mode screen is used to display the sequence number of each record. In relative mode, the sequence number area is used to display 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?jopt 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

DISPLAYENVIRONMENT

The DISPLAYENVIRONMENT option controls "environment mode". For example, "JOPTIONS 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 will automatically attempt to load xreffiles during initialization of the Editor session. This option has no effect if xreffiles are not loaded (refer to the LOAD command).

* See also
  Cross-Referencing Commands
  LOAD Command

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, "IOPTIONS 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?IOPT 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
  FILEKINDs and Record Formats
  COLUMN Command

CASED

The CASED option establishes a default value for the FIND command CASED parameter. For example, "JOPTIONS 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

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, "JOPTIONS 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 until and 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._________
 >>>> ?jopt dup on               >>>> ?1._________
000300                          000300
```

In this example, the offered line was an insert line before line 300, where the user entered "jopt 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 will offer another insert line, this time with the contents duplicated from the previously-inserted line (that is, "2._________"). This process continues until the user enters "jopt 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
  Insert-line Operation
  Edit-line Operation

SPLIT

The SPLIT option controls "split mode". For example, "JOPTIONS SPLIT" turns split mode ON. When in split mode, pressing the SPCFY key in the text field of a displayed line causes the line on which the cursor is positioned to be split at the specified location, 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 operations are affected as follows:

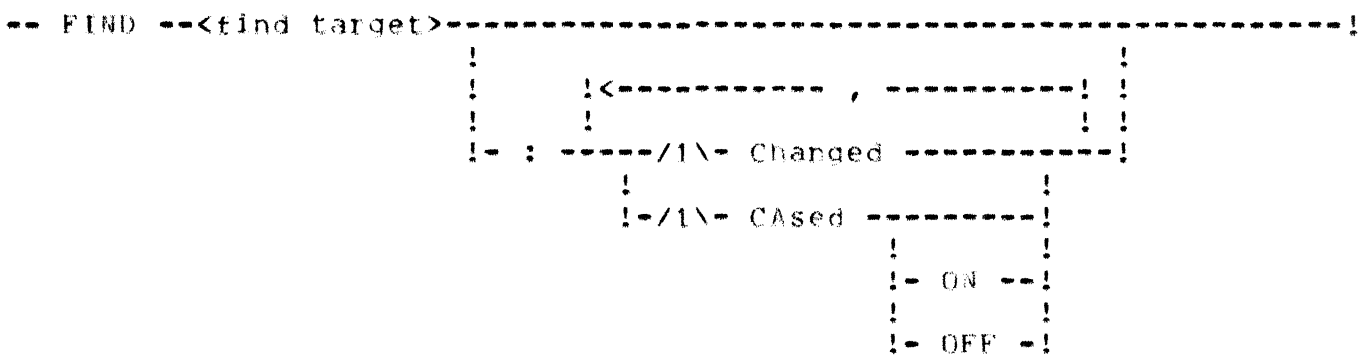
<offer-line operation>
The offer-line operation is not available in split mode.

<split-line operation>
This operation works normally in split mode.

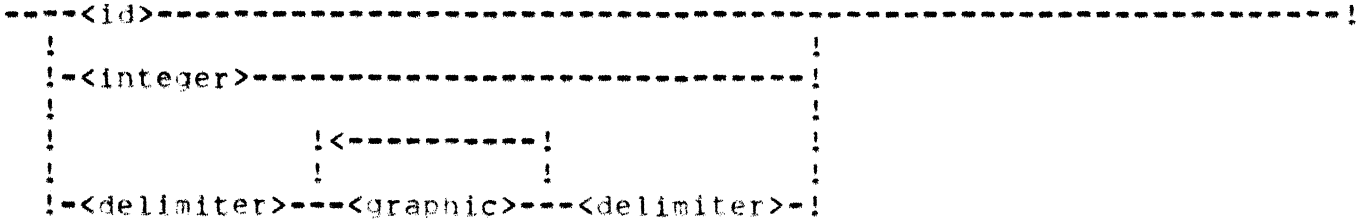
<insert-line operation>
The "insert-after" form of this operation works normally in split mode (that is, SPCFY on the sequence or flag field of a line will enter insert mode after that line). The "insert-before" form is not available in split mode (a SPCFY anywhere on the offered line will split the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

* See also
  Offer-Line Operation
  Insert-Line Operation
  Split-Line Operation

3.4.2 FIND Command



<find target>



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 will perform 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.

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

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

If a string of <graphic>s within <delimiter>s is given as the <find target>, a literal search for the target will be 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 both to literal and to 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 upper or lower case. A case-sensitive search will find only those occurrences of the target that are identical to the <find target> with respect to upper and lower case 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
  GO Command
  OPTIONS Command
  CANCEL Command

3.4.3 LABEL Command

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.

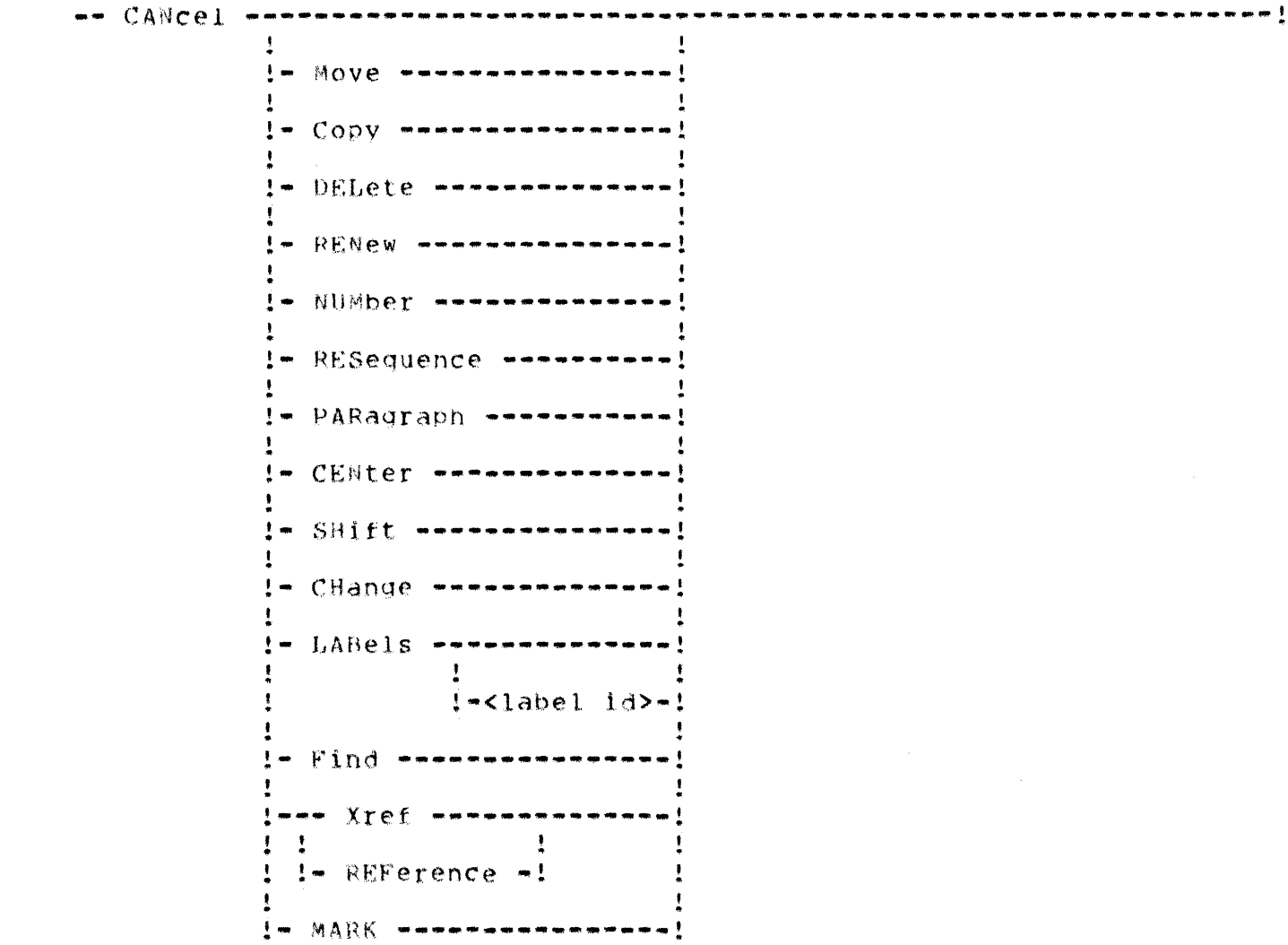
If a labeled line appears on the edit-mode page, the label is displayed in place of the sequence number for the line. Moving a labeled line also moves the label. Copying a labeled line has no effect on the label (the label remains with the original source). Deleting or renewing a labeled line causes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

"LABEL" 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 will cause 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 may be defined at one time. The "CANCEL LABEL" command may be used to discard labels.

* See also
  Insert-line Operation
  GO Command
  CANCEL Command

3.4.4 CANCEL Command



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 xref 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.

CANCEL MOVE, COPY, DELETE, RENEW, NUMBER, RESEQUENCE, PARAGRAPH, CENTER, SHIFT, or CHANGE discards 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. "CANCEL LABELS" 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
  Pending Commands
  FIND Command
  LABEL Command
  MARK Command
  Cross-Referencing Commands

3.4.5 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
  Insert-line Operation
  Editing Commands
  CHANGE Command
  CANCEL Command

3.4.6 COLUMN Command

The COLUMN command controls "column mode". When offering a line for editing, the Editor normally places the cursor where the SPCFY to offer the line was done, on the first non-blank character of the line, or on the find or xref target, depending on how the line became the offered line. In column mode, the Editor unconditionally places the cursor on the specified column.

"COLUMN" by itself turns column mode ON and sets the target column to the column in which the "]" character was entered (as the first non-blank 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".

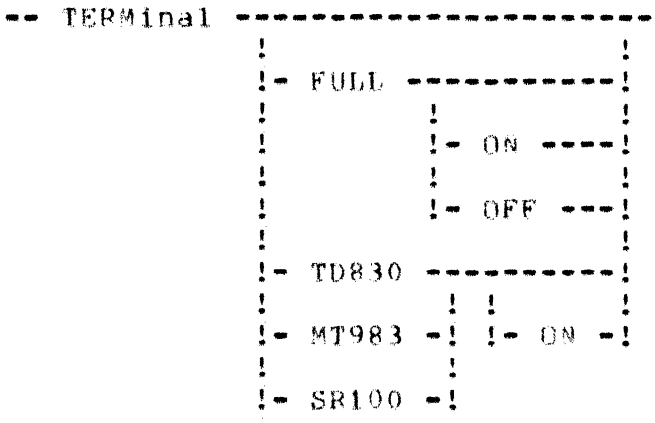
"COLUMN <column>" turns column mode ON and sets the target column to the specified <column>. For example, the command "]COL 65" causes the Editor to always place the cursor in column 65 of the offered line, regardless of where the "]" character of the command was entered. The specified <column> must 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

3.4.7 TERMINAL Command



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

"TERMINAL" 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, and SR100 options are used to 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 or an MT983 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 Use of the Terminal and Installing the Editor.

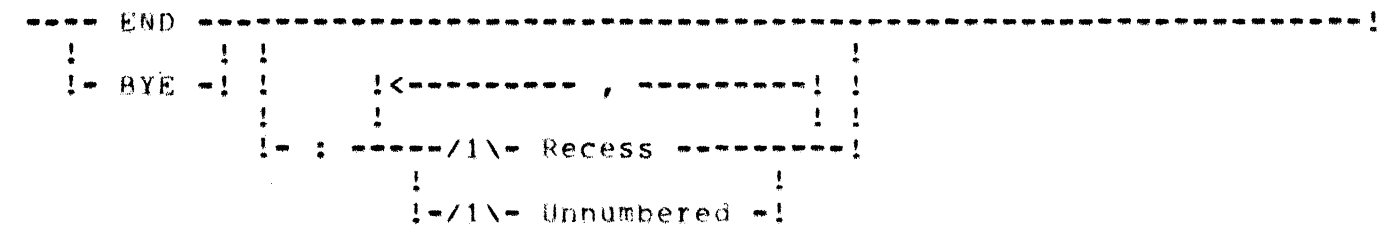
* See also
  Use of the Terminal
  Installing the Editor

3.5 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 command	writes an updated CANDE workfile, (usually) discards the recovery file, and exits.
FORGET command	Does not write an updated CANDE workfile, discards the recovery file, and exits.
RECESS command	Does not write an updated CANDE workfile, saves the recovery file, and exits.

3.5.1 END Command



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
  Patch Mode
  OPTIONS Command

3.5.2 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.

In order to give you a chance to reconsider, the Editor will not purge the recovery file until you have confirmed that this action is what you intended. When FORGET is entered, 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, in which case the recovery file is discarded and the session ends, or pressing XMT, in which case the FORGET command is aborted and the Editor returns to edit mode.

3.5.3 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

3.6 CROSS-REFERENCING COMMANDS

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

The ALGOL, DCALGOL, FORTRAN, 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.

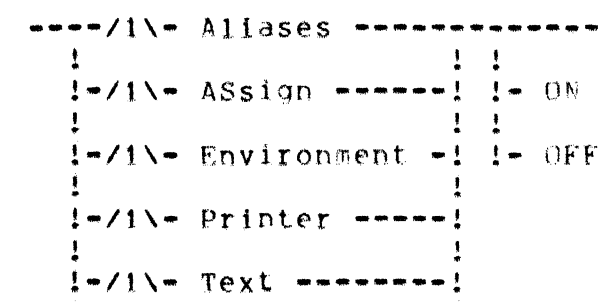
Usage Note: Cross-referencing commands are very useful when patching large programs. If your program is more than a few thousand lines, you might consider using the xreffiles facility.

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

* See also
LOAD Command

3.6.1 Xref Options

<xref option>



Xref options allow settings to be specified for options that affect the cross-referencing commands. These options may 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 may 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 turned ON if the option name is followed by ON or nothing and is turned OFF if followed by OFF. For example, "T" would turn TEXT ON, while "E OFF" would turn 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 causes the displayed list of references to be limited 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

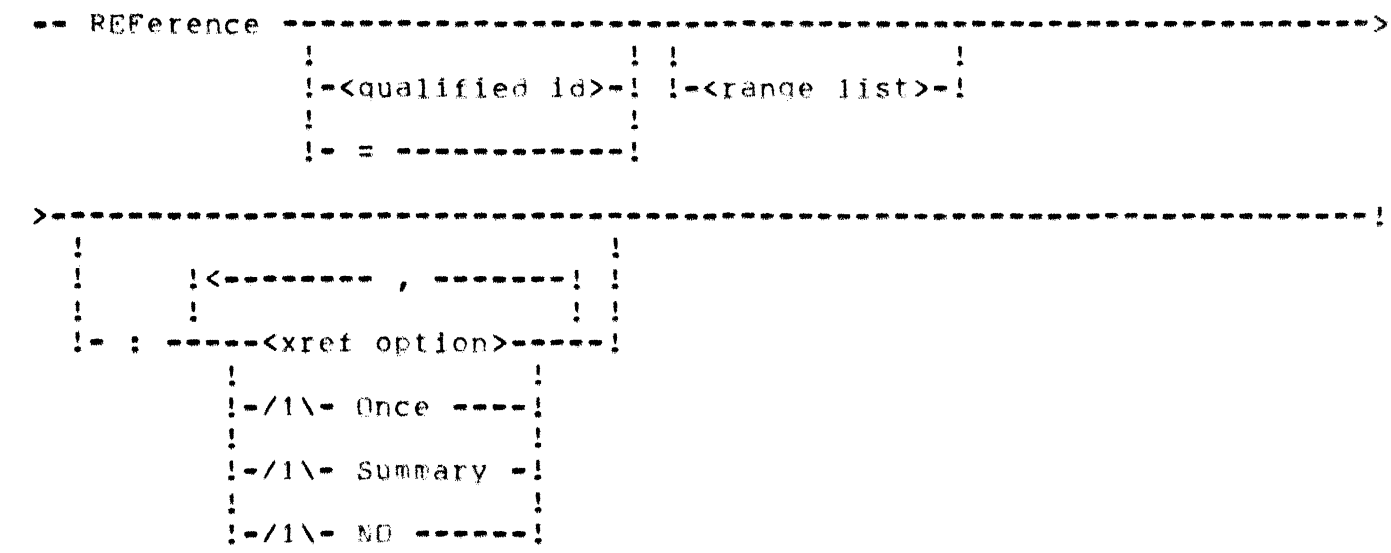
OFF in the command itself.

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
  OPTIONS Command
  REFERENCE Command
  ALL Command
  INTERSECTION and UNION Commands

3.6.2 REFERENCE Command



The REFERENCE command establishes an 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 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.

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.

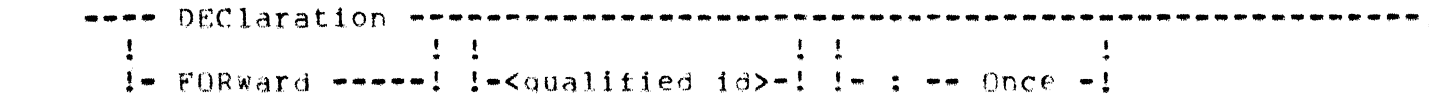
Depending on the options specified in the OPTIONS command or in the REFERENCE command itself, the reference display may include additional information, such as the actual text of the line for each reference or the environment of each reference. For NEWP programs that include module declarations, references 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). Pressing SPCFY without moving the cursor will cause the Editor to display the next page, if there is one. If there isn't another page, the Editor returns to edit mode. Moving the cursor to the sequence number of a displayed reference and pressing SPCFY will cause the Editor to terminate the reference display and offer that line for editing. Any other input will cause the Editor to return to edit mode.

Usage Note: 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 SPCFY on the sequence number after "OCCURS @".

* See also
OPTIONS Command
Xref Options
RANGE Command
RECENT Command

3.6.3 DECLARATION and FORwARD Commands

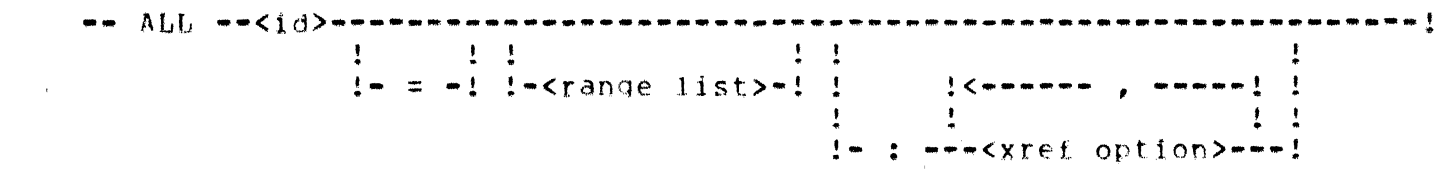


The DECLARATION command moves the offered line to the declaration of the specified <qualified id> or, if null, to the declaration of the current xref target. The FORwARD command moves the offered line to the forward declaration of the specified <qualified id> or xref target; if there is no forward declaration,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

3.6.4 ALL Command



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 options for the ALL command are ALIASES and PRINTER.

After displaying the identifier list, the Editor waits for input before displaying the next page (if any). Pressing SPCFY without moving the cursor will cause the Editor to display the next page, if there is one. If there isn't a next page, the Editor returns to edit mode. Pressing SPCFY when the cursor is on a displayed identifier will cause the Editor to assign the specified identifier as the xref target and then go to the declaration of that identifier (as if a DECLARATION command had been entered for that identifier). Pressing SPCFY when the cursor is on a displayed sequence number will cause the Editor to offer that line. Any other input will cause the Editor to return to edit mode.

* See also
  OPTIONS Command
  Xref Options
  DECLARATION and FORWARD Commands
  RANGE Command

3.6.5 RANGE Command

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

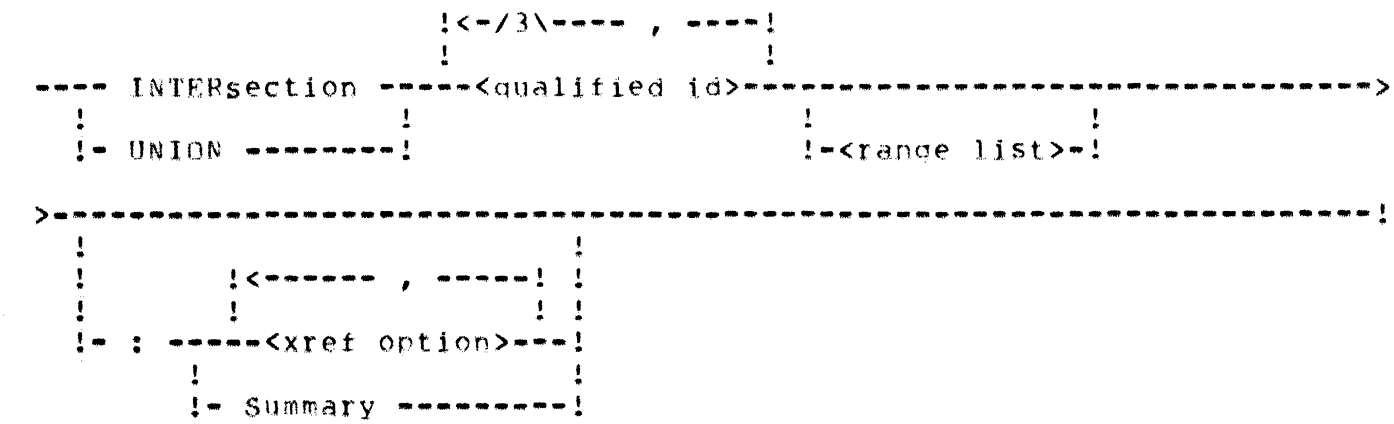
"RANGE" by itself displays the current range specifications.

"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
  REFERENCE Command
  ALL Command
  INTERSECTION and UNION Commands

3.6.6 INTERSECTION and UNION Commands



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 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). Pressing SPCFY without moving the cursor will cause the Editor to display the next page, if there is one. If there isn't another page, the Editor will return to edit mode. Moving the cursor to the sequence number of a displayed reference and pressing SPCFY will cause the Editor to terminate the reference display and offer that line for editing. Any other input will cause the Editor to return to edit mode.

* See also
  OPTIONS Command
  Xref Options
  REFERENCE Command
  RANGE Command

3.6.7 RECENT Command

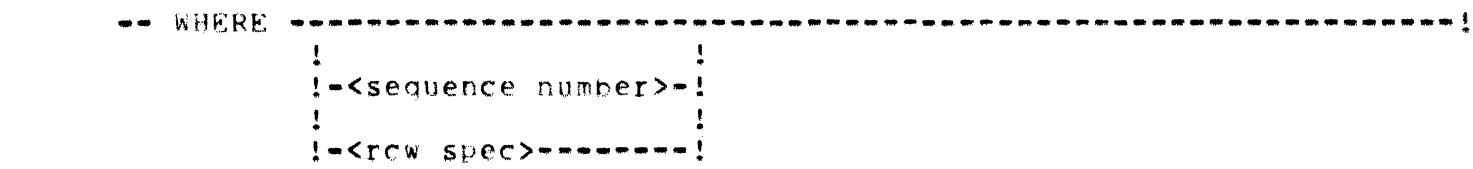
```
-- RECent ----------------------------------------------------------!
```

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

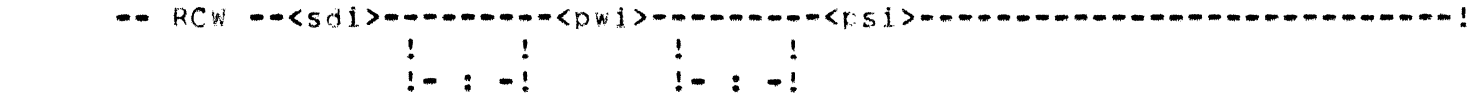
After displaying the identifier list, the Editor waits for input. Pressing SPCFY without moving the cursor will cause the Editor to return to edit mode. Moving the cursor to a displayed identifier and then pressing SPCFY will cause the Editor to assign the specified identifier as the xref target and then go to the declaration of that identifier (as if a DECLARATION command had been entered for that identifier). Moving the cursor to a displayed sequence number and pressing SPCFY will cause the Editor to offer that line (as if a "GO <sequence number>" command had been entered). Any other input will cause the Editor to return to edit mode.

* See also
  GO Command
  REFERENCE Command
  DECLARATION and FORWARD Commands

3.6.8 WHERE Command



<rcw spec>



<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 codefile, which must have been loaded previously (refer to the LOAD command).

* See also
  LOAD Command

3.7 SPECIAL-PURPOSE COMMANDS

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

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

3.7.1 TEACH Command

<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 entered, you will be shown some page of the textbook. [Once in teach mode, press SPCFY to move to the next page of the textbook.] "TEACH" by itself will display 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, SPCFYing in the "Next Topic" box and SPCFYing in the form on the top line (meaning "go to next page") will both 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 XMT. If the topic appears anywhere on the screen, you can move the cursor to where it appears and press SPCFY, instead of entering the topic in the form. For example, if the word COPY appears on the screen, placing the cursor anywhere on that word and pressing SPCFY will cause the Editor to move to the information about the COPY command. If you enter or SPCFY on an unrecognized topic, the Editor will display "Unknown topic" and will remain 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.

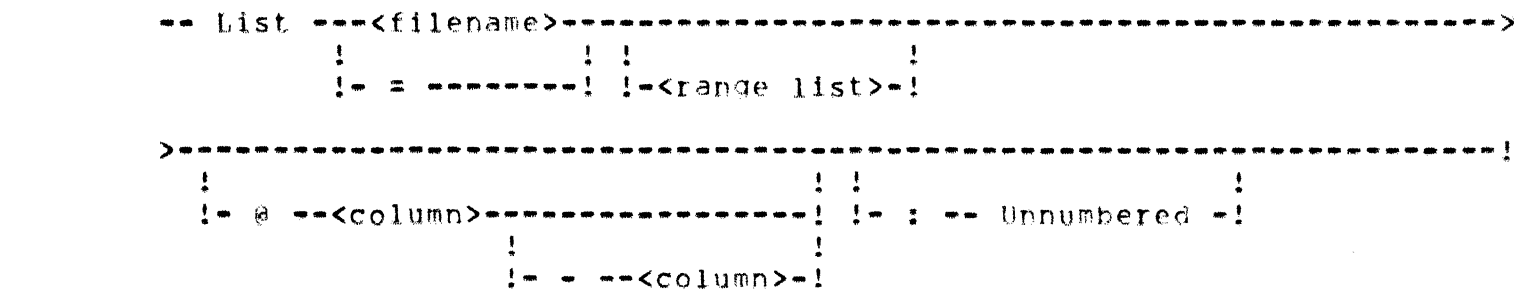
To refresh the page, enter one or more blanks in the form and press XMT or enter CTRL 00 (the CTRL-key synonym for REFRESH).

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.

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

* See also
  PRINT Command
  CTRL-Key Synonyms for Editor Commands

3.7.2 LIST Command



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).

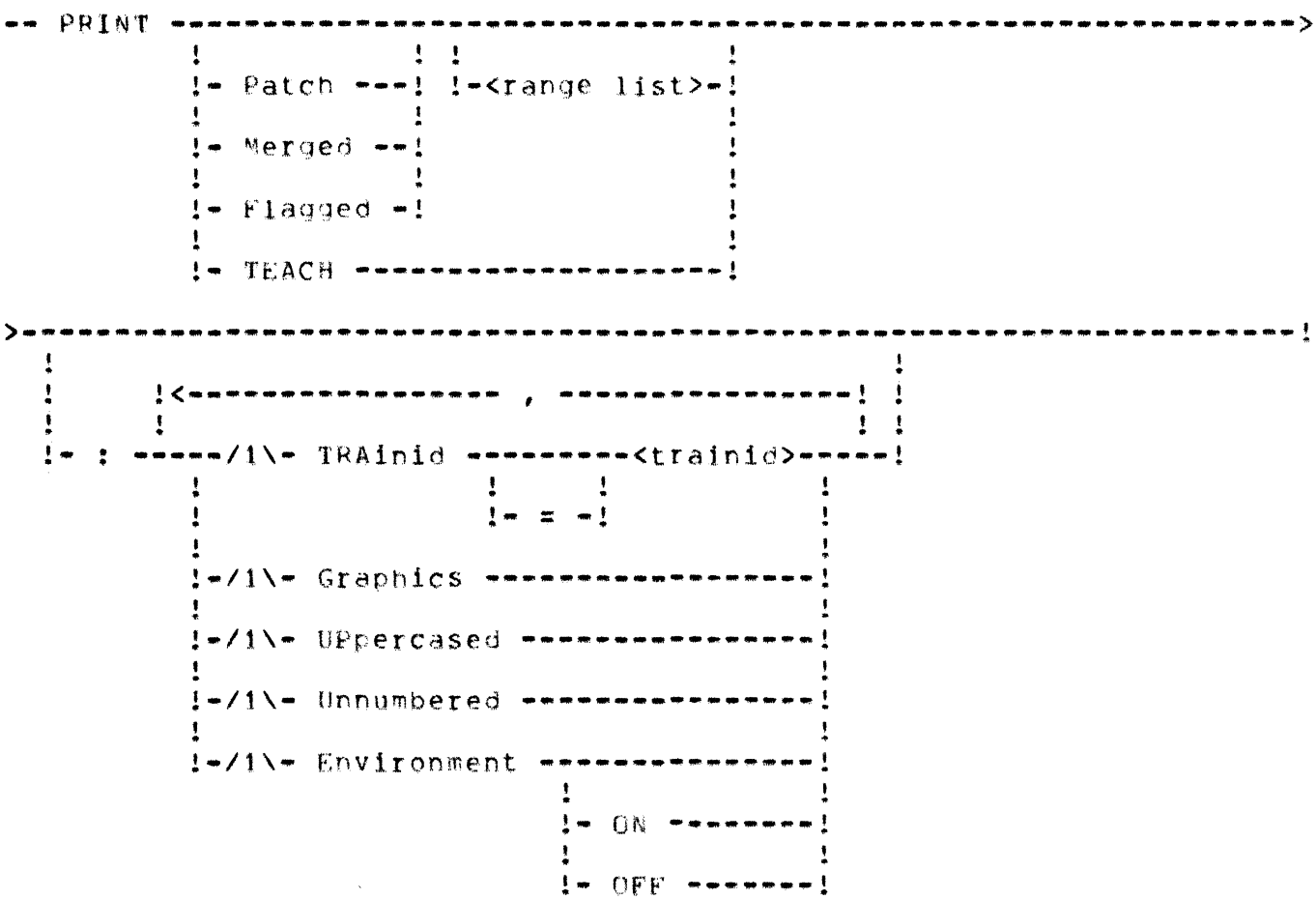
If the "@<column>" syntax appears, only the specified column range will be listed.

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
  LOAD Command

3.7.3 PRINT Command



<trainid> Any one of the mnemonics defined for the TRAINID file attribute in the B 5000/B 6000/B 7000 Series I/O Subsystem Reference Manual (form number 5001779). 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 information to be printed is selected by the PATCH, MERGED, FLAGGED, or TEACH keywords. If PATCH is specified, all changed records in the workfile are printed in a format similar to the COMPARE output of SYSTEM/PATCH. If MERGED is specified, the effective workfile is printed. If FLAGGED is specified, the effective workfile is printed, and any lines that were deleted or modified are also shown in their original form. If TEACH is specified, the entire teach textbook is printed. If none of these keywords appears, 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 <range list> is not specified, it defaults to the entire file. The RANGE specifications established by the RANGE command do not apply to the PRINT command.

TRAINID

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

GRAPHICS

The GRAPHICS option is used to improve the mapping between ASCII (terminal) graphics and EBCDIC (printer) graphics. The following table 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 with GRAPHICS ON	EBCDIC96 Printer 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
non-graphics	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 set.

UPPERCASED

The UPPERCASED option causes the Editor to translate all lower case letters to upper case before printing.

UNNUMBERED

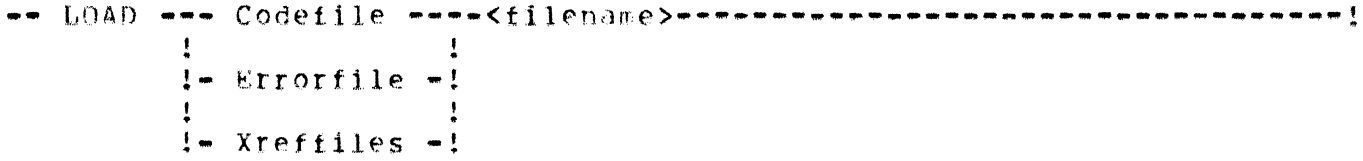
The UNNUMBERED option causes the Editor to leave records that are unnumbered in the workfile unnumbered on the printer output. If UNNUMBERED is not specified, the Editor will automatically assign temporary sequence numbers to all unnumbered records as they are printed (refer to the MAXIMUM and MINIMUM 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.

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, sets the option; ENVIRONMENT OFF resets 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). ENVIRONMENT is invalid for PRINT TEACH, PRINT MERGED (explicit or default), and PRINT FLAGGED operations.

* See also
  FILEKINDs and Record Formats
  Patch Mode
  OPTIONS Command
  Xref Options
  LOAD Command

3.7.4 LOAD Command



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 code file 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 will automatically attempt 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, 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 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) Search for a file called "XREFFILES/<filename>/DECS". If this file is not present, then

2) if the first <name> in the <filename> is SYMBOL, search for a file with SYSTEM as the first <name> (for example, if the <filename> is SYMBOL/ABC, search for XREFFILES/SYSTEM/ABC/DECS). If this file is not present, then

3) search for a file called "XREFFILES/OBJECT/<filename>/DECS". If this file is not present, then

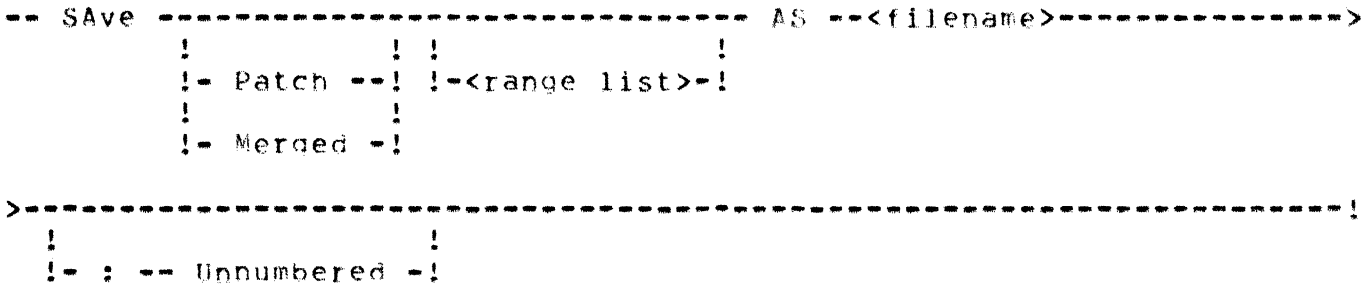
4) replace the first <name> in the <filename> by OBJECT and search for the file (for example, if the <filename> is SYMBOL/ABC, search for XREFFILES/OBJECT/ABC/DECS). If this file is not present, then report an error.

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 may be entered at any time. However, the command causes the current xref target (and RECENT xref targets) to be discarded.

* See also
  GO Command
  Cross-Referencing Commands
  RECENT Command
  WHERE Command

3.7.5 SAVE Command



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 <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
  Patch Mode
  OPTIONS Command

3.7.6 WHAT Command

```
-- WHAT ----------------------------------------------------------!
```

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

3.7.7 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

4 CTRL-KEY SYNONYMS FOR EDITOR COMMANDS

TD830, MT983, 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 CTRL sequences may be entered at any time. The cursor position is irrelevant; the cursor will not move while you are entering the CTRL sequence, nor will the two <digit>s be displayed on the screen.

Any characters that are entered between a XMT operation and a CTRL 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 CTRL 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

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) or
OPTIONS RELATIVE OFF (if RELATIVE is currently ON)

92 CHANGE 1 : MARK

93 JOIN

94 OPTIONS DENV ON (if DENV is currently OFF) or
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) or
COLUMN OFF (if COLUMN is currently ON)

97 OPTIONS DUPLICATE ON (if DUPLICATE is currently OFF) or
OPTIONS DUPLICATE OFF (if DUPLICATE is currently ON)

5 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 CTRL-Key Synonyms for Editor Commands).

cursor
: The box-like character that marks where on the screen the next character entered 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 "DS". 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 upper-most, left-most 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 Patch Mode 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 Patch Mode 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 in which each pressing of the SPCFY key results in a line being split into two lines (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.

6 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, and SP100 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
  Use of the Terminal
  Recovery Files
  OPTIONS Command
  TERMINAL Command

For more information, put the cursor on any word or token (or in one of the boxes to the left) and push SPCFY. These are some of the topics:

}ctrl}	down	INTERsection	PARagraph	scroll
}SPCFY}	DUPlicate	introduction	patchmode	SEQCHeck
}xmt}	editing	Join	pending	SHift
After	END	LABel	Print	SPLit
ALL	entering	Last	<qualified id>	
backward	ENVironment	LEFTMargin	RAnge	split-line
Before	errorfile	LINes	RCW	<subject>
BYE	filekinds	List	RECent	Summary
CANcel	<filename>	LOAD	RECESS	synonyms
capabilities	FIND	margins	recovery	syntax
CENter	First	MARK	REFerence	TEACH
CHange	FORGET	MAXimum	REFResh	TERMinal
COLHeading	FORward	MERge	RELative	UNdo
COLumn	general	MINimum	<relative line>	
commands	glossary	mistakes	RENew	UNION
Copy	GO	modes	RESequence	utility
DECLaration	group	Move	RIGHTJustify	WHAT
DELete	<id>	NUMber	RIGHTMargin	WHERE
<destination>	INsert	offer-line	SAve	window
display	insert-line	Option	screen	xref

EDITOR "TD8XX" CONTROL COMMANDS

82/07/16 PHK

	1	2	3	4	5	6	7	8	9	0
10 1	TEACH: TEACH	TEACH: MOVE/ COPY	TEACH: DELETE/ RENEW	TEACH: RENUMBER	TEACH: PARAG/ ALIGN	TEACH: CHAR. SUBS	TEACH: SCROLL/ FIND BK	TEACH: SCROLL/ FIND FW	TEACH: TEACH	TEACH: MISC
20 2	MOVE FIRST	MOVE LAST	MOVE SINGLE	TOGGLE CONT COPY MODE	COPY FIRST	COPY LAST	COPY SINGLE	INSERT BEFORE	INSERT AFTER	CANCEL MOVE/ COPY
30 3	DELETE FIRST	DELETE LAST	DELETE SINGLE	—	RENEW FIRST	RENEW LAST	RENEW SINGLE	—	—	CANCEL DELETE
40 4	RENUM FIRST	RENUM LAST	RENUM SINGLE	RENUM GROUP	RENUM FIRST (SEQ)	RENUM LAST (SEQ)	RENUM SINGLE (SEQ)	RENUM GROUP (SEQ)	RENUM ALL UNNUMB	CANCEL RENUM
50 5	PARAG FIRST	PARAG LAST	PARAG SINGLE	CENTER LINE	SHIFT FIRST	SHIFT LAST	SHIFT SINGLE	—	SWAP MARGIN SPECS	CANCEL PARAG/ SHIFT
60 6	SUB 7	SUB —	SUB \	SUB |	SUB	SUB ?	SUB	SUB	—	QUERY USER SUB
70 7	GO TO BEGINNING CHANGE	GO TO CHANGE AREA BEGINNING	GO TO UNSEQ BEGIN	GO BACK CURR ENVIR	GO BACK 1 LINE	GO BACK 7 LINES	GO BACK 11 LINES	GO BACK 22 LINES	GO BACK FIND/ XREF	GO TO BEGINNING
80 8	GO TO END CHANGE	GO TO CHANGE AREA END	GO TO UNSEQ END	GO FWD CURR ENVIR	GO FWD 1 LINE	GO FWD 7 LINES	GO FWD 11 LINES	GO FWD 22 LINES	GO FWD FIND/ XREF	GO TO END
90 9	SEARCH SEQ ERROR	MARK AS CHANGED	JOIN	TOGGLE ENVIR MODE	TOGGLE CAPS MODE	TOGGLE COL MODE	TOGGLE DUP MODE	TO EDIT STATE	TO CMD STATE	NO OP
00 0	GO MARKER 1	~~SET GO~~ MARKER 2 1	GO MARKER 3 L	~~SET GO~~ MARKER 4 L	~~GO SET~~ MARKER 3	SET MARKER 2 3	~~GO SET~~ MARKER 3 4	SET MARKER 4	RETRACT LAST CMD	REFRESH SCREEN

Summary of facilities and <subjects>

[If you are a new user, push SPCFY for an introduction to the program]

Command names and TEACH <subjects>

END,RECESS,FORGET: Exit editor; return to CANDE
SAVE: Write output file (merged or patches)
PRINT: Print workfile or tutorial information
INCLUDE: Apply an existing patch to this workfile
FIND(CH): Scan workfile for specified target
LOC(ATE), DEC, FORWARD, LOAD, WHERE, SUMMARY, RANGE
REF(A), MERGE(A), COIN(A): Display XREF info
LEFTM,RIGHTM,INDENT,RIGHTJ, ALIGN: Set margin specs
COL: Set/reset fixed-column mode
MARK: Supply ID for columns 81-90
WHAT,STATUS:Display of information.
MINI,MAXI: Set limits on automatic increment
TEACH: Display tutorial information:
 Introduction to the program: INTRO, MODES, PRIMER
 How to use the terminal: TERM, TD820, TD830, KEYS
 General information: STATES, DISPLAY
 Editing: TEXT ENTRY, SPCFY (SELECT, INSERT, SPLIT)
 Types (FILEKINDs) of files: TYPE
 Character set (graphics): CHARS UPCASE DOWNCASE
 Recovery, ALTER file: ALTER
 Go to a line by sequence number or count: GO

CTRL-digit-digit functions

1m: Teach about CTRL mn
2n: MOVE or COPY line(s)
3n: DELETE or RENEW line(s)
4n: (Re)NUMBER line(s)
5n: PARAGRAPH or ALIGN
6n: Character SUBSTITUTION
7n: SCROLL or FIND backward
8n: SCROLL or FIND forward
90: NOOP (refresh line)
91: CHECK sequence numbers
92: Mark as changed
93: JOIN two lines into one
94: Toggle ENVIRONMENT mode
95: Toggle CAPS mode
96: Toggle COL mode
97: Toggle DUPLICATE state
98: RESUME (prior) editing
99: Go to COMMAND line
00: REFRESH screen
0n: REPOSITION at MARKER
09: RETRACT prior operation

INTRO: Introduction to the program

This program is an editor of card-image files, exploiting/circumventing the idiosyncracies of the TD820 or TD830 terminal. It operates on an arbitrary-size file in either a patch or merged mode. Facilities include: single-line changes, additions and splits; multi-line deletes, moves, copies and renewals; location by sequence number or find by simple token; automatic or specified renumbering.

Simple editing is performed on one line at at time, to minimize exposure of the file to accidental damage due to user, terminal or system error. Any operation may be retracted at once if the results were unsatisfactory.

There are about forty tutorial pages available, in three categories:

1: General subjects, called out by subject name (TERM, SPCFY, etc.)
2: Editor commands, called out by command name (PRINT, FIND, MAXI, etc.)
3: Numeric CTRL functions, called out by first or both digits (3, 95, etc.)

The summary page lists all commands and CTRL functions and the principal subject names; it may be recalled by a CTRL 11 XMT sequence, or by a TEACH command. The entire document may be printed in hard-copy via the PRINT TEACH command.

These pages may be read in order; the general topics and key commands are grouped at the front. A new user should read at least the next five pages.

[Press SPCFY to continue]

STATES: Teach, Command, Edit

The editor operates in one of three basic states:

a: Teach state is what you see now. You get here from command state with the TEACH command, and from command or edit state by CTRL 1m XMT.

b: Command state is when entry is enabled on the top line: you can enter any of the commands listed in the summary. You get to command state via CTRL 99 XMT, or in many other ways. Any lower-case letters input on the command line are translated by the editor to upper case, except in a FIND target.

c: Edit state is when some part of the workfile is exposed for you to operate upon. There are actually three edit states:

   1: Change state, when a line of the workfile is offered for change.
   2: Insert state, when an empty line is offered for possible insertion.
   3: Duplicate state, a variation of insert state for somewhat constant data.

   You get to one of the edit states by a SPCFY input (indicating the line you want to change or insert after), by a locate command, or by CTRL 98 XMT.

The editor starts in insert state if the workfile is empty: just start entering text. Otherwise it starts in command state, so you can enter a locate command.

Editing Modes: MERGED, PATCH

The editor operates upon an effective workfile comprising two parts: a SYMBOL file containing the original file, and an ALTER file containing the changes. (In some cases the SYMBOL file may be absent and/or the ALTER file may be empty of changes.) The output of the editor can be either a new symbol file with all the changes MERGED into it, or a PATCH file which can be applied to the old symbol with SYSTEM/PATCH or a standard compiler.

The editor is invoked with the CANDE command UTILITY <editor name> <parameter>.

If the parameter is empty, the editor runs in MERGED mode: the CANDE worksource (if any) is the SYMBOL; the editor output will be an updated MERGED worksource.

If the parameter is non-empty, the editor runs in PATCH mode: the parameter is the filename of the SYMBOL file and the CANDE worksource (if any) is a patch to be loaded into the the ALTER file. The editor output will be an updated PATCH worksource.

The END command cause the editor to write the updated worksource and return to CANDE; teach END for details. The SAVE command may be used to write either a PATCH or MERGED file with arbitrary filename; teach SAVE for details.

The ALTER file serves as the editor's recovery file; teach ALTER for details.

PRIMER: Typical use of the editor

Here is a typical sequence of operations to edit an existing file:

Use a locate (sequence number) or FIND (token) command to display the point of change.

Proceed through the file by pushing XMT for each line, first changing some lines as appropriate. Select some other line (earlier or several lines later) by putting the cursor on that line and pushing SPCFY.

When you need to insert a new line ahead of the next one offered, push SPCFY. Enter as many new lines as you need; when you're through inserting push SPCFY again. Or select some other line to edit by moving the cursor first.

Use the same techniques to make a new file; the editor starts you "inserting" lines into the empty file.

You can scroll your "window" back and forth along the file, using special sequences like CTRL 78 XMT. Other CTRL sequences permit deleting or moving lines, forcing upper-case letters, and many other functions.

Commands to find specific line numbers, to SAVE the file, etc., are typed only on the top line; you get there with a CTRL 99 XMT sequence. Then you can get back to where you were editing with CTRL 98 XMT.

Command TEACH, CTRL 1m

The TEACH command, entered as just the one word, enters teach mode and displays the summary page. The numeric control sequence CTRL 11 XMT does the same.

The word TEACH may be followed by a <subject> specification to display other tutorial pages. Once in teach mode, you may see other subjects by entering just the <subject>. Information about the numeric control sequence CTRL mn XMT may also be displayed by the sequence CTRL 1m XMT (where m and n are digits). That is, CTRL 1m is like TEACH m (except you don't have to be in command state).

A <subject> may be a word or an integer:

The word may be any command or tutorial topic listed in caps in the summary. Also, many other key words will call up an appropriate page.

The integer may be either one or two digits: the first or both digits of any CTRL function.

You may progress from one tutorial page to the next by pressing SPCFY.

The command PRINT TEACH produces a printer listing of all these pages.

The numeric control sequence CTRL 99 XMT or CTRL 98 XMT is used to exit teach state and return to command or editing state, respectively.

TD830: Features of the terminal

In normal editing, the terminal is operated in "forms" mode, with most of the information on the screen "protected" and user entry restricted to the area(s) between ` and `. Use TAB to go from one field to the next when there are two.

There are 3 kinds of transmission: normal text XMT, CTRL numeric XMT, and SPCFY.

The XMT key, used by itself, sends in the entire unprotected area of the screen. (If there is an ETX on the screen, transmission stops at the ETX.)

By pushing CTRL, two digits, and XMT, one may send a "numeric control input"; these are used extensively with this editor. (The screen isn't involved.)

The SPCFY key sends the cursor position; it is used for that and other purposes.

The following keys are useful in editing individual lines: CHAR INS, CHAR DEL, CLR EOL, HOME, CLEAR, and the left and right arrows. The LINE INS, LINE DEL, and GS keys are ineffective in forms mode. The up and down arrows are used to select other lines but not in editing a given line. Avoid the RET key.

The editor reconfigures your terminal for full-form transmission, among other options. (Typically with CANDE and other programs, home-to-cursor transmission is used.) The editor restores the configuration upon normal exit. The terminal restores its defaults upon power on, or with CTRL, space, shift-D, CTRL.

DISPLAY: How TD830 screen is used

The top line is reserved for status display and entry of commands. CTRL 99 is used to get to the top line to enter a command; a command on any other line is treated as text for the file. In editing state, the top line indicates the current setting of any options whose effect is not obvious elsewhere. The bottom (status) line of the display is also used for editor status information.

The six low-order digits of the sequence number are displayed with each line that has a number. New lines and moved lines are unnumbered and displayed with only a + in the sequence column. If there is a numbered line at or before the top of the screen, its 8-digit number appears at the upper left. (In command mode, just the two high-order digits appear.) If there is a numbered line at or beyond the bottom of the screen, its number appears at the lower left. When a numbered line is offered for editing, its 8-digit number appears upper right.

The space after the number (or +) contains ` for the offered line. Otherwise it is blank for unaltered lines from the symbol file; it is * for modified or inserted lines; it is # for moved or renumbered lines from the symbol file. The column may be pre-empted to show <, > or = when a line has been selected as the first, last or only line in a group to be operated upon by delete, move, etc.

In insert or duplicate state, the offered line has >>>> in the sequence field; this line is offered on the screen but does not yet exist in the file. The line is initially blank (insert) or a copy of the previous line (duplicate).

TEXT ENTRY: Altered and new lines

In the edit states, one line at a time is offered for your entry or modification of text. In change state it's a line from the file; in insert or duplicate state it's a new line. There is a 72-character field between the ` and ` marks. Use the keyboard to set up the line; then push XMT to send it.

In change state, if the line (or partial line) you send is the same as the line already in the file, no change is made. When the input is different, the line in the file is changed and any ID in columns 81-90 of the line image is reset to blanks (or as specified in a MARK command). If you send only a partial line, only the corresponding part of the file line is changed. Remember that the editor sees only the changes you transmit: to be sure the whole line is sent, push HOME before XMT.

In change state, after each XMT the next line is offered. (After the last line, the editor goes to insert state and offers a new line at the end of the file.)

In insert (or duplicate) state, an XMT causes the information on the new line to be inserted into the file at the point corresponding to its position on the screen. After each XMT another new line is offered.

Any editing operation, once done, may be undone by a CTRL 09 entered before the next operation. CTRL 00 may be used to restore the screen. (Teach 0 for more.)

SPCFY (Select Insert Split)

The SPCFY key is used to choose a line to edit, to enter insert mode (after an arbitrary line or before the offered line), to split a line, and to terminate an insertion. (In fixed-column mode, it also sets the column.)

To select a line for editing: put the cursor in the text or number field of any line except the one currently offered; push SPCFY. If the cursor was after the first non-blank, it will be kept at that point as the line is offered.

To insert after a line: put the cursor exactly between the number (or +) field and the text field of any line (it may be on a space, *, `, etc.); push SPCFY.

To insert before the line currently offered: put the cursor at or before the first non-blank of that line; push SPCFY.

To split the line currently offered: put the cursor anywhere after the first non-blank; push SPCFY. Everything from the cursor rightward is moved to a new line, left-justified with it, and inserted after it into the file.

To stop inserting new lines and resume editing at the next line: leave the cursor in the offered insert line and push SPCFY.

CAUTION: All these rules are modified or inapplicable in fixed-column mode. For details, teach COL or 97.

CHARS: Graphic character set

Here are all the ASCII-67 graphic characters, as represented here, using a TD830 or TD820 with full ASCII keyboard.

ABCDEFGHIJKLMNOPQRSTUVWXYZ		upper-case letters				26	
abcdefghijklmnopqrstuvwxyz		lower-case letters				26	
0123456789		digits				10	
() parens	[] brackets	{? braces	<> less,greater			8	
+ plus	- hyphen	/ diagonal	= equals			4	
. period	, comma	: colon	; semicolon			4	
" quote	' apostrophe	% percent	& ampersand			4	
@ at	* asterisk	? question	# crosshatch			4	
$ dollar	_ underscore	\ backslant	} exclamation			4	
! splitbar	¯ tilde	^ carat	~ grave			4	
		space	? rubout (DEL)			2	total=96

The ? is used to represent any other non-graphic character found in the file.

[Teach EBCDIC for discussion of ASCII/EBCDIC mapping on train printers.]

Type (FILEKIND) of workfile

The editor treats COBOLSYMBOL and JOBSYMBOL workfiles specially:

COBOL files have sequence numbers in columns 1-6, text in 7-72, and optional marks in 73-80.

JOB files have text in columns 1-80, blanks in 81-82, and sequence numbers in 83-90. Because of screen-width limitations, the editor will treat only the 1-72 as text; 73-82 are treated as optional mark: unless a MARK value is SPECIFIED, columns 73-82 of all entered or altered lines will be blanked.

All other types of workfile are assumed to have text in columns 1-72, sequence numbers in 73-80, and optional marks in 81-90. If the workfile contains 14-word records (which is the CANDE MAKE default for SEQDATA and FORTRAN files), columns 85-90 are assumed blank on input and are lost on output.

INCLUDE files must match the workfile in placement of text and sequence-number fields.

ALTER file and workfile recovery

The ALTER file contains card-image text of all the new and modified lines in the workfile, copies of the editor table (with sequences and linkages), and assorted auxiliary data. It is a permanent file with special end-of-file handling; it is recoverable after a halt/load, RECESS, or other contingency. The file is EXCLUSIVE, to prevent one user from stepping on the ALTER file of another user with the same usercode and workfile name; instead he gets WAITING FOR .....

The title of the ALTER file is constructed by appending to "ALTER/" the CANDE workfile name (or the date/time as eight digits, mmddhhmm, if the workfile is unnamed). If the ALTER file already exists when the editor is started, the prior editing session is recovered; the current parameter and worksource are ignored (except that in MERGED mode the worksource may be the SYMBOL part of the effective workfile). The SYMBOL file is checked for consistency with the ALTER file; discrepancies cause user warnings and accomodation of the ALTER file.

To recover an aborted editing session:

1: Log onto CANDE again.
2: RECover your CANDE workfile, or reMAKE or reGET it if there is no REC file.
3: Invoke the editor with the Utility command, as before. To recover an ALTER file with a name other than ALTER/<workfile>, specify a Utility parameter containing exclamation point (!) followed by the ALTER file title.

You can terminate a session and rename its ALTER file; teach RECESS for details.

Terminal features and differences

The editor runs on several variations of TD820 and TD830. There are some differences in editor operation due to the terminal characteristics.

TD830 (Firmware level 2, to be released mid-77)
The most general set of facilities are available for this terminal.

TD830[1.2] (Firmware level 1.2; never B6700 qualified)
Fixed-column and CAPS modes and full-line transmission unavailable.

TD820 (Hardwired terminal)
Full-line transmission, bright video, and status line unavailable.

Short memory: a TD830 may be configured with 4K of memory arranged as 24 80-character lines. Its datacom buffer is too short to receive a whole screen, so the editor transmits two blocks when needed.

Modified-ASCII: Some keyboards have only 90 of the 95 ASCII "graphic" characters, and some of those are rearranged. Software translation unscrambles the rearrangement; character substitution is available.

Forms vs. braces: By default, modified-ASCII TD820 terminals convert the { and ? characters to ` and ~ when in forms mode. The editor hides the braces by displaying them as peculiar control graphics.

The editor recognizes terminal types by a dictionary of station names. Some TEACH pages have alternate versions selected according to the user's terminal. Teach TD for more information about using your particular model.

Command PRINT: Output to printer

The PRINT command causes the workfile, or tutorial data, to be written to a line-printer file.

PRINT TEACH causes all the TEACH pages to be written; GRAPHICS is forced.

All other forms write the workfile:

PRINT PATCH causes only the alterations to be shown, in a format similar to the COMPARE output of SYSTEM/PATCH.

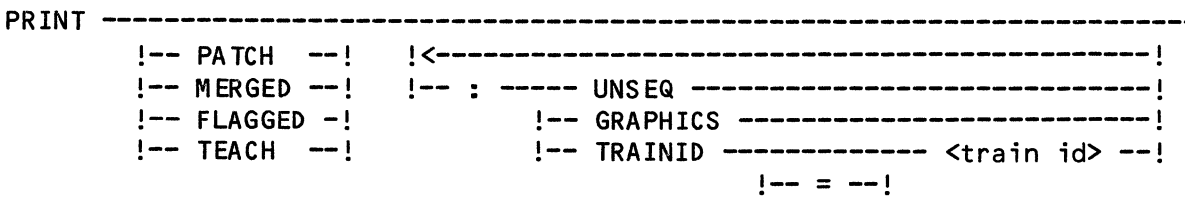
PRINT MERGED causes the effective workfile to be written. The source of each line is noted (PATCH or SYMBOL). Modified or new lines are flagged "*"; moved or resequenced symbol lines are flagged "#".

PRINT FLAGGED causes the effective workfile to be written as above, but any lines which were deleted or modified are also shown in their original form.

PRINT by default uses the PATCH or MERGED format according to the editing mode.

[Continued: SPCFY for syntax details and options.]

PRINT [2nd of 2]



If :UNSEQ appears, any unnumbered lines are printed with blank sequence-number fields; otherwise they are temporarily assigned numbers as though a CTRL 49 were done before the output operation and a CTRL 09 after.

If :TRAINID appears, the specified <trainid> attribute mnemonic will be supplied in the print file; otherwise default (drum or drum-simulated) printing occurs.

If :GRAPHICS appears, TRAINID=EBCDIC96 is inferred and software translation is invoked to cause the ASCII characters { and } to print those graphics. Some other ASCII graphics which do not appear on the EBCDIC96 train are rearranged; SPCFY for details.

The following abbreviations are available: FLAG, G(raphics), TRA.

Printer Graphics

As of February, 1977,
the EBCDIC96 print train relates as follows to the ASCII graphics:

ASCII graphic	EBCDIC96 (and EBCDIC72) character	EBCDIC96 character translated by PRINT:GRAPHICS option
? right brace	exclamation point	right brace
} exclamation point	vertical bar	exclamation point
^ carat	not sign	not sign
! split bar	lozenge	vertical bar
| tilde	cent sign	lozenge
` grave accent	centered dot	cent sign
? rubout (DEL)	question mark	black square

All other ASCII graphic characters have the same representation on the printer.
EBCDIC characters which have no ASCII graphics print as follows:

hex 70	right brace	centered dot
hex FF	black square	centered dot
all others	question mark	centered dot

Conventional ASCII:EBCDIC mapping equates } with vertical bar and ^ with not.

CTRL 7n, 8n: Scrolling

CTRL 7n and CTRL 8n respectively move the "window" backward and forward through the file, either by a fixed displacement or to a particular occurrance. The second digit specifies the displacment amount or the type of occurance:

Displacement: 4 moves to the beginning or end of the current environment.
5 moves 1 line.
6 moves 7 lines.
7 moves 11 lines (half the screen).
8 moves 22 lines (the whole screen less 1 line).

Destinations: 0 moves to the beginning or end of the file (without scrolling).
1 moves to the beginning or end of the next area of any change.
2 moves to the beginning or end of the next group of modified or inserted records, but not moved, renumbered or deleted ones.
3 moves to the next boundary of sequenced with unsequenced lines.
9 moves to the next instance of the current FIND or XREF target.

Scrolling is bounded by the beginning and end of the file; the editor attempts to keep the screen full. If the line offered for change or insertion goes off the screen, the editor reverts to command state.

Displacement over longer distances, or to a specific line number, may be accomplished with the locate command, defined on the next page. [SPCFY]

Go to line number or count

A flexible command provides for locating parts of the file by sequence number, by displacement relative to the current view, or by displacement relative to a sequence number. More formal syntax and semantics appear on the next page. Here are some examples:

123400	Reset display so the line with sequence number 00123400 appears on the line 12; offer the line for editing.
123400-7	Same as above, but show 7 more lines before that one.
|123400-7	Same as above, but stay in command state; don't edit.
+15	Scroll forward 15 lines (same as CTRL 87 plus CTRL 83).
-60	Reset screen 60 lines prior to its current position.

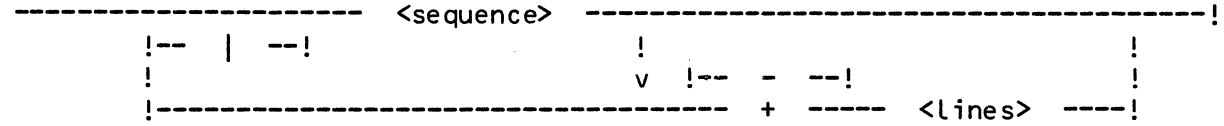
The | option keeps the editor in command state, making it convenient to enter a number of locating commands while browsing. It also suppresses the warning if there is no exact match for the sequence number.

When a sequence number appears, the screen is reset. Otherwise, the screen is scrolled for displacements of 0-22 lines and reset for larger displacements.

[Continued: press SPCFY]

Go [2nd of 2]

Syntax diagram for the locate command:



The <sequence> and <lines> are integers: sequence number and number of lines.

The <sequence> option permits locating a specific line, by number.
The <lines> option moves forward (+) or backward (-) from the current display.
The options may be used together to find a point relative to a numbered line.
The | causes the editor to find the lines nearest to <sequence> and display that neighborhood, but to stay in command mode.

When <sequence> appears, the editor will go into editing (change) state at the specified line, unless:

a: The preface | appeared.
b: The <lines> option pushed the <sequence> line off the screen.
c: There is no such line in the file (an error condition without |).

On keyboards where the _ is more convenient, it may be used instead of +.

Commands FIND and FINDCH

The FIND commands establish a target token, which may be any alphanumeric string or any special character, or any literal enclosed in matching delimiters. Then CTRL 89 or CTRL 79 causes the workfile to be scanned in a forward or backward direction, respectively, for the target. The FIND command scans the entire workfile; the FINDCH command scans only the altered lines.

The target may be a simple token or a literal string. A simple token is an alphanumeric string or a single character. For an alphanumeric token, the editor will match an occurrance in the file which is delimited by blanks or non-alphanumeric characters. A literal target is entered between matching special-character delimiters. For literal (or special-character) targets, the editor will match any occurrance of the target regarless of the context.

Examples: Alphanumeric tokens: GETSPACE Sam x Y
Special-character tokens: $. & (+ ' ? "
Literal tokens: 'GETSPACE' .A+B. /oken/
(These will match FORGETSPACE and GETSPACE;
ZA+BC but not A + B; and Token, token, tokens, etc.)

Unlike other command-line input, the FIND target is not upcased by the editor, although it is affected by the CAPS or SHIFT option.

[SPCFY to continue]

FIND and FINDCH [2nd of 2]

When the editor is in command mode, CTRL 79/89 starts searching with the bottom/top line on the screen (the whole screen is searched). When the editor is in an edit mode, CTRL 79/89 starts searching with the line immediately before/after the offered line (the upper/lower part of the screen is searched). Then the search continues in a backward/forward direction through the effective workfile.

If a target is found, its line is offered for editing. If that line is edited (or just retransmitted), the editor will leave the cursor on that line, so that another CTRL 79/89 will resume scanning with the adjacent line.

If the editor scans to the beginning or end of the file, an appropriate message is displayed.

About every 2-3 seconds, the editor shows the sequence number of the line being scanned. If the editor receives any input from the user terminal, the scanning stops and the input is ignored.

A FIND or FINDCH command with no target cancels any FIND target. If there is no target, CTRL 79/89 may also be used for xrefing. If there is neither FIND target nor XREF identifier, CTRL 79/89 is an error. [SPCFY for XREF discussion]

The editor has the ability to utilize the information in the interactive xref files built by the compilers and XREFANALYZER. The information can be displayed or used to move through the file. [It is important to note that xref files relate to a specific file and do not reflect any subsequent changes.]

To use the xref information to move through the file, a work identifier must be established. This is done with the commands LOC, LOCATE, REF, REFA, DEC, and FORWARD. Once a work identifier is established CTRL 89/79 will move the offered line to the next occurance of the work identifer forward or backward respectively. [There must be no FIND target specified, otherwise CTRL 79/89 are used for FIND.]

If the name of either the symbol file or the workfile name is SYMBOL/xxx, then the editor will try to load the xref files titled XREFFILES/SYSTEM/xxx/DECS and XREFFILES/SYSTEM/xxx/REFS at initialization time.

The commands that use the xref information are:

LOC, LOCATE, REF, REFA, DEC, FORWARD, COIN,
COINA, MERGE, MERGA, ENV, WHERE, SUMMARY, and LOAD

Xref: LOAD

LOAD ---------- <file suffix> ----------}

The load command loads the interactive xref files. The <file suffix> is a standard file title. If the first name of the <file suffix> is "SYMBOL" (with or without a "*" or a user code) it will be changed to "SYSTEM". The editor then uses XREFFILES/<file suffix>/DECS and XREFFILES/<file suffix>/REFS.

Example: LOAD (MADDOG)MY/MCP ON ARK causes the editor to use (MADDOG)XREFFILES/MY/MCP/DECS ON ARK and the corresponding REFS file.

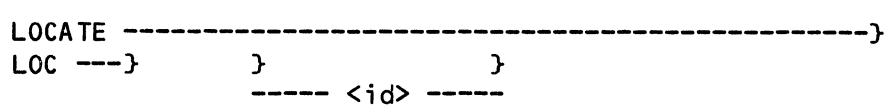
Example: LOAD SYMBOL/MCP causes the editor to use XREFFILES/SYSTEM/MCP/=.

The LOAD command may be entered at any time. A LOAD command causes the current work id, if any, to be discarded and the procedure environment table to be reinitialized before it is used again [TEACH ENV].

If the specified files are not found or are not proper files an error will be given and none of the xref commands will work until valid files are loaded.

An automatic xref file load will be done when the editor is initialized if appropriate files can be found using the symbol file title for <xref suffix>.

Xref: LOC and LOCATE



LOC simply establishes <id> as the work identifier.
LOCATE establishes a work identifier and also displays a description of the id.

<id> may be an identifier, a qualified identifier, or empty.

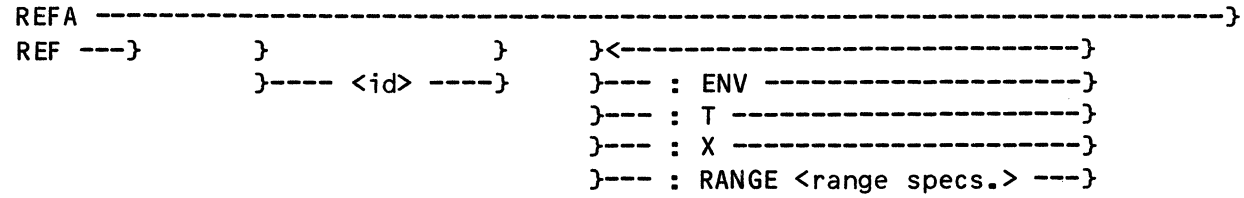
An id may be qualified with 'OF', 'IN', or 'AT'('@'). The string following 'OF' specify the complete qualification of the id; the string following 'IN' may specify only part of the qualification; a number following 'AT' means to locate the id used at that sequence number (@ FIRST means use the first declaration). If no qualification is given, the editor locates the most locally declared id valid at the current editing point.

LOC or LOCATE with an empty <id> simply cancels the current work identifier.

Examples:

```
LOC XYZ @ FIRST
LOC GETSPACE                    LOC I IN FILEHANDLER
LOCATE GETSPACE                 LOC ARG AT 12345678
LOC CLOSE OF FIBSTACK           LOC J @ 44844884
```

Xref: REF and REFA



<id> may be an identifier, a qualified identifier, or empty.
<range specs.> is either a procedure identifier or a pair of sequence numbers.

REF clears the screen and displays a description of and a list of references to <id>. If :T appears, the text lines of each ref are displayed. If :ENV appears environment info is displayed. A # next to the sequence number means <id> was used in a define on that line, an = means it was used in an address-equation, and a * means it was assigned to. If the list of references takes more than a whole screen, the editor will display it a screen at a time waiting for some input between pages. After the last page is displayed the editor waits for input and then re-displays the screen as it was before the REF command. For modularized programs, any ref to <id> outside the module it was declared in is shown in reverse vidio.

[Continued: press SPCFY]

REFA does the same thing as REF except that references to all the aliases of <id> are also listed and the : options are not allowed.

REF or REFA with and empty <id> lists references to the work identifier. If <id> is not empty and :X does not appear, <id> is established as the work identifier.

:X is intended for use when you want to see the references to something but not change your work identifier.

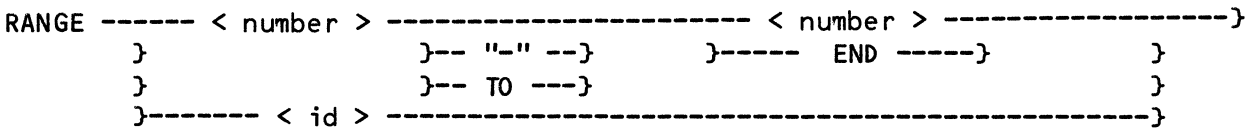
If :RANGE appears only references in the specified range will be listed; if no range option appears, the default range (set by the RANGE command) is used.

Examples:

```
REF GETSPACE                        REF GETSPACE:T
REFA DO3:ENV                        REF I OF LEIBNITZ:ENV
REF                                 REFA
REF A OF B OF C                     REF LINX:T:ENV:X
REF A:RANGE 1234566-68888832        REF GETSPACE:RANGE INITIALIZATION
```

[SPCFY for more on RANGEs]

Xref: RANGE



<id> may be qualified; <number>s must be between 0 and 99999999.

RANGE sets the default range specs for REF.

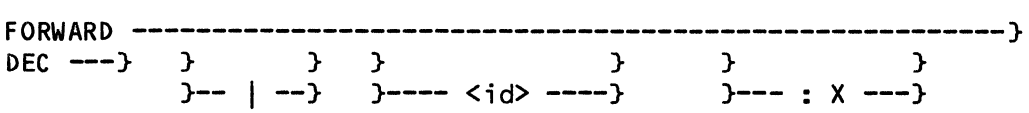
Initially the range is 0-END.

If no :RANGE appears in a REF command, the default range specs set by this command are used.

Examples:

```
RANGE 0-END                         RANGE PRIMARYINITIALIZE
RANGE 12345678 TO 23456789          RANGE SWAPPERSEARCH OF GETSPACE
RANGE 3456 - END
RANGE 1000 2000
```

Xref: DEC and FORWARD



<id> may be an identifier, a qualified identifier, or empty.

DEC moves the offered line to the declaration of <id>.

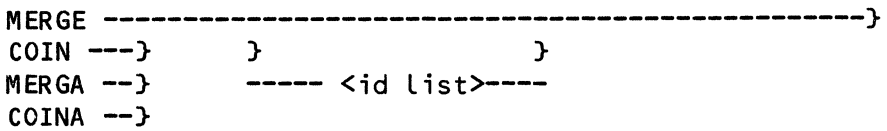
FORWARD move the offered line to the forward declaration of <id>, if any. If no forward declaration exists for <id> then FORWARD moves to the declaration.

If a | appears the cursor is left on the top line.

DEC or FORWARD with an empty <id> moves to the declaration or forward of the work identifier. If <id> is not empty and : X does not appear <id> is established as the work identifier.
Examples:

```
DEC GETSPACE                        DEC
FORWARD FIBSTACK                    DEC A OF B OF C OF D
DEC J OF LEIBNITZ                   FORWARD
DEC| CLOSE                          FORWARD|ACTUALCLOSESERIALWRITE
```

Xref: COIN and MERGE



<id list> is a list of identifiers or qualified identifiers.

MERGE displays a list of references which is the union of the lists of references to each <id> in the <id list>.

COIN displays a list of references which is the intersection of the lists of references to each <id> in the <id list>.

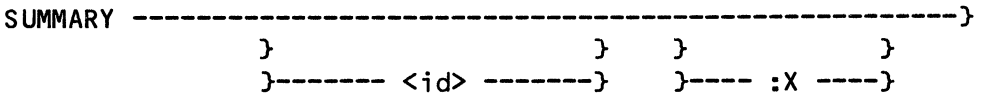
MERGA and COINA do the same thing with aliases included.

Paging action is similar to REF.

Examples:

```
MERGE GETSPACE,FORGETSPACE          COIN GETSPACE,ADDRESSF
MERGE A,B,C OF D                    COIN A OF B OF C OF D,Q
MERGA D03,ARRAYDEC                  COINA TC,UINFO
```

Xref: Summary



<id> may be an identifier, a qualified identifier, or empty.

SUMMARY displays a summary of the references to <id>.

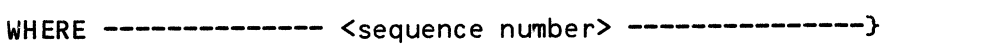
If <id> is empty, the work identifier is used.

If <id> is not empty and :X does not appear, <id> is established as the work identifier.

Examples:

```
SUMMARY GETSPACE                    SUMMARY UINFO:X
SUMMARY
```

Xref: WHERE, ENV mode



The WHERE command clears the screen and displays the inner most environment that contains the <seqeunce number>. The editor then waits for some input and restores the screen to its previous contents.

CNTL 94 toggles environment mode. When environment mode is on, the top line of the screen contains a description of the environment of the offered line. Environment mode is initially off.

If there is no environment for the line in question "not in any procedure" will be displayed.

Example:

```
WHERE 12345678
```

CTRL 2n: Move or Copy

Moving or copying of one or more lines is accomplished with the CTRL 2n family:

21/25 marks the offered line as the first to be moved/copied.
22/26 marks the offered line as the last to be moved/copied.
23/27 marks the offered line as the only one to be moved/copied (first & last).
28 marks the predecessor of the offered line as the destination: insert "before"
29 marks the offered line as the destination: insert "after"
20 cancels any move specifications. 24 toggles copy-continuous mode.

First, last and destination may be marked in any order. While any but not all are specified, the status line shows MOVE or COPY followed by one or two of the letters F, L and D. As soon as all three items are defined, the move or copy takes place: the "first" through the "last" line are inserted (unnumbered) after the "destination." Move deletes the lines from their former site. Copy resets the mark fields of copied lines to blank or the MARK value. In copy-continuous mode, the first and last line markers are retained after the operation is done.

If the "first" line comes after the "last" line, or if the "destination" lies within the group of lines being moved, the operation is aborted. No move or copy is accepted if a delete/renew, number, or paragraph control is pending.

Remember to do a SPCFY to offer the target lines before doing the CTRL 2n. If the offered line is an insertion, the CTRL input is rejected.

CTRL 3n: Delete or Renew

Deletion of one or more lines, or renewal of the symbol file without changes, is accomplished by the CTRL 3n family.

31/35 marks the offered line as the first to be deleted/renewed.
32/36 marks the offered line as the last to be deleted/renewed.
33/37 marks the offered line as the only one to be deleted/renewed (1st & last).
30 cancels any delete specifications. 34, 38 and 39 are invalid.

First and last may be specified in either order. While one but not both is specified, the status line shows DELETE or RENEW followed by letter F or L. As soon as both items are specified, the action takes place: Delete merely removes the "first" through the "last" line from the effective workfile. Renew removes those lines, and then restores any source-file lines that appeared between the predecessor of "first" and the successor of "last". A renew is rejected unless both the predecessor and successor have sequence numbers.

If the "first" line comes after the "last" line, the operation is aborted.

Remember to do a SPCFY to offer the target lines before doing the CTRL 3n. If the offered line is an insertion, the CTRL input is rejected.

A single-line delete or renew control (33 or 37), but no other, will be accepted when a move/copy or number (2n or 4n) control is pending.

CTRL 4n: (Re)Number lines

Numbering (or renumbering) of lines is accomplished with the CTRL 4n family. The range may be specified explicitly, or may be the "current" unnumbered area, or may be all unnumbered areas. The base and increment may be selected automatically, or specified by the user (options 45 through 48).

41/45 marks the offered line as the first to be numbered.
42/46 marks the offered line as the last to be numbered.
43/47 marks the offered line as the only one to be numbered (first & last).
44/48 causes a group of consecutive unnumbered lines to be numbered; the offered line must be in or immediately follow the group intended.
49 causes numbering of all groups of unnumbered lines in the file.
40 cancels any number specifications.

For explicit range selection, first and last may be marked in either order. When one but not both is specified, the status line shows NUMBER followed by letter F or L. As soon as both items are specified, the numbering takes place. If the "first" line comes after the "last" line, the operation is aborted.

If the CTRL input that initiates numbering is 45 through 48, you will be asked to provide the increment and optionally the base. If you provide no base, a multiple of the increment near the beginning of the available range will be chosen. If your increment and/or base don't fit, you will be asked again.

[Continued: push SPCFY]

CTRL 4n [2nd of 2]

If the CTRL input that initiates numbering is 41 through 44, a base and increment will be chosen automatically. The increment will be the largest one of the form 1, 2 or 5 times a power of ten that will fit. The base will then be chosen to center the numbers in the available gap. When an explicit range is specified for automatic numbering, the lines on each side of that range must have numbers; you can't automatically number part of an unnumbered area. (The boundaries of the whole file are "numbered" -1 and 100000000.)

The automatic increment calculation is bounded by parameters MINI and MAXI, whose default values are 1 and 2000. They may be altered by the MINI and MAXI commands. Whenever a NUMBER is pending or the limits are non-default, the limits are displayed on the status line in the form "mini <= I <= maxi". The interpolated increment will not exceed MAXI. If the numbering cannot be accomplished with an increment >= MINI, the boundaries of the numbering range will be extended until the number gap is big enough. No warning is given.

Remember to do a SPCFY to offer the target lines before doing the CTRL 3n. If the offered line is an insertion, the new line is discarded and the function is performed as though the line following the insertion were offered.

No number control is accepted while a move/copy or delete/renew (2n or 3n) control is pending.

CTRL 5n: Paragraph, Shift

A group of lines can be reformatted with margins, indentation, centering, and right-justification, or moved left or right unchanged:

CNTL 51/55 marks the offered line as the first line to paragraph/shift.
52/56 marks the offered line as the last line to paragraph/shift.
53/57 marks the offered line as the only line to paragraph/shift.
54 centers the offered line between the margins.
58 is invalid. 59 swaps margins specifications (see margins).

A paragraph is reformatted with as many words per line as possible. A word in this context is anything between blanks. The old paragraph is deleted and the new one is inserted (unsequenced) in its place. If right-justification mode is on, blanks are inserted randomly between words to fill the line out to the right margin.
Shifting simply causes the specified lines to be shifted left or right. If a line wont fit the operation is suspended; the line can then be fixed manually and the operation restarted.
The commands LEFTM, RIGHTM, SHIFT, and INDENT set up the margin specifications, RIGHTJ toggles right-justification mode.
A paragraph or shift control is rejected if a delete/renew, move/copy, or renumber control is pending; 54 and 59 are OK.

[SPCFY for details on margins]

Paragraph, Align: Margins

Margins for Paragraphing mode are set by the commands LEFTM, RIGHTM, and INDENT.

LEFTM specifies the column number of the left margin.
RIGHTM specifies the column number of the right margin.
INDENT specifies how many columns to indent the first line in addition to LEFTM.
SHIFTL and SHIFTR specify how many columns to shift left/right.

An error will be given if an attempt is made to violate the relation
1 <= LEFTM+INDENT < RIGHTM <= 72.

Column 1 is the leftmost column of the text field of an offered line; column 72 is the rightmost.

SHIFTL and SHIFTR must be in the range 1-71.

The initial specs are LEFTM=1; RIGHTM=72; INDENT=0; RIGHTJ=FALSE.
The initial saved specs are LEFTM=5; RIGHTM=65; INDENT=5; RIGHTJ=TRUE.

Whenever a paragraph command is pending the margin specifications are displayed on the top line.

CNTL 59 saves the current margin specs and installs the last set of saved specs.

CTRL 6n: Character substitution

Several ASCII-67 characters are unavailable on the "modified ASCII" keyboard. A character-substitution mechanism is provided to make these characters available when essential. The facility is redundant with full-ASCII keyboards.

The character | is used as the substitutable character. Two modes of operation are provided. When a | is encountered on a new or modified line:
  a: the character will automatically be replaced by a preset substitute, or
  b: the user will be asked to specify a substitute for this instance.
The default case does nothing: it leaves | as |.

CTRL 61 through 68 preset the substitution as follows; CTRL 69 is not used.

CTRL	61	62	63	64	65	66	67	68
substitutes	^	_	\	!	~	|	{	?

CTRL 60 causes querying the user. Then when a new | is encountered, the editor will put the cursor on it and ask the user to enter a CTRL sequence to specify his choice; the menu of characters will be shown on the top line.

The status line shows "|=\" or the like when direct substitution is armed, or "| ?" when query is armed. The default case |=| (CTRL 66) is not displayed. The current substitution specification is retained in the ALTER file.

CTRL 91: Check sequence numbering

CTRL 91 causes the editor to scan the workfile forward from the current display position, seeking any line with no number or whose number does not exceed that of the preceding line. The first such line is then offered. It (or other lines) may be deleted, moved or numbered using the CTRL 2n, 3n or 4n facilities.

If the end of the workfile is reached without any sequence error, an appropriate message is displayed.

About every 2-3 seconds, the editor displays the sequence number of the line being checked. If the editor receives any input from the user terminal, the scan stops and the input is ignored.

Note that the CTRL 91 facility scans the unchanged symbol file as well as the ALTER file, so its time is proportional to the length of the effective workfile. The CTRL 73 or CTRL 83 facility will quickly locate unsequenced areas in the alterations without scanning; teach 8 for details.

CTRL 93: Join two lines into one

CTRL 93 causes the editor to append the text on the offered line to that of the preceding line and then delete the offered line; it leaves the cursor at the junction. If the last character on the previous line and the first character on the offered line are both alphanumeric, one blank is interposed.

If there is insufficient room on the preceding line for the combined text, the function is rejected.

CTRL 95: CAPS (upcasing) on TD830

The CTRL 95 function toggles the CAPS mode. If the mode is set, CAPS is displayed on the status line, and the terminal is caused to upcase all letters upon entry.

The terminal can automatically translate lower-case letters to upper-case as they are typed. Unfortunatly, it also alters some other characters: {, ?, !, |, ~ and ? become [,], \, ^, @, and _.

You may switch the terminal upcasing on and off by entering CTRL Y or CTRL T, respectively. (Do not XMT; the effect is immediate upon the terminal. You may use lower-case y or t as well as upper case Y or T. Do not hold shift when you push CTRL, or the terminal will remain in CTRL state. If the CTRL light remains on, push LOCAL.) Any change made with CTRL T while in CAPS mode will be temporary; the editor will set terminal upcasing again for the next line.

The CAPS mode is saved in the ALTER file for recovery.

CAPS is initialized true when the CANDE workfile is of a symbolic TYPE. Setting CAPS mode resets SHIFT mode.

[CTRL 94 provides another case-shift mechanism; SPCFY for details.]

Upcase and Downcase

UPCASE toggles UPCASE mode and resets DOWNCASE mode. DOWNCASE toggles DOWNCASE mode and resets UPCASE mode.

These modes provide a simple way of changing the case of a few lines of your file.

In UPCASE mode all letters are translated to upper case when the offered line is displayed. If the line is then retransmitted, it will be stored in upper case.

In DOWNCASE mode all letters are translated to lower case when the offered line is displayed. If the line is then retransmitted, it will be stored in lower case.

The UPCASE and DOWNCASE modes are independent of CAPS mode, but are inhibited when in SHIFT mode.

(These commands deal with up- or down-casing lines in the file. There is a facility to upcase the input text from the keyboard; teach CAPS for details.)

Command COL, CTRL 96: Fixed column

The CTRL 96 function and the COL command control fixed-column mode. CTRL 96 toggles the mode; the command explicitly sets or resets it:

COL OFF	resets column mode.
COL	sets column mode at column 1.
COL = n	sets column mode at column n, where n is an integer (1 to 72) and the = is optional.
CTRL 96	resets column mode if it is set, sets column mode at the current (or most recent) cursor position if the mode is reset.

In column mode, the cursor is placed at the fixed column instead of being set at the first non-blank position. The feature is convenient for making changes at the same place in a group of lines.

In column mode, any SPCFY to a line of the file (offered or not), or to the offered new line in duplicate mode, resets the column number. Therefore the line-split feature and the insert-before feature of SPCFY are unavailable, as is the ability to exit duplicate mode with SPCFY on the offered line. To select another line without changing column, SPCFY with the cursor in the number field.

While column mode is active, COL=n is displayed on the status line.

CTRL 97: Duplication

CTRL 97 controls duplicate mode:

Duplicate mode is the same as insertion mode, except that each new line offered is preset to be a copy of the preceding line. You make the changes desired and then XMT the new line; a copy of that becomes the starting point for the next new line.

If you are already in inserting mode, CTRL 97 simply turns on (or off) the duplication variation. If you in normal editing mode, CTRL 97 puts you into insertion mode by presenting a new line just ahead of the offered line, with a duplicate of its predecessor.

The same actions that terminate insertion mode terminate duplicate as well, except that SPCFY on the offered duplicate line when in fixed-column mode resets the column rather than terminating duplication/insertion.

CTRL 99 & 98: To command or edit

CTRL 99 puts the editor into command state, so that a command may be entered. It is useful in edit or teach state. If the editor was in insertion or duplication state, the offered new line is discarded.

CTRL 98 puts the editor back into edit state (change, insert or duplicate) at the previous point. It may be used in command, teach, or edit state.

When used in edit state, CTRL 98 will go to a prior edit point. When a new editing point is determined by a SPCFY on some line other than the current offering (or its immediate predecessor), or by a locate command or other action, the current editing point is saved. A later CTRL 98 will return to that point (saving the then current point). You can interrupt a sequence of editing to go fix some problem just noticed, and then conveniently return to your main stream.

When the editor enters command state (by CTRL 99 or other means), it saves the prior edit point. If you return to that point by CTRL 98, another immediate CTRL 98 will stay there: the "current" is the same as the "previous" point.

When the editor is first started, or when editing proceeds past the last line of the file, the "previous" point is defined as an insertion after the last line.

For a more flexible repositioning mechanism, teach REPOSITION.

CTRL 00, 09, 90: Refresh, Retract

CTRL 00 will refresh the screen. Ordinarly, the editor sends only as much information as necessary to show changed information and repair the damage from your transmisssion and automatic response; the rest of the screen remains valid. The screen may become incorrect as a result of your receiving messages directly from CANDE, or by your error (for instance, moving the cursor to another line and XMTing without SPCFYing), or by data-transmission retry on a TD820. Any time you question the validity of the screen, use CTRL 00.

CTRL 09 will return the workfile to the state before the most recent operation (other than CTRL 0n). CTRL 00 has no effect on the restorability of a prior command. Once CTRL 09 has acted, it will do nothing more until after some other operation. Any other operation renders all prior operations irretrievable.

CTRL 09 provides a second chance for an overwide deletion, an unfortunate move, an intended insert-before that became a split, etc.

CTRL 90 is a NO-OP command; it causes the editor to restore the line currently offered and the top line, but makes no changes. It does not refresh the full screen.

CTRL 01 through CTRL 08 are used for markers; SPCFY for details.

Repositioning with markers

The inputs 01-08 are used to place markers in the effective file which can then be used to reposition the display.

There are 4 markers available. To set a marker the even numbers are used : these set a marker to the current offered line. The odd numbers are used for moving around: these reposition the display to offer a previously-marked line.

01 goes to marker 1	02 sets marker 1
03 goes to marker 2	04 sets marker 2
05 goes to marker 3	06 sets marker 3
07 goes to marker 4	08 sets marker 4

An attempt to go to a marker that has not been set is a noop.

Setting a marker when on the command line cancels the marker.

A marker is canceled if the line it points to is deleted. A marker is moved if the line it points to is moved. Renumbering does not affect markers.

End-of-editing commands

The END command (or its synonyms BYE, STOP or QUIT) causes the editor to update the worksource and return control to CANDE. The recovery (ALTER) file is purged unless :REC appears. Any unnumbered lines are assigned sequence numbers as though CTRL 49 were used, unless the editor is in PATCH mode and :UNSEQ appears; in that case any unnumbered lines are written with blanks in the sequence-number field.

The FORGET command causes the editor to purge the ALTER file and return to CANDE without updating the worksource.

The RECESS command causes the editor to return to CANDE without updating the worksource, but it leaves its ALTER file for later recovery, so editing may be resumed at a later opportunity. Teach FILES for more data on recovery.

A file name may follow the word RECESS, in which case the recovery file title will be changed as specified. To resume editing this workfile, you may run the editor with the recovery file title specified in the UTILITY command parameter, preceded by an exclamation point (!). The full recovery-file title used, including usercode and family if any, is displayed by the RECESS command.

The CANDE control command ?END has essentially the same effect as RECESS.

Command SAVE

```
SAVE ------------------------------------------------------------------>
       !-- PATCH  --!    !-- : UNSEQ --!    !-- AS <filename> --!
       !-- MERGED --!
```

The SAVE command causes the editing results to be written to a file. If PATCH or MERGED appears, a patch or a merged new file is written, respectively; if neither appears, the editing mode determines the type of output.

If the SAVE mode is PATCH and :UNSEQ appears, any unnumbered lines will be written with blank sequence-number fields; otherwise they will be temporarily numbered as though CTRL 49 had preceded the output and CTRL 09 followed.

If an AS clause appears, that filename is used for the output file and the editor continues; otherwise a new worksource is written and the editor returns control to CANDE, purging the recovery (ALTER) file.

If an ALTER file has been recovered, the attributes it specifies are used for the output file of a SAVE. These attributes were supplied by CANDE to the invocation of the editor that created the ALTER file, according to the workfile attributes at that time; they may differ from the attributes supplied to the current invocation of the editor. In this respect a simple SAVE differs from an END command, which uses the current CANDE-supplied attributes.

INCLUDE command: incorporate patch

The INCLUDE command will cause a specified patch file to be included in the current workfile. Through this facility, you may make several patch files against the same symbol file, and then combine them.

The name of the file to be INCLUDEd follows the word INCLUDE in the command.

The INCLUDE file must be in the format of a PATCH output of the editor or the PATCH file from SYSTEM/PATCH. Specifically, it should not involve SYSTEM/PATCH controls or editing controls like $SET SEQ or $GO TO.

Any patch marks in the INCLUDEd file will be retained; none will be supplied.

The changes effected in the INCLUDEd patch are henceforth displayed as changes, as though they had been entered through the keyboard or picked up in an initial PATCH-mode worksource.

Each INCLUDE operation applies to the effective workfile, so the effect of successive INCLUDEs is the same as that of a SYSTEM/PATCH run with the same patches in the same order (which is the same as successive compilations with one patch at a time).

WHAT and STATUS commands

The WHAT command causes the names of all relevant files to be displayed. The format is much like that used in the initial display when the editor is started.

The STATUS command causes display of assorted numbers that reflect the state of the current invocation of the editor, the alter file, and the workfile.

Any input received (e.g. a SPCFY) causes the previous workfile display to be refreshed; the input is ignored.

Commands MAXI, MINI

The MINI and MAXI commands specify minimum and maximum values for automatically generated sequence increments. The syntax is

MINI=n	or	MAXI=n	where n is an integer (1 to 999999) and the = is optional.

If the new assignment would make MINI > MAXI or MINI = 0, it is rejected.

For the semantics of these limits, see the number facility: teach 4 or CTRL 14.

The MAXI and MINI values are shown on the status line if either value differs from its default (2<=I<=2000), unless there is a non-blank MARK specification occupying that part of the line. MAXI and MINI are shown unconditionally if if there is a numbering (CTRL 4n) pending.

MAXI and MINI values are retained in the recovery file.

Command MARK: Provide ID in 81-90

Many symbol files consist of 90-column images; columns 81-90 are not edited directly but are altered whenever a line is explicitly changed or a new line entered. By default, these columns are blanked.

The MARK command specifies the string to be used in the field. At most ten characters are used; leading blanks are ignored and trailing blanks are assumed if needed; there may be imbedded blanks.

If no text follows MARK, the mark field is reset to blanks.

```
Examples:  MARK 29.048.011
           MARK AUG 20
           MARK    <retained><discarded>
           MARK
```

The current mark value is displayed on the status line if it is non-blank, unless there is a number (4n) control pending (in which case that space is preempted for the MAXI/MINI data).

The mark value is retained in the ALTER file for recovery.

The mark field of a line is not affected by the line being moved or renumbered.

DUMP, TRACE, BUG, PRINT TEACHER

Three debugging commands exist:

DUMP causes a PROGRAMDUMP(ARRAYS).

TRACE causes a formatted listing of the editor main table to be written.

BUG n or BUG -n causes certain diagnostic actions:

```
bit 0: Print all remote-file input/output to the line printer.
bit 1: Print the main table whenever it is written to workfile.
bit 2: Print the interpolation data and results for automatic renumbering.
bit 3: Don't purge the workfile after an END or a default SAVE.
sign:  Purge the printer backup file upon normal termination if n <= 0.
```

All debugging output is written to the TASKFILE of the editor task.

The PRINT TEACHER command is a variation of PRINT TEACH: it prints all the tutorial pages, including each version of the terminal-selected pages. The page numbers, selection codes and keywords for each page are also shown.

The TD830 Keyboard

TD830 Keyboard (full ASCII)