CRAY

RESEARCH, INC.

RECORD OF REVISION

CRAY RESEARCH, INC.

PUBLICATION NUMBER SG-0055

Each time this manual is revised and reprinted, all changes issued against the previous version in the form of change packets are incorporated into the new version and the new version is assigned an alphabetic level. Between reprints, changes may be issued against the current version in the form of change packets. Each change packet is assigned a numeric designator, starting with 01 for the first change packet of each revision level.

Every page changed by a reprint or by a change packet has the revision level and change packet number in the lower righthand corner. Changes to part of a page are noted by a change bar along the margin of the page. A change bar in the margin opposite the page number indicates that the entire page is new; a dot in the same place indicates that information has been moved from one page to another, but has not otherwise changed.

Requests for copies of Cray Research, Inc. publications and comments about these publications should be directed to:

CRAY RESEARCH, INC.,
1440 Northland Drive,
Mendota Heights, Minnesota 55120

Revision	Description
	May, 1982 - Original printing.
01	March, 1983 - This change packet brings the manual into agreement with version 1.12 of COS.
A	June, 1984 - This reprint incorporates the original printing with change packet 01. No other changes have been made.
A-01	December, 1984 - Change packet deleted 'SYS' command and brings documentation into agreement with COS 1.14. (Appendix A was deleted and Appendixes B and C are now A and B respectively.)

PREFACE

This publication describes the external features of TEDI, the Cray Research, Inc., interactive text editor. TEDI is designed to edit text online.

It is assumed the reader has knowledge of the Cray Operating System (COS) and is familiar with the CRAY-OS Version 1 Reference Manual, publication SR-0011.

Cray Research, Inc., gratefully acknowledges the work of Dr. Clair Neilson in the development of the predecessor to TEDI.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CONTENTS

3. COMMAND FUNCTIONS (continued)

COMMAND SYNTAX (continued)

COMMAND SYNTAX (continued)

APPENDIX SECTION

FIGURE

TABLES

INTRODUCTION 1

Text Editor (TEDI) is an interactive line editor on a Cray Computer System operating under control of the Cray Operating System (COS). TEDI can be used to edit computer programs, data, documentation, or any other text files.

TEDI CONTROL STATEMENT

Format:

```
TEDI,DN=dn,I=idn,L=ldn.
```

Parameters:

DN=*dn* Name of the dataset to be edited. If this parameter is omitted, TEDI uses the first input command of the dataset *idn* as the name of the dataset to be edited. Keyword with value unassigned is not allowed.

I=*idn* Name of the dataset containing the editor commands. (Input is entered at the terminal keyboard in interactive mode.) The default is $IN.

L=*ldn* Name of the dataset used for TEDI output listing. Default is $OUT.

TEDI DATASETS

TEDI operates in command and enter modes and edits COS-blocked datasets through TEDI commands. TEDI datasets are called input dataset, current dataset, output dataset (depending on the stage of processing), run-time switch dataset, and command dataset. The user defines the *input dataset* through the COS control statement TEDI,DN=*dn*. When the input dataset is being edited, it is referred to as the *current dataset*.

(The *output dataset* contains the text TEDI has updated.)

The user might want to call an input dataset composed of frequently needed commands rather than repeating the commands. The user can call such a dataset (referred to as a *command dataset*) while editing the current dataset by using the U command (see section 3). The *run-time switch dataset* ($TEDISW) records run-time switch settings for an editing session (refer to Run-time switches, this section, and Run-time switch commands, section 3).

Figure 1-1 shows the interrelationship of the TEDI datasets.

TEDI REGISTERS

TEDI maintains symbolic registers (R0 through R9) as they are altered by the user. These registers permit the user to access a particular value repeatedly and can be set to a value that replaces a numeric value in a command. The Set register (SR) command loads the registers.

Example:

SR 1,.	Loads the current line number into register R1
T R1	Displays the line pointed to by the contents of register R1

RUN-TIME SWITCHES

The user can exit from TEDI without losing switch settings by using a *run-time switch* dataset. A run-time switch is a symbolic switch that sets and saves parameters in TEDI (for example, tab settings). If the dataset name is not provided with the control statement when execution begins, TEDI reloads the existing run-time switch settings. If the switch dataset is nonexistent or the user provides a dataset name, TEDI resets all switches to their default values. The user can examine and modify the switch settings with TEDI commands. (Refer to sections 2 and 3 for detailed information on using the run-time switches.)

Releasing the run-time switch dataset or editing a new dataset forces the switches to their default conditions.

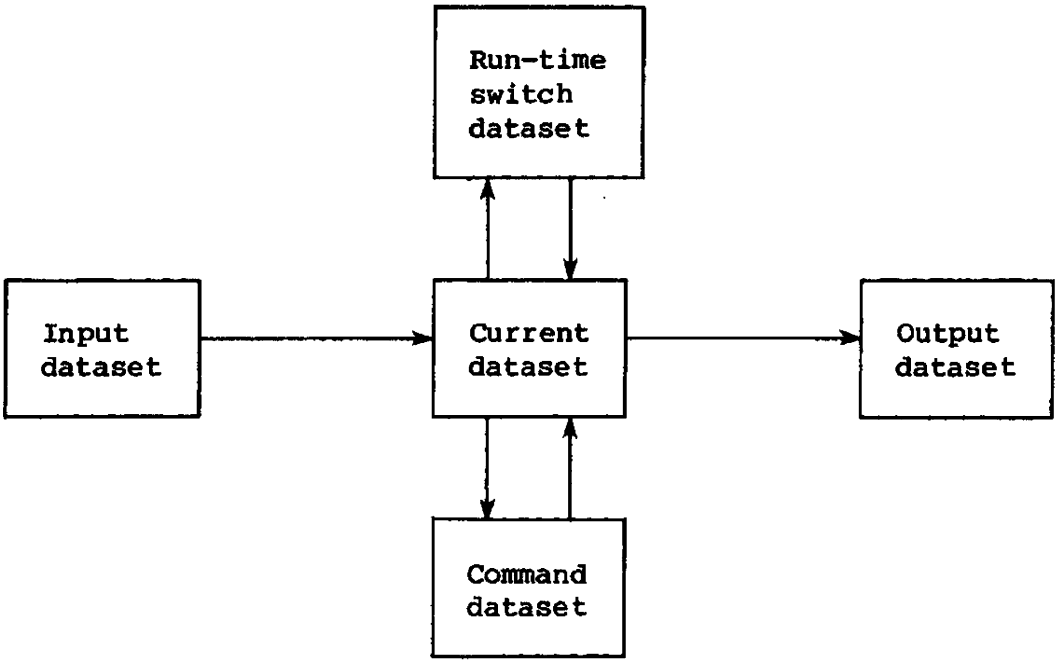

Figure 1-1. TEDI data flow

GENERAL DATASET PROCESSING CHARACTERISTICS

When TEDI overwrites, or alters the length of, a permanent dataset, maintenance permission and unique access are required. TEDI prompts for permission (YES or NO) from the user before the command is continued. TEDI terminates execution without updating the dataset when there is no input after END command in a batch job. When TEDI edits a permanent dataset, the edition number is not updated.

During initialization, COS ends of file are replaced by an end-of-file character string (<EOF>). The user can add or delete end-of-file character strings while editing. During post-processing, TEDI converts the <EOF> strings back into COS EOFs.

TERMINAL ATTENTION INTERRUPT

When an entry is terminated by a Terminal Attention Interrupt, TEDI handles the interrupt as follows:

- All dataset editing commands are continued to completion. Only the terminal output is suppressed.
- An interrupt during enter mode cancels all additions. No update occurs.
- Iterative commands (see section 3) are suspended after the iteration in which the interrupt was received.

COMMAND RANGE

TEDI uses line numbers to reference the contents of the dataset by assigning a number to each line of the current dataset. Each time the dataset is modified, TEDI reassigns the numbers. Therefore, a line number is not permanently assigned to a line but is implied by its relative position.

Line numbers are consecutive (1, 2, 3, ...]) to the last line. Every editing operation that adds or deletes or moves lines renumbers the dataset.

Commands that modify or display text can be restricted to a range of lines. Additional restrictions can be placed on the command range by specifying a column range, or a pattern or symbol that must match.

TEDI restricts string, line, and column length in the following ways.

- Insertion and replacement strings are limited to 64 characters.
- The maximum TEDI line length is 150 characters.
- The maximum TEDI line range is $1 < m \leq]$ (] indicates the last line number of the dataset).
- The TEDI column range is $1 < c \leq 150$.
- In commands requiring column number specifications, *m,n* must specify the line range.

Certain commands accept column numbers as input. The column numbers must be entered with the command mnemonic (they are not prompted). Some commands, such as **RC** (see section 4), accept either a column range or a single column number; other commands require only a single column number (for example, see command **AC** in section 4).

COMMAND STRUCTURE

The basic format of a TEDI command is a verb optionally followed by numeric arguments and string arguments. TEDI commands are described in detail in section 4 of this manual. Appendix B contains a basic set of commands for the new user.

A TEDI verb is a 1-, 2-, or 3-character alphabetic mnemonic, for example, TA (type after). Entry of a command is explained in section 2.

Numeric arguments specify numeric information such as line number, column number, increment value, and number of display lines. Numeric arguments, separated from each other by commas or spaces, can immediately follow the verb.

String arguments include symbols, patterns, and meta-string symbols and patterns and are required by some commands. String arguments are separated by escape characters or line feeds.

Numeric and string arguments are discussed in more detail later.

CONVENTIONS

In this manual the following conventions are used to illustrate commands and TEDI's replies.

UPPERCASE	Identifies the command verb or literal parameter
Italics	Define generic terms that represent the words or symbols to be supplied by the user
<> Angle brackets	Enclose optional portions of a command format
... Ellipsis	Indicates the omission of one or more words, terms, or characters that are obviously understood
BOLD	Highlights information in the text

$	User-supplied escape character
*	Prompt displayed in command mode
&	Prompt displayed in enter mode
-	Prompt displayed in system mode

NOTE

The preceding characters *, &, and - are often dependant on the front end and can be altered (by default) by the site.

USING TEDI

This section explains how to use TEDI for creating and editing a dataset and terminating an editing session. Command entry and termination are discussed.

MODE

TEDI operates in command mode and enter mode. In command mode, TEDI edits existing text and displays an * when ready to accept the next command.

Enter mode permits entry of new text. In this mode, TEDI prompts entries with &. The following commands put TEDI in enter mode.

AL	after line
BL	before line
RL	replace line

To exit from enter mode to command mode, enter a period in column 1, followed by a carriage return.

COMMAND ENTRY

Enter a command verb in uppercase or lowercase letters or a combination; enter patterns, symbols, and line additions exactly as needed. End the command with a carriage return or an escape character. Using a carriage return with a string argument elicits the appropriate prompt (see prompts later in this section) on the screen. Using an escape character suppresses the prompt, allowing the command and string argument information to be strung together. (Escape characters are explained later in this section.)

Use commas or blanks to separate elements in a numeric argument. Use a carriage return or an escape character to separate numeric arguments from string arguments and string arguments from each other.

Example:

```
tb 206,9
```

tb	Type before
206	Line number
9	Maximum number of lines to be typed

SPECIAL CHARACTERS

When TEDI is in enter mode, special characters can be used as instructions or for indentation control. For indentation control, automatic line indentation must be enabled (see the SI command, section 4). Then the first character of each added line is recognized as the indentation control character.

Table 2-1 lists the special characters and their descriptions. The \ , */abcde/*, !, and _ characters can be used only with the X command. Their special meanings cannot be altered.

ESCAPE CHARACTER

The *escape character* can be used instead of the carriage return to separate string arguments. Using the escape character causes TEDI to suppress prompts between each string argument.

The escape character can be either the escape key or a user-supplied character that functions in the same way. (In this manual, both of these characters are represented by **$**.) If an escape character is used, indicate the additional argument on the same line and suppress the prompt.

The *user-supplied escape character* is determined on a line-by-line basis only in command mode. The first occurrence of a nonalphanumeric character not recognized by TEDI as a special character (refer to Special characters, this section) is the escape character. The user-supplied escape character stays the same until another command line supplies an escape character.

Table 2-1. Special characters

Character	Description
Tab	Skip to the next tab stop. (See tab commands, STS and STC, in section 4.)
"	Establish the next character as text input. " allows possible control characters to be entered as text. A doubled quote ("") specifies a single quote.
.	Do not indent this line.
+	Increase indentation level.
-	Decrease indentation level.
0	Return to first indentation level.
:	FORTRAN labeled line; indent after label.
,	FORTRAN continued line; indent to first level.
\	Delete character.
Λ*abcde*Λ	Insert characters *abcde* where indicated by first Λ .
!	Split line before the indicated character.
_	Replace character with blank.

PROMPTS

When a carriage return is used after verb and parameter entry, TEDI prompts for the additional data needed to complete the command. Table 2-2 lists the prompts TEDI displays and description of the action to be taken by the user.

To elicit a prompt, enter the verb and parameters, followed by a carriage return. To suppress a prompt, enter the verb and parameters, followed by an escape character.

Table 2-2. Prompts

Prompt	Description
*	Enter a TEDI command.
&	Enter a line to add to current dataset.
CHAR:	Enter any character including control characters.
DN:	Enter a legal COS dataset name.
I:	Enter a string to be inserted.
R:	Enter a replacement string.
P:	Enter a pattern used as a search target.
S:	Enter a symbol used as a search target.
X:	Enter a pattern used as a search target to search for a meta-string construction.
Y:	Enter a symbol used as a search target to search for a meta-string construction.

Example:

```
AP 5,8
```

TEDI searches lines 5 through 8 to insert a new string after a specified pattern. If a carriage return is entered after the 8, the prompt, P:, appears on the screen, allowing for entry of the pattern. Another carriage return used after the pattern is entered causes a second prompt, I:, to appear. Now enter the string to be inserted after the pattern. TEDI searches for all occurrences of the specified pattern within the line range and adds the specified string.

In the following example, by choosing the escape character rather than a carriage return after the 8, the user can enter the pattern, another escape character, and the string to be inserted on a single line.

AP 5,8$*file*$*dataset*

where *file* is the pattern and *dataset* is the string to be inserted.

USING NUMERIC ARGUMENTS

Use the special characters shown in table 2-3 to replace line numbers in a command. For example, * is a shortened version of the line number and line increment designation (represented as *m,n* in command descriptions, section 4) when the entire dataset is indicated.

Table 2-3. Command line number replacements

Character	Description
*	Indicates the entire dataset
]	Indicates the last line in the dataset
.	Indicates the current line, that is, the last line that was displayed or changed

A question mark used as the first argument in a command displays the command's format and function. When it is used alone in place of the command mnemonic, it displays all TEDI commands.

EXPRESSIONS

TEDI recognizes simple arithmetic expressions using the addition (+) and subtraction (-) operators. An expression can replace any simple numeric argument.

Examples:

The expression **.-10** addresses a line 10 lines above the current line.

The expression **.-10,.+10** specifies a 21-line range encompassing the current line.

USING STRING ARGUMENTS

A command statement can require *string arguments*. The string arguments include symbols, patterns, and meta-string symbols and patterns. String arguments are separated by escape characters.

SYMBOL AND PATTERN

TEDI distinguishes between patterns and symbols. A *pattern* is any string of characters; a *symbol* is one character or a string of contiguous characters delimited by spaces or separators (not 0-9 or A-Z). This distinction provides a means for locating and replacing variables in a dataset. Specific symbol-matching commands (for example, **TS** means display symbol or **RS** means replace symbol) can distinguish all occurrences of the specified symbol from identical characters or strings embedded in symbols or in patterns. For example, all occurrences of the FORTRAN variable I can be distinguished from the letter I both in the symbol PI and in the FORTRAN statement **IF**.

META-STRING ARGUMENT

The *meta-string argument* is a powerful tool in the TEDI command inventory. It uses nonalphanumeric characters to match single characters and to construct pattern- or symbol-matching character groups and strings embodying some optional feature, usually generalized character identity or quantity.

The characters in table 2-4 are recognized as meta-string constructions when entered as the pattern or symbol part of an appropriate command.

Examples:

Consider the following meta-strings and their matches.

?ickory matches **hickory** and **dickory**. **??ck...** matches **hickory**, **dickory** and **dock**.

wh...@? and **wh...!** both match **who?**, **what?**, **when?**, **where?** and **why?**.

The meta-string **ma...** matches **ma**, **mama** and **mamama** in the string **mamamamamam**. The string containing the least number of characters, **ma**, is selected. Therefore, the result of the command **RX$ma$+$** when applied to the string **mamaxx** is **++xx**

Table 2-4. Meta-string characters

Character	Description
?	Matches any one character (including separators). For example, **g??e** could mean give or gone.
!	Matches any nonalphanumeric character (separator)
@	The character that follows @ is the actual keyed-in character and is not part of a meta-string construction.
[*xyz*]	Matches any one of the enclosed characters. For example, **[abc]** matches either the **a**, the **b**, or the **c**, independent of each other.
...	Matches any character string, including words and phrases. For example, **t...s** matches the string beginning with **t** and ending with **s**.

COMMAND TERMINATION

To terminate the execution of a command, use the terminal attention interrupt on your specific terminal. See section 1 for TEDI's handling of interrupts.

INTERACTIVE EDITING SESSION

The process for using TEDI follows:

1. Log on to the Cray mainframe. For logging-on procedures for the interactive IOS station, refer to the I/O Subsystem (IOS) Operator's Guide, CRI publication SG-0051.

2. When the prompt ! appears, enter the TEDI job control statement, specifying the name of the dataset to be edited. Typing TEDI without a dataset name either loads the last dataset updated or prompts for a name if TEDI cannot find a dataset name.

   ! TEDI, DN=*dn*.

3. When the prompt * is displayed, you are in command mode. Enter any command. (See section 4 for a description of all TEDI commands.)

TERMINATING AN EDITING SESSION

To terminate an editing session, enter the END or the QUI command.

> Enter END to output the current dataset with all the changes recorded. (The edited version of the dataset is saved.)
>
> Enter QUI to leave the original dataset intact. QUI is used if changes made during the editing session are no longer desired.

TYPICAL EDITING SESSION

```
! TEDI, DN=RUN.
- RUN   n lines                       Call TEDI

* ts 1,10$CAT                         Request for lines between first
                                      and tenth (inclusive) containing a
                                      specific symbol

* AL 4                                Add lines after specified location
& FILE

.                                     Return to command mode
```

SETTING RUN-TIME SWITCHES

To examine and modify run-time switch settings, enter the appropriate command after TEDI prompts with *. To display the contents of a run-time switch, use the information command (I).

Saving the switches allows exit and reentry to TEDI without specifying the dataset name or resetting any of the run-time switches in the same interactive session (or batch job).

Example:

To set a tab character for the current dataset, use the STC (set tab character) command.

```
*STC $@

     $    User-supplied escape character

     @    Symbol to set the tab characters for the current dataset
```

To elicit a TEDI prompt, enter this command without $.

```
*STC

CHAR:@
```

COMMAND FUNCTIONS 3

In this section, the commands are described according to function. For a detailed explanation of each command and command entry, refer to section 4, Command Descriptions. Basic commands are discussed in Appendix B.

Table 3-1 contains definitions of the numeric arguments for TEDI commands in this section. Refer to section 2, table 2-3 for command line number replacements.

Table 3-1. Numeric arguments

Argument	Description
c	Column number; must be present.
d	Column number or increment. If *d* is less than *c*, the command applies to columns *c* through *c+d* (increments the column count). If *d* is greater than or equal to *c*, the command applies to columns *c* through *d*. If *d* is omitted, the command applies only to column *c*.
f	Line number
i	Positive integer
m	Line number. If *m* is omitted, the command applies only to the current line.
n	Line number or increment. If *n* is less than *m*, the command applies to lines *m* through *m+n* (increments the line count). If *n* is greater than or equal to *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

COPY COMMANDS

Copy line commands (CA, CB) duplicate the line range *m,n* either before or after line *f*. Copy dataset commands (CDA, CDB) copy the line range *m,n* of **DN:** either before or after line *f*. The dataset to be copied must be local to the job. During a dataset copy, TEDI replaces each logical end of file with the end-of-file string (EOF).

Command	Description
CA *m,n,f*	Copy lines after
CB *m,n,f*	Copy lines before
CDA *m,n,f*	Copy dataset after
CDB *m,n,f*	Copy dataset before

DATASET MANIPULATION COMMANDS

Dataset manipulation commands control the current dataset updating process.

The open dataset command (OD) discards all updates to the current dataset and opens a second dataset for editing. The open command (O) updates the current dataset before opening a second dataset.

Write commands (W, WF, WFN, WR) output all or a portion of the edited current dataset to a second dataset. The **W** command rewrites the current dataset including all changes. The **WD** command allows a specified line range to be written to the second dataset. The **WDN** command prefixes each line written with a 1-word ASCII line number. The **WR** command rewrites the entire dataset to a second dataset. **WD** and **WDN** allow line ranges; **W** and **WR** do not.

The use command (U) directs TEDI to accept commands from a command dataset. The command dataset is read sequentially from its current position until the end of the dataset or a command error is encountered. Control then reverts to the console. Nested **U** commands (that is, a **U** command in a command dataset) are illegal.

Command	Description
O	Open dataset
OD	Open dataset, discard updates

Command	Description
U	Use command
W	Rewrite dataset
WD	Write dataset
WDN *<m,n>*	Write dataset numbered
WR	Write dataset to another dataset

DELETE COMMANDS

Delete commands delete all or parts of lines in a specified range. Delete commands referring to before or after, delete from the beginning of the line (before) or from a specified point to the end of the line (after).

Command	Description
DL *<m,n>*	Delete line
DC *m,n,c<,d>*	Delete column
DP *<m,n>*	Delete pattern
DAP *<m,n>*	Delete after pattern
DBP *<m,n>*	Delete before pattern
DS *<m,n>*	Delete symbol
DAS *<m,n>*	Delete after symbol
DBS *<m,n>*	Delete before symbol

Command	Description
DX *<m,n>*	Delete meta-string pattern
DAX *<m,n>*	Delete after meta-string pattern
DBX *<m,n>*	Delete before meta-string pattern
DY *<m,n>*	Delete meta-string symbol
DAY *<m,n>*	Delete after meta-string symbol
DBY *<m,n>*	Delete before meta-string symbol

EXCHANGE COMMAND

The exchange command is a special form of replacement that can modify several parts of a line at a time. In response to this command, TEDI presents each line of the range followed by a colon prompt. The user can then key in a replacement on a character-by-character basis. A space used as a replacement signifies the character is to remain unchanged. TEDI also recognizes the special exchange characters , , ! and _. (See table 2-1.)

Command	Description
X *<m,n>*	Exchange

INSERT COMMANDS

Insert commands add lines or character strings either before or after a designated point. These commands put TEDI in enter mode. To insert at the beginning, BL1! must be used.

TEDI prompts line insert commands with &.

Command	Description
AL *<m>*	After line
BL *<m>*	Before line
AC *m,n,c*	After column

Command	Description
BC *m,n,c*	Before column
AP *<m,n>*	After pattern
BP *<m,n>*	Before pattern
AS *<m,n>*	After symbol
BS *<m,n>*	Before symbol
AX *<m,n>*	After meta-string pattern
BX *<m,n>*	Before meta-string pattern
AY *<m,n>*	After meta-string symbol
BY *<m,n>*	Before meta-string symbol

ITERATIVE COMMANDS

Iterative commands allow a series of commands to be executed repetitively. The commands are specified on the same line and separated by a user-supplied escape character. Iterate command nesting is not allowed and can produce unexpected results. The **U** command cannot be used in an iterate command.

Command	Description
DO *<m,n>*$*t*...	Do commands unconditionally
DOP *<m,n>*$*pat*$*t*...	Do commands if pattern condition
DOS *<m,n>*$*sym*$*t*...	Do commands if symbol condition
DOX *<m,n>*$*mpat*$*t*...	Do commands if meta-string pattern condition
DOY *<m,n>*$*msym*$*t*...	Do commands if meta-string symbol condition
DNP *<m,n>*$*pat*$*t*...	Do commands if no pattern condition
DNS *<m,n>*$*sym*$*t*...	Do commands if no symbol condition

Command	Description
DNX *<m,n>*$*mpat*$*t*...	Do commands if no meta-string pattern condition
DNY *<m,n>*$*msym*$*t*...	Do commands if no meta-string symbol condition

LINE DIVISION COMMANDS

Line division commands divide each line in a specified range into two lines at a designated point. The join line (JL) command is the exception; it reverses the action of the division, appending all lines in the range to form one line.

Command	Description
JL *<m,n>*	Join lines
VAC *m,n,c*	Divide after column
VBC *<m,n>*	Divide before column
VAP *<m,n>*	Divide after pattern
VBP *<m,n>*	Divide before pattern
VAS *<m,n>*	Divide after symbol
VBS *<m,n>*	Divide before symbol
VAX *<m,n>*	Divide after meta-string pattern
VBX *<m,n>*	Divide before meta-string pattern
VAY *<m,n>*	Divide after meta-string symbol
VBY *<m,n>*	Divide before meta-string symbol

LOCATE COMMANDS

Locate commands search for a specified line number and display those lines. These commands are particularly useful in conjunction with the U command in the command dataset. They function identically to the type commands except locate commands allow no console output.

Command	Description
L *<m,n>*	Locate line
LA *m,i*	Locate after
LB *m,i*	Locate before
LP *<m,n>*	Locate pattern
LPA *m,i*	Locate pattern after
LPB *m,i*	Locate pattern before
LS *<m,n>*	Locate symbol
LSA *m,i*	Locate symbol after
LSB *m,i*	Locate symbol before
LX *<m,n>*	Locate meta-string pattern
LXA *m,i*	Locate meta-string pattern after
LXB *m,i*	Locate meta-string pattern before
LY *<m,n>*	Locate meta-string symbol
LYA *m,i*	Locate meta-string symbol after
LYB *m,i*	Locate meta-string symbol before

REPLACE COMMANDS

Replace commands replace the data within a specified line range.

The replace data command (RD) provides a convenient method of replacing fixed field data (such as assembler fields). The position of the tab stop (see the STC command, sections 2, 3, and 4) determines whether the field is right- or left-justified (positive or negative, respectively). TEDI prompts line replacement commands (RL) with &.

Command	Description
RD *m,n,i*	Replace data field
RL *m,n*	Replace line

Command	Description
MA *m,n,f*	Move lines after
MB *m,n,f*	Move lines before
RC *m,n,c*<,*d*>	Replace column
RAC *m,n,c*	Replace after column
RBC *m,n,c*	Replace before column
RP <*m,n*>	Replace pattern
RAP <*m,n*>	Replace after pattern
RBP <*m,n*>	Replace before pattern
RS <*m,n*>	Replace symbol
RAS <*m,n*>	Replace after symbol
RBS <*m,n*>	Replace before symbol
RX <*m,n*>	Replace meta-string pattern
RAX <*m,n*>	Replace after meta-string pattern
RBX <*m,n*>	Replace before meta-string pattern
RY <*m,n*>	Replace meta-string symbol
RAY <*m,n*>	Replace after meta-string symbol
RBY <*m,n*>	Replace before meta-string symbol

RUN-TIME SWITCH COMMANDS

Run-time switch commands manipulate and display run-time switches. The information command (I) reports the status of the following:

- Current dataset name and length in lines
- Current tab character
- Tab stop settings

- Symbolic register settings
- Verification switch settings
- Translation mode
- Line number echo
- Column match and type settings
- Initial indentation level
- Indentation level increment

Case control commands (SUC, SDC) allow all text insertion lines from the terminal to be forced to uppercase or accepted as both uppercase and lowercase. Certain input such as dataset names is translated to uppercase regardless of the case setting.

Tab stop commands (STC, STS) set a maximum of 10 tab stops. If the column number is negative, display information is right-justified to the tab stop. When the tab character is designated space, a double space on input signals tab. Pressing the space bar twice is the default tab character.

The set indentation command (SI) automatically controls the indentation of a group of lines added in enter mode.

The set-columns-for-match command (SCM) limits pattern and symbol searches to the designated column ranges *c* through *d*. The default values are *c*=1 and *d*=150.

The set-columns-to-type command (SCT) controls the column range, which is displayed by such commands as T, TP, and TS.

The set register command (SR) alters the symbolic registers R0 through R9.

Command	Description
NN	No line number echo
SN	Set line number echo
NV	No change verification
SV	Set change verification
I	Information
SUC	Set uppercase
SDC	Set to uppercase and lowercase

Command	Description
STC *char*	Set tab character
STS <c_1,c_2,...>	Set and clear tab stops
SI *c*<,*i*>	Set and clear indentation
SCM *c*<,*d*>	Set column match
SCT *c*<,*d*>	Set columns type
SR *i*,<*expression*>	Set or clear a symbolic register

TERMINATE COMMANDS

Two types of terminate commands are available. The end command terminates TEDI execution after updating the current dataset. The quit command terminates TEDI without updating. For both commands, the current switch settings are retained.

Command	Description
END	End
QUI	Quit

TYPE COMMANDS

Type commands display a line or range of lines on the console. The current line number is set to the last line displayed.

The type delimited command (TD) displays a group of lines followed by carriage returns. This command is useful with a hardcopy terminal. The number of lines typed equals *i*, and the number of carriage returns between the next set of lines typed equals *j*. The default values are *i*=60 and *j*=6. TEDI retains *i* and *j*; therefore, once they have been specified, they need not be repeated for subsequent TD commands.

The type octal (TO) and type hexadecimal (TZ) commands display the specified lines as a series of character codes.

The type column (TC) command displays the specified lines prefixed by a 1-line column heading. This command is particularly useful for data field alignment.

Type after and type before commands (TA, TB, TPA, TPB, TSA, TSB, TXA, TXB, TYA, TYB) set display *i* lines before or after line *m*. If a pattern or symbol match is required, only lines containing the match are displayed. A maximum of *i* lines is displayed by these commands.

Type line range commands (TP, TS, TX, TY) display a line range having the specified matching string.

Command	Description
T <*m,n*>	Type line
TD <*m,n,i,j*>	Type delimited
TO <*m,n*>	Type octal
TZ <*m,n*>	Type hexadecimal
TC <*m,n*>	Type column
TA <*m,i*>	Type line after
TB <*m,i*>	Type line before
TP <*m,n*>	Type pattern
TPA <*m,i*>	Type pattern after
TPB <*m,i*>	Type pattern before
TS <*m,n*>	Type symbol
TSA <*m,i*>	Type symbol after
TSB <*m,i*>	Type symbol before
TX <*m,n*>	Type meta-string pattern
TXA <*m,i*>	Type meta-string pattern after
TXB <*m,i*>	Type meta-string pattern before
TY <*m,n*>	Type meta-string symbol
TYA <*m,i*>	Type meta-string symbol after
TYB <*m,i*>	Type meta-string symbol before

COMMAND DESCRIPTIONS 4

Detailed descriptions of TEDI commands are contained in this section and are discussed in alphabetical order by command mnemonic. For command breakdown by function, refer to section 3, Command Functions. Basic commands are discussed in Appendix B.

COMMAND SYNTAX

Every command requires a verb, and most commands require a numeric and/or a string argument.

Commands described in this section have the following general form:

verb Set of characters uniquely identifying the command. TEDI recognizes only the mnemonics in the command formats. For readability, a space can follow the verb.

numeric Specifies numeric information such as line and column numbers. Numeric arguments are separated by a space or a comma.

string A single character or group of characters such as a dataset name or a pattern. String arguments are separated by escape characters.

NOTE

Execution of TEDI commands applies only to whole lines in the current dataset.

AC - AFTER COLUMN

FUNCTION: Searches for a designated line range and, within that line range only, inserts a specified string after the specified column

FORMAT: AC *m*,*n*,*c*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

c Column number

PROMPT: I: Insert the new string.

EXAMPLE: AC 1,5,1$END

Insert the string **END** after column 1, for lines 1 through 5 only.

AL - AFTER LINE

FUNCTION: Adds lines after a designated line; puts TEDI in enter mode

FORMAT: AL <*m*>

m Line number. If *m* is omitted, the command applies only to the current line.

PROMPT: &

EXAMPLES: AL

Adds lines after the current line. Used for entering lines into a new dataset.

AL]

Add lines after the last line in the dataset.

To terminate the insert, type a period as the first character of the next line and follow it with a carriage return.

AP - AFTER PATTERN

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated pattern. The specified string is inserted after each occurrence of the pattern within the line range.

FORMAT: AP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: P: Insert the pattern used as a search target.

I: Insert the new string.

EXAMPLE: AP 1 .

In line 1 through the current line, insert a specified string after the target pattern. TEDI prompts for the target pattern and the insertion string.

AS - AFTER SYMBOL

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated symbol. The specified string is inserted after each occurrence of the symbol within the line range.

FORMAT: AS *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: S: Insert the symbol used as a search target.

I: Insert the new string.

EXAMPLE: AS]$DATA$=

Insert = after the symbol **DATA** in the last line of the dataset. No prompts appear since the escape character **$** is used.

AX - AFTER META-STRING PATTERN

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated meta-string pattern. The specified string is inserted after each occurrence of the pattern within the line range.

FORMAT: AX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: X: Insert the pattern used to search for a meta-string construction.

I: Insert the new string.

EXAMPLES: AX .$B...$00

In the current line, insert 00 after any pattern beginning with B.

AX .B00

In the current line, insert 00 after the pattern B.

AY - AFTER META-STRING SYMBOL

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated meta-string symbol. The specified string is inserted after each occurrence of the symbol within the line range.

FORMAT: AY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: Y: Insert the symbol used to search for a meta-string construction.

I: Insert the new string.

EXAMPLE: AY 1,]$[TB]$00

Insert 00 after each occurrence of the symbols T or B throughout the dataset.

BC - BEFORE COLUMN

FUNCTION: Searches for a designated line range and, within that line range only, inserts a specified string before the specified column

FORMAT: BC *m*,*n*,*c*

m	Line number
n	Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*.
c	Column number

PROMPT: I: Insert the new string.

EXAMPLE: BC 5,2,1$*.

Insert the string *. before column 1 in lines 5 through 7 only.

BL - BEFORE LINE

FUNCTION: Adds lines before a designated line; puts TEDI in enter mode.

FORMAT: BL <*m*>

m Line number. If *m* is omitted, the command applies only to the current line.

PROMPT: &

EXAMPLES: BL

Add any number of lines before the current line.

BL 1

Add lines at the beginning of the dataset.

BL .

Insert lines before the current line. To terminate the insert, type a period as the first and only character of the next line.

BP - BEFORE PATTERN

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated pattern. The specified string is inserted before each occurrence of the pattern within the line range.

FORMAT: BP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: P: Insert the pattern used as a search target.

I: Insert the new string.

EXAMPLE: BP 1$ first name $ last name,

In the first line, insert the string **last name** before the pattern **first name**.

BX - BEFORE META-STRING PATTERN

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated meta-string pattern. The specified string is inserted before each occurrence of the pattern within the line range.

FORMAT: BX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPTS: X: Insert the pattern used to search for a meta-string construction.

I: Insert the new string.

EXAMPLE: BX

Insert a string before the specified pattern in the current line only. TEDI prompts for search target and insertion string.

BY - BEFORE META-STRING SYMBOL

FUNCTION: Searches for the designated line range and, within those lines only, searches for the designated meta-string symbol. The specified string is inserted before each occurrence of the symbol within the line range.

FORMAT:

```
BY <m<,n>>
```

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS:

Y: Insert the symbol used to search for a meta-string construction.

I: Insert the new string.

EXAMPLE:

```
BY1,]$[xy]$I
```

Insert an I before all occurrences of the symbols x and y throughout the dataset.

CA - COPY LINE AFTER

FUNCTION: Copies the line range *m*,*n* after line *f*

FORMAT: CA *m*,*n*,*f*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

f Line number after which the lines are copied

PROMPT: None

EXAMPLE: CA 1,.,]

Copy the first line through the current line after the last line of the dataset.

CB - COPY LINE BEFORE

FUNCTION: Copies the line range *m*,*n* before line *f*

FORMAT: CB *m*,*n*,*f*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through $\bar{n}$.

f Line number before which the lines are copied

PROMPT: None

EXAMPLE: CB 1] 1

Copy first line through last line of dataset before line one.

CDA - COPY DATASET AFTER

FUNCTION:	Copies the line range *m,n* of DN after line *f* of the current dataset. The dataset to be copied must be local to the job. During the copy, TEDI replaces each physical end of file with an end-of-file string (EOF).	
FORMAT:	CDA *m,n,f*	
	m	Line number
	n	Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.
	f	Line number after which the dataset is copied
PROMPT:	DN:	Insert the 1- to 7-character dataset name.
EXAMPLE:	CDA 1 5 1	
	Copy lines 1 through 5 of the TEDI-prompted dataset after line 1.	

DAS - DELETE AFTER SYMBOL

FUNCTION: Within the specified line range, deletes all characters after the first occurrence of specified symbol

FORMAT: DAS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: DAS $FILE

In the current line, delete everything after the symbol FILE.

DAX - DELETE AFTER META-STRING PATTERN

FUNCTION: Within the specified line range, deletes all characters after the first occurrence of the specified meta-string pattern

FORMAT: DAX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE:

```
DAX1,]$ft
```

For all lines in the dataset, delete text after the pattern ft.

DAS - DELETE AFTER SYMBOL

FUNCTION: Within the specified line range, deletes all characters after the first occurrence of specified symbol

FORMAT: DAS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n > m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: DAS $FILE

In the current line, delete everything after the symbol FILE.

DAX - DELETE AFTER META-STRING PATTERN

FUNCTION: Within the specified line range, deletes all characters after the first occurrence of the specified meta-string pattern

FORMAT: DAX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE:

```
DAX1,]$ft
```

For all lines in the dataset, delete text after the pattern ft.

DAY - DELETE AFTER META-STRING SYMBOL

FUNCTION: Within the specified line range, deletes all characters after the first occurrence of specified symbol in a meta-string construction

FORMAT: DAY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: DAY *$@?

For lines containing the symbol ?, delete after the symbol to the end of the line.

DBP - DELETE BEFORE PATTERN

FUNCTION: Within the specified line range, deletes all characters before the first occurrence of the specified pattern

FORMAT: DBP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE: DBP 1

In the first line only, delete text before the TEDI-prompted pattern.

DBS - DELETE BEFORE SYMBOL

FUNCTION: Within the specified line range, deletes all characters before the first occurrence of the specified symbol

FORMAT: DBS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: DBS *

Delete everything before the TEDI-prompted symbol in every line of the current dataset.

DBX - DELETE BEFORE META-STRING PATTERN

FUNCTION: Within the specified line range, deletes all characters before the first occurrence of the specified pattern in a meta-string construction

FORMAT: DBX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: DBX $!rewind!

In the current line, delete everything before the pattern **rewind**, which is delimited by separators (same as **DBS$rewind**).

DBY - DELETE BEFORE META-STRING SYMBOL

FUNCTION: Within the specified line range, deletes all characters before the first occurrence of the specified symbol in a meta-string construction

FORMAT: DBY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: DBY $END@!

In the current line, delete everything before the symbol **END!**.

DC - DELETE COLUMN

FUNCTION: Deletes specified columns in specified line range

FORMAT: DC *m,n,c<,d>*

m Line number

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n.

c Column number

d Column number or increment. If $d < c$, the command applies to columns c through $c+d$ (increments the column count). If $d \geq c$, the command applies to columns c through d. If d is omitted, the command applies only to column c.

PROMPT: None

EXAMPLES:

```
DC 1,],73,150
```

Delete columns 73 through 150 from all lines.

```
DC .,.,1,6
```

Delete columns 1 through 6 from the current line.

DL - DELETE LINE

FUNCTION: Deletes specified line or line range

FORMAT: DL <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLES: DL

Delete current line.

DL .,]

Delete current through last line.

DL *

Delete all lines.

DNP - DO IF NO PATTERN

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line not containing the pattern

FORMAT: DNP *<m<,n>>$pat$xyz*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

$ User-supplied escape character

pat Pattern used as search target

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DNP *$COMMENT$DP.CMA..1

If the pattern **COMMENT** is not contained in the line, execute the series of commands (**DP.,C,MA..1**) throughout the dataset.

DNS - DO IF NO SYMBOL

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line not containing the symbol

FORMAT: DNS *<m<,n>>symxyz*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m+n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

$ User-supplied escape character

sym Symbol used as search target

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DNS*$*.$DL

Delete every line where the symbol *. is not found.

DNX - DO IF NO META-STRING PATTERN

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line not containing the meta-string pattern

FORMAT: DNX *<m<,n>>$mpat$xyz*

m	Line number. If *m* is omitted, the command applies only to the current line.
n	Line number or increment. If *n* < *m*, the command applies to lines *m* through *m+n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.
$	User-supplied escape character
mpat	A pattern used to search for a meta-string construction
xyz	Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DNX1,]$mod...$*xyz*

For every line in the dataset where a pattern beginning with **mod** is not found, execute the series of type commands.

DNY - DO IF NO META-STRING SYMBOL

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line not containing the meta-string symbol

FORMAT: DNY *<m<,n>>*$*msym*$*xyz*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m+n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

$ User-supplied escape character

msym A symbol used to search for a meta-string construction

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DNY.$[C*]$*xyz*

For the current line only, if symbols C or * are not found, execute the series of type commands.

DO - UNCONDITIONAL DO

FUNCTION: Within a specified line range, unconditionally allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line

FORMAT: DO *<m<,n>>$xyz*

m Line number. If *m* is omitted, the command applies only to the entire file.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies to the entire file.

$ User-supplied escape character

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DO 1 5 $ CA 1 10 1

After line 1, copy 1 through 10 lines five times.

DOP - DO IF PATTERN

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line containing the pattern

FORMAT:

```
DOP <m<,n>>$pat$xyz
```

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

\$ User-supplied escape character

pat Pattern used as a search target

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE:

```
DOP 1,.$FT$RAP.$FT$00$DC.,.,73,150
```

If the pattern FT exists in line 1 through the current line, execute a series of commands separated by user-supplied escape characters.

DOS - DO IF SYMBOL

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line containing the symbol

FORMAT: DOS *<m<,n>>*$*sym*$*xyz*

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

$ User-supplied escape characters

sym Symbol used as a search target

xyz Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DOS1,]$STOP$*xyz*

For all lines in the dataset containing the symbol **STOP**, execute the series of type commands.

DOX - DO IF META-STRING PATTERN

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line containing the meta-string pattern

FORMAT: DOX <*m*<,*n*>>$*mpat*$*xyz*

m — Line number. If *m* is omitted, the command applies only to the current line.

n — Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

$ — User-supplied escape characters

mpat — A pattern used to search for a meta-string construction

xyz — Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: DOX 1,10$*$DL.

If the pattern * exists in lines 1 through 10, delete the lines where it exists.

DOY - DO IF META-STRING SYMBOL

FUNCTION: Within a specified line range, allows a series of commands, specified on the same line and separated by user-supplied escape characters, to be executed on every line containing the meta-string symbol

FORMAT: `DOY m,n$msym$xyz`

m	Line number. If m is omitted, the command applies only to the current line.
n	Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.
$	User-supplied escape characters
msym	A symbol used to search for a meta-string construction
xyz	Series of commands separated by user-supplied escape characters

PROMPT: None

EXAMPLE: `DOY*$@?$xyz`

In every line containing the symbol ?, execute the series of type commands.

DP - DELETE PATTERN

FUNCTION: Deletes every occurrence of a specified pattern within a specified line range

FORMAT: DP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE: `DP*$PROGRAM ONE`

Delete all occurrences in the dataset of the pattern **PROGRAM ONE**.

DS - DELETE SYMBOL

FUNCTION: Deletes every occurrence of a specified symbol within a specified line range

FORMAT: DS <*m*<,*n*>>

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: DS]$EXIT

Delete the symbol **EXIT** in the last line of the dataset.

DX - DELETE META-STRING PATTERN

FUNCTION: Deletes every occurrence of a specified pattern in a meta-string construction within a specified line range

FORMAT: DX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: DX *$MOD1...

Delete all patterns beginning with MOD1.

DY - DELETE META-STRING SYMBOL

FUNCTION: Deletes every occurrence of a specified symbol in a meta-string construction within a specified line range

FORMAT: DY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: DY *$MOD1...

Delete all symbols beginning with **MOD1**.

END - TERMINATES TEDI EXECUTION

FUNCTION: Terminates TEDI execution after updating the current input dataset

FORMAT: END

PROMPT: None[†]

[†] Prompting occurs only if input dataset is permanent and has write permission.

I - INFORMATION

FUNCTION: Displays run-time switch settings:

- Current dataset name
- Length in lines
- Current tab character
- Tab stop settings
- Symbolic register settings
- Verification switch setting
- Translation mode
- Line number echo
- Column match and type settings
- Initial indentation level
- Indentation level increment
- Symbolic register values

FORMAT: I

PROMPT: None

JL - JOIN LINES

FUNCTION: Appends all lines in a specified range to form one line

FORMAT: JL <*m*<,*n*>>

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n > m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

PROMPT: None

EXAMPLE: JL .,1

Append the next line to the current line.

L - LOCATE LINE

FUNCTION: Resets the current line pointer to a specified line

FORMAT: L <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLES:

```
L ]
```

No line echo; current line points to the last line.

```
L 1,5
```

Locate line 5.

```
L 5,3
```

Locate line 8.

LA - LOCATE LINE AFTER

FUNCTION: Locates a maximum number of lines after a specified line number. Output is not displayed.

FORMAT: LA *m*,*i*

m	Line number after which the lines are located
i	Maximum number of lines to be located

PROMPT: None

EXAMPLE: LA 1 3

Locate a maximum of three lines after line 1. Current line points to line 4. No lines are displayed.

LB - LOCATE LINE BEFORE

FUNCTION: Locates a maximum number of lines before a specified line number. Output is not displayed.

FORMAT: LB *m*,*i*

m Line number before which the lines are located

i Maximum number of lines to be located

PROMPT: None

EXAMPLE:

```
LB ],1
```

Assign current line pointer to the last line number minus 1 in the dataset.

LP - LOCATE PATTERN

FUNCTION: Within a specified line range, locates all lines containing the specified pattern. Output is not displayed.

FORMAT: LP *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE: LP *$TARGET

Locates the pattern **TARGET** in the dataset. The current line points to the last line found containing the search target.

LPA - LOCATE PATTERN AFTER

FUNCTION:	For a maximum number of lines after the designated line, silently locates each line containing the specified pattern	
FORMAT:	LPA *m*,*i*	
	m	Line number after which the lines are located
	i	Maximum number of lines to be located
PROMPT:	P:	Insert the pattern used as a search target.
EXAMPLE:	LPA .,1	
	After the current line, locate the first line containing the specified pattern. The current line points to the line with the pattern.	

LPB - LOCATE PATTERN BEFORE

FUNCTION: For a maximum number of lines before the designated line, locates each line containing a specified pattern. Output is not displayed.

FORMAT: LPB *m*,*i*

m	Line number before which the lines are located
i	Maximum number of lines to be located

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE:

```
LPB 10,1$QUIT
```

Search for the pattern **QUIT** in line 9. The command ends when the first line containing the pattern is found. The current line points to the line containing the pattern. No output is echoed.

LS - LOCATE SYMBOL

FUNCTION: Within a specified line range, locates all lines containing a specified symbol. Output is not displayed.

FORMAT: LS *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: LSR1,R2

Locate the prompted symbol within the line range contained in the symbolic registers R1 and R2.

LSA - LOCATE SYMBOL AFTER

FUNCTION: For a maximum number of lines after the designated symbol, locates each line containing the specified symbol

FORMAT: LSA *m*,*i*

m	Line number after which the lines are located
i	Maximum number of lines to be located

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE:

```
lsa.,1$BYE
```

After the current line, locate the first line containing the symbol **BYE**.

LSB - LOCATE SYMBOL BEFORE

FUNCTION: For a maximum number of lines before the designated line, locates each line containing the specified symbol

FORMAT: LSB *m,i*

m	Line number before which the lines are located
i	Maximum number of lines to be located

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: LSB],1$EOF

Search backward and locate the first line containing EOF.

LX - LOCATE META-STRING PATTERN

FUNCTION: Within a specified line range, locates all lines containing a specified meta-string pattern. Output is not displayed.

FORMAT: LX *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE:

```
lx .-5,.+5$([abc])
```

Within the line range five lines before and five lines after the current line, locate the pattern (a), (b), or (c).

LXA - LOCATE META-STRING PATTERN AFTER

FUNCTION: For a maximum number of lines after the designated line, locates each line containing the specified meta-string pattern. Output is not displayed.

FORMAT: LXA *m*,*i*

m	Line number after which the lines are located
i	Maximum number of lines to be located

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: LXA 1,1

After line 1, locate the first line containing the TEDI-prompted meta-string pattern.

LXB - LOCATE META-STRING PATTERN BEFORE

FUNCTION: For a maximum number of lines before the designated line, locates each line containing the specified meta-string pattern. Output is not displayed.

FORMAT: LXB *m*,*i*

m Line number before which the lines are located

i Maximum number of lines to be located

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: LXB .,1

Before the current line, locate the first line containing the TEDI-prompted meta-string pattern.

LY - LOCATE META-STRING SYMBOL

FUNCTION: Within a specified line range, locates all lines containing a specified meta-string symbol. Output is not displayed.

FORMAT: LY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: LY*$@?

Locate all occurrences of the symbol ? in the dataset. The current line points to the last occurrence of the symbol.

LYA - LOCATE META-STRING SYMBOL AFTER

FUNCTION: For maximum number of lines after the designated line, locates each line containing the specified meta-string symbol. Output is not displayed.

FORMAT: LYA *m*,*i*

m	Line number after which the lines are located
i	Maximum number of lines to be located

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: LYA1,]$?eof?

After line 1, locate the first line containing the symbol ?eof? where ? matches any character.

LYB - LOCATE META-STRING SYMBOL BEFORE

FUNCTION: For maximum number of lines before the designated line, locates each line containing the specified meta-string symbol. Output is not displayed.

FORMAT: LYB *m*,*i*

m	Line number before which the lines are located
i	Maximum number of lines to be located

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: LYB .,1

Before the current line, locate the first line containing the TEDI-prompted meta-string symbol.

MA - MOVE LINES AFTER

FUNCTION: Moves a specified line range after a specified line

FORMAT: MA *m*,*n*,*f*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

f Line number after which the lines are moved

PROMPT: None

EXAMPLE: MA .,],1

Move the current line through the last line after the first line in the dataset.

MB - MOVE LINES BEFORE

FUNCTION: Moves a specified line range before a specified line

FORMAT: MB m,n,f

m Line number

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n.

f Line number before which the lines are moved

PROMPT: None

EXAMPLE: MB .,.,1

Move the current line to the beginning of the dataset.

NN - NO LINE NUMBER ECHO

FUNCTION:	Suppresses display of line numbers as the lines are echoed to the terminal
FORMAT:	NN
PROMPT:	None

NV - NO CHANGE VERIFICATION

FUNCTION: Suppresses display of changes made to a dataset. Set verification (SV) allows display.

FORMAT: NV

PROMPT: None

O - OPEN DATASET

FUNCTION:	Opens a new dataset after updating the current dataset	
FORMAT:	O	
PROMPT:	DN:	Insert the 1- to 7-character dataset name.
EXAMPLE:	o$DN	
	Update the current dataset and open DN for editing.	

OD - OPEN DATASET, DISCARD UPDATES

FUNCTION:	Discards all updates to the current dataset and opens a new dataset for editing	
FORMAT:	OD	
PROMPT:	DN:	Insert the 1- to 7-character dataset name.
EXAMPLE:	OD$FILE2	

Open **FILE2** for editing without updating the current dataset.

NOTE

If the dataset that is loaded from the current dataset is not the desired one, use the OD command to open a different dataset.

QUI - QUIT

FUNCTION:	Terminates TEDI without updating the current input dataset
FORMAT:	QUI
PROMPT:	None

RAC - REPLACE AFTER COLUMN

FUNCTION: Within the specified line range, replaces the designated data with the replacement string after a specified column

FORMAT: RAC *m*,*n*,*c*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

c Column number

PROMPT: R: Insert the replacement data.

EXAMPLE: RAC 1,],72

Throughout the dataset, replace data after column 72 with the TEDI-prompted replacement data.

RAP - REPLACE AFTER PATTERN

FUNCTION: Within the specified line range, replaces the data with the replacement string after the first occurrence of a specified pattern only.

FORMAT: RAP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: P: Insert the pattern used as a search target.

R: Insert the replacement data.

EXAMPLE: RAP *$DATA =$ABC

Replace the data in the current line after the pattern DATA = with ABC.

RAS - REPLACE AFTER SYMBOL

FUNCTION: Replaces the data with the replacement string after the first occurrence of a specified symbol

FORMAT: RAS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: S: Insert the symbol used as a search target.

R: Insert the replacement data.

EXAMPLE: RAS $XYZ$ABC

In the current line only, replace the data after the symbol *xyz* with ABC.

RAX - REPLACE AFTER META-STRING PATTERN

FUNCTION: Within the specified line range, replaces the data with the replacement string after the first occurrence of a meta-string pattern

FORMAT: RAX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: X: Insert the pattern used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE:

```
RAX *$[XY] = $ABC
```

Replace the data in the current line after X = or Y = with ABC.

RAY - REPLACE AFTER META-STRING SYMBOL

FUNCTION: Within the specified line range, replaces the data with the replacement data after the first occurrence of the specified meta-string symbol

FORMAT: RAY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: Y: Insert the symbol used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE:

```
RAY 1$ 10$ FORMAT (1X)
```

In line 1, replace all the data after the symbol 10 with **FORMAT (1X)**.

RBC - REPLACE BEFORE COLUMN

FUNCTION: Within the specified line range, replaces the data with the replacement string before the first occurrence of a specified column

FORMAT: RBC *m,n,c*

m Line number.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

c Column number

PROMPT: R: Insert the replacement data.

EXAMPLE: RBC 1,],10

Replace all data in dataset before column 10.

RBP - REPLACE BEFORE PATTERN

FUNCTION:	Within the specified line range, replaces the data with the replacement string before the first occurrence of a specified pattern	
FORMAT:	RBP <*m*<,*n*>>	
	m	Line number. If *m* is omitted, the command applies only to the current line.
	n	Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.
PROMPTS:	P:	Insert the pattern used as a search target.
	R:	Insert the replacement data.
EXAMPLE:	RBP],]EODEOF	
	In the last line of the dataset, replace any text before the pattern EOD with EOF.	

RBS - REPLACE BEFORE SYMBOL

FUNCTION: Within the specified line range, replaces the designated data with the replacement string before the first occurrence of a specified symbol

FORMAT: RBS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: S: Insert the symbol used as a search target.

R: Insert the replacement data.

EXAMPLE: RBS 10,5$VL

Replace any data before the symbol VL in lines 10 through 15 with TEDI-prompted replacement data.

RBX - REPLACE BEFORE META-STRING PATTERN

FUNCTION: Within the specified line range, replaces the designated data with the replacement data before the first occurrence of a specified meta-string pattern

FORMAT: RBX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: X: Insert the pattern used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE: RBX .,]

From the current line through the last line of the dataset, replace the designated data before the specified meta-string pattern.

RBY - REPLACE BEFORE META-STRING SYMBOL

FUNCTION: Within the specified line range, replaces the designated data with the replacement data before the first occurrence of a specified meta-string symbol

FORMAT: RBY *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: Y: Insert the pattern used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE:

```
RBY.$...ING$new code
```

In the current line only, replace the data before the symbol ending in ING with new code.

RC - REPLACE COLUMN

FUNCTION: Within the specified line range, replaces the designated column with the replacement data

FORMAT: RC *m*,*n*,*c*<,*d*>

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

c Column number

d Column number or increment. If $d < c$, the command applies to columns *c* through $c+d$ (increments the column count). If $d \geq c$, the command applies to columns *c* through *d*. If *d* is omitted, the command applies only to column *c*.

PROMPT: R: Insert the replacement data.

EXAMPLE:

```
RC ],],1,10$LABEL
```

In the last line, replace columns 1 through 10 with LABEL.

RD - REPLACE DATA

FUNCTION: Within the specified line range, replaces the specified data field with the replacement data

FORMAT: RD *m*,*n*,*i*

m Line number

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*.

i Data field number

PROMPT: R: Insert the replacement data.

EXAMPLE: RD .,.,2

In the current line only, replace data field 2 with the TEDI-prompted replacement data.

RL - REPLACE LINE

FUNCTION: Replaces the specified line or line range; puts TEDI in enter mode.

FORMAT: RL *m*,*n*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n > m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: &

EXAMPLE: RL .,]

Replace the current line through the last line of the dataset. These lines are deleted and TEDI is in enter mode.

To terminate the insert, type a period as the first character of the next line and follow it with a carriage return.

RP - REPLACE PATTERN

FUNCTION: Within the specified line range, replaces every occurrence of the designated pattern with the replacement data

FORMAT: RP <m<,n>>

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

PROMPTS:

P: Insert the pattern used as a search target.

R: Insert the replacement data.

EXAMPLE: RP *$EOI$EOD

Throughout the dataset, replace the pattern EOI with EOD.

RS - REPLACE SYMBOL

FUNCTION: Within the specified line range, replaces every occurrence of the designated symbol with the replacement data

FORMAT: RS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: S: Insert the symbol used as a search target.

R: Insert the replacement data.

EXAMPLE: rs

Replace the designated symbol (prompted by TEDI) in the current line with the TEDI-prompted replacement data.

RX - REPLACE META-STRING PATTERN

FUNCTION: Within the specified line range, replaces every occurrence of the designated meta-string pattern with the replacement data

FORMAT: RX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: X: Insert the pattern used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE:

```
RX 1,]$[XY]+
```

Replace all occurrences of the patterns X+ or Y+ with the TEDI-prompted replacement data.

RY - REPLACE META-STRING SYMBOL

FUNCTION: Within the specified line range, replaces every occurrence of the specified meta-string symbol with the replacement string

FORMAT: RY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPTS: Y: Insert the symbol used to search for a meta-string construction.

R: Insert the replacement data.

EXAMPLE: RY .,. $"/EOF$

In the current line, replace the symbol /EOF with a blank string.

SCM - SET COLUMNS FOR MATCH

FUNCTION: Limits pattern and symbol searches to a designated column range

FORMAT: SCM c<,d>

c Column number. Default value is 1.

d Column number or increment. If $d < c$, the command applies to columns c through $c+d$ (increments the column count). If $d \geq c$, the command applies to columns c through d. If d is omitted, the command applies to c, 150. Initial value is 150.

PROMPT: None

EXAMPLES: SCM R1,R2

Set columns for a match according to the values found in TEDI registers R1 and R2.

SCM

Returns set columns for match to default values of [1, 150].

SCT - SET COLUMNS TO TYPE

FUNCTION: Controls the column range displayed by the type commands

FORMAT: SCT *c*<,*d*>

c Column number. Default value is 1.

d Column number or increment. If $d < c$, the command applies to columns *c* through $c+d$ (increments the column count). If $d \geq c$, the command applies to columns *c* through *d*. If *d* is omitted, the command applies to *c*, 150. Initial value is 150.

PROMPT: None

EXAMPLE: SCT

Set default values of [1, 150].

SDC - SET TO UPPERCASE AND LOWERCASE

FUNCTION:	Allows input from the terminal to be in both uppercase and lowercase. Default is dual case.
FORMAT:	SDC
PROMPT:	None

SI - SET INDENTATION

FUNCTION: Automatically controls the indentation of a group of lines added in enter mode. No value in *c* and *i* clears the indentation. This command is used in conjunction with the indentation control characters (refer to Appendix A), and can be used to define data fields for use with the RD command.

FORMAT: SI *c*<,*i*>

c	Column number for initial indentation level
i	Increments amount for additional indentation levels

PROMPT: None

EXAMPLES: SI 7,2

Set initial indentation level at column 7. Each additional indentation increments by 2.

SI

Clear all previous indentation levels.

SN - SET LINE NUMBER ECHO

FUNCTION:	Allows display of line numbers
FORMAT:	SN
PROMPT:	None

SR - SET REGISTER

FUNCTION: Sets or clears the symbolic registers R0 through R9. A null value in the *expression* field causes the register to be set to 0. Registers can be used as arguments.

FORMAT: SR *i<,expression>*

i	An integer value of 0 through 9
expression	The expression to be placed in the register. It can be a line number or a special character.

PROMPT: None

EXAMPLES:

```
SR 0,.
```

Set TEDI register R0 to the current line number.

```
SR0
```

Clear TEDI register R0 (set to zero).

STC - SET TAB CHARACTER

FUNCTION:	Designates the tab settings. The default tab character is double space.
FORMAT:	STC
PROMPT:	CHAR: Any character
EXAMPLE:	STC $@
	Set the tab character to @.

STS - SET TAB STOPS

FUNCTION: Sets tab stops at specified columns. If no column is specified, tabs are cleared. Up to 10 tab stops can be set. If a negative column number is used, the data is right-justified within the field.

FORMAT: STS <c_1,c_2,c_3...>

c Column number or increment. If $|c_i+1| < |c_i|$ then the actual column for the tab stop is $|c_i| + |c_i+1|$ (increment from the preceding column).

PROMPT: None

EXAMPLES:

```
STS 10,20,-30
```

Set tab stops at columns 10 and 20 (left-justified), and column 30 (right-justified).

```
STS 10,5,-5
```

Set tab stops at columns 10 and 15 (left-justified), and column 20 (right-justified)

```
STS
```

Clear tab stops.

SUC - SET UPPERCASE

FUNCTION: Forces input from the terminal to be in uppercase only. All lowercase characters are translated to uppercase.

FORMAT: SUC

PROMPT: None

SV - SET CHANGE VERIFICATION

FUNCTION:	Forces display of all changes made to a dataset. No verification (NV) suppresses the display.
FORMAT:	SV
PROMPT:	None

T - TYPE LINE

FUNCTION: Displays the specified line range on the console

FORMAT: T <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* $\geq$ *m*, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLES: T.

Display the current line.

t*

Display the entire dataset.

TA - TYPE LINE AFTER

FUNCTION:	Displays a maximum number of lines after a specified line number	
FORMAT:	TA <*m*,*i*>	
	m	Line number after which the lines are displayed
	i	Maximum number of lines to be displayed. If *i* is omitted, default is 1.
PROMPT:	None	
EXAMPLE:	TA.,2	
	Display a maximum of two lines after the current line.	

TB - TYPE LINE BEFORE

FUNCTION: Displays a maximum number of lines before a specified line number ending with the specified line

FORMAT: TB *m*,*i*

m Line number before which the lines are displayed

i Maximum number of lines to be displayed. If *i* is omitted, default is 1.

PROMPT: None

EXAMPLE: tb],R1

Display the number of lines found in R1 before the last line in the dataset.

TC - TYPE COLUMN

FUNCTION: Displays a line with column numbers followed by the specified line range

FORMAT: TC <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLES:

```
TC
```

Display a column heading over the current line.

```
TC .,]
```

Display the column heading, and list the current line through the last line of the dataset under the column heading.

TD - TYPE DELIMITED

FUNCTION: Types a group of lines followed by line feeds. Primarily used for hard-copy terminals

FORMAT: TD <*m*,*n*,*i*,*j*>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

i Number of lines per page. Default is 60

j Number of line feeds at end of page. Default is 6.

PROMPT: None

EXAMPLE: TD 1] 50,5

Display or print the entire dataset with 50 lines per page and five line feeds at the end of each page.

TO - TYPE OCTAL

FUNCTION: Displays the specified lines as a series of octal character codes

FORMAT: TO <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLE: T 1 $ TO 1

In line one, ASCII AB CD is displayed as octal format 101 102 040 103 104.

TP - TYPE PATTERN

FUNCTION: Within the specified line range, displays all lines containing the specified pattern

FORMAT: TP <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE:

```
TP *$tedi
```

Display all lines in the dataset containing the pattern tedi.

TPA - TYPE PATTERN AFTER

FUNCTION:	After the designated line, displays a maximum number of lines containing the specified pattern	
FORMAT:	TPA *m*,*i*	
	m	Line number after which the lines are displayed
	i	Maximum number of lines to be displayed
PROMPT:	P:	Insert the pattern used as a search target.
EXAMPLE:	TPA 1,5$variable	

After the first line, display a maximum of five lines containing the pattern **variable**.

TPB - TYPE PATTERN BEFORE

FUNCTION: Before the designated line, displays a maximum number of lines containing the specified pattern

FORMAT: TPB *m,i*

m	Line number before which the lines are displayed
i	Maximum number of lines to be displayed

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE:

```
TPB .,1$=
```

Display the first line (preceding the current line) containing the pattern =.

TS - TYPE SYMBOL

FUNCTION: Within a specified line range, displays all lines containing the designated symbol

FORMAT: TS *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m+n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: TS R0,]$FORMAT

From the line number contained in TEDI register R0 through the last line of the dataset, display all lines containing the symbol FORMAT.

TSA - TYPE SYMBOL AFTER

FUNCTION: For a maximum number of lines after the designated line, displays each line containing the specified symbol

FORMAT: TSA *m,i*

m	Line number after which the lines are displayed
i	Maximum number of lines to be displayed

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE:

```
tsa 1,1$EOF
```

Beginning with the second line of the dataset, display the first line containing the symbol EOF.

TSB - TYPE SYMBOL BEFORE

FUNCTION: For a maximum number of lines before the designated line, displays each line containing the specified symbol

FORMAT: TSB <*m*,>*i*

m Line number before which the lines are displayed

i Maximum number of lines to be displayed

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: TSB],2

From the second line through the last line of the dataset, search backward and display the first two lines encountered that contain the TEDI-prompted symbol.

TX - TYPE META-STRING PATTERN

FUNCTION: Within the specified line range, displays all lines containing the specified meta-string pattern

FORMAT: TX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: TX 10,5$E+00

In lines 10 through 15, display all lines containing the pattern E+00.

TXA - TYPE META-STRING PATTERN AFTER

FUNCTION: For a maximum number of lines after the designated line, displays each line containing the specified meta-string pattern

FORMAT: TXA *m*,*i*

m	Line number after which the lines are displayed
i	Maximum number of lines to be displayed

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE:

```
txa .,1
```

After the current line, display the first line containing the TEDI-prompted meta-string pattern.

TXB - TYPE META-STRING PATTERN BEFORE

FUNCTION:	For a maximum number of lines before the designated line, displays each line containing the specified meta-string pattern	
FORMAT:	TXB *m*,*i*	
	m	Line number before which the lines are displayed. If *m* is omitted, the command applies only to the current line.
	i	Maximum number of lines to be displayed
PROMPT:	X:	Insert the pattern used to search for a meta-string construction.
EXAMPLE:	TXB],6	
	For a maximum of six lines before the last line, display all lines containing the TEDI-prompted meta-string pattern.	

TY - TYPE META-STRING SYMBOL

FUNCTION: Within the specified line range, displays all the lines containing the specified meta-string symbol

FORMAT: TY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: TY *

Display all lines containing the TEDI-prompted meta-string symbol.

TYA - TYPE META-STRING SYMBOL AFTER

FUNCTION: For a maximum number of lines after the designated line, displays each line containing the specified meta-string symbol

FORMAT: TYA *m*,*i*

m Line number after which the lines are displayed

i Maximum number of lines to be displayed

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: TYA 1]$d...a

After the first line, find all occurrences of the symbol d...a where ... matches any character string.

TYB - TYPE META-STRING SYMBOL BEFORE

FUNCTION: For a maximum number of lines before the designated line, displays each line containing the specified meta-string symbol

FORMAT: TYB *m*,*i*

m Line number before which the lines are displayed

i Maximum number of lines to be displayed

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE:

```
tyb ],1$FORMAT
```

Display one line before the last line containing the symbol **FORMAT.**

TZ - TYPE HEXADECIMAL

FUNCTION: Displays the specified lines as a series of hexadecimal character codes

FORMAT: TZ <*m*<,*n*>>

m Line number after which the lines are displayed. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: None

EXAMPLE: TZ .

Display the current line as a series of hexadecimal character codes.

U - USE

FUNCTION: Directs TEDI to accept commands from command dataset. The command dataset is read sequentially from its current position until the end of the dataset or a command error. Control then reverts to the console.

FORMAT: U

PROMPT: DN: Insert the 1- to 8-character dataset name.

EXAMPLE: U$CMD

Accept commands from dataset CMD.

VAC - DIVIDE AFTER COLUMN

FUNCTION: Within a specified line range, divides the line into two lines after a specified column

FORMAT: VAC *m*,*n*,*c*

m	Line number
n	Line number or increment. If $n < m$, the command applies to lines *m* through *m*+*n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.
c	Column number after which the line is divided

PROMPT: None

EXAMPLE: VAC.,.,10

Divide the current line after column 10.

VAP - DIVIDE AFTER PATTERN

FUNCTION: Within a specified line range, divides the line into two lines after every occurrence of the specified pattern

FORMAT: VAP <*m*<,*n*>>

m Line number. If m is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through $\overline{n}$. If n is omitted, the command applies only to line m.

PROMPT: P: Insert the pattern used as a search target.

EXAMPLE: VAP * $ EOL

Divide all lines in the dataset after the pattern **EOL**.

VAS - DIVIDE AFTER SYMBOL

FUNCTION: Within a specified line range, divides the line into two lines after every occurrence of the specified symbol

FORMAT: VAS <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE:

```
vas 10,2$"/EOF
```

Divide lines 10, 11, and 12 after the symbol /EOF.

VAX - DIVIDE AFTER META-STRING PATTERN

FUNCTION: Within a specified line range, divides the line into two lines after every occurrence of the specified meta-string pattern

FORMAT: VAX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE: VAX

Divide the current line after the TEDI-prompted meta-string pattern.

VAY - DIVIDE AFTER META-STRING SYMBOL

FUNCTION: Within a specified line range, divides the line into two lines after every occurrence of the specified meta-string symbol

FORMAT: VAY <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: VAY]$STOP

Divide the last line after the symbol **STOP**.

VBC - DIVIDE BEFORE COLUMN

FUNCTION: Within a specified line range, divides the line into two lines before a specified column

FORMAT: VBC *m*,*n*,*c*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n > m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

c Column number before which the line is divided

PROMPT: None

EXAMPLE: VBC 1,],73

Divide all lines in the dataset before column 73.

VBP - DIVIDE BEFORE PATTERN

FUNCTION: Within a specified line range, divides the line into two lines before every occurrence of the specified pattern

FORMAT: VBP *<m<,n>>*

m — Line number. If *m* is omitted, the command applies only to the current line.

n — Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: P: — Insert the pattern used as a search target.

EXAMPLE: VBP$XXX

Divide the current line before **XXX**.

VBS - DIVIDE BEFORE SYMBOL

FUNCTION: Within a specified line range, divides the line into two lines before every occurrence of the specified symbol

FORMAT: VBS *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through $\overline{n}$. If *n* is omitted, the command applies only to line *m*.

PROMPT: S: Insert the symbol used as a search target.

EXAMPLE: VBS.$10

Divide the current line before the symbol 10.

VBX - DIVIDE BEFORE META-STRING PATTERN

FUNCTION: Within a specified line range, divides the line into two lines before every occurrence of the specified meta-string pattern

FORMAT: VBX <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: X: Insert the pattern used to search for a meta-string construction.

EXAMPLE:

```
VBX 1,20$x[=+-]
```

Divide lines 1 through 20 before **x=**, **x+**, and **x-**.

VBY - DIVIDE BEFORE META-STRING SYMBOL

FUNCTION: Within a specified line range, divides the line into two lines before every occurrence of the specified meta-string symbol

FORMAT: VBY *<m<,n>>*

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: Y: Insert the symbol used to search for a meta-string construction.

EXAMPLE: VBY1,]$?eof

Divide all lines in the dataset before the meta-string symbol **eof** where ? is any character including separators.

W - WRITE

FUNCTION: Rewrites the input dataset with all changes included. This rewrite should be done periodically during the editing session to prevent loss of current changes due to system failure.

FORMAT: W

PROMPT: None

WD - WRITE DATASET

FUNCTION: Allows a specified line range to be written to the designated dataset

FORMAT: WD <*m*<,*n*>>

m Line number

n Line number or increment. If $n < m$, the command applies to lines m through $m+n$ (increments the line count). If $n \geq m$, the command applies to lines m through n. If n is omitted, the command applies only to line m.

PROMPT: DN: Insert the 1- to 8-character dataset name.

EXAMPLE: WD$DN

Rewrite the entire dataset to a second dataset, DN.

NOTE

If both m and n are omitted, the command affects the entire dataset.

WDN - WRITE DATASET NUMBERED

FUNCTION: Prefixes each line written with a 1-word ASCII line number

FORMAT: WDN <*m*<,*n*>>

m Line number

n Line number or increment. If *n* < *m*, the command applies to lines *m* through *m*+*n* (increments the line count). If *n* ≥ *m*, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: DN: Insert the 1- to 8-character dataset name.

EXAMPLE: WDN 1,.$DN

Write the first line through the current line to dataset DN. Prefix each line with a line number.

NOTE

If both *m* and *n* are omitted, the command affects the entire dataset.

WR - WRITE DATASET

FUNCTION: Writes the current dataset to a new dataset

FORMAT: WR

PROMPT: DN: Insert the 1- to 8-character dataset name.

EXAMPLE: WR$FILE

Rewrite the current dataset into **FILE**.

X - EXCHANGE

FUNCTION: A special replacement modifying several parts of a line at a time. After the prompt, the user can enter a replacement on a character-by-character basis. A space used as a replacement signifies that the character is to remain unchanged. (Refer to Special Characters, section 2.)

FORMAT: X <*m*<,*n*>>

m Line number. If *m* is omitted, the command applies only to the current line.

n Line number or increment. If $n < m$, the command applies to lines *m* through *m+n* (increments the line count). If $n \geq m$, the command applies to lines *m* through *n*. If *n* is omitted, the command applies only to line *m*.

PROMPT: : Insert the replacement data on a character-by-character basis.

EXAMPLE: X 1,.

Do a character-by-character replacement of the first line through the current line.

APPENDIX SECTION

MESSAGES A

TE001 - INVALID TEDI COMMAND ARGUMENT

User specified an invalid command argument. Either the line numbers are not within the legal range ($m > 0$ and $n \geq 0$), the column number references are not contained within the legal range [1,150], or the argument list contained a special character unknown to TEDI.

TE002 - INCORRECT TEDI COMMAND MNEMONIC

TEDI verb is syntactically incorrect.

TE003 - MISSING TEDI COMMAND ARGUMENTS

TEDI command found a required argument or arguments missing.

TE004 - DATA FIELD NOT FOUND

Replace data command could not find the specified data field.

TE005 - U COMMAND NOT NESTABLE

A U command can not be executed in another U dataset or an iterate command.

TE006 - UNKNOWN TEDI COMMAND

User specified a command unknown to TEDI.

TE007 - INVALID TAB STOP REQUEST

Tab stop is out of the range [-150,+150].

TE008 - INVALID INITIAL INDENTATION LEVEL

Initial indentation level request is not in the range [1,150].

TE009 - MISSING INDENTATION LEVEL INCREMENT

Set indentation command was missing the indentation level increment.

TE010 - REGISTER SYMBOL INVALID

Register symbol was not contained in the range [0,9].

TE011 - WRITING OVER A PERMANENT DATASET

A permanent dataset will be overwritten if the command is continued. TEDI gives the option to continue.

TE012 - NO MAINTENANCE PERMISSION

Specified dataset does not have maintenance permission.

TE013 - PATTERN OR SYMBOL NOT FOUND
TEDI did not find specified pattern or symbol.

TE014 - DATASET NOT LOCAL
A TEDI command requested access to a dataset that is not local to the job.

TE015 - SWITCHES RECOVERED
TEDI's run-time switches are recovered from the previous editing session.

TE016 - ILLEGAL DATASET NAME
Dataset name contains an invalid character or is longer than seven characters.

TE017 - NEW DATASET
Local dataset has been created by TEDI.

TE018 - NO UPDATE WILL BE DONE
Dataset does not have write permission. No update can be done.

TE019 - READ ONLY DATASET
Dataset does not have write permission and unique access.

TE020 - APPROACHING MAXIMUM DATASET SIZE
User is approaching maximum dataset size TEDI can handle (varys depending on use of commands, scratch area, and other factors).

TE021 - DATASET TOO LARGE
Dataset is too large for TEDI to handle.

TE022 - BAD META-STRING CONSTRUCT
TEDI found an illegal meta-string construct.

TE023 - TOO MANY TEDI COMMAND ARGUMENTS
Too many command arguments were found.

TE024 - CANNOT EDIT AN INTERACTIVE DATASET
User attempted to edit an interactive dataset.

TE025 - UNKNOWN SYSTEM COMMAND
User specified an unknown system command.

TE026 - *n* NULL INPUT RECORDS SKIPPED
n null records were encountered on input and subsequently skipped.

TE027 -*n* LONG INPUT RECORDS TRUNCATED
n records containing over 150 characters were truncated. Maximum record length for TEDI is 150 characters.

TE0999 - *subroutine* OR *function name*
Serious TEDI execution error occurred.

BASIC COMMANDS B

Although TEDI has a large instruction set designed to allow a user to carry out any editing operation with a minimum of effort, the following subset may be sufficient for a new user. In the representation of commands and their arguments (as described in section 1), the dollar sign indicates either a carriage return or escape character. If the carriage return is used, TEDI prints the argument prompt shown in the quotes. If an escape character is used (and the user enters an appropriate argument), TEDI does not output the argument prompt. The space immediately following the command mnemonic is optional. It is used here for readability.

Commands requiring line numbers follow these conventions:

- *m* designates a single line or the first line of a range.
- *n* designates the end of a line range if greater than or equal to *n* and designates an increment (such as *m+n*) if *n* is less than *m*.
- *p* designates an arbitrary line number.

The basic commands are:

AL *m*	Enter input mode[†] and insert lines following line *m*.
BL *m*	Enter input mode[†] and insert lines preceding line *m*.
DL *m,n*	Delete lines *m,n*.
RL *m,n*	Delete lines *m,n*, then enter input mode[†] and insert remaining lines.
MA *m,n,p*	Move lines *m,n* after line *p*.
MB *m,n,p*	Move lines *m,n* before line *p*.
CA *m,n,p*	Copy lines *m,n* after line *p*.
CB *m,n,p*	Copy lines *m,n* before line *p*.
DP *m,n*$"P:"	Delete pattern "P:" in lines *m,n*.
RP *m,n*$"P:"$"R:"	Replace pattern "P:" with pattern "R:" in lines *m,n*.
T *m,n*	Type lines *m,n*.
TP *m,n*$"P:"	Type lines *m,n* containing pattern "P:".

[†] Input mode is terminated by keying in a single period in column 1 immediately followed by end of line.

INDEX

INDEX

Page numbers appearing in italics designate references of major importance.

READERS COMMENT FORM

Text Editor (TEDI) User's Guide SG-0055 A-01

Your comments help us to improve the quality and usefulness of our publications. Please use the space provided below to share with us your comments. When possible, please give specific page and paragraph references.

NAME ______________________________

JOB TITLE ___________________________

FIRM ______________________________

ADDRESS ____________________________

CITY _______________ STATE ______ ZIP ______

CRAY
RESEARCH, INC.

CUT ALONG THIS LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.
1440 Northland Drive
Mendota Heights, MN 55120
U.S.A.

Attention:
PUBLICATIONS

READERS COMMENT FORM

Text Editor (TEDI) User's Guide SG-0055 A-01

Your comments help us to improve the quality and usefulness of our publications. Please use the space provided below to share with us your comments. When possible, please give specific page and paragraph references.

NAME ______________________________

JOB TITLE ______________________________

FIRM ______________________________

ADDRESS ______________________________

CITY ______________ STATE ________ ZIP ________

CRAY RESEARCH, INC.

CUT ALONG THIS LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO 6184 ST PAUL MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

Attention:
PUBLICATIONS

1440 Northland Drive
Mendota Heights, MN 55120
U.S.A.

READERS COMMENT FORM

Text Editor (TEDI) User's Guide SG-0055 A

Your comments help us to improve the quality and usefulness of our publications. Please use the space provided below to share with us your comments. When possible, please give specific page and paragraph references.

NAME ______________________________

JOB TITLE ______________________________

FIRM ______________________________

ADDRESS ______________________________

CITY ______________ STATE ________ ZIP ________

CRAY RESEARCH, INC.

CUT ALONG THIS LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST PAUL MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

1440 Northland Drive
Mendota Heights, MN 55120
U.S.A.

Attention:
PUBLICATIONS

READERS COMMENT FORM

Text Editor (TEDI) User's Guide SG-0055 A

Your comments help us to improve the quality and usefulness of our publications. Please use the space provided below to share with us your comments. When possible, please give specific page and paragraph references.

NAME ______________________________

JOB TITLE ______________________________

FIRM ______________________________

ADDRESS ______________________________

CITY ______________ STATE ________ ZIP ________

CRAY RESEARCH, INC.

CUT ALONG THIS LINE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD
FIRST CLASS PERMIT NO 6184 ST PAUL MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

1440 Northland Drive
Mendota Heights, MN 55120
U.S.A.

Attention:
PUBLICATIONS