Price: $3.00

TYMSHARE TYMCOM-X

TAPE

REFERENCE MANUAL

TYMSHARE, INC.
CUPERTINO, CALIFORNIA

TYMSHARE®

FEBRUARY 1978
VERSION 1

CONTENTS

SECTION 1

INTRODUCTION

TAPE, Tymshare's paper tape package on the TYMCOM-X, provides an outstanding set of paper tape handling features, including the ability to read and punch tape and to convert character codes between input and output.

The TAPE user may read and punch information in the usual symbolic form, or he may choose among several alternative binary forms.

Symbolic information stored on a file may easily be converted to a different set of codes before it is punched on tape, and character codes read from paper tape may be converted to different characters before they are stored on a file.

TAPE contains many features which extend the convenience and flexibility of paper tape handling:

- TITLE enables the user to punch an identifying title at the beginning of a paper tape.
- EDITING lets the user define his own editing characters to edit the incoming data during symbolic paper tape reading.
- PARITY allows the user to request even or odd parity checking during paper tape reading or punching and to define the form used for binary reading or punching.
- LENGTH permits the user to specify the length of blank leader and trailer to be punched at the beginning and end of a paper tape.
- SPLICE and SSPLICEREAD enable the user to make a simple recovery if symbolic paper tape reading is interrupted by a system failure or other emergency.
- SUBSTITUTE provides an easy, convenient method of specifying code conversions to be performed during paper tape reading or punching.
- REPRESENTATION allows the user to convert the characters on a paper tape being read to their octal or decimal ASCII codes before storing them on a symbolic file, and, conversely, to convert the characters on a symbolic file to their octal or decimal ASCII codes before punching them on paper tape. REPRESENTATION also permits the user to convert a symbolic file or paper tape containing octal or decimal ASCII codes to a standard symbolic paper tape or file.

TAPE also contains the excellent interactive features traditional in Tymshare software. These include:

- The ability to correct errors made while typing commands.
- Command abbreviations.
- The HELP and ? commands, which may be typed anywhere in a TAPE command to provide a list of all arguments available to the user at that point in the command.
- Command files, which allow the user to store an entire sequence of commands on a file and then execute them with one simple command.

ABOUT THIS MANUAL

Since TAPE includes a wide variety of features, not all of which are needed by every user, the user should first read Section 2 to familiarize himself with the basic concepts and overall design of TAPE. He can then refer to subsequent sections of this manual, as needed, to accomplish his paper tape handling tasks.

Section 3 describes symbolic paper tape reading and punching, including the EDITING, ECHO, SPLICE, SSPLICEREAD, TITLE, LENGTH, and PARITY commands. Section 4 explains binary paper tape reading and punching. Code conversions are discussed in Section 5, and Section 6 describes command files.

Appendix A explains how to interpret the codes on punched paper tape. It is suggested that users who are unfamiliar with interpreting punched tape read Appendix A first. A table of various code conversions, including octal and decimal ASCII codes and their standard ASCII character names, is presented in Appendix B.[1] Appendix C contains a summary of TAPE commands, and Appendix D explains TAPE error messages.

SYMBOL CONVENTIONS

In all examples and command forms in this manual, everything typed by the user is underlined. The special symbols for user-typed characters are:

Carriage Return: ↵
Line Feed: ↴
Alt Mode/Escape: ⊕

Control characters are denoted by a superscript c. For example, A^c denotes Control A. The method of typing a control character depends on the type of terminal being used. Consult the literature for your particular terminal or see your Tymshare representative.

The general form of a TAPE command is

p: command↵

where p is the command level prompt number, and the colon (:) is the prompt character. TAPE numbers the commands given in a TAPE session sequentially; for example, successive commands appear as:

1: command↵
2: command↵
3: command↵

Lowercase letters in a command form describe the actual information to be typed. For example, *file name* in the command form

p: READ file name↵

indicates that the user types any valid file name at that position in the command.

1 - ASCII is the American Standard Code for Information Interchange.

Braces in a command form indicate that the user must enter one of the items within the braces. The braces are not part of the command. For example,

p: **RADIX** { **OCTAL** / **DECIMAL** }↵

indicates that the user may enter either:

p: **RADIX OCTAL**↵

or

p: **RADIX DECIMAL**↵

SECTION 2

USING TAPE

This section presents the basic information needed to use TAPE and introduces the commands and options available to the user. Two sample sessions, which illustrate symbolic paper tape reading and punching, are presented first. Included with these examples is basic information about reading and punching symbolic tape. The interactive features shown in the examples, such as calling TAPE and specifying files, are explained later in this section.

SAMPLE SESSIONS

Example 1

The user wishes to read a paper tape and store the information read on a symbolic file named FPROG.[1] Before reading the tape, the user positions the tape in the paper tape reader within the leader section, following any title punched on the tape. The leader is a length of blank tape at the beginning of the paper tape; paper tapes also may have a length of blank tape at the end, called the trailer. The title consists of readable text which may be punched on the paper tape for the purpose of visual identification.

The user calls TAPE and reads the paper tape as follows:

-R TAPE⏎	*The user calls TAPE from TYMEX.*
YOUR JOB ID IS 17	*TAPE prints the user's job number.*
1: READ⏎	*The READ command specifies symbolic reading.*
TO FILE: FPROG⏎	*The user wants the information stored on a file named FPROG.*
NEW FILE⏎	
TURN ON READER	*TAPE prints this message, and the user turns on the paper tape reader.*
3382 CHARACTERS ON FILE	*After the tape is read, TAPE waits 10 seconds and then prints the number of characters read.*
2: QUIT⏎	*The user types QUIT to return control to TYMEX.*
-	

After the paper tape is read, TAPE waits 10 seconds to verify that no more characters are forthcoming, prints the number of characters read, and returns to command level.

NOTE: Any character typed at the terminal during the 10-second wait, including an Alt Mode/Escape, is stored on the file.

1 - A symbolic file is a file in which information is stored in the standard ASCII character representation. See page 21 for details.

Example 2

The user has information stored on a symbolic file named TRANS that he wishes to punch on paper tape. The procedure used for punching the tape is similar to that used for reading in the above example. The user calls TAPE and executes the PUNCH command as follows:

```
-R TAPE⊃
YOUR JOB ID IS 12

1: PUNCH⊃                    The PUNCH command specifies symbolic punching.

FROM FILE: TRANS⊃            The file TRANS contains the information to be punched.

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
⊃                            The user types a Carriage Return and turns on the paper tape punch.

ABC SUPPLIES,4031,730215,150.66          The information is punched on the
CARTER PRINTING,4018,730220,500.00       tape and printed at the terminal.
DAILY DESIGN,3921,730225,150.00
KAY DEALERS,4251,730217,550.00
KUBLA PRINTS,399,730213,200.00
T T & G,3877,730228,271.30

2: QUIT⊃                     After punching the information, TAPE returns control to command level.

-
```

After the user types the PUNCH command and identifies the file, TAPE requests that he type either a Carriage Return or a Line Feed and then turn on the paper tape punch. Typing a Line Feed instructs TAPE to punch the file name, the type (symbolic or binary), and the value of the PARITY option in readable text at the beginning of the paper tape.[1] In the above example, the user bypasses this option by typing a Carriage Return. After all the data is punched, control returns to TAPE command level; the user turns off the paper tape punch.

RUNNING TAPE

TAPE is called from TYMEX by typing an R followed by a space, TAPE, and a Carriage Return. TAPE responds by printing the user's job number and the first command level prompt.[2]

```
-R TAPE⊃
YOUR JOB ID IS 7

1:
```

1 - This information, if requested, is punched in addition to any user-specified title. See the discussions of the TITLE option, page 31, and the PARITY option, page 33, for more information.

2 - TAPE uses the job number in naming two temporary files, the interruption recovery file and the parity error file, discussed on page 28 and page 34, respectively.

The user may now enter his first TAPE command. All TAPE commands are terminated by a Carriage Return. Whenever control is transferred to command level, TAPE prints the appropriate prompt number and a colon (:); each time TAPE executes a command, the prompt number is incremented by 1.

When entering a command, the user may type the first word in a command and let TAPE prompt for the missing component. For example, the READ command, which has the form

p: READ file name↵

may be entered as:

p: READ↵
TO FILE: file name ↵

The user may interrupt execution of any TAPE command by typing an Alt Mode/Escape. TAPE prints

<ESC>

and returns to command level, repeating the previous prompt number to indicate that execution of the command was not completed. For example:

```
1: TITLE 'DAILY TRANSACTIONS'↵
OK

2: PUNCH↵

FROM FILE: TRANS↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.⊕

<ESC>


2:
```

The user specifies a title to be punched on a paper tape.

He enters the PUNCH command.

He interrupts the PUNCH command by typing an Alt Mode/Escape.

To return to TYMEX from TAPE, the user types the QUIT command:

2: QUIT↵

—

FILES

All commands for paper tape reading and punching require that the user specify a file name. When punching tape, the user specifies the name of the file that contains the information to be punched; when reading tape, he specifies the name of the file that is to contain the information read.

A file name may contain as many as six characters, with any combination of letters and digits, such as TEST12. In addition, an extension containing as many as three characters may follow a file name. The extension may be a standard one, recognized by TYMEX, or it may be created by the user as a reminder of the contents of the file.[1] A period (.) separates the file name from the extension. For example, DATA.TMP has the file name DATA and the extension TMP. This extension is recognized by TYMEX; it denotes a temporary file which the system automatically deletes when the user logs out.

When the user specifies a file during paper tape reading, the system prints one of these messages:

NEW FILE	*Printed if there is no file with the specified name in the user's directory.*
OLD FILE	*Printed if there is a file with the specified name in the user's directory.*

The user may type a Carriage Return to confirm the file name or he may type an Alt Mode/ Escape to abort the command.

NOTE: If OLD FILE is confirmed with a Carriage Return, the information read is written on the file, and the previous contents of the file are lost.

Example

```
1: READ↵

TO FILE: FPROG↵
OLD FILE⊕                The user aborts the READ command with an Alt Mode/Escape.
CANNOT OPEN FPROG - NEW/OLD FILE MESSAGE NOT CONFIRMED

1: READ↵

TO FILE: F4PROG↵
NEW FILE↵                This time, he confirms the file name, and TAPE executes the READ command.

TURN ON READER
```

1 - For additional information on file name extensions and other TYMEX features for creating and manipulating files, see the *Tymshare TYMCOM-X TYMEX Reference Manual.*

COMMAND EDITING

TAPE contains a complete built-in editing capability similar to Tymshare's EDITOR. All the editing control characters available in EDITOR, except Control B and Control G, may be used to edit TAPE commands as they are typed. A few of the most useful control characters are discussed below.[1]

Control A deletes the preceding character. On most terminals, when the user types Control A, the system prints a back slash (\) and the character deleted. The user may type Control A repeatedly to delete several characters; each repeated use causes the character deleted to be printed. When the user types the next character in the command, another back slash is printed, so that all characters deleted are enclosed in back slashes. For example:

1: PUMA^C\M\NCH↵

FROM FILE: DAT22A^C\2A^C2\A12↵ *The user wants to punch the file DATA12.*

Control Q deletes the entire line being typed. On most terminals, when the user types Control Q, the system prints an up arrow (↑), returns the carriage, and repeats the current prompt number. For example:

2: PUNCHQ^C↑

2: READ↵ *The READ command is entered.*

Control W deletes the immediately preceding blanks, if any, and the preceding word in the line being typed. On most terminals, when Control W is typed, a back slash, the characters deleted, and another back slash are printed. Control W may be used repeatedly to delete successive words in a line. For example:

3: READ↵

TO FILE: 3TESTW^C\TSET3\4TEST↵ *The user specifies the file 4TEST.*

Control A, Control Q, and Control W can be used only in the line being typed. They do not delete any characters typed before the previous Carriage Return.

When the user enters a command and receives an error message, he may use the incorrect command as an editing image for reentering the correct command. For example:

1: PUNCH STA12↵

CANNOT OPEN STA12 – FILE NOT FOUND

1: Z^CAPUNCH STAE^C<TD^C>12

In this example, the user types a complete command, not realizing he omitted a T from the file name STAT12. When he receives the error message and prompt, he types Control Z followed by an A which copies the command up to and including the A. He then uses Control E to insert the missing T and, finally, Control D to terminate the insertion and copy the rest of the command.

1 - The *Tymshare EDITOR Reference Manual* contains complete documentation of control characters and their functions.

ABBREVIATING COMMANDS

All TAPE commands may be shortened to as few characters as necessary to identify the command uniquely. For example,

1: PUNCH↵

may also be typed as:

1: PU↵

It may not be typed as

1: P↵

since this does not distinguish the PUNCH command from the PARITY command.

Separate command words may be shortened individually. For example, the command

1: PARITY EVEN↵

may be shortened to:

1: PA E↵

The user may also abbreviate a response to a prompt. For example:[1]

```
3: PROMPT↵
SYMBOLIC OR BINARY? BIN↵        The user types an abbreviation of BINARY.
```

OPTIONS AVAILABLE DURING READING AND PUNCHING

The TAPE user may read and punch either symbolic or binary information using the following commands:[2]

Command	Description
READ PUNCH	Reads or punches information in symbolic form.
BREAD BPUNCH	Reads or punches information in binary form.

1 - The PROMPT command is discussed on page 12.

2 - These commands are discussed in detail in Sections 3 and 4 of this manual.

In addition, various options extend the convenience and flexibility of paper tape reading and punching. Each option may be set simply by entering it as a command. If an option is not set by the user, TAPE uses a preset value for the option; such a standard, preset value is called a *default* value. TAPE options are summarized in the following table.[1]

SUMMARY OF READ AND PUNCH OPTIONS		
Option	**Description**	**Default or Preset Value**
ECHO	Allows data being read from tape to be printed at terminal.	Information being read is not printed at terminal.
EDITING	Permits user to define characters to edit the incoming data during symbolic paper tape reading.	No editing.
TITLE	Allows user to specify a title of readable text to be punched at beginning of tape.	No title.
LENGTH	Enables user to specify length of blank leader and trailer to be punched on tape.	Leader and trailer are each 5 inches long.
PARITY	Controls use of parity bit (eighth level of paper tape). Even or odd parity may be specified. Also defines the form to be used for binary reading and punching.	Eighth level ignored when reading, and not punched when punching. Binary information is read or punched in 6-bit bytes.
SUBSTITUTE	Defines character code conversions to be performed during symbolic reading or punching.	No conversions.
REPRESENTATION	Allows literal representation of character codes; that is, allows symbolic reading or punching of the numeric ASCII codes for characters instead of the actual characters.	No literal representation; characters are read or punched in usual manner.

NOTE: Once an option is set, it remains at that value throughout the TAPE session unless the user resets the option.

1 - These options are discussed in detail in Sections 3, 4, and 5 of this manual.

Option Prompting

The user may enter each of the options in the preceding table as a specific command. For example, the TITLE command is used to request that the title INTEREST RATES be punched at the beginning of a paper tape:

```
1: TITLE 'INTEREST RATES'↵
OK

2:
```

Instead of typing a command for each option to be set, the user may enter the PROMPT command to instruct TAPE to prompt for each option. This command may be used for either symbolic or binary reading or punching. It has the form:

p: PROMPT↵

and causes TAPE to prompt for all applicable options.

Example

The user reads a paper tape and stores the information read on the file DATA10. The PARITY option is set to perform parity checking for even parity; all other options are set to their default values.

```
1: PROMPT↵

SYMBOLIC OR BINARY? SYMBOLIC↵        The user specifies symbolic form.

PARITY: EVEN↵                        He requests even parity checking.

INPUT FROM:  T↵                      T, for terminal, specifies the paper tape reader.

OUTPUT TO:   DATA10↵
NEW FILE↵

LITERAL REPRESENTATION? NO↵

EDITING: NO↵

SUBSTITUTE:
FILE,TAPE
↵                                    Since the user does not wish to perform any code conversions,
                                     he terminates the SUBSTITUTE prompt with a Carriage Return.

ECHO: NO↵

TURN ON READER                       TAPE instructs the user to turn on the paper tape reader.

194 CHARACTERS ON FILE               After the tape is read, TAPE waits 10 seconds and then prints
                                     the number of characters read.

2:
```

During PROMPT, typing only a Carriage Return in response to an option prompt causes TAPE to assume the default value or, if that option has been set by the user previously, to assume the most recent value. To set any option except LENGTH and SUBSTITUTE to its default value, irrespective of a value assigned previously, the word NO or OFF is typed in response to the option prompt.[1] These features are illustrated in the following example.

Example

The information on the symbolic file DATA10 is punched on paper tape. The user requests that the title TEST DATA be punched at the beginning of the tape, and that the leader and trailer be 10 inches long. All other options are set to their default values.

```
2: PROMPT⏎

SYMBOLIC OR BINARY?⏎
PARITY: NO⏎
INPUT FROM: DATA10⏎
OUTPUT TO: ⏎
LITERAL REPRESENTATION?⏎
SUBSTITUTE:
FILE,TAPE
⏎
TITLE: 'TEST DATA'⏎
LENGTH OF LEADER AND TRAILER: 10⏎
TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
⏎
```

Typing only a Carriage Return causes TAPE to assume symbolic information.

Since the PARITY option was set to EVEN in the preceding example, it must be set to its default value by typing NO and a Carriage Return.

Typing only a Carriage Return causes TAPE to assume the terminal, which specifies the paper tape punch in this example.

Typing only a Carriage Return causes these options to assume their default values or the values most recently set.

After the user types a Carriage Return or Line Feed and turns on the punch, the tape is punched as usual.

NOTE: If an Alt Mode/Escape is typed during PROMPT, TAPE retains all option values set before the Alt Mode/Escape is typed.

An additional command, RUN, is available for symbolic reading or punching only. This command is included in TAPE to maintain compatibility with the Tymshare TYMCOM-IX TAPE program. The RUN command causes TAPE to prompt for information about the reading or punching operation to be performed.

1 - LENGTH and SUBSTITUTE can be reset to their default values as described on page 15.

When used for symbolic paper tape reading, RUN has the following form:

p: RUN↵
INPUT FROM: T↵ — *T, for terminal, specifies the paper tape reader.*
OUTPUT TO: file name↵
NEW FILE *or* OLD FILE {↵ / ⊕}
EDITING? {YES / B / NO}↵
TURN ON READER
n CHARACTERS ON FILE — *TAPE terminates paper tape reading in the same way as during the READ command.*

The question

EDITING?

allows the user to specify that certain characters be used to edit the incoming data, as shown in the following table.

User Response	Meaning
YES	Control A is used for character deletion; Control Q, for line deletion.
B	A back arrow (←) is used for character deletion. No line deletion is performed unless the user has defined a line-deleting character previously with the EDITING option.[1]
NO	No editing is performed.

In all cases, the value of the EDITING option is changed from its previous setting, unless that setting is identical to the new values specified.

When used for symbolic paper tape punching, RUN has the following form:

p: RUN↵
INPUT FROM: file name↵
OUTPUT TO: T↵
TITLE: 'title'↵
TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.

1 - See page 23 for a description of the EDITING option, which allows the user to define his own editing characters.

After the user types a Carriage Return or a Line Feed and turns on the punch, paper tape punching is performed just as during the PUNCH command. If the user specifies a title in response to the TITLE prompt, it is punched in readable text at the beginning of the paper tape, changing the value of the TITLE option from its previous setting. Alternatively, the user may type only a Carriage Return in response to the TITLE prompt, to cause the TITLE option to assume its default value or the value most recently set. He may also type NO or OFF to restore the TITLE option to its default value, no title.

Resetting an Option

Once the user has set an option, he may reset it to its default value with the NO command. The form is:

p: NO option name ↵

For example,

4: NO TITLE ↵
OK

resets the TITLE option to no title, and

5: NO LENGTH ↵
OK

resets the leader and trailer length to 5 inches.

Alternatively, any option except SUBSTITUTE and LENGTH may be reset to its default value by typing NO or OFF after the name of the option. For example:

6: EDITING NO ↵
OK

7: PARITY OFF ↵
OK

During the PROMPT command, NO or OFF may also be typed after any option prompt except LENGTH and SUBSTITUTE. For example,

8: PROMPT ↵
⋮
PARITY: OFF ↵
⋮

resets the PARITY option to its default value.

To reset the LENGTH option, the user must either type the NO command or type the actual default value. For example,

```
9: LENGTH 5↵
```

resets the LENGTH option at command level, and

```
10: PROMPT↵
:
LENGTH OF LEADER AND TRAILER: 5↵
:
```

resets the LENGTH option during the PROMPT command.

The SUBSTITUTE option may be reset to its default value, no conversions, only with the NO command.

All options may be reset to their default values simultaneously by using the command:

```
p: RESET↵
```

This command also resets the RADIX and HUSH commands.[1]

Listing Option Status

The LIST command prints the current status of all TAPE options and of the RADIX and TALK commands.[1] For example:

```
1: LIST↵

TALK ON                                  These are the standard, or default, values.
SUBSTITUTION:    OFF
USING:           TERMINAL
ECHO:            OFF
REPRESENTATION:  NONE
NO TITLE
NO EDITING
NO PARITY
LENGTH:          5 INCHES
RADIX:           DECIMAL

2:
```

In the example, USING: TERMINAL means that the user is reading or punching paper tape using the paper tape controls at his terminal. An option is available for users with access to a TYMCOM-X installation to use the paper tape reader and punch on the TYMCOM-X central processor. Consult your Tymshare representative for more information.

1 - RADIX is described on page 17; HUSH and TALK, on page 20.

USING OCTAL AND DECIMAL NUMBERS

Normally, numbers typed by the user during a TAPE session are interpreted as decimal numbers. For example,

```
2: LENGTH 10↵
OK
```

specifies a leader and trailer length of 10 inches.

There are times, however, when it is more convenient to enter an octal (base 8) number rather than a decimal (base 10) number, such as when specifying character codes for a code conversion. The RADIX command allows the user to change to octal mode or to return to the usual decimal mode. This command has the form:

p: RADIX { OCTAL / DECIMAL }↵

Example

```
3: RADIX↵
DEFAULT I-O RADIX: OCTAL↵          The user specifies octal mode.
OK

4: RADIX DECIMAL↵                  He resets the mode to decimal.
OK
```

In octal mode, *all* numbers typed by the user are interpreted as octal numbers; thus, in octal mode,

```
5: LENGTH 12↵
OK
```

specifies a leader and trailer length of 12 octal inches (equal to 10 decimal inches).

If the user wishes to enter a decimal number while in octal mode, he may follow the number with the letter D, for example, 15D.

To specify an octal number while in decimal mode, the user follows the number with the letter B, for example, 17B.

If TAPE prints a number at the terminal which is not in the current mode, it is followed by a B, if octal, and a D, if decimal. There is one exception: the command level prompt number is always a decimal number and is not followed by a D when in octal mode.

UTILITY COMMANDS

The following utility commands are available in TAPE.

Command	Description
CAPABILITIES	Describes program capabilities.
INSTRUCTIONS	Prints operating instructions for TAPE.
HELP	Prints a list of all TAPE commands, with a brief description of each. May be used within a command to obtain a description of the arguments available at that point in the command.
?	Prints a list of available commands, without descriptions. May be used within a command to obtain a list of arguments available at that point in the command.
HUSH	Suppresses or abbreviates usual messages printed by TAPE.
TALK	Terminates effect of HUSH.
CREDITS	Lists the author of the program; TAPE was written by Tymshare.
CHARGES	Lists any additional charges for using the program; TAPE has no additional charge.
VERSION	Prints the number of the current version of TAPE.
QUIT *or* Q	Returns control to TYMEX.

The HELP, ?, HUSH, and TALK commands are described in greater detail below.

HELP and ?

The command

p: HELP↵

prints a detailed list of all TAPE commands, with a brief description of each. The command

p: ?↵

prints a list, without descriptions, of all TAPE commands.

HELP and ? may also be used within a command, in place of any command word. When used in this manner, HELP prints a detailed list and descriptions of all arguments available at that point in the command; ? prints a list of these arguments. For example:

```
1: TITLE HELP↵

ARGUMENTS ARE:

<TITLE>  THE TITLE (ENCLOSED IN SINGLE OR DOUBLE QUOTES)
         THAT IS TO BE PUNCHED ON EVERY PUNCH OPERATION.  MAY
         OPTIONALLY BE PRECEEDED BY THE WORD "PUNCH" IN WHICH
         CASE THE TITLE GIVEN WILL BE PUNCHED IMMEDIATELY
         WITHOUT CHANGING ANY TITLE PREVIOUSLY ENTERED.
NO       TURNS THE TITLE OPTION OFF (DEFAULT)
OFF      SAME AS NO

1: PARITY ?↵

VALID ARGUMENTS ARE:
NONE
OFF
EVEN
ODD
DATA
HELP
PACK

1:
```

When HELP and ? are used in a command, the command level prompt number is not incremented, since the specified command is not executed.

If the user types HELP or ? at the end of a complete command, such as

1: TITLE OFF HELP↵

TAPE prints the message:

THE INPUT IS COMPLETE: NO FURTHER ARGUMENTS AVAILABLE

NOTE: HELP and ? cannot be used in place of a file name in any command used for reading or punching, because TAPE would interpret them as file names. Similarly, HELP and ? cannot be used in place of the file name in SSPLICEREAD, SAVE, LOAD, DO, and @.

HUSH and TALK

The HUSH command suppresses or abbreviates the usual conversation printed by TAPE, including the prompt number and colon (:). The TALK command terminates the effect of HUSH. For example:

```
1: HUSH↵
PUNCH TRANS↵


TURN PUNCH ON
↵                    The user types a Carriage Return and turns on the punch.

ABC SUPPLIES,4031,730215,150.66          The information being punched
CARTER PRINTING,4018,730220,500.00       is printed at the terminal.
DAILY DESIGN,3921,730225,150.00
KAY DEALERS,4251,730217,550.00
KUBLA PRINTS,399,730213,200.00
T T & G,3877,730228,271.30
TALK↵                TAPE terminates the PUNCH command and returns to command level; however, no prompt
OK                   is printed since HUSH is still in effect. The user types TALK to return to conversational
                     mode. Alternatively, he may type NO HUSH followed by a Carriage Return.
4:
```

SECTION 3

SYMBOLIC READING AND PUNCHING

TAPE may be used to read paper tape and store the information on a file in symbolic form, or to punch symbolic information from a file on paper tape. In addition, many convenient features are available during symbolic reading and punching.

When reading symbolic tape, the user may define his own editing characters to edit the incoming data. If desired, the contents of the tape being read can be displayed on the terminal. To protect the user from an interruption of tape reading due to a system failure or other emergency, the SPLICE and SSPLICEREAD commands are available.

During symbolic paper tape punching, the user can specify a title of readable text to be punched at the beginning of the tape and can specify the length of the leader and trailer to be punched on the tape.

A parity checking option is also available during paper tape reading and punching.

ASCII CODES

Symbolic form is simply the standard character representation, the same representation in which characters appear at the terminal. Specifically, each character is represented by a number known as its ASCII code. A character stored on a file is represented by its internal ASCII code; a character punched on paper tape is represented by its external ASCII code.[1] *NOTE: The internal and external ASCII codes for a character have the same numeric value on the TYMCOM-X.*

Certain special conventions are used to represent Carriage Returns and Line Feeds. A Carriage Return is represented by a Carriage Return followed by a Line Feed (octal codes 15 and 12); a Line Feed is represented by a Line Feed followed by a Carriage Return (octal codes 12 and 15). This rule always applies when typing characters on line; when punching tape off line, however, a Carriage Return is punched as a 15 alone, and a Line Feed is punched as a 12 alone. Thus, the following user-typed characters are equivalent:

Typed Off Line		Typed On Line
↵ ↴	equivalent to	↵
↴ ↵	equivalent to	↴

The user may instruct TAPE to print the octal or decimal ASCII code for any character by using the TYPE command. The form used is

p: TYPE character name↵

where the character name is a standard ASCII character name.[1]

1 - See Appendix A for an explanation of how to interpret punched tape. A table of ASCII codes and standard ASCII character names may be found in Appendix B.

This command causes TAPE to print the decimal code for the specified character if the user is in decimal mode, or the octal code if the user is in octal mode. For example:

```
1: TYPE A↵
A (65)
```
A is the standard ASCII character name for A. The decimal ASCII code for A is 65.

```
2: RADIX OCTAL↵
OK

3: TYPE A↵
A (101)
```
The octal code is 101.

Several character names, separated by commas, may be specified in one TYPE command. In the following example, octal mode is assumed.

```
4: TYPE B,ONE,COLON,BELL↵
B (102)
ONE (61)
COLON (72)
BELL (7)
```
These are the standard ASCII character names for B, 1, :, and G^C, respectively.

The TYPE command may also be used to print the character name for any ASCII code by appending a plus sign (+) to the code specified. For example, assuming that the user is in octal mode, the character whose octal code is 101 can be printed as follows:

```
5: TYPE 101+↵
A (101)
```

PAPER TAPE READING

Paper tape may be read in symbolic form using the READ, PROMPT, or RUN command.[1] In all cases, TAPE reads one 7-bit character per paper tape frame and writes it on a file in standard ASCII form. Five 7-bit characters are stored in one machine word. The eighth level of the paper tape is ignored unless otherwise specified with the PARITY option.[2] Rubouts (frames of tape with all bits punched) and nulls (frames of tape with no bits punched) are also ignored.

The READ command has the form:

```
p: READ file name↵
```

1 - The PROMPT and RUN commands are discussed on page 12 and page 13, respectively.

2 - See page 33 for details on parity.

After the command is entered, TAPE prints the NEW FILE or OLD FILE message, which may be confirmed or aborted, as described on page 8. TAPE then instructs the user to turn on the paper tape reader as in the following example.

```
1: READ PROBAB↵
NEW FILE↵

TURN ON READER

275 CHARACTERS ON FILE

2:
```

Paper tape reading terminates after a 10-second wait with no further characters read. *NOTE: Any characters typed at the terminal during this wait, including Alt Mode/Escape, are written on the file.* TAPE then prints the number of characters read and returns to command level.

When the READ command is used, all available options, such as ECHO, EDITING, and PARITY, assume either their default values or the values most recently set. If the user wishes to change an option value before using READ, he must use the appropriate option command first. For example, to request that TAPE print at the terminal the contents of a tape being read, he uses the commands:

```
2: ECHO ON↵
OK

3: READ TEST4↵
```

If the user has set ECHO ON previously and has not reset it, he may simply enter the READ command. The ECHO option still remains ON, since all options remain set until specifically reset with the appropriate command.

Two options useful during symbolic paper tape reading, the EDITING option and the ECHO option, are discussed below.

The EDITING Option

The EDITING option allows the user to define his own editing characters to edit the incoming data during symbolic paper tape reading. The user may define two editing characters: a line-deleting character and a character-deleting character. When read from paper tape, the line-deleting character deletes the entire line read so far, and the character-deleting character deletes the last character read. Thus, these characters perform the same functions that Control Q and Control A do during command editing.[1]

The EDITING option may be set either with the EDITING command or in response to the EDITING prompt during the PROMPT command.

1 - Control Q and Control A are discussed on page 9.

The EDITING command has the form

p: EDITING LINE='l' CHARACTER='c' ↵

where l and c are the line-deleting and character-deleting characters that the user wishes to define. Both l and c may be surrounded by single or double quotation marks; quotation marks are required only when specifying a numeric character. The command may be used to define only the line-deleting character, only the character-deleting character, or both.

NOTE: The user must type a space after the word EDITING and also between the definitions of the line- and character-deleting characters, but spaces may not be typed between the word CHARACTER or LINE and the following equals sign (=).

Examples

1: EDITING LINE=↑ ↵

Defines ↑ as the line-deleting character. When read from paper tape, ↑ deletes the entire line read so far.

2: EDITING CHARACTER=← ↵

Defines ← as the character-deleting character. When read from paper tape, ← deletes the last character read.

3: EDITING LINE=↑ CHARACTER=← ↵

Defines the line-deleting character to be ↑ and the character-deleting character to be ←.

4: EDITING CHARACTER='1' LINE=< ↵

Defines a 1 as the character-deleting character and a < as the line-deleting character. Note that the parts of the command may be interchanged.

The line deleted by the line-deleting character is defined as all preceding characters up to but not including the preceding Carriage Return or Line Feed read. The line-deleting character cannot delete more than one line.

The character-deleting character may be used repeatedly to delete several characters; however, it cannot be used to delete characters in preceding lines.

When setting the EDITING option, the user may define the editing characters by entering the equivalent ASCII codes or the standard ASCII character names instead of entering the actual characters.[1] The ASCII codes or character names must not be surrounded by quotation marks. The decimal form of the ASCII codes is assumed; however, octal codes may be entered by following each code typed with the letter B or by changing to octal mode with the RADIX command.

1 - See Appendix B for a table of ASCII codes and standard ASCII character names.

Examples

5: EDITING CHARACTER=BACKARROW↵

Uses the standard ASCII character name to define the back arrow (←) as the character-deleting character.

6: EDITING CHARACTER=1 LINE=17↵

Uses the decimal ASCII codes for Control A and Control Q to define them as the character-deleting character and the line-deleting character, respectively.

7: EDITING LINE=21B↵

Uses the octal ASCII code for Control Q to define it as the line-deleting character.

8: EDITING LINE=DC1 CHARACTER=SOH↵

Uses the standard ASCII character names for Control Q and Control A to define them as the line-deleting character and the character-deleting character, respectively.

A control character may be defined as an editing character by entering the ASCII code or character name for the desired control character or by typing a Control V before the control character itself to inhibit its usual function. When the user types Control V followed by a control character, the system prints an ampersand (&) followed by the letter for that control character. For example,

9: EDITING CHARACTER=V^CA^C&A LINE=V^CQ^C&Q↵

defines Control A as the character-deleting character and Control Q as the line-deleting character.

The word NOTHING may be used to reset an editing character so that it no longer exists. For example, after the commands

```
10: EDITING CHARACTER=\ LINE=L↵
OK

11: EDITING CHARACTER=NOTHING↵
OK
```

the line-deleting character L is still defined, but there is no longer a character-deleting character.

To set the EDITING option during PROMPT, the user defines his editing characters in response to the EDITING prompt, using the same rules as during the EDITING command. For example:

```
1: PROMPT↵
:
EDITING: CHARACTER=← LINE=17↵
:
```

Uses the character back arrow (←) and the ASCII code 17 to define the editing characters.

The ECHO Option

The ECHO option allows data being read from paper tape to be printed at the terminal. It may be set using the ECHO command in the form

p: ECHO ON↵

or during PROMPT in response to the ECHO prompt as follows:

1: PROMPT↵
⋮
ECHO: ON↵
⋮

CAUTION: Users with terminals that read information at 30 characters per second should avoid using ECHO, since doing so may cause erroneous information to be stored on the file due to buffering requirements.

Interruption Recovery

The TAPE program is designed so that a simple recovery can be made if symbolic paper tape reading is interrupted by a system failure, a telephone line disconnection, or other emergency interruption. Two commands are available for this purpose:

Command	Description
SPLICE	Used to recover from a system failure.
SSPLICEREAD	Used to recover from a telephone line disconnection or other emergency interruption.

When paper tape is being read, the characters read are periodically written on the file. Thus, if an interruption occurs when reading a lengthy paper tape, part of the information read is stored on the file. Paper tape reading may then be completed, without rereading the entire tape, by using SPLICE or SSPLICEREAD, as appropriate.

To recover from a system failure, the user logs in again, calls TAPE, and positions the tape at what he estimates to be the last 10 inches read before the interruption occurred. He then uses SPLICE in the form

p: SPLICE job number↵

where the job number is the job number printed by TAPE during the session in which the failure occurred, not the one printed when TAPE is called again to do the recovery.

After the SPLICE command is entered, TAPE prints the last characters in the file being spliced to aid the user in finding the right place to splice the tape. TAPE then prints:

READ IN ABOUT 10 INCHES OF TAPE

The user now turns on the paper tape reader, reads in about 10 inches of tape, and turns off the reader. If TAPE can match the information read with that already stored in the file, it prints:

FOUND MATCH
TURN ON READER

When the tape has been read, TAPE prints both the total number of characters in the file and the number of characters read during SPLICE.

If a match is not found, TAPE prints:

NO MATCH FOUND
TRY AGAIN?

This question may be answered by typing YES or NO, followed by a Carriage Return. If the user does not wish to continue the recovery attempt, he types NO followed by a Carriage Return, and control returns to TAPE command level. To continue searching for a match, the user should reposition the tape about 5 feet back before typing YES and a Carriage Return. TAPE then repeats the request

READ IN ABOUT 10 INCHES OF TAPE

and again attempts to find a match. This process may be repeated as many times as necessary to find a match, or until the beginning of the tape is reached.

Example

A system failure occurs while the user is reading a paper tape to the file named SSR. His job number at the time the failure occurs is 15. To recover from the interruption, he logs in again and proceeds as follows:

```
-R TAPE↵
YOUR JOB ID IS 20

1: SPLICE 15↵

THE LAST 48 CHARACTERS ON THE FILE ARE:

"ED AND THE LAST FILE NAME
  BY A CARRIAGE RE"


READ IN ABOUT 10 INCHES OF TAPE
```

The user turns on the reader and reads about 10 inches of tape.

```
NO MATCH FOUND
```

```
TRY AGAIN? YES↵                    He positions the tape about 5 feet back and types YES.

READ IN ABOUT 10 INCHES OF TAPE

FOUND MATCH

TURN ON READER                     The user turns on the reader and reads the rest of the tape.

7512 CHARACTERS ON FILE            There are 7512 characters on the file,
808 TOTAL CHARACTERS READ          808 of which were read during SPLICE.

SPLICE COMPLETED.

2:
```

If only a small number of characters were read before the failure, TAPE responds to the SPLICE command as follows:

```
1: SPLICE 5↵
ONLY 194 CHARACTERS WERE READ INTO THE OUTPUT FILE
IT WOULD BE FASTER TO RE-READ THE TAPE
```

When a system failure occurs during paper tape reading, TAPE automatically creates a temporary file containing the status of all options used. The file is named BxxTAP.TMP, where xx is the user's job number. Since the file is temporary, it is automatically deleted when the user logs out; however, it is not deleted in the event of a system failure. TAPE uses the information in the temporary file to perform the recovery and then deletes the file.

Temporary files are deleted when a telephone line disconnection occurs; hence, SPLICE cannot be used to recover from such an interruption. In this case, recovery can be made using the SSPLICEREAD command. In fact, SSPLICEREAD can be used to recover from any interruption in which a BxxTAP.TMP file is not retained in the user's directory. For example, if the paper tape being read becomes tangled, the user can stop the paper tape reader, untangle the tape, and resume reading with SSPLICEREAD.

Two steps are needed to use the SSPLICEREAD command:

1. The user must set all applicable options to the same values used when the interruption occurred.
2. He then uses the SSPLICEREAD command in the form

   p: SSPLICEREAD file name↵

   where the file name is the name of the file to be spliced.

TAPE then proceeds just as during the SPLICE command.

Example

An interruption occurs while the user is reading a symbolic tape to the file SSR, with even parity checking and a back arrow (←) defined as the character-deleting character. He recovers as follows:

```
1: EDITING CHARACTER=BACKARROW↵
OK

2: PARITY EVEN↵
OK

3: SSPLICERFAD SSR↵

THE LAST 48 CHARACTERS ON THE FILE ARE:

"ED AND THE LAST FILE NAME
  BY A CARRIAGE RE"


READ IN ABOUT 10 INCHES OF TAPE

FOUND MATCH

TURN ON READER

8032 CHARACTERS ON FILE
1003 TOTAL CHARACTERS READ

SPLICE COMPLETED.

4:
```

First, the user sets the options to the same values used when the interruption occurred.

Then he enters the SSPLICEREAD command.

PAPER TAPE PUNCHING

Symbolic information stored in a file may be punched on paper tape using the PUNCH, PROMPT, or RUN command.[1] Each character in the file is punched on one paper tape frame in 7-bit ASCII code.[2] The eighth level of paper tape is not punched unless specifically requested with the PARITY option.[3]

The PUNCH command has the form:

p: PUNCH file name↵

After the user enters the command, TAPE instructs the user:

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.

1 - The PROMPT and RUN commands are described on page 12 and page 13, respectively.

2 - See Appendix A for a description of how to interpret punched tape.

3 - See page 33 for a discussion of parity.

If the user types a Line Feed, TAPE punches the file name, the file type, and the value of the PARITY option in readable text at the beginning of the tape.[1] The user may bypass this option by typing a Carriage Return. In either case, TAPE punches any title specified by the user. TAPE then punches the information on paper tape and prints it at the terminal. After the tape is punched, TAPE returns to command level; the user turns off the paper tape punch.

Example

```
1: PUNCH EXPD.F4↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵                                   The user types a Carriage Return and turns on the punch.

00010          ACCEPT 10,A,B,C
00015    10    FORMAT(3F8.3)
00020          D=SQRT(A**B/2*C)
00030          ANS=EXP(D)
00035          TYPE 20,ANS
00040    20    FORMAT(F12.4)
00045          END

2:                                  After punching the information, TAPE returns control to command level.
```

TAPE uses the following format, in the following order, when punching a paper tape:

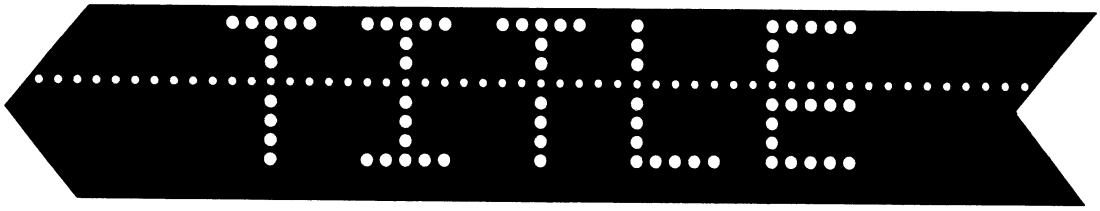
1. The file name, the file type, and the value of the PARITY option, if requested by typing a Line Feed before turning on the punch.
2. The user-specified title, if any.
3. Blank leader of the length specified by the user. The default value is 5 inches.
4. Three rubouts.[2]
5. The actual information.
6. Three rubouts.
7. Blank trailer of the same length as the leader.

1 - The file type is punched as SYM, for symbolic, or BIN, for binary.

2 - A rubout is a frame of paper tape with all eight holes punched.

The TITLE Option

A title of readable text may be punched on a paper tape to identify it to the user. For example:



The title may be specified either with the TITLE command or in response to the TITLE prompt during the PROMPT command.

The TITLE command may be used either to specify a title to be punched during the RUN, PUNCH, PROMPT, or BPUNCH command or to punch only a title.[1] The following forms of the command are used:

p: TITLE 'title'↵ *Specifies a title to be punched during paper tape punching.*

p: TITLE PUNCH 'title'↵ *Punches only a title, without changing any title previously defined to be punched during paper tape punching.*

In either case, the title must be surrounded by single or double quotation marks. If the user wishes to include single quotation marks in the title, the title must be surrounded by double quotation marks, and vice versa. For example:

```
1: TITLE "BOB'S PROGRAM"↵
OK

2:
```

If necessary, a Line Feed may be used to continue the title onto the next line at the terminal; the Line Feed is not punched in the title. A title may contain as many as 240 characters.

1 - BPUNCH is discussed on page 40.

When TITLE is used to punch only a title, the procedure is as follows:

```
2: TITLE PUNCH 'TAPE'↵

TYPE CARRIAGE RETURN OR LINE FEED.      In this case, a Carriage Return and a Line Feed
THEN TURN ON PUNCH.                     are equivalent; only the specified title is punched.
↵

IIII                                    Extraneous characters appear on the terminal when the title is being punched.
3:
```

After the title is punched, TAPE returns control to command level.

To specify a title during PROMPT, the user types a title in response to the TITLE prompt. The title is surrounded by single or double quotation marks, as in the following:

```
1: PROMPT↵
:
TITLE: 'INVENTORY'↵
:
```

The LENGTH Option

TAPE normally punches 5 inches of blank leader and 5 inches of blank trailer on a paper tape. If the user wants a leader and a trailer of a different length, he may specify this with the LENGTH command or during the PROMPT command.

The LENGTH command has the form:

```
p: LENGTH number of inches↵
```

The number of inches specifies the length of both the leader and the trailer; any number of inches, from 0 through 511, may be specified. Thus,

```
1: LENGTH 0↵
OK

2:
```

specifies no leader and no trailer.

To set the LENGTH option during PROMPT, the user types the number of inches desired in response to the prompt:

```
LENGTH OF LEADER AND TRAILER:
```

For example,

```
1: PROMPT↵
:
LENGTH OF LEADER AND TRAILER: 10↵
:
```

sets the leader and the trailer lengths to 10 inches.

PARITY

The symbolic codes punched on paper tape are all 7-digit binary numbers; however, there are eight levels, or bits, available on a paper tape frame.[1] During symbolic reading and punching, the eighth level of the paper tape is not used and, therefore, is available for other purposes.[2] This extra level, called the parity bit, may be used for parity checking.

TAPE contains several PARITY options which allow the user to define how the parity bit is to be handled. During symbolic reading and punching, the following PARITY options are available:

Option	Description	
	During Punching	During Reading
EVEN	Parity bit is punched only if there is an odd number of bits punched in levels 1–7, thus making the total number of bits in a frame even.	Tape being read must have even parity; if not, an error message is printed.
ODD	Parity bit is punched only if there is an even number of bits punched in levels 1–7, thus making the total number of bits in a frame odd.	Tape being read must have odd parity; if not, an error message is printed.
DATA	Parity bit is always punched.	Tape being read must always have the parity bit punched; if not, an error message is printed.
OFF *or* NO (default value)	Parity bit is never punched.	Parity bit is ignored.

These options may be set with the PARITY command or in response to the PARITY prompt during the PROMPT command.

1 - Level and frame are defined in Appendix A.

2 - When paper tape is punched off line, the eighth level is always punched.

Example

1: PARITY EVEN↵ **OK**	*Specifies even parity.*
2: PARITY ODD↵ **OK**	*Specifies odd parity.*
3: PARITY DATA↵ **OK**	*Specifies that the eighth level is always punched; checks that the eighth level is always punched during reading.*

To set the PARITY option during PROMPT, the user simply types the desired option after the PARITY prompt. For example,

```
4: PROMPT↵
:
:
PARITY: OFF↵
:
:
```

resets the PARITY option to its default value.

When parity errors are detected, TAPE finishes reading the tape, creates a file in the user's directory containing the location(s) of the parity error(s) found, and prints the message

THERE WERE n PARITY ERROR(S) WHILE READING YOUR TAPE
PLEASE READ THE FILE AxxTAP.TMP FOR DETAILS

where n is the number of parity errors found, and xx is the user's job number. The extension TMP means that the file is a temporary file; it is automatically deleted when the user logs out. Thus, the user should examine the file before logging out or rename the file with the TYMEX command RENAME.[1] For example:

```
-RENAME A16TAP.TMP AS PAR16↵
```

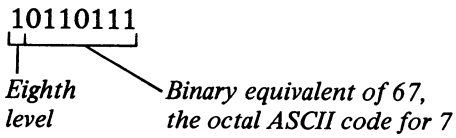
Example

Suppose that the user punches the following symbolic file on paper tape with even parity

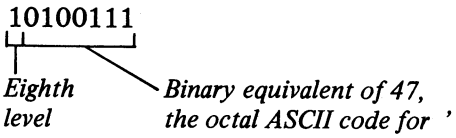
```
-TYPE DATA1↵
A26
B79
C43

-
```

1 - The RENAME command is discussed in the *Tymshare TYMCOM-X TYMEX Reference Manual.*

but an error occurs during data transmission so that the code for 7, normally punched with even parity as



is punched instead as:



Since the error caused a character to be punched on the tape with odd parity, the following occurs when the user reads the tape with even parity.

```
-R TAPE↵
YOUR JOB ID IS 13

1: PARITY EVEN↵
OK

2: READ DATA2↵
NEW FILE↵

TURN ON READER

15 CHARACTERS ON FILE

THERE WERE 1 PARITY ERROR(S) WHILE READING YOUR TAPE
PLEASE READ THE FILE A13TAP.TMP FOR DETAILS

3: QUIT↵

-TYPE A13TAP.TMP↵          This is the parity error file.
PARITY ERRORS WHILE READING FILE DATA2


IN LINE 2
B*9(CR)     TAPE prints (CR) or (LF) to show a Carriage Return or a Line Feed in a line containing a parity error.
 ↑          The up arrow (↑) points to the parity error. It would point to the left parenthesis
            if the Carriage Return or Line Feed contained the parity error.

-TYPE DATA2↵                This is the file created during READ.
A26
B*9
C43

-
```

Thus, the user discovers the bad character in his file and can now correct it.

SECTION 4

BINARY READING AND PUNCHING

In addition to the standard symbolic form, TAPE offers the user a choice among several different binary forms for paper tape reading and punching.

The only difference between symbolic and binary forms is the way in which information is represented. On a symbolic file, characters are represented internally as 7-bit ASCII codes; five characters are stored in one internal machine word.[1] A binary file is simply a file in which information is represented internally in some form other than the standard ASCII character representation. Binary files cannot be listed meaningfully in TYMEX or in EDITOR, since these programs are designed to list ASCII characters only.

AVAILABLE BINARY FORMS

The TAPE user may specify the desired binary form using the PARITY option. Both 6-bit and 8-bit representations are available; when 6-bit representation is used, even or odd parity checking can be performed. The following table describes the PARITY options used for binary reading and punching.

PARITY Option	Description	
	During Reading	During Punching
OFF or NO (6-bit; this is the default value)	Six-bit codes are read from levels 1–6 of each paper tape frame and written on a file, six 6-bit bytes per machine word, filling the word. Bits 7–8 on paper tape are ignored.	Six-bit bytes from a file are punched on paper tape, one per frame. Six 6-bit bytes are used from each word in the file; bits 7–8 are not punched.
EVEN (6-bit)	Same as OFF, except parity bit (eighth level) is checked for even parity; bit 7 is ignored.	Same as OFF, except parity bit is punched when there is an odd number of bits punched in levels 1–6. Bit 7 is never punched.
ODD (6-bit)	Same as EVEN, except parity bit is checked for odd parity.	Same as EVEN, except parity bit is punched when there is an even number of bits punched in levels 1–6.

1 - One word contains 36 bits, so one bit remains unused in each word when storing symbolic information.

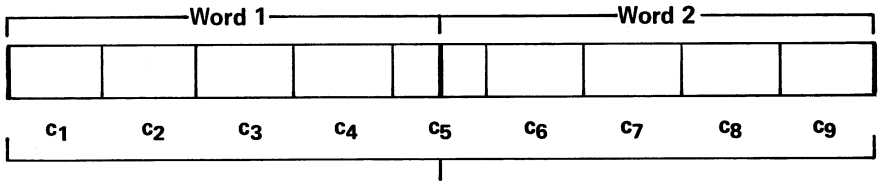
PARITY Option	Description	
	During Reading	During Punching
PACK (8-bit)	Eight-bit codes are read from tape and written on a file, overlapping word boundaries. (See Note 3 below.)	Eight-bit bytes from a file are punched, one per frame of paper tape. (See Note 3 below.)
DATA (8-bit)	Eight-bit codes are read from tape and written on a file, one code per word right-justified in the word.	One 8-bit byte per machine word is read from a file and punched on paper tape, one per paper tape frame. Only the right-most 8 bits are used from each word on the file.

Notes

1. If a PARITY option is not set, the default form for binary reading and punching is equivalent to PARITY OFF in the above table; that is, 6-bit bytes are used when reading or punching tape.

2. No parity checking can be performed when using the PACK and DATA options, since these options specify that all eight levels of the paper tape are to be used for storing information.

3. When reading tape with the PACK option, the first four 8-bit codes from the paper tape are stored in the first 32 bits, bits 0—31, of a machine word. The next code read from paper tape is stored overlapping a word boundary, the first 4 bits being stored in bits 32—35 of the current word and the next 4 bits in bits 0—3 of the next word. The next four 8-bit codes read are stored in bits 4—35 of the same word, so that a code ends at a word boundary in every other word, as in the following diagram.



Nine 8-bit codes, c_1 through c_9, stored in two 36-bit words

When punching tape with the PACK option, the same scheme is used; that is, the first 8 bits from the first word in the file are punched on one frame of tape, then the second 8 bits, and so on.

The options in the preceding table may be set with the PARITY command or during the PROMPT command, as shown in the following examples.

Example 1

```
1: PARITY DATA↵
OK
```
Specifies that 8-bit bytes right-justified in a word be used for binary reading or punching.

```
2: PARITY EVEN↵
OK
```
Specifies that 6-bit bytes be used for binary reading or punching; the parity bit is used for even parity.

```
3: PARITY OFF↵
OK
```
Resets PARITY option to its default value. Six-bit bytes are used for binary reading or punching; no parity checking is performed.

Example 2

```
4: PROMPT↵

SYMBOLIC OR BINARY? BINARY↵

PARITY: PACK↵
```
Specifies that 8-bit bytes overlapping word boundaries be used for binary reading or punching.

PAPER TAPE READING

Paper tape may be read in binary form using either the BREAD command or the PROMPT command. The EDITING, SUBSTITUTE, and REPRESENTATION options may not be used when reading binary paper tape. The ECHO option can be used, but the characters printed at the terminal are meaningless, since ECHO is designed to print ASCII characters.

NOTE: To read a binary paper tape, the user must position the tape on the paper tape reader at the first frame following the leading three rubouts.[1] *In addition, he must tear off the paper tape at the last trailing rubout. These steps are necessary to avoid reading the rubouts and nulls in the leader and trailer, since they are not ignored during binary paper tape reading.*

The BREAD command is analogous to the READ command, except that it prints the number of machine words stored on the file instead of the number of characters. The command form is:

p: BREAD file name↵

1 - The format of leader, rubouts, and trailer for binary paper tape punched by TAPE is the same as that for symbolic tape described on page 30.

For example:

```
1: BREAD BIN↵
NEW FILE↵

TURN ON READER

45 WORDS ON FILE          After the tape is read, TAPE waits 10 seconds and
                          then prints the number of words on the file.
2:
```

In this example, since no PARITY command is entered prior to using BREAD, 6-bit codes are read from the paper tape and stored on the file BIN.

The following example illustrates using PROMPT for binary paper tape reading.

```
2: PROMPT↵

SYMBOLIC OR BINARY? BINARY↵

PARITY: PACK↵             Specifies 8-bit codes to be read from tape and
                          stored on file overlapping word boundaries.
INPUT FROM:  T↵

OUTPUT TO:   BTEST↵
NEW FILE↵

ECHO:↵

TURN ON READER

268 WORDS ON FILE

3:
```

PAPER TAPE PUNCHING

The BPUNCH command or the PROMPT command may be used for binary paper tape punching. The TITLE and LENGTH options available with symbolic paper tape punching are also available with binary paper tape punching; they are used the same way in both cases.[1] The format of leader, rubouts, and trailer when punching paper tape is the same as that for symbolic paper tape described on page 30.

The BPUNCH command is analogous to the PUNCH command. The command form is:

p: BPUNCH file name↵

1 - See page 31 for a discussion of the TITLE option, and page 32 for a discussion of the LENGTH option.

For example:

```
1: BPUNCH EXPD.REL⊃

ALWAYS PUNCH 8-TH LEVEL? NO⊃        The user does not want the eighth level of the paper tape
                                    punched.
TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
⊃                                   The user types a Carriage Return and turns on the punch.

#"%???' ???(0 ??<???(???3 ???(831*???  ???(???177 ???(8 ????????
                                    The contents of the tape appear as meaningless characters, since it is not symbolic.

#$"

2:                                  After punching the information, TAPE returns control to command level.
```

In this example, 6-bit bytes from the file are punched on paper tape; the eighth level, or parity bit, is never punched, since the PARITY option was not set before using BPUNCH. Note, however, that the user has the option to always punch the eighth level. To do this, he simply answers the following question as illustrated:

ALWAYS PUNCH 8-TH LEVEL? YES⊃

This question is asked during BPUNCH or PROMPT only when the PARITY option is at its default value (OFF or NO), since all other PARITY options specify a different use for the parity bit.

The following example illustrates using PROMPT for binary paper tape punching.

```
1: PROMPT⊃

SYMBOLIC OR BINARY? BINARY⊃

PARITY: DATA⊃          Specifies right-most 8 bits of each word on the file are to be punched on tape.

INPUT FROM:  PDATA⊃

OUTPUT TO:   T⊃

TITLE: 'PERSONNEL'⊃                 The user specifies a title to be punched on the tape.

LENGTH OF LEADER AND TRAILER: 15⊃   He specifies leader and trailer lengths of 15 inches.

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
⊃

IIII)F&III2>AAA>  IIII@@@@&@FXL@(HN@\@D@B@2HNN\@(RD@@FNP\

2:
```

SECTION 5

CODE CONVERSIONS

Two code conversion options are featured in TAPE: SUBSTITUTE and REPRESENTATION.

The SUBSTITUTE option provides a general-purpose conversion routine. Using SUBSTITUTE, any characters on a symbolic file may be read, converted to any other set of codes, and punched on paper tape. Conversely, a paper tape may be read, the codes converted, and the converted contents written on a symbolic file.

REPRESENTATION is a special-purpose conversion option which allows the user to represent characters on a file or a tape literally; that is, characters can be represented by their numeric ASCII codes instead of their usual character representations. Using REPRESENTATION, each character read from a paper tape can be converted to its octal or decimal ASCII code and the converted codes stored on a symbolic file, separated by Carriage Returns. Conversely, each character on a symbolic file can be converted to its octal or decimal ASCII code, and the codes punched on paper tape, separated by Carriage Returns. REPRESENTATION can also be used to convert a file or a tape containing octal or decimal ASCII codes separated by Carriage Returns to a standard ASCII symbolic tape or file.

THE SUBSTITUTE OPTION

The SUBSTITUTE option is used to convert character codes during symbolic reading or punching. The desired conversions may be specified with the SUBSTITUTE command or during the PROMPT command. Features of SUBSTITUTE include the ability to:

- Specify code conversions by typing the characters themselves, the desired codes in octal or decimal form, or the standard ASCII character names.
- Ignore any character codes.
- Substitute a string of character codes for one code.
- Substitute the same character code for several different character codes.
- Save a table of conversions on a file for use at a later time.

Command Forms

The SUBSTITUTE command has the form:

p: SUBSTITUTE↵

After the user enters the command, TAPE responds:

SUBSTITUTE:
FILE,TAPE

TAPE then rings the bell and waits for the user to specify his conversions. The user may specify as many as 200 conversions. Each conversion is entered on a separate line in the form:

file representation,tape representation↵

After all conversions are entered, the user types a Carriage Return to terminate the command.

Example

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE      Conversions specified using:
A,L↵                 Characters
'1','6'↵
107,110↵             Decimal ASCII codes
DC1,Q↵               Standard ASCII character names
↵              This Carriage Return terminates the command.

2:
```

There are three different ways to specify the codes to be converted:

1. By typing the ASCII characters for the codes, for example:

   A,B↵

   The characters may be surrounded by single or double quotation marks; nonalphabetic characters *must* be surrounded by quotation marks, as in:

   '6','!'↵

   If a single quotation mark is specified, it must be surrounded by double quotation marks; a double quotation mark must be surrounded by single quotation marks.

2. By typing the character codes themselves. For example, the octal ASCII code for A is 101; the code for B is 102.[1] Thus:

```
1: RADIX OCTAL↵     The user specifies octal mode so that he
OK                  can enter octal rather than decimal codes.

2: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
101,102↵            Equivalent to A,B followed by a Carriage Return.
↵

3:
```

   Any code in the range 0—177 octal, inclusive, may be specified.

1 - Appendix B contains a table of ASCII codes.

3. By typing the standard ASCII character names for the codes.[1] For example, the standard ASCII character name for Control Q is DC1; that for a back arrow (←) is BACKARROW. Thus:

```
3: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
DC1,BACKARROW↵
↵

4:
```

Code conversions are specified during PROMPT in the same way as during the SUBSTITUTE command. TAPE prompts

SUBSTITUTE:
FILE,TAPE

and the user enters the desired conversions. A Carriage Return terminates the SUBSTITUTE prompt and causes TAPE to prompt for the next option.

A short form of the SUBSTITUTE command is available for specifying only one conversion. The form is:

p: SUBSTITUTE file representation,tape representation↵

The word FOR may be substituted for the comma in the long form or the short form of SUBSTITUTE. FOR is equivalent to the comma.

Example

```
1: SUBSTITUTE '6',*'↵                    Short form.
OK

2: SUBSTITUTE↵                           Long form with FOR.

SUBSTITUTE:
FILE,TAPE
W FOR '9'↵
↵

3: SUBSTITUTE '&' FOR 62↵                Short form with FOR.
OK

4:
```

1 - Appendix B contains a table of standard ASCII character names.

Using SUBSTITUTE

When using SUBSTITUTE, the first character representation entered in each line is always the file representation, and the second is always the tape representation. For example, consider the command:

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
A,B↵
↵

2:
```

If the conversion A,B defined above is used for paper tape reading, each B on the paper tape is converted to A and stored on the file. If A,B is used for paper tape punching, each A on the file is converted to B and punched on the tape. In other words, A is substituted for B during paper tape reading, but B is substituted for A during paper tape punching.

If an Alt Mode/Escape is typed during SUBSTITUTE, the specified conversions are still defined.

The leading and trailing rubouts and blank leader and trailer on a paper tape are ignored when conversions are performed.

The following examples illustrate using SUBSTITUTE for paper tape punching and reading.

Example 1

The user has the following symbolic file, named CHARS1:

```
-TYPE CHARS1↵
RBCDQFGHI:KLMNO6

-
```

He wishes to make the following changes when the file is punched on paper tape:

R changed to A
Q changed to E
: changed to J
6 changed to P

To make these changes, the user proceeds as follows:

```
1: SUBSTITUTE↵
```

First, the user defines the conversions to be made.

```
SUBSTITUTE:
FILE,TAPE
R,A↵
Q,E↵
':',J↵
'6',P↵
↵
```

```
2: PUNCH CHARS1↵
```

Then he punches the information on paper tape.

```
TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵
```

```
ABCDEFGHIJKLMNOP
```

The information is punched with the specified conversions.

```
3:
```

Example 2

The user wishes to make the same conversions as in the above example, but the original data is on paper tape and the converted data is to be stored on the file CHARS2. He has a paper tape containing the codes for

RBCDQFGHI:KLMNO6

and wishes to make the following conversions:

R changed to A
Q changed to E
: changed to J
6 changed to P

This time, the conversions are specified in the reverse order as follows:

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
A,R↵
E,Q↵
J,':'↵
P,'6'↵
↵
```

```
2: READ CHARS2↵
NEW FILE↵

TURN ON READER

18 CHARACTERS ON FILE

3: QUIT↵

-TYPE CHARS2↵
ABCDEFGHIJKLMNOP          This is the converted information.
-
```

Ignoring Characters

The word NOTHING specifies that a character is to be ignored during a code conversion. For example, the user specifies that the character Q is to be ignored during paper tape reading.

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
NOTHING,Q↵
↵

2:
```

Using this conversion, the information LQS is read from paper tape and stored on a file as LS.

During paper tape punching, the SUBSTITUTE command shown above is meaningless and causes an error message to be printed. To ignore a character during paper tape punching, the order must be reversed in the SUBSTITUTE command. Thus,

```
5: SUBSTITUTE Q,NOTHING↵
```

specifies that Q is to be ignored during paper tape punching, but causes an error message to be printed during paper tape reading.

Substituting a String of Character Codes for One Code

SUBSTITUTE allows the user to convert one character code to a string of character codes when reading or punching tape. He may enter a string of character codes either as a string of characters enclosed in single or double quotation marks or as a list of characters separated by commas.

SUBSTITUTE does not, however, perform the reverse; that is, the user cannot convert a string of codes to one code. If such a conversion is specified, only the first character code in the string is used. In particular, this occurs if the same substitutions are used for both reading and punching. Thus, if the substitutions below are used for both reading and punching, ← is

converted to ABC when reading a tape, but, when punching a tape, A is converted to ←. Similarly, 6 is converted to 23 when punching a tape, but, when reading a tape, 2 is converted to 6.

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
'ABC','←'↵
'6','23'↵
↵

2:
```

When the user enters a list of characters separated by commas, the word FOR must be used to separate the file representation from the tape representation. For example:

```
3: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
A,B,C FOR TWO↵
24 FOR 62,14↵
CR,LF FOR CR↵
↵

4:
```

TWO is the standard ASCII character name for 2.

CR and LF are the standard ASCII character names for a Carriage Return and a Line Feed, respectively.

The advantage of this method is that the conversions may be specified by typing the ASCII codes or the standard ASCII character names as well as the characters themselves. When quoted strings are used, however, only the characters themselves may be typed.

Substituting One Character Code for Several Codes

Using SUBSTITUTE, several different character codes may be converted to the same character code. When the user specifies such conversions, TAPE prints a message indicating that a potential conflict exists; however, the conversions are still made where possible. The message simply warns the user that, while the substitutions specified may be valid when reading tape, they result in a conflict when punching tape, or vice versa. For example, suppose the user has the following information on a file named DATA1:

ABC6644422

He wishes to convert A, B, and C to the character 1 when he punches the file on paper tape, so he uses SUBSTITUTE as follows:

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
A,'1'↵
B,'1'↵
CONFLICT WITH RIGHT PART.  "1" ALREADY SUBSTITUTES FROM:
"A"
C,'1'↵
CONFLICT WITH RIGHT PART.  "1" ALREADY SUBSTITUTES FROM:
"A"
↵

2: PUNCH DATA1↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

1116644422        This is the converted information.

3:
```

The desired conversions are made. If the same conversions are specified for paper tape reading, however, a conflict exists. When such a conflict exists, TAPE uses the first conversion specified and ignores the remaining conflicting conversions. For example, if the SUBSTITUTE command in the above example were used when reading the tape punched above, an A would be substituted for each 1 on the tape; the other two conversions would be ignored. Thus:

```
4: READ CTEST↵
NEW FILE↵

TURN ON READER

12 CHARACTERS ON FILE

5: QUIT↵

-TYPE CTEST↵
AAA6644422        This is the converted information.

-
```

The Substitution Table: PRINT, SAVE, and LOAD

TAPE stores the code conversions specified during a SUBSTITUTE command in a substitution table. The user may display this table by using the PRINT command. For example:

```
1: SUBSTITUTE↵
```
The user defines a substitution table.

```
SUBSTITUTE:
FILE, TAPE
A, L↵
'1', ':'↵
75, 78↵
DC1, Q↵
↵
```
These are the decimal ASCII codes for K and N, respectively.

```
2: PRINT↵
```
The user displays the substitution table at the terminal.

```
SUBSTITUTE:
FILE, TAPE

"A", "L"
"1", COLON
"K", "N"
DC1, "Q"

3:
```

TAPE uses the following conventions when printing the substitution table:

1. Alphabetic and numeric characters are surrounded by double quotation marks.
2. Nonalphanumeric characters are represented by their standard ASCII character names, such as COLON and DC1 above.

If the SUBSTITUTE command is used more than once during a TAPE session, the additional conversions are appended to the existing substitution table; the existing table is not deleted.

Example

```
3: PRINT↵

SUBSTITUTE:
FILE, TAPE

"A", "L"
"1", COLON
"K", "N"
DC1, "Q"
```
This substitution table is already defined.

```
4: SUBSTITUTE↵
```

The user defines additional conversions.

```
SUBSTITUTE:
FILE,TAPE
'6','2'↵
X,Y↵
↵

5: PRINT↵

SUBSTITUTE:
FILE,TAPE

"A","L"
"1",COLON
"K","N"
DC1,"Q"
"6","2"
"X","Y"

6:
```

The new conversions are appended to the existing table.

To delete an existing substitution table, the user types:

p: NO SUBSTITUTE↵

Example

```
3: PRINT↵

SUBSTITUTE:
FILE,TAPE

"A","L"
"1",COLON
"K","N"
DC1,"Q"

4: NO SUBSTITUTE↵
OK

5: SUBSTITUTE↵
```

The current substitution table is deleted.

The user specifies new conversions.

```
SUBSTITUTE:
FILE,TAPE
'6','2'↵
X,Y↵
↵
```

```
6: PRINT↵

SUBSTITUTE:
FILE,TAPE

"6","2"          This is the new substitution table.
"X","Y"

7:
```

If the same substitution table is used to punch a paper tape and then to read that tape back onto a file, or vice versa, the resulting tape or file is the same as the original tape or file except when:

1. A character is ignored.
2. A string of characters is substituted for one character.
3. The same character is substituted for several different characters.
4. A character is converted to a character that is also on the original file or tape. For example, if the file

   ABC

   is punched on paper tape with the substitution table

   "B", "A"

   the resulting paper tape contains the characters:

   AAC

   When this tape is read with the same substitution table, the file created is:

   BBC

A substitution table may be saved on a symbolic file in the user's directory with the SAVE command. The form used is:

p: SAVE file name↵

TAPE responds with OLD FILE or NEW FILE, which is confirmed or aborted as described on page 8.

Example

```
9: SAVE STBL↵
NEW FILE↵
OK

10: QUIT↵
```

```
-TYPE STBL⊃
"A","L"
"1",COLON
"K","N"
DC1,"Q"

-
```

This is the substitution table that was saved.

A saved substitution table may be loaded using the LOAD command. This command has the form:

p: LOAD file name⊃

Example

```
1: LOAD STBL⊃
OK

2: PRINT⊃

SUBSTITUTE:
FILE,TAPE

"A","L"
"1",COLON
"K","N"
DC1,"Q"

3:
```

LOAD appends the specified substitution table to any existing substitution table; it does not delete the existing substitution table. If the user wishes to delete the current substitution table and load a new one, he enters

p: NO SUBSTITUTE⊃

before typing the LOAD command.

Another way to load a saved substitution table is to use it as a command file. Command files may be executed by typing the file name preceded by @.[1] When this is done during SUBSTITUTE, TAPE loads the substitution table and then prints

```
COMMAND FILE EOF:

SUBSTITUTE:
FILE,TAPE
```

1 - See page 67 for a discussion of command files.

thus allowing the user to specify additional conversions either by typing them at the terminal or by specifying another file. To terminate the SUBSTITUTE option at this point, the user types a Carriage Return.

Example

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
@STBL↵                    The substitution table stored on the file STBL is loaded.

COMMAND FILE EOF:

SUBSTITUTE:
FILE,TAPE
↵                         The user terminates the SUBSTITUTE option. At this point, he could also enter more
                          conversions directly from the terminal or load them from another file by typing the
2: PRINT↵                 file name preceded by @.

SUBSTITUTE:
FILE,TAPE

"A","L"
"1",COLON
"K","N"
DC1,"Q"

3:
```

THE REPRESENTATION OPTION

REPRESENTATION is a special-purpose conversion option which allows the user to convert the standard ASCII characters on a symbolic file or a paper tape to their numeric ASCII codes when the file or tape is punched or read. Conversion to decimal or octal ASCII codes may be specified. In addition, REPRESENTATION allows the user to convert a symbolic file or paper tape containing numeric octal or decimal ASCII codes to a standard ASCII symbolic paper tape or file.

The representation of characters on a file or tape as numeric ASCII codes instead of the usual character representation is called *literal representation.* Information in literal representation consists of a list of either octal or decimal codes separated by Carriage Returns. For example:

```
-TYPE LIT↵
101
65
177
0
-
```

The codes may be any numbers in the range 0 through 177 octal or 0 through 127 decimal.

The REPRESENTATION option may be set with the REPRESENTATION command or during the PROMPT command.

The REPRESENTATION command has the form

p: REPRESENTATION {INTERNAL / EXTERNAL / BOTH}↵

where the arguments are as defined in the following table.

SUMMARY OF REPRESENTATION ARGUMENTS

Type of Operation \ Argument	EXTERNAL (Literal representation on the tape)	INTERNAL (Literal representation on the file)	BOTH (Literal representation on the tape and the file)
FILE→TAPE (Paper Tape Punching)	Converts each character on the file to a literal ASCII code and punches the results on paper tape.	Converts each literal ASCII code on the file to a standard ASCII character and punches the results on paper tape. The information being punched must be in literal representation.	Punches each literal ASCII code on the file on paper tape. The information being punched must be in literal representation.
TAPE→FILE (Paper Tape Reading)	Converts each literal ASCII code on the tape to a standard ASCII character and stores the results on a file. The information being read must be in literal representation.	Converts each character code on the tape to its literal ASCII code and stores the results on a file.	Reads each literal ASCII code from the tape and stores it on a file. The information being read must be in literal representation.

To set the REPRESENTATION option during PROMPT, the user types the desired argument in response to the question

LITERAL REPRESENTATION?

as shown in the following example:

```
1: PROMPT
:
LITERAL REPRESENTATION? EXTERNAL↵
:
```

If REPRESENTATION is used during decimal mode, the literal representation is in decimal ASCII codes. To specify octal codes, the user enters the RADIX command, as shown in the following example:

```
1: REPRESENTATION INTERNAL↵
OK

2: READ DEC↵                    This command creates a symbolic file containing a literal decimal
NEW FILE↵                       ASCII code for each character read from the paper tape.

TURN ON READER

36 CHARACTERS ON FILE

3: RADIX OCTAL↵                 The user changes to octal mode.
OK

4: READ OCT↵                    This command creates a symbolic file containing a literal octal
NEW FILE↵                       ASCII code for each character read from the paper tape.

TURN ON READER

47 CHARACTERS ON FILE

5:
```

NOTE: The SUBSTITUTE option may be used with the REPRESENTATION option. When this is done, the substitutions are performed first, and then the conversion to or from literal representation is performed.

The remainder of this section details the various REPRESENTATION options. The examples used in this discussion refer to paper tapes and files containing the following information:

ASCII Character	Octal Code	Decimal Code
A	101	65
B	102	66
C	103	67
,	54	44
1	61	49
2	62	50
3	63	51
↵	15, 12 = ↵	13, 10 = ↵

Internal Representation

The command

p: REPRESENTATION INTERNAL↵

specifies that the file used in reading or punching is to contain information in literal representation.

During paper tape reading with INTERNAL representation, the binary code on each frame of a paper tape is converted to its octal or decimal ASCII code; the converted codes, separated by Carriage Returns, are stored on a symbolic file.

Example

Suppose the paper tape to be read with INTERNAL representation normally would be read as

ABC,123

followed by a Carriage Return. The user may create a file containing the literal decimal ASCII codes for these characters as follows:

```
1: REPRESENTATION INTERNAL↵
OK

2: READ INTDEC↵                    The user reads the paper tape.
NEW FILE↵

TURN ON READER

36 CHARACTERS ON FILE

3: QUIT↵

-TYPE INTDEC↵
65                                 These are the converted decimal codes.
66
67
44
49
50
51
13
10

-
```

Alternatively, the user may enter the RADIX command to specify that the characters are to be converted to octal codes instead of decimal codes. For example:

```
1: RADIX OCTAL↵
OK

2: REPRESENTATION INTERNAL↵
OK

3: READ INTOCT↵                    The user reads the same tape again.
NEW FILE↵

TURN ON READER

47 CHARACTERS ON FILE

4: QUIT↵

-TYPE INTOCT↵                      These are the converted octal codes.
101
102
103
54
61
62
63
15
12
-
```

If INTERNAL representation is used to read a paper tape to a file and then punch the resulting file information on paper tape, the resulting tape is the same as the original tape.

When punching paper tape with INTERNAL representation, each ASCII code on the file is converted to a standard ASCII character and punched on paper tape. The information to be punched must be in literal representation; that is, it must consist of octal or decimal ASCII codes, separated by Carriage Returns. If the information is not in literal representation, an error message is printed and punching is aborted.

NOTE: When the codes on the file are octal, the RADIX command should be used to change to octal mode before punching the tape; otherwise, the codes on the file are interpreted as decimal.

Example

The codes on the files INTDEC and INTOCT created in the preceding examples may be converted to standard ASCII characters and punched on paper tape as follows:

```
1: REPRESENTATION INTERNAL↵
OK

2: PUNCH INTDEC↵          The codes on the file to be punched are decimal.

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

ABC,123                   These are the converted characters.

3: RADIX OCTAL↵           The user requests octal mode, since the codes on the file
OK                        to be punched are octal.

4: PUNCH INTOCT↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

ABC,123                   These are the converted characters.

5:
```

External Representation

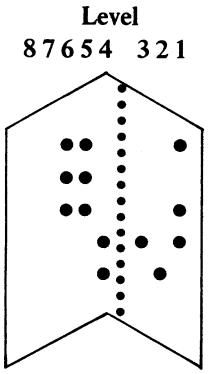
The command

p: REPRESENTATION EXTERNAL↵

specifies that the paper tape to be punched or read is to contain information in literal representation.

Paper tape punching with EXTERNAL representation converts each character on a symbolic file to an octal or decimal ASCII code and punches the converted codes on paper tape. Each converted code is punched on one to three paper tape frames as needed, followed by the Carriage Return/Line Feed combination. The Carriage Return/Line Feed combination is equivalent to a Carriage Return typed at the terminal. For example, the octal ASCII code for A is 101. With EXTERNAL representation, A is punched on paper tape in octal mode as the binary ASCII codes for 1, 0, and 1, followed by the Carriage Return/Line Feed combination, as follows:[1]

1 - See Appendix A for an explanation of interpreting punched tape, and Appendix B for a table of ASCII codes.

Paper Tape Code	Equivalent Octal ASCII Code	Equivalent Character
	61 60 61 15 12	1 0 1 ↵ ↓

= 101↵ = A

In decimal mode, A is punched as the binary codes for 6 and 5 followed by the Carriage Return/Line Feed combination, since the decimal code for A is 65.

Example

Suppose the symbolic file SYM contains the characters

ABC,123

followed by a Carriage Return. Using EXTERNAL representation, the literal decimal and octal ASCII codes for these characters can be punched on paper tape as follows:

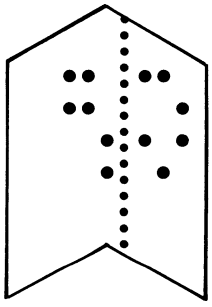
```
1: REPRESENTATION EXTERNAL↵
OK

2: PUNCH SYM↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

65
66
67
44
49
50
51
13
10
```

The decimal ASCII codes are punched on paper tape.

```
3: RADIX OCTAL↵          The user changes to octal mode.
OK

4: PUNCH SYM↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

101                      The octal ASCII codes are punched.
102
103
54
61
62
63
15
12

5:
```

If EXTERNAL representation is used to punch a symbolic file on paper tape and then to read the resulting paper tape back onto a file, the resulting symbolic file is the same as the original file punched.

When reading a paper tape with EXTERNAL representation, the binary codes on each set of one to three paper tape frames, terminated by the Carriage Return/Line Feed combination, are converted to a character, and the resulting characters are stored on a symbolic file. For example, in octal mode, the following codes are converted to the character whose octal code is 61, namely 1.

Paper Tape Code	Equivalent Octal ASCII Code	Equivalent Character
Level 8 7 6 5 4 3 2 1 	66 61 15 12	6 1 ↵ ⤓ } = 61↵ = 1

The tape being read must be in literal representation; that is, it must contain octal or decimal ASCII codes, separated by the Carriage Return/Line Feed combination. If the paper tape is not in literal representation, an error message is printed.

Example 1

This example illustrates using EXTERNAL representation to read a paper tape containing decimal ASCII codes, converting the literal codes to their character equivalents. The ECHO option is set to show the original codes on the paper tape.

```
1: ECHO ON↵
OK

2: REPRESENTATION EXTERNAL↵
OK

3: READ SYM1↵
NEW FILE↵

TURN ON READER
65                  These are the codes on the paper tape.
66
67
44
49
50
51
13
10

9 CHARACTERS ON FILE

4: QUIT↵

-TYPE SYM1↵
ABC,123             These are the converted characters.

-
```

Example 2

In this example, the RADIX command is used to change to octal mode, because the codes on the paper tape are octal. The ECHO option is set to show the original codes on the tape.

```
1: ECHO ON↵
OK

2: RADIX OCTAL↵        The user requests octal mode.
OK

3: REPRESENTATION EXTERNAL↵
OK

4: READ SYM2↵
NEW FILE↵

TURN ON READER
101                    These are the codes on the paper tape.
102
103
54
61
62
63
15
12

11 CHARACTERS ON FILE

5: QUIT↵

-TYPE SYM2↵
ABC,123                These are the converted characters.
-
```

Internal and External Representation

The command

p: REPRESENTATION BOTH↵

specifies that the information on both the paper tape and the file is to be in literal representation. No conversions are performed when BOTH is specified; REPRESENTATION BOTH is equivalent to REPRESENTATION OFF, except that the file information being punched or the tape information being read must be in literal representation. For example, suppose the user wishes to punch on paper tape literal octal ASCII codes from the following file:

```
-TYPE CODES↵
101
107
130
144
-
```

The first time he punches the codes, he forgets to specify octal mode. TAPE interprets the codes as decimal; thus, the codes 130 and 144 are interpreted as being greater than the maximum allowable decimal value of 127. TAPE prints the octal values of the codes in error and punches a 127 on the paper tape for each code in error. The user then changes to octal mode and punches the tape again. This time, the codes are punched correctly.

```
1: REPRESENTATION BOTH↵
OK

2: PUNCH CODES↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

101
107
NUMBER MUST BE LESS THAN 177B - MAXIMUM VALUE ASSUMED: 202B
127
NUMBER MUST BE LESS THAN 177B - MAXIMUM VALUE ASSUMED: 220B
127

3: RADIX OCTAL↵
OK

4: PUNCH CODES↵

TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵

101
107
130
144

5:
```

SECTION 6

COMMAND FILES

The user may instruct TAPE to take commands from a file rather than from the terminal; such a file is called a command file. Command files can be especially useful when a particular sequence of commands is to be executed repeatedly, since the user can store the commands on a file and then execute them as needed by typing one simple TAPE command.

CREATING A COMMAND FILE

The user may create a command file in EDITOR or in TYMEX.[1] He enters each command as if he were entering it directly in TAPE; however, there is one exception: When a command that causes information to be stored on a file, such as the READ command, is included, the user should not include the Carriage Return response to the NEW FILE or OLD FILE message. TAPE suppresses this message during command file execution and opens the specified file without confirmation. For example, a command file named CREAD, which reads a symbolic paper tape, checking for even parity and using the EDITING option, is created in TYMEX as follows:

```
-COPY T TO CREAD↵
PARITY EVEN↵
EDITING CHARACTER=↑ LINE=←↵
READ TDATA↵
D^C
-
```

No extra Carriage Return is included for the NEW FILE or OLD FILE message, since it is suppressed during command file execution.

When any command for paper tape reading or punching is included in a command file, the output file name or the input file name must also be included; it cannot be entered from the terminal.

OPENING A COMMAND FILE

The user may instruct TAPE to take commands from a command file by opening the command file either at TAPE command level or within any TAPE command.

To open a command file at TAPE command level, the user may enter either:

p: **@command file name↵**

or

p: **DO command file name↵**

For example:

1: **@CREAD↵**

or

1: **DO CREAD↵**

1 - See the *Tymshare TYMCOM-X TYMEX Reference Manual* for instructions on creating files in TYMEX or EDITOR.

To open a command file within a command, the user must type the character @ followed by the command file name. For example:

```
1: TITLE @T2↵
```

The DO command may not be used to open a command file within a command.

When a command file is opened, TAPE begins taking commands from the file immediately and executes the commands sequentially until it reaches the end of the command file. TAPE then closes the command file and returns to taking commands from the terminal.

Example 1

The commands in the command file CREAD, created above, may be executed as follows:

```
1: @CREAD↵          The user may alternatively type DO CREAD followed by a Carriage Return.
TURN ON READER      The user turns on the paper tape reader.
20 CHARACTERS ON FILE
COMMAND FILE EOF:   TAPE prints this message to indicate the end of the command file.
4:
```

The command level prompt number is incremented by the number of commands in the command file if @ was used to open the command file, or by the number of commands plus 1 if the DO command was used.

Example 2

This example illustrates opening a command file within a command. The command file PSPECS contains responses to all the option prompts in the PROMPT command.

```
-TYPE PSPECS↵
SYMBOLIC
EVEN
FILE1
T
NO
                    This line contains a Carriage Return which terminates the SUBSTITUTE prompt.
'TEST DATA'
10
-
```

The command file is opened within the PROMPT command as follows:

```
1: PROMPT↵

SYMBOLIC OR BINARY? @PSPECS↵        The user opens the command file PSPECS and TAPE
                                    takes the responses to the option prompts from it.
TYPE CARRIAGE RETURN OR LINE FEED.
THEN TURN ON PUNCH.
↵                                   The user types a Carriage Return and turns on the punch.

IIII&III2AAA>ABCDEFG

COMMAND FILE EOF:

2:
```

Example 3

```
1: SUBSTITUTE↵

SUBSTITUTE:
FILE,TAPE
@TBL3↵                              A substitution table is loaded from the file TBL3.[1]
:
:
:
```

MESSAGES IN COMMAND FILES

During command file execution, most messages normally printed at the terminal, such as INPUT FROM: and OK, are suppressed. The following messages, however, are printed at the terminal during command file execution.

Message	Comments
TURN ON READER	The user must turn on the reader. After the tape is read, TAPE continues to take commands from the command file.
n {CHARACTERS / WORDS} ON FILE	This message is printed after a tape is read.

1 - See page 54 for a complete description of this technique.

Message	Comments
TYPE CARRIAGE RETURN OR LINE FEED. THEN TURN ON PUNCH.	The Carriage Return or Line Feed required before turning on the punch is always typed at the terminal. It cannot be taken from the command file.
COMMAND FILE EOF:	Indicates the end of the command file.

The user may also print his own messages during command file execution by using the following form of the TYPE command in the command file:

TYPE 'message to be printed'↵

The message to be printed must be surrounded by single or double quotation marks. When the TYPE command is encountered, the message is printed at the terminal. For example, if the command

TYPE 'PAPER TAPE TO BE READ TO FILE TDATA'

were included at the beginning of the command file CREAD, created on page 67, execution would appear as follows:

```
1: @CREAD↵
PAPER TAPE TO BE READ TO FILE TDATA

TURN ON READER

20 CHARACTERS ON FILE

COMMAND FILE EOF:

5:
```

NESTED COMMAND FILES

A command file may contain a command that opens another command file; a command file opened in this manner is called a nested command file. When nesting occurs, control is transferred immediately to the nested command file. When all commands in this file have been executed, control is returned to the previous command file, and the remaining commands are executed. For example, consider the two command files CM1 and CM2:

```
-TYPE CM1↵
TYPE 'PAPER TAPE TO BE READ TO FILEA'
READ FILEA
@CM2
TYPE 'JOB COMPLETED'
```

```
-TYPE CM2↵
TYPE 'PAPER TAPE TO BE READ TO FILEB'
TYPE 'PARITY MUST BE EVEN'
PARITY EVEN
READ FILEB

-
```

When the user opens the command file CM1, TAPE takes commands from this file until it reaches the command @CM2. At that point, TAPE immediately opens the command file CM2 and takes commands from it until it reaches the end of this command file. Then TAPE closes CM2 and returns to taking commands from CM1, starting with the command:

TYPE 'JOB COMPLETED'

This procedure is shown below.

```
1: @CM1↵                                  The user opens CM1.
PAPER TAPE TO BE READ TO FILEA            TAPE begins taking commands from CM1.

TURN ON READER

275 CHARACTERS ON FILE
PAPER TAPE TO BE READ TO FILEB            Now, TAPE is taking commands from CM2.
PARITY MUST BE EVEN

TURN ON READER

194 CHARACTERS ON FILE
JOB COMPLETED                             TAPE reaches the end of CM2 and returns to taking commands from CM1.

COMMAND FILE EOF:                         TAPE reaches the end of CM1 and returns to taking commands from the terminal.

8:
```

Both the files CM1 and CM2 could contain additional commands to open other command files; however, the user may not open more than 10 command files at once.

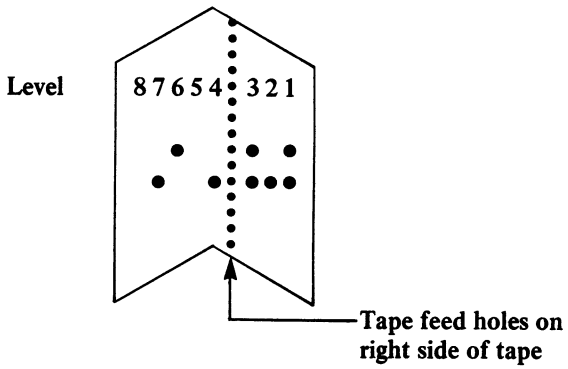
APPENDIX A

HOW TO INTERPRET PUNCHED TAPE

A punched paper tape contains codes punched in rows, called frames, across the paper tape. There are eight positions in each paper tape frame which may or may not be punched; each of these positions is called a level, a channel, or a bit.

A code that is punched on a paper tape is a binary number. On a symbolic paper tape, this number represents an ASCII character. To determine the character represented, the binary number is converted to an octal number. This octal number is the octal ASCII code for the character punched on the tape.

To convert a paper tape code to an octal ASCII code, the user should position the tape as shown below.



A punch represents a 1; no punch represents a 0. The first number punched above is 00100101.

A binary number is converted to an octal number by grouping in threes the digits from right to left, and converting each group to one octal digit. Each octal digit is the sum of the values of the numbers punched in each group. For example:

```
00    100   101    Binary number
      |     | |
21    421   421    Value if punched
       \     V
0      4     5     Octal digits
```

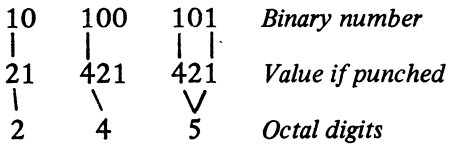
The code punched is 45, the octal ASCII code for the character %.

The second number punched on the tape above is converted as follows:

```
01    001   111    Binary number
 |      |   |||
21    421   421    Value if punched
 /     /     V
1     1     7      Octal digits
```

The code punched is 117, the octal ASCII code for the letter O.

TAPE punches the eighth level of the paper tape only when the user specifically requests this with the PARITY option; however, paper tape punched off line always has the eighth level punched. When interpreting a tape code with the eighth level punched, the octal ASCII code is found by subtracting 200 from the octal code punched on the tape. For example, as shown on the previous page, the code punched for % is 45; off line, a % is punched as 245:



APPENDIX B

TABLE OF CODE CONVERSIONS

Character[1]	Standard ASCII Character Name	Octal ASCII Code	Decimal ASCII Code	Binary[2]	FLEX, N/C, EIA
P^{cs}	NULL	0	0	0000.000	
A^{c}	SOH	1	1	0000.001	1
B^{c}	STX	2	2	0000.010	2
C^{c}	ETX	3	3	0000.011	
D^{c}	EOT	4	4	0000.100	4
E^{c}	ENQ	5	5	0000.101	
F^{c}	ACK	6	6	0000.110	
G^{c}	BELL	7	7	0000.111	7
H^{c}	BS	10	8	0001.000	8
I^{c}	HT	11	9	0001.001	
J^{c}	LF	12	10	0001.010	
K^{c}	VT	13	11	0001.011	REW STOP
L^{c}	FF	14	12	0001.100	
M^{c}	CR	15	13	0001.101	
N^{c}	SO	16	14	0001.110	
O^{c}	SI	17	15	0001.111	
P^{c}	DLE	20	16	0010.000	SP
Q^{c}	DC1	21	17	0010.001	
R^{c}	DC2	22	18	0010.010	
S^{c}	DC3	23	19	0010.011	3
T^{c}	DC4	24	20	0010.100	
U^{c}	NAK	25	21	0010.101	5
V^{c}	SYN	26	22	0010.110	6
W^{c}	ETB	27	23	0010.111	
X^{c}	CAN	30	24	0011.000	
Y^{c}	EM	31	25	0011.001	9
Z^{c}	SUB	32	26	0011.010	
K^{cs}	ESC	33	27	0011.011	
L^{cs}	FS	34	28	0011.100	
M^{cs}	GS	35	29	0011.101	

1 - Superscript s denotes a shift character.

2 - This is the code actually punched on the tape. The decimal point represents the tape feed hole on the paper tape. The number 1 represents a punched hole.

Character	Standard ASCII Character Name	Octal ASCII Code	Decimal ASCII Code	Binary	FLEX, N/C, EIA
N^{cs}	RS	36	30	0011.110	
O^{cs}	US	37	31	0011.111	
blank	SPACE	40	32	0100.000	0
!	EXCLAMATION	41	33	0100.001	
"	QUOTE	42	34	0100.010	
#	POUND	43	35	0100.011	t
$	DOLLAR	44	36	0100.100	
%	PERCENT	45	37	0100.101	v
&	AMPERSAND	46	38	0100.110	w
'	APOSTROPHE	47	39	0100.111	
(	LFPAREN	50	40	0101.000	
)	RTPAREN	51	41	0101.001	z
*	ASTERISK	52	42	0101.010	
+	PLUS	53	43	0101.011	
,	COMMA	54	44	0101.100	
-	HYPHEN	55	45	0101.101	
.	PERIOD	56	46	0101.110	
/	SLASH	57	47	0101.111	
0	ZERO	60	48	0110.000	
1	ONE	61	49	0110.001	/
2	TWO	62	50	0110.010	s
3	THREE	63	51	0110.011	
4	FOUR	64	52	0110.100	u
5	FIVE	65	53	0110.101	
6	SIX	66	54	0110.110	
7	SEVEN	67	55	0110.111	x
8	EIGHT	70	56	0111.000	y
9	NINE	71	57	0111.001	
:	COLON	72	58	0111.010	
;	SEMICOLON	73	59	0111.011	,
<	LESSTHAN	74	60	0111.100	
=	EQUALS	75	61	0111.101	
>	GREATERTHAN	76	62	0111.110	TAB
?	QUESTIONMARK	77	63	0111.111	
@	ATSIGN	100	64	1000.000	-

Character	Standard ASCII Character Name	Octal ASCII Code	Decimal ASCII Code	Binary	FLEX, N/C, EIA
A	A	101	65	1000.001	
B	B	102	66	1000.010	
C	C	103	67	1000.011	l
D	D	104	68	1000.100	
E	E	105	69	1000.101	n
F	F	106	70	1000.110	o
G	G	107	71	1000.111	
H	H	110	72	1001.000	
I	I	111	73	1001.001	r
J	J	112	74	1001.010	
K	K	113	75	1001.011	
L	L	114	76	1001.100	
M	M	115	77	1001.101	
N	N	116	78	1001.110	
O	O	117	79	1001.111	
P	P	120	80	1010.000	
Q	Q	121	81	1010.001	j
R	R	122	82	1010.010	k
S	S	123	83	1010.011	
T	T	124	84	1010.100	m
U	U	125	85	1010.101	
V	V	126	86	1010.110	
W	W	127	87	1010.111	p
X	X	130	88	1011.000	q
Y	Y	131	89	1011.001	
Z	Z	132	90	1011.010	
[	LFBRACKET	133	91	1011.011	
\	BACKSLASH	134	92	1011.100	
]	RTBRACKET	135	93	1011.101	
↑	UPARROW	136	94	1011.110	
←	BACKARROW	137	95	1011.111	
`	GRAVE	140	96	1100.000	

Character	Standard ASCII Character Name	Octal ASCII Code	Decimal ASCII Code	Binary	FLEX, N/C, EIA
a	a	141	97	1100.001	a
b	b	142	98	1100.010	b
c	c	143	99	1100.011	
d	d	144	100	1100.100	d
e	e	145	101	1100.101	
f	f	146	102	1100.110	
g	g	147	103	1100.111	g
h	h	150	104	1101.000	h
i	i	151	105	1101.001	
j	j	152	106	1101.010	
k	k	153	107	1101.011	
l	l	154	108	1101.100	
m	m	155	109	1101.101	
n	n	156	110	1101.110	
o	o	157	111	1101.111	
p	p	160	112	1110.000	+
q	q	161	113	1110.001	
r	r	162	114	1110.010	
s	s	163	115	1110.011	c
t	t	164	116	1110.100	
u	u	165	117	1110.101	e
v	v	166	118	1110.110	f
w	w	167	119	1110.111	
x	x	170	120	1111.000	
y	y	171	121	1111.001	i
z	z	172	122	1111.010	
{	LFBRACE	173	123	1111.011	
|	VERTBAR	174	124	1111.100	
}	RTBRACE	175	125	1111.101	
~	TILDE	176	126	1111.110	
RUBOUT	DEL	177	127	1111.111	DEL
		200	128	10000.000	EOB

APPENDIX C

TAPE COMMAND SUMMARY

The following symbols are used in this summary:

p: Indicates the command level prompt.

c Represents the character-deleting character defined with the EDITING option; it can be any of the following:

- A character enclosed in quotation marks, such as 'A'.
- An ASCII code for a character, such as 101.
- A standard ASCII character name for a character, such as BELL or DC1.

l Represents the line-deleting character defined with the EDITING option; it can assume the same forms as c, above.

c_f Represents a character on a file specified during SUBSTITUTE; it can assume the same forms as c, above, and also that of a string of characters enclosed in quotation marks, such as 'XYZ'.

c_t Represents a character on paper tape specified during SUBSTITUTE; it can assume the same forms as c_f, above.

NOTE: Wherever a pair of single quotation marks is shown, a pair of double quotation marks can be used instead.

PAPER TAPE READING AND PUNCHING

Command	Form	Description
READ	p: READ file name↵	Reads tape in symbolic form.
BREAD	p: BREAD file name↵	Reads tape in binary form.
PUNCH	p: PUNCH file name↵	Punches tape in symbolic form.
BPUNCH	p: BPUNCH file name↵	Punches tape in binary form.
PROMPT	p: PROMPT↵	Initiates conversational reading or punching in either symbolic or binary form.
RUN	p: RUN↵	Reads or punches tape in symbolic form only. Prompts user for information required.

SUMMARY OF PROMPT COMMAND RESPONSES

Each response must be terminated by a Carriage Return.

Prompt	Valid User Response(s)	Remarks
SYMBOLIC OR BINARY?	SYMBOLIC BINARY	Only a Carriage Return means SYMBOLIC.
PARITY:	EVEN ODD DATA PACK OFF NO	PACK applies only to binary reading or punching.
INPUT FROM:	T file name	T or only a Carriage Return specifies reading; a file name specifies punching.
OUTPUT TO:	file name T	T or only a Carriage Return specifies punching; a file name specifies reading.
LITERAL REPRESENTATION?	INTERNAL EXTERNAL BOTH OFF NO	Prompted only during symbolic reading or punching.
EDITING:	CHARACTER=c LINE=l OFF NO	Prompted only during symbolic reading.
SUBSTITUTE: FILE,TAPE	c_f, c_t c_f, c_t :	Prompted only during symbolic reading or punching. Only a Carriage Return terminates the prompt.
ECHO:	ON OFF NO	Prompted only during reading.
TITLE:	'title' OFF NO	Prompted only during punching.
LENGTH OF LEADER AND TRAILER:	number of inches	Prompted only during punching.
ALWAYS PUNCH EIGHTH LEVEL?	YES NO	Prompted only during binary punching with the PARITY option set to its default value (OFF or NO).

OPTION COMMANDS

PARITY, SUBSTITUTE, and REPRESENTATION are not included in this table, but are given in separate tables below.

Command	Form(s)	Description
NO	p: NO option name↵	Resets specified option to its default value. May be used with EDITING, ECHO, TITLE, LENGTH, PARITY, SUBSTITUTE, REPRESENTATION, HUSH, and TALK.
RESET	p: RESET↵	Resets all options to their default values; also resets HUSH and RADIX.
LIST	p: LIST↵	Lists current values of all options.
ECHO	p: ECHO ON↵	Causes information on tape being read to be printed at terminal.
	p: ECHO {NO / OFF}↵	Terminates effect of ECHO ON.
EDITING	p: EDITING CHARACTER=c↵ p: EDITING LINE=l↵ p: EDITING CHARACTER=c LINE=l↵	Defines the character-deleting character and/or the line-deleting character to edit the incoming data during symbolic paper tape reading.
	p: EDITING {NO / OFF}↵	Resets EDITING to no editing characters.
TITLE	p: TITLE 'title'↵	Defines the title to be punched at the beginning of a paper tape.
	p: TITLE PUNCH 'title'↵	Punches the specified title immediately without changing any previously defined title.
	p: TITLE {NO / OFF}↵	Resets the TITLE option to no title.
LENGTH	p: LENGTH number of inches↵	Defines the lengths of leader and trailer to be punched on a tape. The default value is 5 inches.

PARITY

The PARITY command is used both to request parity checking and to define the format to be used for binary reading and punching.

Form	Description	
	Symbolic Reading and Punching	Binary Reading and Punching
p: PARITY NO↵	The parity bit (eighth level of paper tape) is ignored.	Information is read or punched in 6-bit bytes. Bits 7 and 8 of the paper tape are ignored.
p: PARITY EVEN↵	The parity bit is punched to make the total number of bits punched on a paper tape frame even. During reading, each frame of paper tape is checked for even parity.	Same as above, except parity bit is used for even parity checking.
p: PARITY ODD↵	The parity bit is punched to make the total number of bits punched on a paper tape frame odd. During reading, each frame of paper tape is checked for odd parity.	Same as above, except parity bit is used for odd parity checking.
p: PARITY DATA↵	The parity bit is always punched. During reading, TAPE checks that the parity bit is always punched.	Information is read or punched in 8-bit bytes right-justified in a machine word. No parity checking is performed.
p: PARITY PACK↵	Not used.	Information is read or punched in 8-bit bytes, overlapping word boundaries. No parity checking is performed.

CODE CONVERSION

To specify that octal ASCII codes are to be used with SUBSTITUTE and REPRESENTATION, the user enters the command

p: RADIX OCTAL↵

before typing the appropriate SUBSTITUTE or REPRESENTATION option. Otherwise, numeric codes are interpreted as decimal.

Command	Form(s)	Description
SUBSTITUTE	p: **SUBSTITUTE** ↵ **SUBSTITUTE:** **FILE,TAPE** c_f,c_t ↵ c_f,c_t ↵ ⋮ ↵	Defines conversions to be made during reading or punching. For each pair c_f,c_t, c_f is substituted for c_t during reading, but c_t is substituted for c_f during punching. The word FOR may be used in place of the comma, in which case c_f or c_t may consist of a list of character specifications, separated by commas, as in CR, LF FOR 40.
	p: **SUBSTITUTE** c_f {, / FOR} c_t ↵	Short form of SUBSTITUTE for specifying only one conversion.
PRINT	p: **PRINT** ↵	Prints current substitution table at terminal.
SAVE	p: **SAVE file name** ↵	Saves current substitution table on specified file.
LOAD	p: **LOAD file name** ↵	Loads substitution table from specified file without deleting any currently defined substitutions. The substitutions loaded are appended to the existing ones.
REPRESENTATION	p: **REPRESENTATION INTERNAL** ↵	**During reading:** Converts each character code on the tape to its octal or decimal ASCII code and stores the converted codes, separated by Carriage Returns, on a symbolic file. **During punching:** Converts a symbolic file containing octal or decimal ASCII codes separated by Carriage Returns to a standard symbolic paper tape.
	p: **REPRESENTATION EXTERNAL** ↵	**During reading:** Converts a paper tape containing octal or decimal ASCII codes separated by the Carriage Return/Line Feed combination to a standard symbolic file. **During punching:** Converts each character on a symbolic file to its octal or decimal ASCII code and punches each converted code on one to three paper tape frames, as required, followed by the Carriage Return/Line Feed combination.
	p: **REPRESENTATION BOTH** ↵	Specifies that both the paper tape and the file used during reading or punching contain octal or decimal ASCII codes separated by Carriage Returns.
	p: **REPRESENTATION** {NO / OFF} ↵	Resets REPRESENTATION option to none.

INTERRUPTION RECOVERY

Command	Form	Description
SPLICE	p: SPLICE job number↵ where the job number is the one printed by TAPE during the session in which the failure occurred.	Allows recovery when symbolic paper tape reading is interrupted by a system failure. When a system failure occurs, TAPE creates a file containing information necessary to do the SPLICE. The file is named BxxTAP.TMP, where xx is the user's job number.
SSPLICEREAD	p: SSPLICEREAD file name↵	Allows recovery when symbolic paper tape reading is interrupted by any emergency in which a BxxTAP.TMP file is not created. Before attempting the recovery, the user should set all TAPE options to the same values as when the interruption occurred.

COMMAND FILES

Form	Description
@command file name↵	Opens the specified command file. May be used either at command level or within a command.
p: DO command file name↵	Opens the specified command file at command level only.
TYPE 'message'↵	May be included in a command file to print the specified message at the terminal during command file execution.

UTILITY COMMANDS

Command	Form(s)	Description
RADIX	p: RADIX { OCTAL / DECIMAL }↵	Sets radix to either octal (base 8) or decimal (base 10). Default value is decimal.
TYPE	p: TYPE c↵	Types ASCII code for character(s) specified by c, where c is either a standard ASCII character name or a list of character names separated by commas.
	p: TYPE n+↵	Types name of character whose ASCII code is n.
	p: TYPE 'message'↵	Types specified message at terminal.
HELP	p: HELP↵	Prints a list of TAPE commands, with descriptions. May be used within a command to obtain descriptions of the arguments available at that point in the command.
?	p: ?↵	Prints a list of TAPE commands, without descriptions. May be used within a command to obtain just a list of arguments available at that point in the command.
HUSH	p: HUSH↵	Abbreviates or suppresses the usual messages printed by TAPE, including the prompt number and colon (:).
TALK	TALK↵	Terminates effect of HUSH.

The usual standard utility commands are also available; these are shown in the table on page 18.

APPENDIX D

ERROR MESSAGES

This appendix contains a list of TAPE error messages. Each message is followed by an explanation of what may possibly cause the error condition. When the explanation is system error, the user probably did not cause the error and should reenter the information, making sure it is correct. If the user encounters a system error again, he should contact his Tymshare representative. For some errors, TAPE prints more than one message to give the user a more complete description of the error.

The messages listed first begin with a particular command abbreviation, character string, or command argument. The remaining error messages are arranged in alphabetical order.

"command abbreviation" AMBIGUOUS
CANNOT DISTINGUISH AMONG: list of commands

The user has typed too few characters in a command abbreviation.

"character" (octal code) IS AN UNRECOGNIZABLE DELIMITER OR TERMINATOR

The user has typed an illegal nonalphanumeric character to separate command parts or to terminate a command. For example, typing a space between the word CHAR and the equals sign in the invalid command EDITING CHAR =← produces this message.

"argument" NOT LEGAL HERE

The user has specified an invalid argument in the given command.

"characters" NOT RECOGNIZED

The user has entered an invalid command word, an invalid response during PROMPT, or an invalid character specification during SUBSTITUTE. The characters in quotation marks are the invalid entry.

ANSWER YES OR NO:

The user has typed something other than YES or NO, or an abbreviation of these, in response to a question requiring a YES or NO answer.

BINARY PUNCH CANNOT HAVE LITERAL REPRESENTATION

The user has set the REPRESENTATION option before attempting a binary punch. This is illegal.

BINARY READ CANNOT HAVE LITERAL REPRESENTATION.

The user has set the REPRESENTATION option before attempting a binary read. This is illegal.

BOTH SOURCES CANNOT BE SAME

The user has entered the same response to the prompts INPUT FROM: and OUTPUT TO: during the PROMPT or RUN command. For example, he may have specified T in response to both prompts.

CANNOT OPEN command file name – FILE NOT AVAILABLE

The user has instructed TAPE to open a command file illegally. Possibly the file is not in the user's directory, the file security controls do not permit its use, an illegal file name has been typed, etc.

CANNOT OPEN file name – FILE BUSY

The specified file is not available, since it is in use elsewhere for output.

CANNOT OPEN file name – FILE NOT FOUND

The specified file is not in the user's directory.

CANNOT OPEN file name – ILLEGAL FILE NAME

The user has specified an illegal file name, such as ?.

CANNOT OPEN file name – ILLEGAL OPERATION ON THE TERMINAL

The terminal has been specified as a file name illegally, as in the invalid command BPUNCH T.

CANNOT OPEN file name – INVALID USER NAME OR PPN

The user has specified an invalid user name or file directory number when attempting to use a file in another user's directory.

CANNOT OPEN file name – NEW/OLD FILE MESSAGE NOT CONFIRMED

The user has aborted the NEW/OLD FILE message with an Alt Mode/Escape.

CANNOT OPEN file name – PROTECTION FAILURE

The file security controls on the file do not permit using it as requested.

CANNOT OPEN file name – TOO MANY FILES OPEN

Ten files can be open at once. The user has exceeded this limit.

CANNOT OPEN SYS:TAPE.HLP – SYSTEM ERROR, CANNOT COMPLETE COMMAND

System error.

CANNOT RECOVER CRASH PROTECTION FILE FOR THAT JOB ID

There is no interruption recovery file BxxTAP.TMP, where xx is the user's job number, in the user's directory. Try recovery with SSPLICEREAD instead of SPLICE.

CANNOT RECOVER FROM BINARY READ

The user has attempted a recovery after an emergency interruption occurred during a binary read. This is illegal.

CANNOT RECOVER FROM PUNCH

The user has attempted a recovery after an emergency interruption occurred during paper tape punching. This is illegal.

COMMAND FILE TERMINATED AT FOLLOWING POINT
IN LINE line number COMMAND FILE # file number
line in which error occurred
explanatory message

An illegal command has been included in a command file. TAPE prints the line number, the command file number, the line in error, and a message indicating the nature of the error.

COMMAND REQUIRED

At command level the user has typed something other than a command, such as a quoted string or a nonalphanumeric character.

CONFLICT WITH LEFT PART. "character" ALREADY SUBSTITUTES TO: "character"
CONFLICT WITH RIGHT PART. "character" ALREADY SUBSTITUTES FROM: "character"

One or both of these messages is printed when the user enters conflicting conversions during SUBSTITUTE. The message is simply a warning to the user; the specified conversions are still made where possible.

ERROR DURING PUNCH IN "characters"
PUNCHING ABORTED

Paper tape punching has been terminated because there are illegal characters on the input file, such as when the user attempts to punch a file containing nonnumeric characters with REPRESENTATION INTERNAL.

ERROR DURING READ IN "characters"

The tape being read contains illegal characters, such as when the user attempts to read nonnumeric characters with REPRESENTATION EXTERNAL.

ILLEGAL CHARACTER "number" FOR INPUT RADIX

The user has entered a number which is not valid for the current base, for example, entering a number that contains the digit 8 when the base is octal.

ILLEGAL WORD IN INPUT

A word has been entered in a SUBSTITUTE command which makes no sense to TAPE.

IMPROPER TERMINATION: string

The user has not used quotation marks properly in terminating a quoted string. He may have forgotten the final quotation mark, or begun the string with a single quotation mark and ended it with a double quotation mark.

INVALID CHARACTER "character" IN NUMBER

The user has entered a nonnumeric character as part of a number, such as 5Y.

NO SUBSTITUTIONS IN EFFECT

The user has typed the PRINT or SAVE command when no substitutions have been defined.

NULL STRING IS ILLEGAL

The user has specified a null string, denoted by NOTHING, '', or "", where it is not allowed.

NUMBER MUST BE LESS THAN 177B–MAXIMUM VALUE ASSUMED: n

The user has entered a numeric ASCII code n which is greater than the maximum 177 octal allowed. This can occur when ASCII codes are specified during EDITING, SUBSTITUTE, and REPRESENTATION.

NUMERIC ARGUMENT ILLEGAL HERE

The user has specified a number as a command argument where this is not allowed, for example, in the invalid command TITLE 5.

ONLY n CHARACTERS WERE READ INTO THE OUTPUT FILE
IT WOULD BE FASTER TO RE–READ THE TAPE

The user has attempted a recovery where n, the number of characters read, is so small that the recovery would be inefficient.

RADIX MUST BE BETWEEN 2 AND 16, INCLUSIVE

The user has specified an invalid radix.

STRING ARGUMENT ILLEGAL HERE

The user has specified a quoted string as a command argument where this is not allowed, for example, in the invalid command REPRESENTATION 'EXTERNAL'.

STRING ERROR:
string
STRING ARGUMENT CANNOT END WITH "character"

The user has terminated a string with some character other than a single or double quotation mark, for example, 'TEST'3.

SUBSTITUTION STORAGE EXCEEDED

The user has specified more than 200 code conversions.

SYSTEM ERROR: ATTEMPT TO OPEN TOO MANY COMMAND FILES
THE MAX # OF ALLOWABLE COMMAND FILES IS CURRENTLY 10

As many as 10 command files can be open at once. The user has exceeded this number.

SYSTEM ERROR: CXXTAP.TMP TEMP FILE IS BAD ON DEVICE DSK:
READING ABORTED

System error.

SYSTEM ERROR ON FILE:

System error.

SYSTEM ERROR: REQUIRED HELP MESSAGE NOT IN SYS:TAPE.HLP

System error.

THE INPUT IS COMPLETE: NO FURTHER ARGUMENTS AVAILABLE

The user has typed HELP or ? after a complete command, as in PARITY DATA HELP.

THERE WERE n PARITY ERROR(S) WHILE READING YOUR TAPE
PLEASE READ THE FILE AxxTAP.TMP FOR DETAILS

TAPE has found n parity error(s) while reading a tape. Any parity errors are described in the file AxxTAP.TMP, where xx is the user's job number.

YOU DIDN'T READ IN ENOUGH OF THE TAPE

Printed during SPLICE or SSPLICEREAD. The user has not read enough tape when requested to read in about 10 inches of tape. He may try again.

YOU READ IN MORE THAN 20 INCHES OF TAPE

Printed during SPLICE or SSPLICEREAD. The user has read in too much tape when requested to read in about 10 inches of tape. He may try again.

INDEX

NOTE: Page numbers which appear in **bold** *face type refer to those pages where the listed item receives the most detailed discussion.*

USER EVALUATION

Tymshare would like to improve the quality and usefulness of its publications. However, to achieve this goal, we need your help and critical evaluations. Will you please provide us with such constructive information by filling out this questionnaire and mailing it back to us?

1. (a) Is this manual a useful document? ☐ Yes ☐ No
   (b) If your answer is Yes, what features make it useful.

   (c) If your answer is No, what features prevent it from being a useful document.

2. (a) Is the text clear and readily understandable? ☐ Yes ☐ No
   (b) If your answer is No, please cite the sections, subsections, or paragraphs that are unclear or difficult to understand.

3. (a) Are you pleased with the organization of this manual? ☐ Yes ☐ No
   (b) Should the organization be changed? ☐ Yes ☐ No
   (c) What changes do you suggest?

4. (a) Are the example problems helpful and easy to understand? ☐ Yes ☐ No
   (b) Should more examples be added when this manual is revised? ☐ Yes ☐ No
   (c) What kind of programs would you like to see added?

5. (a) Should anything be deleted from this manual when it is revised? ☐ Yes ☐ No
   (b) If your answer is Yes, give us your suggestions.

6. List any further suggestions you have for the improvement of this manual.

OPTIONAL INFORMATION:

Name _____ Company _____

Street Address _____

City _____ State _____ Zip _____

Occupation _____

THANK YOU FOR YOUR ASSISTANCE.

To mail this questionnaire, simply cut from manual, fold Part A back, fold Part B back, and staple where indicated. **Part A**

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

First Class
Permit No.
300
Los Altos,
California

POSTAGE WILL BE PAID BY

TYMSHARE, INC.
10340 BUBB ROAD
CUPERTINO, CALIFORNIA 95014
ATTN: Documentation Group

Part B

STAPLE HERE