GENERAL UTILITY PROGRAMS

PART 12

THE DOS/BATCH FILE UTILITY PACKAGE

PIP

PART 12

CHAPTER 1

INTRODUCTION TO THE PERIPHERAL INTERCHANGE PROGRAM (PIP)

The Peripheral Interchange Program (PIP) performs transfers of data files from standard PDP-11 I/O devices to other standard PDP-11 I/O devices and performs simple editing and control functions as well. PIP handles all standard data formats used in the DOS/BATCH system.

The major features provided by PIP follow.

- Transfer a file or group of files from one device to another in either linked or contiguous format (refer to Section 12-3.1.4).
- Merge a group of input files into a single (new) output file.
- Allocate a contiguous file on disk or DECtape.
- Rename an existing file.
- Cause an input file having the same name as a file on the output device to replace that output file.
- Delete a file or group of files.
- Selectively perform an operation on each file of a group of files.
- Change the protection code on a file.
- Print filename, file size, creation date, and protection code of each file in a disk, DECtape, cassette, or magtape directory.
- Print a listing of the filenames on disk, DECtape, cassette, or magtape directory.
- List the number of free blocks remaining on the device specified.
- Zero (initialize) a disk (except RPØ3), DECtape, cassette, or magtape.
- Perform fast copy and/or verify operations for DECtape.
- Perform read-after-write verification for cassette tape.

1.1 CALLING PIP

The user must be logged into DOS/BATCH before running PIP. Any previous program in core must be removed with the KILL command. (The user should be familiar with the operating system commands before attempting to use PIP; particularly the RUN, LOGIN, KILL, and FINISH commands.)

In response to the DOS/BATCH dollar sign character, the user can type the RUN PIP command. For example:

```
$RUN PIP
PIP Vxx    (xx is PIP version number)
#
```

Following the RUN command, PIP is called into core, after which it prints its version number and a number sign. The number sign indicates that PIP is ready to accept a command string. Following completion of a command, PIP prints another # character and awaits further commands.

PART 12

CHAPTER 2

COMMAND STRINGS

All commands to PIP are processed by the operating system's Command String Interpreter (CSI). This ensures a uniform command format among the various system programs operating under DOS/BATCH. This chapter briefly describes the command string format of PIP.

2.1 GENERAL COMMAND STRING FORMATS

The most general format of a PIP command string is

```
output file<input file(s)
```

where a single input or output file is completely specified by the following format.

```
dev:filnam.ext[uic]
```

Example:

```
DTØ:MACRO.OBJ[2ØØ,2ØØ]
```

Either an output or an input file specification can be followed by a switch specification in the command string. A switch specification indicates that some special PIP action is to be performed. (See Appendix J for a list of PIP switches.) The switch specifications have two forms:

```
/sw
```

or

```
/sw:n
```

where

/	signals a PIP switch specification.
sw	indicates the two letter switch code.
:n	indicates a decimal or octal value or, in some cases, a one or two letter code.

PIP commands can take a variety of formats, depending upon the action desired. Some PIP actions are caused when a switch is the only specification in the command string. Other PIP actions are caused by a file specification followed by one or more switch specifications in the command string. Examples of typical command strings are shown throughout this part of the handbook.

2.2 DEVICE SPECIFICATION

The device specification consists of two letters, followed where necessary by an octal digit indicating one of multiple units, and terminated by a colon. (The standard device specifications are shown in Appendix C.) An octal digit is required in the device specification to distinguish between units on multiple-unit devices. For example, DT1: and DT2: could be used to specify two DECtapes on the same controller. When no digit is indicated for a device, unit Ø is assumed by the system.

When no device is specified for a file, the desired device is assumed to be the last device specified on that side (input or output) of the angle bracket (<). For example:

```
#DK1:<DT1:FILE1,FILE2
```

where FILE1 and FILE2 are on DECtape unit 1.

When no device has been specified in the command, the system disk is assumed to be the default I/O device (except for /DI, /BR, and /FR switches, where the terminal is the default output device and the system disk is the default input device). For example, /DI will list the current user's system disk directory on the terminal.

2.3 FILENAME SPECIFICATION

A filename specification consists of from one to six letters or digits or an asterisk. All characters in excess of six are ignored. The asterisk feature is described in Section 12-2.5. Examples of filenames are shown below.

```
SEVENLE      is interpreted as SEVENL
A34B
FILE
*            indicates all (filenames or extensions)
```

Null fields in a command are ignored.

```
*SY:<DK1:A,,,B
```

This command causes files A and B to be read from the DK1: disk to the system disk as if the superfluous commas were not present; i.e., filenames are separated by a comma.

2.4 FILENAME EXTENSION SPECIFICATION

The filename extension specification consists of a period followed by one to three letters or digits, or an asterisk. The asterisk feature is described in Section 12-2.5. Extensions may be used to specify the format in which a certain file is stored, the version of a particular file, or any other information the user desires. Examples of filenames used with filename extensions are shown below.

```
FORT.1A
ABC.PAL
FILE.TMP
MODS.ØØ1
MODS.ØØ2
MAT.*          all files with the name MAT
```

Filename extensions, where associated with a filename, must be used to specify that particular file, unless the asterisk feature is used. A filename extension may not be used without a filename (except with an asterisk substituted for a filename, as described below).

2.5 ASTERISK FEATURE

The asterisk (*) character can be used in a command string to replace either the filename or filename extension specification or the user number or user group number in the UIC. The asterisk is interpreted as all files with the filename or filename extension indicated or all users with the user number or user group number specified. For example:

```
*.TMP          indicates all files with the extension .TMP
FILE.*         indicates all files with the name FILE
[*,13]         indicates all users in user group 13
[1Ø,*]         indicates all user groups with user number 1Ø
```

The asterisk can appear in both positions:

```
*.*
```

which denotes all files on the specified device belonging to the user identification code specified (or the current user UIC if no UIC is specified).

Except where noted, the asterisk feature can be used in transfer operations, directory listing operations, deletion operations, and rename operations.

PIP can reference all files with null extensions through use of the asterisk feature. For example, to delete all files with null (no) extensions, use the following command.

```
#*./DE
```

Explicit noncurrent UIC specification and/or the asterisk feature are allowed in the UIC field for functions that are not explicitly prohibited by the other user.

If the user attempts to transfer a series of files (with the *) in which one has a protection code that does not allow transfer, a fatal Monitor error will result when PIP attempts to transfer that file, and successive files designated for transfer in that command string will not be transferred.

If a file appears on disk or DECtape with a name not acceptable to the CSI, it must be deleted using the * feature. For example,

```
ABC.  A      (2 spaces in extension)
```

is not acceptable and should be deleted.

```
ABC.*/DE
```

The above works unless the file is locked; in which case it must be unlocked. See Section 12-4.5.

To copy files from one device to another using the *, the * must be in the input field of the command string. For example:

```
dev1:<dev2:*.*
```

or

```
dev1:<dev2:*.PAL
```

When transferring files, the * cannot be specified in the output field of the command string.

2.6 SWITCH SPECIFICATIONS

The switch specification consists of a slash followed by two letters, and is optionally followed by a value specification of octal or decimal digits or one or two letters separated from the switch code by a colon. Switch names can be of any length; but, only the first two characters are significant. Table 12-1 lists the PIP switches.

Nontransfer operations in PIP are selected by use of switches. If more than one switch is used, each switch is preceded by a slash. A summary of the switch options is in Appendix J.

When present, a switch must follow any dataset specification; that is, a switch cannot appear before the filename, extension, or UIC of the file on which the switch is to operate. However, some switches may be specified without any file specification.

Some options require a numeric value to be associated with the switch. Whether the value is octal or decimal depends upon the switch. For example: /PR is the switch to change the file protection code. The new protection code is preceded by a colon.

```
#DT1:ABC/PR:155
```

This command changes the protection code of file ABC on DECtape unit 1 to 155. Refer to Section 12-3.6 for a description of protection codes and their uses.

In general, PIP switch options can be grouped into two categories: action switches and qualifying switches. Action switches are generally associated with some action or operation. One one action switch can be used in any one PIP command. The action switches follow.

/AL	Allocate
/BR	Brief Directory
/DE	Delete
/DI	Directory
/EN	Enter
/FC	Fast Copy
/FR	Free
/PR	Protect
/RE	Rename
/RU	Rewind and Unload
/RW	Rewind
/SU	Supersede
/UN	Unlock
/VE	Verify
/ZE	Zero

Qualifying switches are used to give PIP particular information about how to perform an operation. These switches can be used in conjunction with other switches in a single command string. The qualifying switches follow.

/CK	Checksum
/CO	Contiguous
/FA	Formatted ASCII
/FB	Formatted Binary
/IN	Inspect
/RW	Rewind
/SU	Supersede
/UA	Unformatted ASCII
/UB	Unformatted Binary
/VE	Verify
/VW	Verified Write

Another type of distinction that is made among switches is that some can be used alone, some operate upon a single file designation, and some require both an input and an output specification.

Table 12-1

PIP Switches

Used Alone	Used with One or More File Designations	Used with Both an Input and Output Designation
/BR	/AL	/BR
/CK	/BR	/CK
/DI	/CK	/CO
/EN	/DE	/DI
/FR	/DI	/FA
/ZE	/EN	/FB
	/FR	/FR
	/IN	/PR
	/PR	/RE
	/RE	/RW
	/SU	/UA
	/UN	/UB
	/VW	
	/ZE	

PART 12

CHAPTER 3

FILE MANIPULATION

3.1 FILE TRANSFERS

The transferring of files between devices is a primary function of PIP. PIP uses the device independent features of the operating system and can either copy a file directly onto another device or merge several files from the same or different devices into a single file on one device.

3.1.1 File Copy Operation

In the simplest case, PIP can transfer a file from one peripheral onto another. For example, to list a file, transfer a copy of the file to the line printer, as shown here.

```
#LP:<DT2:FILE1.PAL      where FILE1.PAL is on DECtape
#LP:<MT1:FILE2.PAL      where FILE2.PAL is on magtape
#LP:<FILE3.PAL          where FILE3.PAL is on the system disk
```

The following command transfers the data on a paper tape from the high-speed reader and creates a file named INPUT.TST on the system disk.

```
#INPUT.TST<PR:
```

The following command duplicates a paper tape.

```
#PP:<PR:
```

There can be only one output file designation in a PIP command, although there can be any number of input files (limited only by the terminal line length). For example, to copy three files (ONE.PAL, TWO.PAL, and THREE.PAL) from the system disk onto DECtape unit Ø, type the following command.

```
#DT:<ONE.PAL,TWO.PAL,THREE.PAL
```

In the case shown above, the files are copied directly to DECtape unit Ø, using the same filenames as on disk.

In order to change a filename, the RENAME (/RE) switch can be used.

```
#DK:<DT1:FILE1,PR:ABC,MT:DEF.ØØ1
#DAVE1/RE<FILE1
```

The first command transfers three files onto the RK11 disk unit Ø; FILE1 from DECtape unit 1, ABC from the paper tape reader, and DEF.ØØ1 from magtape unit Ø. Each file retains its original name. Note the ability to indicate the name of a file from the high-speed paper tape reader. The second command renames the file FILE1 on DKØ: to DAVE1.

3.1.2 Fast Copy and/or Verify Operations

The fast copy (/FC) and verify (/VE or /FC/VE) operations are applicable only to DECtape. Input and output device types must be the same; unit numbers must be different. The S26Ø or S231 error message will result if the preceding two conditions are not met. The format of these operations is illustrated in the three command strings shown below.

```
#DT1:<DT2:/FC/VE
```

or

```
#DT1:<DT2:/FC
```

or

```
#DT1:<DT2:/VE
```

/FC does a block-by-block transfer, unlike a normal transfer at the file level. PIP uses the largest amount of core memory available; consequently, the larger the main memory, the faster the operation. On 28K machines, for example, a DECtape can be copied in about two minutes; verifying it takes two and a quarter minutes.

The verify operation as shown above can be performed alone. If the data on the two devices differ, a message indicating the block and word in disagreement is printed at the console in the following octal format.

Block	Word
N	M

3.1.3 File Merge Operation

When there are several input files involved in a transfer, the files can be transferred individually, as the previous examples illustrate, or they can be merged

into a single output file. When no output filename is specified on the output device, input files are copied without being combined.

```
#DT3:<ABC,DT2:RASP,PR:DEF
```

Three files are copied onto DECtape unit 3; one from the system disk, one from DECtape unit 2, and one from the high-speed paper tape reader. Each retains the same name given on the input side.

When a filename is specified on the output device, all input files are combined to create a single new output file. In order to combine the three files shown above into a single file named MERGE, the following command is used.

```
#DT3:MERGE<ABC,DT2:RASP,PR:DEF
```

Numerous variations on this operation are possible. One or more of the input files can be specified as being under another UIC, input files can be on the same or different devices, etc.

An operation to merge a single file is equivalent to creating another copy of that file with a different name. The command

```
#ABC<DT1:FILE1
```

takes FILE1 on DECtape unit 1 and puts in on the system disk under the name ABC.

3.1.4 File Transfer Modes

The modes of transfer are either formatted or unformatted, either ASCII or binary. Refer to Part 3, the DOS/BATCH Monitor, for further details on the modes of transfer.

In PIP, the mode of transfer of any file is determined by the format switch specified in the command string. If a format switch is not specified in a command string, the mode assumed for the transfer is unformatted binary or image mode (unformatted ASCII if either device does not support binary). This means, for example, that files transferred from paper tape without a mode specification include blank frames, leader and trailer tape. A switch specified in the command string may be used to force a particular mode of transfer. The switches that specify format are /FA, /FB, /UA, and /UB. Only one type of format switch can be specified on one side of a command. The switch can be used with both input and output files. For example, to transfer the files ABC.DAT and ABC.TST in formatted binary mode, use either of these commands.

```
#DT4:<ABC.DAT/FB,ABC.TST/FB
```

or

```
#DT4:/FB<ABC.DAT,ABC.TST
```

In the case of the paper tape reader, use of these switches guards against putting the tape in the reader backwards. For example, if the paper tape were in backwards, the command

```
#SPRMAN.OBJ<PR:/FB
```

would cause PIP to print a S2Ø2 error message on the terminal. To transfer ASCII cards from the card reader to disk or DECtape, the /FA switch must be used.

3.1.5 Contiguous Files

Contiguous files are handled differently from linked files. Blocks of contiguous files are guaranteed to be physically adjacent, whereas linked files are generally separated by a minimum fixed distance (called the interleave factor) to take advantage of the rotational delay inherent in a disk. Contiguous files are generally used for random accessing of data, a feature that is difficult with linked files. A contiguous file is noted in a directory listing by a C following the number of blocks for that file.

Unless otherwise specified, file transfers will produce linked files. The /CO switch is used to specify an output file as contiguous. Three cases exist where use of the /CO switch is valid. All other uses of the /CO switch are invalid. The first two cases apply to a merge operation and a copy operation. For each operation, a new filename is specified in the output field of the command string. The /CO switch must be specified in the output side of the command string if the new file is to be created as a contiguous file, as shown below.

1. A merge operation where the new file is to be made contiguous. For example,

   ```
   #SY:FILEME.RGE/CO<DK1:FIL1,DT1:FIL2
   ```

2. A copy operation where the new file is to be made contiguous. For example,

   ```
   #SY:FILB/CO<DK1:FILA
   ```

   NOTE

   In the case where the output file is explicitly named, the /CO switch must be on the output side.

In both of the preceding cases, it makes no difference whether the input file(s) is contiguous or linked. The output file must be designated contiguous with the /CO switch.

The third case applies to a simple transfer operation where no new filename is specified explicitly but the new output file(s) is to be made or is to remain contiguous. In this case, the /CO switch must be given on the input side of the command string for the file(s) desired to be contiguous on the output device. For example,

```
#SY:<DK1:FIL1/CO,FIL2,FIL3/CO
```

both FIL1 and FIL3 are created as contiguous files on the output device. FIL2 is created as a linked file, whether or not it is linked or contiguous. The same applies to the simplest case. For example,

```
#SY:<DK1:FILA/CO
```

FILA is created as a contiguous file on the output device. If the /CO switch is not used, FILA would be created as a linked file on the output device, whether or not it is linked or contiguous on the input device.

3.1.6 Transfers from the Terminal

Occasionally, the user may wish to transfer a file from the keyboard to another device. This is accomplished as follows.

```
#PP:<KB:
```

Subsequent characters typed at the keyboard are output to the high-speed paper tape punch. (Punching is performed in bursts as the tape punch buffer fills.) To indicate an end-of-file with such an operation (i.e., to stop transferring data to the punch), the DOS convention is to type CTRL/C followed by the END command.

In order to execute the command, the terminal must be in Monitor mode. Consequently, the sequence is:

```
↑C      (type CTRL/C)
.END    (followed by the RETURN and LINE FEED keys respectively)
```

The processing of the LINE FEED Key is necessary to force control back to PIP after the END command is executed. PIP returns with the # character when the above sequence is completed.

This same sequence is necessary when transferring from the low-speed paper tape reader. However, it is necessary to indicate the end-of-file from the user keyboard; the END PT command is given in this case. If the user types

```
#DF:FILEIN.MAC<PT:
```

it is necessary to type

```
↑C          (type CTRL/C)
.END PT     (followed by the RETURN key and LINE FEED key respectively;
             no colon is used after PT)
```

after the tape has been read and the low-speed reader stops.

Binary transfers are not allowed from the low-speed paper tape reader. The device is effectively the console, and the binary data input may contain control characters that will stop the data transfer.

3.2 RENAMING FILES

The rename switch allows the user a means of changing the name of a file. It may be used on either side of the command string. Only one file at a time can be renamed. If no device is specified, the system disk is assumed. For example,

```
#MAIN.OBJ/RE<TESTX.OBJ
```

changes the name of the system disk file TESTX.OBJ to MAIN.OBJ.

When renaming, it is mandatory that the same device be used on both sides of the angle bracket. The following is allowed.

```
#DT:FILE1<FILE2/RE
```

The device for FILE2 is assumed to be the DECtape. The same command could be written as the following.

```
#DT:FILE1:<DT:FILE2/RE
```

3.3 DELETING FILES

The delete switch is provided to allow the user a means of deleting one or more files. The simple form of the command consists of a file designator followed by the /DE switch. For example,

```
#DT:MAIN.LDA/DE
```

deletes the file MAIN.LDA from DECtape unit Ø.

Several files can be deleted by specifying a sequence of filenames before the DELETE switch. For example,

```
#AB,DT:CD,DT1:EF.OBJ/DE
```

deletes three files on three different devices: SY:, DT:, DT1.

The delete switch supports the asterisk in the filename and/or extension fields of the file specifications. The asterisk is interpreted as specified in Section 12-2.5. For example,

```
#DT1:*.LDA/DE
```

deletes all files with the extension .LDA from DECtape unit 1.

```
#*.*/DE
```

deletes all files under the current UIC from the system disk.

To delete all files from a DECtape, magtape, or cassette tape, it is recommended that the zero switch be used, since that operation is much faster (see Sections 12-4.4.1 and 12-5.4). The zero switch must be used to delete files from magtape as no form of the delete switch is allowed in magtape processing.

3.4 INSPECT SWITCH

The inspect switch (/IN) is an auxiliary switch that instructs PIP to type out the name of each eligible file, followed by a question mark, before executing the main operation. The user can then perform the operation on that particular file, by typing a Y (for yes) followed by the RETURN key; or he may bypass the file by simply typing the RETURN key.

In general, /IN is used with file transfers and with the /SU, /DE and /RE switches which is particularly useful when used in conjunction with asterisk-type commands. For example

```
#*.*/DE/IN
```

allows the user to edit (accept or reject) the contents of a whole directory of the current UIC; for example,

```
ABC.OBJ? Y          ;Delete this file
DEF.PAL?            ;Do not delete this file
.
.
.
etc.
```

In the examples above, typing Y at the console caused the deletion of file ABC.OBJ. PIP listed another filename; the RETURN key was typed, file is left intact; PIP lists another file, etc.

3.5 SUPERSEDE OPERATION

When used on the input side of a transfer command, the supercede switch (/SU) causes any input file having the same name as a file on the output device to replace that output file. Input files not existing on the output device will simply be transferred as usual. The /SU switch will not delete an output file if the input file to replace it is not present on the input device. In this instance, an S error condition will be given, and PIP will proceed to transfer any remaining files.

The command is normally used in association with specific filenames specified only on the input side of the command string.

```
#dev:<devl:A,B,C/SU
```

or for asterisk-type transfers,

```
#dev:<devl:*.*/SU
```

If the filename specification appears in the output side of the command string and the file already exists on the output device, the transfer is not made, and an S257 error occurs.

In addition to its use as a stand-alone command, the supersede mode is invoked with a /SU:on command, which will remain in effect until the mode is turned off with a /SU:OFF, or until the program is killed. Enabling the mode in this way obviates entering a /SU switch following each command.

3.6 PROTECT SWITCH

The protect switch is provided to allow the user to change the protection of a file. It is not applicable to cassette tape.

The protection code is specified as an octal value argument to the switch and is treated as three fields corresponding to owner, user group, and all others. Each field is assigned an octal digit as described in the DOS/BATCH Monitor, Section 3-4.2.2. The general form of this command follows.

```
dev:filnam.ext/PR:value
```

Example:

```
#FILE3/PR:355
```

This command changes the protection code of FILE3 to 355. The protection switch works for DECtape and disk.

3.7 ALLOCATING A CONTIGUOUS FILE

The allocate switch provides a means of creating a contiguous file. This option requires a filename and a decimal integer value (being the number of 64-word segments of storage to be allocated). For example,

```
#DF:FILE/AL:1Ø
```

allocates 64Ø contiguous words of file storage on the RF11 disk. A subsequent PIP command,

```
#FILE<PR:
```

causes a paper tape to be read into the contiguous file previously created.

The allocate switch can be used to create a contiguous file on disk or DECtape. Note that the segments allocated are always 64 words and do not depend upon the block size of the device. For example,

```
#DKØ:FILE/AL:1ØØ
```

creates a blank contiguous file named FILE on device DKØ. Since an RK11 device contains 256 words per block, the allocated file contains 25 blocks, each 256 words in length. Each block contains four 64-word file segments. The number 25 followed by C is shown as the number of blocks of the file on the full directory listing for the device on which the contiguous file was created. For example:

```
#DKØ:FILE/DI
DIRECTORY DKØ:  [2ØØ,2ØØ]
24-AUG-72
FILE      25C       24-AUG-72     <233>
```

whenever it is necessary to create an empty, contiguous file, the identifier :E is appended to the command string. For example,

```
#DF:FILE1/AL:1Ø:E
```

allocates 64Ø contiguous words of file storage on the RF11 disk as normally. However, pointers are set in the UFD file entry such that the space is apparently empty. A file allocated in this manner can accept partial sequential output (through a Monitor .OPENC call). The pointers are updated to reflect the end of the actual data entered in part of the allocated space. The same file can subsequently be extended (through a Monitor .OPENE call) within the unfilled allocated space.

Use of the allocate switch allows data to be transferred from a non-file-structured device to the blank contiguous file. Without the allocate switch, it would be impossible for the system to determine the size of the file from a non-file-structured input device. Data can also be transferred from a file-structured device to a blank contiguous file. However, allocation of the file space is not necessary, since the file-structured input device contains the necessary size information in its directory. (See the description of the /CO, contiguous, switch in Section 12-3.1.5.)

PART 12

CHAPTER 4

DIRECTORY MANIPULATION

A significant number of PIP features center around the manipulation of files within directories and of the directories themselves.

4.1 ENTERING USER IDENTIFICATION INTO THE MASTER FILE DIRECTORY

The User Identification Code (UIC) must appear in the Master File Directory (MFD) before a user is able to create his own files on the disk or DECtape. This code is generally entered into the MFD by using the enter switch in PIP, although the DOS system building program creates a few fundamental User File Directories (UFD's). Refer to the DOS/BATCH System Manager's Guide for further details.

The UIC of the current user of the system is passed to the operating system by the LOGIN command. For example,

```
$LOGIN 11,11
```

sets the current UIC to [11,11]. In using the enter switch, the UIC of the current user is obtained from the system and written into a spare slot in the MFD. Here is the recommended sequence.

```
$LOGIN uic
$RUN PIP
PIP Vxx
#/EN
```

The LOGIN UIC is now a valid UIC on the system disk.

If no spare slots exist in the MFD, PIP prints a fatal error message, and no attempt is made to extend the MFD.

This procedure is not always necessary for DECtapes because the zero switch enters the UIC of the current user when the directory is initialized (see Section 12-4.4.1).

If a user other than the one originally zeroing the DECtape wishes to access files on the tape, he must enter his UIC on the DECtape or explicitly reference the UIC field of the original user during each reference. For example, if the DECtape on unit 1 had been originally zeroed and filled by UIC [5Ø,5Ø], user UIC [2ØØ,2ØØ] could reference those files by logging in under [2ØØ,2ØØ] and by giving the following command.

```
#DT1:/EN
```

In general, then, an ENTER operation performs one of two services: for DECtapes it allows any user to access all files on the DECtape as though he had initialized the DECtape; and on disk it creates a UFD for the current (logged in) UIC (or the UIC specified), allowing that user to store files on the system disk with whatever protection is desired.

4.2 DIRECTORY LISTINGS

A listing of the directory of all or certain selected files on any file structured device can be printed on the terminal or line printer; PIP assumes the terminal is the output device unless another device is specified in the command string. The information listed in a directory is a subset of all the information in the UFD entry for each file.

4.2.1 Full Directories

A full directory (/DI) listing contains two title lines to identify the device, UIC, and the date, followed by all filenames and extensions, the number of blocks per file (followed by a C if the file is contiguous), file creation date, and file protection code (enclosed within angle brackets), plus the total number of blocks and files for that UIC. See the example on the following page.

```
#/DI

DIRECTORY DKØ: [  1,6  ]

Ø3-APR-74

FORTRN.OBJ     72  20-FEB-74 <233>
OVØ   .OBJ     72  20-FEB-74 <233>
OV1   .OBJ     82  20-FEB-74 <233>
OV2   .OBJ    102  20-FEB-74 <233>
OV3   .OBJ     95  20-FEB-74 <233>
OV4   .OBJ     47  20-FEB-74 <233>
FORLIB.OBJ    167  20-FEB-74 <233>
EAEMOD.OBJ     17  20-FEB-74 <233>
EISMOD.OBJ     18  20-FEB-74 <233>
FISMOD.OBJ      3  20-FEB-74 <233>
FPUMOD.OBJ     31  20-FEB-74 <233>
EAEPC .OBJ      4  20-FEB-74 <233>
EISPC .OBJ      4  20-FEB-74 <233>
FISPC .OBJ      4  20-FEB-74 <233>
FPUPC .OBJ      4  20-FEB-74 <233>
TRCLIB.OBJ     12  20-FEB-74 <233>
FORPC .OBJ      4  20-FEB-74 <233>
COMDGN.SRC     15  20-FEB-74 <233>
RUNDGN.SRC     14  20-FEB-74 <233>
FORLIB.BAT      1  20-FEB-74 <233>
FORPC .BAT      1  20-FEB-74 <233>
EAELIB.BAT      1  20-FEB-74 <233>
EAEPC .BAT      1  20-FEB-74 <233>
EISLIB.BAT      1  20-FEB-74 <233>
EISPC .BAT      1  20-FEB-74 <233>
FISLIB.BAT      1  20-FEB-74 <233>
FISPC .BAT      1  20-FEB-74 <233>
FPULIB.BAT      1  20-FEB-74 <233>
FPUPC .BAT      1  20-FEB-74 <233>
FORDGN.FTN      6  20-FEB-74 <233>
TRACEF.FTN      8  20-FEB-74 <233>
DVB   .MAC      9  20-FEB-74 <233>
FORCOM.DGN     21  20-FEB-74 <233>
FORRUN.DGN     16  20-FEB-74 <233>

TOTL BLKS:   837
TOTL FILES:   34

#
```

Notice that the system device (DKØ: in this case) was assumed and the user was logged in under UIC [1,6].

Whenever a file is transferred onto disk, cassette tape, or DECtape, the date in the directory reflects the date on which the transfer took place, not the date on which the file was created.

The full directories of multiple devices can also be obtained.

```
#DK0:,DT1:/DI

DIRECTORY DK0: [  4,3  ]

01-APR-74

DIREC .LST     98  04-FEB-74 <233>
DIREC .OBJ     13  04-FEB-74 <233>
DIREC .CRF     15  04-FEB-74 <233>
PIP1          116  04-FEB-74 <233>
PIP   .LIB    116  04-FEB-74 <233>
DIREC .MAC     32  04-FEB-74 <233>
PIPEX .MAC     35  04-FEB-74 <233>
PIPEX .LST    109  04-FEB-74 <233>
PIPEX .OBJ      9  04-FEB-74 <233>
PIP   .LDA     31  04-FEB-74 <233>
PIP   .MAP      9  04-FEB-74 <233>

TOTL BLKS:   583
TOTL FILES:   11

DIRECTORY DT1: [  4,3  ]

01-APR-74

DUMMY .MAC      3  01-APR-74 <233>
MTTEST.LDA     11  01-APR-74 <233>
TRNTST.LDA     11  01-APR-74 <233>

FREE BLKS:   537
FREE FILES:   53

#
```

When a full DECtape directory is obtained, the listing is concluded with the total number of free blocks and files, whereas with a disk, magtape, or cassette tape, the number of used blocks and files are reported.

The full directory listing can be limited to produce a partial list of files on one or more devices. The following example illustrates the use of the asterisk to list only those files with the extensions specified on the two devices specified.

```
#DT1:*.LDA,DK0:*.LIB,*.OBJ/DI

DT1:

[  4,3  ]

MTTEST.LDA     11  01-APR-74 <233>
TRNTST.LDA     11  01-APR-74 <233>

DK0:

[  4,3  ]

PIP    .LIB   116  04-FEB-74 <233>

[  4,3  ]

DIREC .OBJ     13  04-FEB-74 <233>
PIPEX .OBJ      9  04-FEB-74 <233>
#
```

Notice that the UIC, date, and number of blocks and files for that UIC are not reported in partial /DI listings.

Directory information about a particular file can be obtained by specifying the filename and extension with the /DI switch. For example:

```
#PIPEX.OBJ/DI

DK0:

[  4,3  ]

PIPEX .OBJ      9  04-FEB-74 <233>

#
```

This obtains all information about PIPEX.OBJ if it is in the DKØ directory. (If the file is not in the directory, there is no listing, and no error message.)

A directory of any UIC on any device can be listed by specifying that UIC in the command to PIP. For example, if a user is logged in under UIC 2ØØ,2ØØ, he can give the following command (the results are shown on the following page).

```
#DK1: [1ØØ,32]/DI

DIRECTORY DK1: [100,32 ]

Ø3-APR-74

KNIGHT.ODL      1   17-JAN-74 <233>
ADL    .ODL      1   17-JAN-74 <233>
MAIN   .FTN      1   17-JAN-74 <233>
ADL1   .FTN      1   17-JAN-74 <233>
ADL2   .FTN      1   17-JAN-74 <233>
A2A    .FTN      1   17-JAN-74 <233>
A2B    .FTN      1   17-JAN-74 <233>
A3A    .FTN      1   17-JAN-74 <233>
A3B    .FTN      1   17-JAN-74 <233>
A4A    .FTN      1   17-JAN-74 <233>
A4B    .FTN      1   17-JAN-74 <233>
ADL    .BLD      1   17-JAN-74 <233>
ADLDK1.BLD       1   17-JAN-74 <233>
A3A    .OBJ      2   17-JAN-74 <233>
A3B    .OBJ      1   17-JAN-74 <233>
A4B    .OBJ      1   17-JAN-74 <233>
OVRLY  .OBJ      2   17-JAN-74 <233>

TOTL BLKS:    19
TOTL FILES:   17

#
```

If the user requests information on any particular file or uses the asterisk feature where no such file exists, no error message is given. The directory is considered blank; and a blank line is printed.

```
#DK0:*.OBJ,DT1:*.OBJ/DI

DK0:

[  4,3  ]

DIREC .OBJ     13   04-FEB-74 <233>
PIPEX .OBJ      9   04-FEB-74 <233>

DT1:

[  4,3  ]

#
```

4.2.2 Brief Directory Listings

The brief directory (/BR) switch can be used to list only the device name, filenames, and extensions. The asterisk feature or specific filenames can be used with the brief directory switch to obtain partial brief directories.

The following PIP example demonstrates the correspondence between /DI and /BR directory listings.

```
#*.LDA/DI

DIRECTORY DK0: [  1,1  ]

18-MAR-74

BADB  .SYS      1  19-FEB-74 <377>
MONLIB.CIL   1750  19-FEB-74 <377>
SYSMAC.SML     26  26-FEB-74 <233>
LDUMP .LDA     29  26-FEB-74 <233>
CL0   .SYS      1  26-FEB-74 <233>
CL1   .SYS      1  26-FEB-74 <233>
CL2   .SYS      1  26-FEB-74 <233>
CL3   .SYS      1  26-FEB-74 <233>
CL4   .SYS      1  26-FEB-74 <233>
CL5   .SYS      1  26-FEB-74 <233>
CL6   .SYS      1  26-FEB-74 <233>
CL7   .SYS      1  26-FEB-74 <233>
CKBLD .BAT      2  26-FEB-74 <233>
CKBLDF.BAT      4  26-FEB-74 <233>
CTKIT1.BAT      2  26-FEB-74 <233>
CTKIT2.BAT      9  26-FEB-74 <233>
CTBLD .BAT      9  26-FEB-74 <233>
CTSAV .BAT      6  26-FEB-74 <233>
CTRES .BAT      4  26-FEB-74 <233>
CTKITF.BAT      6  26-FEB-74 <233>
CTBLDF.BAT      4  26-FEB-74 <233>
CTSAVF.BAT      4  26-FEB-74 <233>
CTRESF.BAT      2  26-FEB-74 <233>
KILLCT.BAT      1  26-FEB-74 <233>
PTKIT1.BAT      9  26-FEB-74 <233>
PTKIT2.BAT      5  26-FEB-74 <233>
PTBLD .BAT      2  26-FEB-74 <233>
PTSAV .BAT      2  26-FEB-74 <233>
PTRES .BAT      1  26-FEB-74 <233>
PTKITF.BAT      4  26-FEB-74 <233>
PTBLDF.BAT      4  26-FEB-74 <233>
PTSAVF.BAT      1  26-FEB-74 <233>
PTRESF.BAT      2  26-FEB-74 <233>
KILLPT.BAT      2  26-FEB-74 <233>
UDT   .OBJ     10  26-FEB-74 <233>
LINK  .LDA    570  26-FEB-74 <233>
FORTRN.LDA     94  04-MAR-74 <233>
FORCOM.OBN    210  04-MAR-74 <233>
MACRO .LDA    390  26-FEB-74 <233>
CREF  .LDA     16  26-FEB-74 <233>
PIP   .LDA    350  26-FEB-74 <233>
CILUS .LDA     59  26-FEB-74 <233>
EDIT  .LDA     13  26-FEB-74 <233>
FILDMP.LDA      7  26-FEB-74 <233>
DSKINT.LDA      9  26-FEB-74 <233>
LIBR  .LDA      9  26-FEB-74 <233>
```

```
FILCOM.LDA      14   26-FEB-74 <233>
EBASIC.LDA       6   26-FEB-74 <233>
VERIFY.LDA     07C   26-FEB-74 <233>
PIUTIL.LDA      11   26-FEB-74 <233>
FORRUN.DGN     16C   04-MAR-74 <233>
FTNLIB.OBJ     157   04-MAR-74 <233>
TRCLIB.OBJ      12   04-MAR-74 <233>
FORTRN.OVR    118C   04-MAR-74 <233>

TOTL BLKS:   1618
TOTL FILES:    54
```

#*.LDA/BR

```
DK0:

[  1,1  ]

BADB  .SYS
MONLIB.CIL
SYSMAC.SML
LOGMF .LDA
CL0   .SYS
CL1   .SYS
CL2   .SYS
CL3   .SYS
CL4   .SYS
CL5   .SYS
CL6   .SYS
CL7   .SYS
DKBLD .BAT
DKBLDF.BAT
CTKIT1.BAT
CTKIT2.BAT
CTBLD .BAT
CTSAV .BAT
CTRES .BAT
CTKITF.BAT
CTBLDF.BAT
CTSAVF.BAT
CTRESF.BAT
KILLCT.BAT
PTKIT1.BAT
PTKIT2.BAT
PTBLD .BAT
PTSAV .BAT
PTRES .BAT
PTKITF.BAT
PTBLDF.BAT
PTSAVF.BAT
PTRESF.BAT
KILLPT.BAT
ODT   .OBJ
LINK  .LDA
FORTRN.LDA
FORCOM.DGN
MACRO .LDA
CREF  .LDA
PIP   .LDA
CILOS .LDA
```

```
EDIT   .LDA
FILDMP.LDA
DSKINT.LDA
LIBR   .LDA
FILCOM.LDA
EBASCI.LDA
VERIFY.LDA
PTUTIL.LDA
FORRUN.OBJ
FTNLIB.OBJ
TRCLIB.OBJ
FORTRN.OVR
```

An example of the /BR switch with the system device assumed is shown below.

```
#/BR

DK0:

[  1,5  ]

PIP16 .LDA
EDIT16.LDA
LNK256.LDA
VER256.LDA
CILUS .LIB
LINK  .LIB
LINK  .ODL
LNKNOV.BAT
LNKOV .BAT
MACRO .LIB
MACRO .ODL
MACNOV.BAT
VERIFY.LIB
VERIFY.ODL
VERNOV.BAT
VEROV .BAT
PIP   .LIB
PIP   .ODL
PIPNOV.BAT
CUSP  .LIB
FILCOM.BAT
DSKINT.BAT
EBASCI.BAT
EDIT  .BAT
LIBR  .BAT
FILDMP.BAT
CREF  .BAT
```

The asterisk (*) feature for filenames and extensions works for directory listings, both full and brief form. For example, the command

```
#*.PAL/BR
```

gives a brief directory listing for all files on the user's disk area with an extension of .PAL. The asterisk can appear in the filename field, the extension field, or both. In addition, groups of file specifiers can be arranged to provide sequential listings.

```
#DTØ:*.MAC,DF:FORTRN.*/DI
```

The asterisk feature can be used in the UIC field when obtaining directory listings.

```
#LP:<DK:[*,*]/DI
```

All directories on RK11 unit Ø are listed on the line printer.

```
#LP:<DF:[*,2ØØ]/DI
```

A listing is produced of all directories associated with user number 2ØØ, in every project.

```
#LP:<DK:*.MAC[*,*]/DI
```

A directory of all files with extension .MAC, on RK11 unit Ø, is produced.

4.3 FREE BLOCKS

Since a full directory listing of a system disk provides the user only with a summary of the number of blocks used by his current UIC, the /FR switch is available for determining the number of free blocks remaining. For example:

```
#/FR

DFØ:
FREE BLKS:  5513
```

This switch can be used with any file-structured device, but is generally only necessary with disk, as the information is provided automatically with a DECtape. Further examples of the /FR switch follow.

```
#/FR

DKØ:
FREE BLKS:  4113
#DF:,DTØ:/FR

DFØ:
FREE BLKS:  3243
DTØ
FREE BLKS:  282
```

4.4 DECTAPE AND RK11 DISK INITIALIZATION

(See Chapter 12-5 for magtape initialization and Chapter 12-6 for cassette initialization.)

4.4.1 DECtape

In order to initialize a new DECtape with the basic file structure information required by the DOS/BATCH Monitor, the zero switch is provided. The zero switch causes PIP to create on the specified DECtape the permanent bit maps, the file bit maps, the MFD, and a UFD for the UIC currently running on the system. The zero switch can be used alone,

```
#DTØ:/ZE
```

or in combination with other actions.

```
#DTØ:ONE/ZE<DT1:A,B,C
```

The latter command zeroes DTØ: first, then creates file ONE on DTØ: by merging files A, B, and C from DECtape unit 1. When used in combination with other actions, zero is performed before any other implied actions.

The /ZE switch can be used to zero only one DECtape per PIP command string. In order to zero two DECtapes, proceed as follows.

```
#DT:/ZE
#DT1:/ZE
```

4.4.2 RK11 Disk

Before initializing an RK11 cartridge with the zero switch, ensure that the cartridge has been formatted using either the RK11 diagnostic program (MAINDEC-11-D5HA-PB2) or the SYSLOD system program. Consult the system administrator for local procedures. The following actions occur during cartridge initialization.

1. The entire disk cartridge is zeroed.
2. The first and second MFD blocks are written.
3. The UIC of the current user (and only this UIC) is entered in the MFD.
4. Bit map blocks are written.

An RK11 disk is initialized by typing

```
#DKn:/ZE
```

where n is the octal number of the appropriate cartridge (or omitted if DKØ is meant). PIP responds by printing

```
CONFIRM:
```

to which the user should type either H (for high-density) or L (for low-density) followed by the RETURN key. Should the user decide not to initialize the disk, he types the SPACE bar before pressing the RETURN key. (High-density cartridges are marked 22ØØ BPI, and low-density cartridges are marked 11ØØ BPI.)

When the disk has been initialized (there is a noticeable pause), PIP again prints the # character to indicate readiness to accept another command.

4.5 RECOVERING FILES

A file can be left in a state that makes it inaccessible for subsequent processing. For example, if a file is open for output and a system crash (hardware or software) occurs causing the Monitor to be reloaded, the file may be left in an inaccessible state. Files declared inaccessible by the Monitor can have any of the following conditions.

1. The LOCK bit in the UFD entry for this file will be set.
2. The USAGE COUNT in the UFD entry for this file will be invalid.
3. Some blocks allocated for this file may not have the proper bits set in the permanent bit map.

PIP provides a partial solution to this problem with the unlock switch. The function of this switch is to restore the lock and usage count fields so that the file can be read. It does not make an attempt to set bits in the bit map for the blocks.

The sequence for recovery is to use the unlock switch, such as

```
$RUN PIP
PIP Vxxx
#DF:FILE.OLD/UN
```

which allows the file to be accessed.

Then, before writing any new data on the disk copy the file to another device and delete it from the disk.

```
#DT:<FILE
#FILE/DE
```

Copy, then delete all files that were open for output when the crash occurred. The disk may then be used safely.

PART 12

CHAPTER 5

MAGTAPE OPERATION

5.1 GENERAL MAGTAPE INFORMATION

Magnetic tape is supported by PIP for the following operations.

a. Directory listings.

```
#MT1:/DI
#MTØ:/BR
```

b. File transfer operations (a magtape reel must be initialized before any files are transferred to the reel for the first time. See Section 12-5.4 for the initialization procedure).

```
#MT1:FILE<DT2:ABC
#MT:MERGE<ABC,DTØ:FACT,MT1:DATA
```

c. Magtape initialization (see Section 12-5.4).

```
#MT1:/ZE
```

d. Rewind, rewind suppression, and rewind and unload (see Section 12-5.2 and 12-5.3).

Transfer and directory operations include all asterisk features used for other file-structured devices. For example, it is possible to type either of the following.

```
#MT1:[*,*]/DI
```

or

```
#MT1:[*,*]/BR
```

This allows the user to obtain a complete listing of all files on the specified magtapes. When this option is used, the UIC specification is printed as part of the file information.

5.2 REWIND

The rewind switch is applicable to magtape units (MTn) and cassette drives (CTn) only, and controls rewind operations for one or more specified units. The specified

unit(s) can still be addressed after use of the rewind switch. The specifications /RW and /RW:YES cause the specified unit(s) to be rewound; the specification /RW:NO prevents rewinding of the specified unit(s).

Examples: #MT1:/RW

causes magtape unit 1 to be rewound.

#DKØ:<MT2:*.*/RW:NO

causes all files under the current UIC on magtape unit 2 to be transferred to disk Ø without rewinding magtape unit 2 between file transfers.

#DKØ:<MT1:FILE.ONE/RW:YES

causes magtape unit 1 to be rewound before and after transferring FILE.ONE to disk Ø.

#MTØ:<MT1:FILE1,FILE2

causes MTØ and MT1 to be rewound before and after each transfer.

#MTØ:/RW:NO<MT1:FILE1,FILE2

will cause MT1 to act the same as the preceding example. MTØ in this case will not rewind.

#MTØ:/RW:NO<MT1:FILE1/RW:NO,FILE2

will cause MTØ: to act as in the preceding example. MT1 will not be rewound before or after the access of FILE1, but MT1 will be rewound on access to FILE 2.

#MT:/RW:NO

inhibits rewinding on magtape unit Ø for all subsequent operations.

#MT:/RW:YES

enables rewinding on magtape unit Ø for all subsequent commands.

If the rewind switch is not specified for an operation, a default value of /RW:YES is assumed.

5.3 REWIND/UNLOAD

The rewind/unload switch is applicable only to tape units. This switch causes the tape to be wound back and the tape unit to be switched off-line. Following this option, the affected tape units cannot be addressed for further processing. For example:

```
#MT:/RU
```

or

```
#MT2:,MT3:/RU
```

5.4 ZERO (INITIALIZATION)

It is mandatory that a magtape be zeroed before use. The /ZE switch writes two end-of-file marks on the tape so that it can be handled correctly by the system monitor. For example:

```
#MT2:/ZE
```

Zeroes (initializes) MT2 (magtape unit 2).

5.5 MAGNETIC TAPE FORMAT

All files on magnetic tape have the general format, exclusive of data format, as shown in Figure 12-1. Any file on the tape may be retrieved by name.

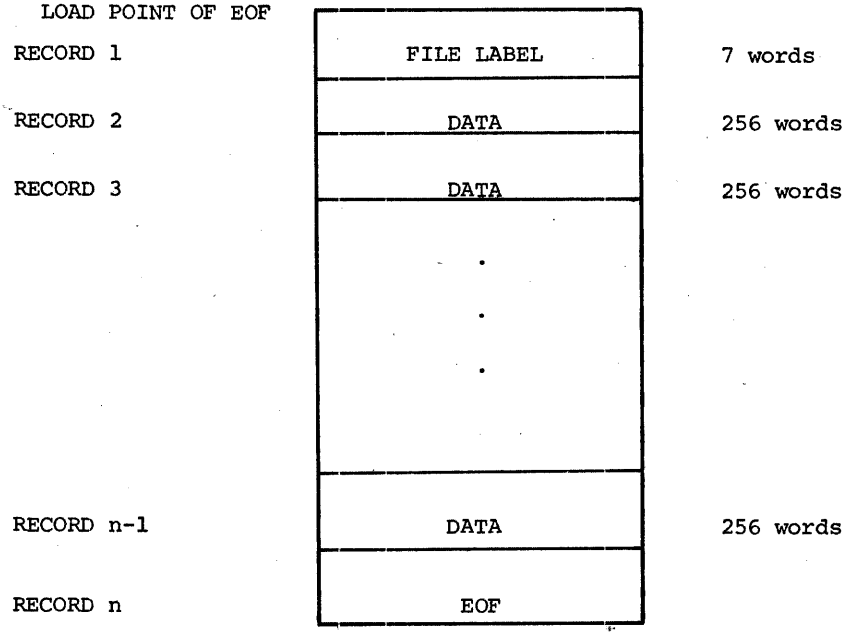

Figure 12-1
Magtape General Format

PART 12

CHAPTER 6

CASSETTE TAPE OPERATION

6.1 GENERAL CASSETTE TAPE INFORMATION

Cassette tape is supported by PIP for the following operations.

a. Directory listings

```
#CTØ:/BR
#CT:/DI
```

b. File transfer operations (a cassette tape must be initialized before any files are transferred to the tape for the first time; refer to Section 12-6.3 for the initialization procedure).

```
#CT:SOURCE.MAC<DK:MACRO.MAC
#CT:MERGE.MAC<CT1:FILE1.MAC,DK1:FILE2.MAC
```

c. Cassette tape initialization (refer to Section 12-6.3)

```
#CT:/ZE
```

d. Rewind specifications (refer to Section 12-6.2).

```
#CT:/RW
#CT:/RW:NO<CT1:*.*/RW:NO
```

e. Read-after-write-verification (refer to Section 12-6.5).

```
#CT:/VW<CT1:FILE.MAC
```

The transfer and directory operations include all asterisk features used for other file-structured devices. This allows the user to obtain a complete listing of all files on the specified cassette.

6.2 REWIND

The rewind switch is applicable only to cassette (CTn) and magtape (MTn) drives. This switch causes the tape to be rewound back onto its original reel. Following this operation, the cassette units still can be addressed. The following are examples.

```
#CT1:/RW
#CTØ:,MT1:/RW
```

The rewind switch enables rewinds during the specified cassette operation. The following example rewinds CT1 before seeking the file specified for the transfer operation.

```
#DKØ:<CT1:FILNAM.EXT/RW:YES
```

The rewind switch also can be used to suppress rewinds during the specified cassette operation. For example,

```
#DKØ:<CT1:*.*/RW:NO
```

transfers all files onto the disk without rewinding CT1.

6.3 ZERO (INITIALIZATION)

A cassette must be zeroed before using the /ZE switch. The /ZE switch writes a sentinel label record on the tape which enables proper handling by the Monitor. The following example,

```
#CT1:/ZE
```

zeroes (initializes) CT1.

6.4 CASSETTE TAPE FORMAT

All files on cassette tape have the general format, exclusive of data format, shown in Figure 12-2. Any file on the tape can be retrieved by name.

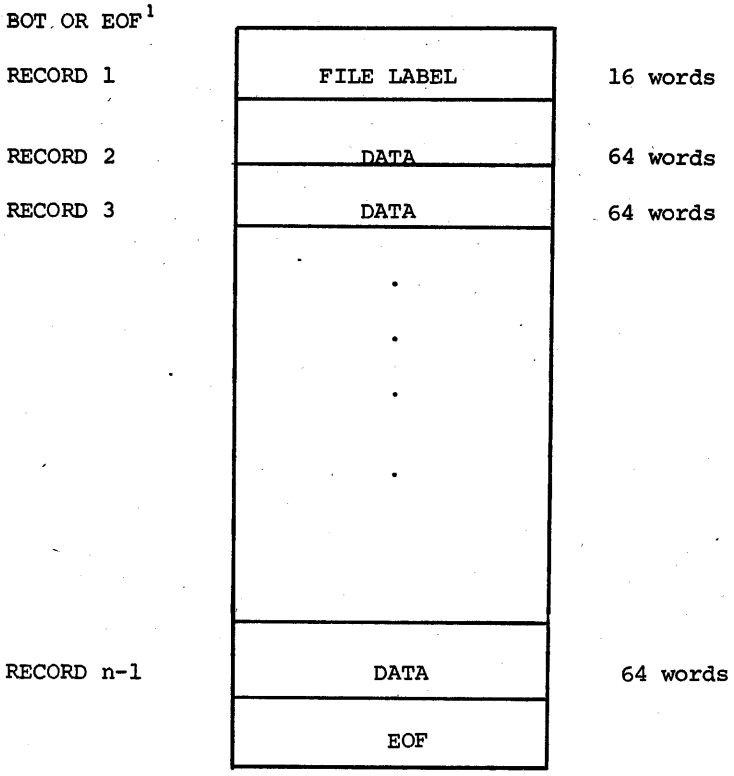

Figure 12-2

Cassette General Format

6.5 READ-AFTER-WRITE VERIFICATION

Throughout the transfer, after writing each record, the cassette is backspaced and the record is read and verified against the contents of the user's buffer. Notification of verification failures are printed on the operator's console as either I1ØØ or FØ76 error diagnostics.

[1]BOT refers to the beginning of tape; EOF refers to the end of the previous file.

PART 12

CHAPTER 7

HELPFUL HINTS

There are some techniques familiar to experienced DOS/BATCH users, but the beginner often has to learn them empirically. This chapter is meant to help the beginner avoid some common pitfalls.

1. If you are having difficulty running PIP, make sure that

   a. You did a KILL command after running the last program prior to running PIP.

   b. The PIP load module is on the disk.

   c. You are accessing the proper UIC field on the DECtape, if attempting to run from DECtape.

2. If you are having difficulty accessing a file that you are positive is there, you are probably logged in under a different UIC than appears on the device. In this case, either

   a. Explicitly specify the UIC in the command.

   b. FINISH and LOGIN under the proper UIC.

   c. ENTER your UIC on the device (See Section 12-4.1).

3. If you are having difficulty transferring to the line printer, make sure the printer is on-line; set the switch to ON LINE.

4. If you are having difficulty reading a file and there has recently been a software or hardware crash, read Section 12-4.5, Recovering Files.

5. If you continue to get file structure errors on output files, you are probably trying to create a file which already exists. The system does not allow this, so delete the file first or use the supersede (/SU) switch.

6. If you are logged in for the very first time and cannot create any files, it is because you have no User File Directory (UFD) in the Master File Directory (MFD). To create one, log in, call PIP, and type

   #SY:/EN

7. If you cannot write anything on a DECtape even though the WRITE switch is on, it is likely that

   a. Your UIC is not entered on the DECtape (see Section 12-4.1).

   b. The tape has not been certified.

   Uncertified DECtapes can be certified using a diagnostic program (TC11 DECtape Formatter MAINDEC-11-YPTB-PB) supplied with the maintenance programs.

8. Use of the /RW:NO option when transferring files from magtape and cassette can save much time normally lost during a rewind operation.

9. If experiencing trouble with transfers from the card reader, check to ensure that (1) control cards are compatible with the card reader type, and (2) end-of-file cards are compatible with the current version of the Monitor.