

Book 6

Utility Programs

Peripheral Interchange Program
(PIP)

Source Compare
(SRCCOM)

Binary Compare
(BINCOM)

DECtape Utility Program
(TENDMP)

File Update Generator
(FUDGE 2)

Cross-Reference Listing
(CREF)

Global Symbol Cross Reference List
(GLOB)

PERIPHERAL INTERCHANGE PROGRAM (PIP)

PIP transfers data files from any standard I/O device to any other standard I/O device and, additionally, performs simple editing and magnetic tape control functions. PIP1, a compact version of PIP, performs a subset of PIP functions. PIP handles all data formats, and eliminates the need for a satellite computer to handle off-line data conversions.

Requirements

PIP	Minimum Core: 3K	Additional Core: 1K if disk is one of the I/O devices; any core above that required is used for extra I/O buffers.
	Equipment Handled:	DECtape, disk, magnetic tape, paper tape reader, paper tape punch, card reader, line printer, and teletype.
PIP1	Minimum Core: 1K	Additional Core: Any core greater than 1K is used for extra input buffers.
	Equipment Handled:	DECtape, disk, magnetic tape, paper tape reader, paper tape punch, card reader, line printer, and teletype.

Initialization

`_R PIP)` or `_R PIP1)`

`*`

Loads PIP (or PIP1) into core.

PIP is ready to receive a command; an asterisk is typed after each requested action has been completed.

Commands

General Command Format

destination-dev:filename.ext ←source-dev:filename.ext , . . .source-n)

destination-dev:
source-dev:

The destination device, to which the data is to be transferred; the source device(s), from which the data is to be read

NOTE

If logical device SYS (the CUSP device) is a DECtape, it must not be modified using the /R or /D switches or any other request requiring it to be initialized for input and output at the same time.

DTAn: (DECtape)
PTR: (paper tape reader)
PTP: (paper tape punch)
DSK: (disk)
CDR: (card reader)
MTAn: (magnetic tape)
LPT: (line printer)
TTY: or (Teletype)
TTYn:

If more than one file is to be transferred from a magnetic tape, card reader, teletype, or paper tape reader, dev: is followed by a comma for each file after the first; these devices can also be followed by * or *.* to indicate all files are to be transferred.

filename.ext (DSK: and DTAn:only)

The filename and filename extension to be assigned to the file on the destination device; the filename and filename extension of the file(s) to be read from the source device.

An asterisk can be used for source files as follows.

filename.* - Transfer all files having the specified filename.
*.ext - Transfer all files having the specified extension.
. - Transfer all files.
* - Transfer all files with null extensions.

The destination descriptors and the source descriptors are separated by the left arrow symbol (+).

Disk File Descriptor Format

DSK:filename.ext [proj,prog] <protection >	
[proj,prog]	Project-programmer number assigned to the disk area to be used, if other than the user's project-programmer number.
<protection >	Protection value to be assigned to the destination file. If omitted, the standard protection is assigned.

NOTE

Standard protection (055) designates that the owner is permitted to read or write, or change the protection of, the file while others are permitted only to read the file.

Standard Assumptions

Unless otherwise changed by switches, all files which are on directory devices and which have a file-name extension of .REL, .SAV, .DMP, or .CHN are copied in binary; all other files are assumed to be in ASCII line mode. Magnetic tape files, unless otherwise changed by switches, are read in odd parity and written in odd parity at 556 bpi.

Examples

<u>Command</u>	<u>Function</u>
_R PIP)	Loads and starts PIP.
*LPT:+DTA1:FILE1)	Transfer the file named FILE1 from DTA1 to the line printer.
LPT:+DTA1:)	Transfer all files with null extensions from DTA1 to the line printer.
*DTA2:FILE2+DTA1:FILE1.TMP)	Transfer the file named FILE1.TMP to DTA2 and give it the name FILE2.

Command	Function
*DTA2:FILE3←DTA1:FILE1,FILE2)	Transfer the files named FILE1 and FILE2 from DTA1 to DTA2, combining them as one file under the name FILE3.
*DTA2:FILE3←DTA1:FILE1,DTA3:FILE2)	Transfer the file named FILE1 from DTA1 and the file named FILE2 from DTA3 to DTA2, combining them as one file under the name FILE3.
*DSK:FILE1←MTA1:)	Transfer the next file from the present position of MTA1 to the user's area on the disk, call it FILE1, and assign the standard protection of 055.
DSK:FILE1<177>←MTA1:)	Transfer all files from MTA1 (starting at the current position of the read head) to the user's area on the disk, combining them into one file called FILE1, and assign protection 177.
*DSK:FILE1[1,3]←MTA1:)	Transfer the next two files from the present position of MTA1 to area 1,3 on the disk, combining them into one file called FILE1, and assign the standard protection (055).
*PTP:←PTR:,,,,)	Transfer five files from the paper tape reader, combining them as one file on the paper tape punch.
*+C)	Return to Monitor.
.	Dot indicates that user is at Monitor level.

Switches

Nonmagnetic-tape switches, when used, are preceded by a slash (if more than one is specified, they may be enclosed by parentheses instead) and can appear anywhere in the command string; however, if the command string contains commas, the switches must be specified prior to the first comma.

Magnetic tape switches are enclosed by parentheses and must appear immediately following the device or file to which they refer.

Switches are used to specify:

- Particular files for transferral or deletion;
- Editing;
- Mode of transfer;
- Directory manipulation (DECtape and DSK); and
- Magnetic tape control.

A listing of PIP switches can be obtained by typing

```
⌘output-dev:/Q+ ⌘
```

where output-dev: may be either LPT: or TTY:

PIP Switch Options

Switch	Meaning
A	Line blocking
B	Process file in Binary mode.
C	Suppress trailing spaces and Convert multiple spaces to tabs.
D	Delete the file.
E	Treat Ending (card) columns 73 through 80 as spaces.
F	List the directory in short form for DSK: or DTAn: only. (Filenames and extensions only.) Useful when disk directory cannot be listed with /L sw.
G	Ignore I/O errors.
H	Process file in image binary mode.
I	Process file in Image mode.
L	List the directory (DSK: or DTAn: only)
M	Magnetic tape switches. A string of one or more magnetic tape switches begins with an M and is enclosed in parentheses.
#nA	Advance the tape n files.
#nB	Backspace the tape n files.
#nD	Advance the tape n records.
#nP	Backspace the tape n records.
2	200 bpi density.
5	556 bpi density.
8	800 bpi density.
A	Advance tape one file.
B	Backspace tape one file.
E	Even parity.
F	Mark End of File.
T	Skip to logical end of Tape.
U	Rewind tape and Unload.
W	Rewind the tape.

NOTE

MTA switches always apply to the device or file immediately preceding the switches. MTA switches should be used only in specific situations. For a more detailed treatment, see PIP Programmer's Reference Manual.

Examples

```

_R PIP )
_DSK:/X+DTA1:*** )
_DSK:(DX)+DTA1:FILE1,*.REL )

_MTA2:/S+CDR: )

_LPT:/P+DTA1,FILE1 )

_DTA2:FILE1/I+PTR: )

_TTY:/L+DTA1: )

[nnnn FREE BLOCKS LEFT )
filename.ext no. blocks creation date
:
_DTA1:/Z+ )
_MTA2:(M8E)+MTA1:(ME8) )
_MTA2:(MW)+ )
_LPT:+MTA1:(M2W),(MA),, )

_MTA1:(M#4A)+CDR: )

_*C )
:

```

Load and run PIP.

Transfer all files from DTA1 to DSK, keeping them separate and retaining their filenames.

Transfer all files, except FILE1 and any files with the extension .REL, from DTA1 to DSK, keeping them separate and retaining their filenames.

Transfer a file from the card reader to MTA2 and add sequence numbers.

Take FILE1 (a FORTRAN output print file), interpret the carriage control characters, and print the file using specified carriage control.

Initialize both DTA2 and the paper tape reader in image mode and transfer one file from the paper tape reader to DTA2, calling it FILE1.

List the directory of DTA1 on the teletype.

Zero the directory of DTA1.

Transfer a file from MTA1 to MTA2 in 800 bpi, even parity mode.

Rewind MTA2.

Set MTA1 to 200 bpi, odd parity, rewind the tape, and transfer the first, third, fourth, and fifth files to the printer.

Advance MTA1 four files before transferring a file from the card reader.

Return to the Monitor.

Diagnostic Messages

PIP Diagnostic Messages

Message	Type	Meaning
?4K NEEDED	S	4K is not currently available but is needed when a disk is present in the system.
?DECTAPE I/O ONLY	S	I/O device for copy block 0.(/U) must be a DECTape.
?DEVICE dev:DOES NOT EXIST	I/O	Either device name has been misspelled or there is no such device.
?DEVICE dev:NOT AVAILABLE	I/O	The device has been assigned to another job.
?DIRECTORY FULL	FR	There is no room for an entry in a DECTape directory.
DISK DIRECTORY READ	I/O	This message is nonfatal if the /G switch is used; otherwise, it is fatal and is preceded by a ?. A second message follows (see Table 6-3).
?DISK OR DECTAPE INPUT REQUIRED	I/O	This command requires a directory device for input.
?DTA TO PTP ONLY	RIM	DTA input and PTP output must be specified for /Y.
FAILURE DURING (/X,/Z,/D,/R) REQUEST	S	Each file requested does exist, but one or more was unavailable for processing. This message is never fatal.
?FILE filename.ext ILLEGAL EXTENSION	RIM	Extension for /Y request must be .RMT, .RTB, or .SAV.
?FILE filename.ext ILLEGAL FORMAT	RIM	1. Zero-length file; or 2. Requisite job data info not available; or 3. Block overlaps previous block (RIM 10) or 4. EOF found when data was expected, or 5. A pointer word was expected but not found in the source file.
?filename.ext (3) FILE WAS BEING MODIFIED	FR	Disk file named is currently being processed by another job.
?filename.ext (0) FILE WAS NOT FOUND	FR	Filename.ext not found during LOOKUP.

PIP Diagnostic Messages

Message	Type	Meaning
?filename.ext (0) ILLEGAL FILE NAME	FR	Indicates that 1. No filename was specified for DTA output file; or 2. A reject occurred on a /R request for disk file; or 3. Illegal filename was specified for a /R request on DTA.
?filename.ext (1) NO SUCH PROJECT-PROGRAMMER NUMBER	FR	The project-programmer number specified for a DSK file is incorrect.
INPUT DEVICE dev: FILE filename.ext	I/O	This message is nonfatal if the /G switch is used; otherwise, it is fatal and is preceded by a ?. A second message follows (see Table 6-3).
?LINE TOO LONG	S	A line >140 characters was detected in the source file.
?LOAD POINT BEFORE END OF (MB) OR (MP) REQUEST	S	Load point on a magnetic tape file has been reached before the tape has been backspaced the number of files or records specified in (M#nB), (M#nP).
?NO BLOCK 0 COPY	C	/U given but PIP assembled without provision for this.
?NO FILE NAMED filename.ext	FR	No such file found during PIP directory search.
?NO FILE NAMED QPIP	S	The data file for the /Q switch is not available.
OUTPUT DEVICE dev: FILE filename.ext	I/O	Followed by a second message (see Table 6-3).
?PIP COMMAND ERROR	C	1. Illegal format for command string; or 2. Nonexistent switch requested; or 3. Filename.ext other than * (or *.*) requested for a non-directory device; or 4. The illegal switch combination RX.
?filename.ext (2) PROTECTION FAILURE	FR	Same as FAILURE DURING ... message except that the processing halts.
?filename.ext (4) RENAME FILENAME ALREADY EXISTS	FR	Tried to rename file with already existing name.
?filename.ext (5) RENAME ERROR	FR	LOOKUP or ENTER not done.
?filename.ext (6)	I/O	Error not yet defined.
?filename.ext (7)	I/O	Error not yet defined.

PIP Diagnostic Messages

Message	Type	Meaning
?TERMINATE /X MAX of 999 FILES PROCESSED	S	The /X switch specified for nondirectory device source files has processed the maximum number of files (999).
?TOO MANY REQUESTS FOR dev:	C	Conflicting parity/density requests have been given for a magnetic tape.
?TRY PIP		During a PIP1 run, a switch or function which is not present in PIP1 has been requested.
NOTES		
All fatal diagnostic messages are preceded by a question mark (?).		
Message types are:		
C	Command string error	
FR	File reference error	
I/O	I/O error	
RIM	Readin Mode specification error	
S	Other types of errors.	

Table 6-3
Secondary PIP I/O Diagnostic Messages

Message	Device	Meaning
BINARY DATA INCOMPLETE	PTR	Length of block disagrees with word count (nonfatal if the /G switch has been specified).
BLOCK TOO LARGE	DTA	DTA link number >1101.
CHECKSUM OR PARITY ERROR	All	Read or write error (nonfatal if the /G switch has been specified).
INPUT BUFFER OVERFLOW	All except DTA	Block too large for buffer (nonfatal if the /G switch has been specified).
DEVICE ERROR	All	The data control unit has detected the loss of data (nonfatal if the /G switch has been specified).

Secondary PIP I/O Diagnostic Messages

Message	Device	Meaning
PHYSICAL EOT	MTA	The end of tape has been reached (nonfatal if the /G switch has been specified).
WRITE (LOCK) ERROR	DTA, DSK, MTA	Attempt has been made to write on a write-locked file.
7-9 PUNCH MISSING	CDR	Binary card lacks 7-9 punch (nonfatal if the /G switch has been specified).

1K Version of PIP (PIP1) Limitations

The following limitations apply to PIP1:

- a. Z and MW requests ignore all source devices.
- b. B switch included since REL, SAV, DMP, and CHN files are not automatically copied in 36-bit bytes.
- c. Error messages assume all I/O devices are DECtape.
- d. Neither project-programmer numbers nor protection can be specified for disk files.
- e. The * cannot be used for filenames or extensions.
- f. SAV files cannot be successfully copied with PIP1.

Monitor Commands

The following Monitor commands perform PIP-type operations.

<u>Desired Result</u>	<u>Monitor Command</u>	<u>Equivalent CUSP Commands</u>
To type the contents of a file on the TTY.	_TYPE dev:filename.ext)	_R PIP) *TTY: +dev:filename.ext)
To list the contents of a file on the line printer.	_LIST dev:filename.ext)	_R PIP) ±LPT: +dev:filename.ext)
To type the directory of a device on the TTY.	_DIRECT dev:)	_R PIP) *TTY: +dev:/L)
To list the directory of a device on the line printer.	_DIRECT dev:/L)	_R PIP) ±LPT: +dev:/L)

<u>Desired Result</u>	<u>Monitor Command</u>	<u>Equivalent CUSP Commands</u>
To delete a file.	<code>_DELETE dev:filename.ext</code>	<code>_R PIP)</code> <code>*dev:filename.ext/D+)</code>
To rename a file.	<code>_RENAME dev:newfn =oldfn</code>	<code>_R PIP)</code> <code>*dev:newfn/R+oldfn)</code>

NOTE

If dev: is omitted in the Monitor commands, DSK: is assumed.

FILE UPDATE GENERATOR (FUDGE2)

FUDGE2 updates files containing one or more relocatable binary programs, and permits the user to manipulate individual programs within program files.

Requirements

Minimum Core: 2K

Additional Core: Dynamically allocates its buffers to utilize as much core as is made available.

Equipment: Two input devices, one for the master file and one for the transaction file; one output device for the updated file. The input device(s) and output device can be the same device (DSK:). The two input devices can be the same DECTape.

Initialization

._R FUDGE2)

_

Loads the File Update Generator program.

FUDGE2 is ready to receive a command.

Commands

General Command Format

new-dev:filename.ext ← master-dev:filename.ext <programe1,programe2,..programe > ,)

transaction-dev:filename.ext <programea,programeb,..programez > (commands) \$

new-dev: The destination device, on which the updated file is written.

DTAn:

DSK:

MTAn:

PTP:

master-dev: The device containing the file to be updated.

DTAn:

DSK:

MTAn:

PTR:

NOTE

If more than one file is to be transferred from a magnetic tape or paper tape reader, dev: is followed by a colon (:) for each file after the first.

transaction-dev: The device containing the file of programs to be used in the updating process.

DTAn:

DSK:

MTAn:

PTR:

NOTE

If more than one file is to be transferred from a magnetic tape or paper tape reader, dev: is followed by a colon (:) for each file after the first.

More than one transaction device, with its associated filenames and program names, can be specified in certain instances (see Switches).

filename.ext (DSK: and DTAn: only) The filename.ext of the new, updated version of the program file.

The filename.ext of the program file containing the programs to be deleted, replaced, or augmented.

<programe,.....> (DSK: and DTAn: only)

The filename.ext of the program file containing the programs to be used in performing additions or replacements to the master file.

If no .ext is given, .REL is assumed.

Program names must be specified in the same relative order in which they appear in the file.

Program names are grouped together within angle brackets <> and are separated by commas.

If it is desired to append, replace, insert, or extract all programs within a file, only the filename.ext need be specified.

Program names cannot be specified for the output file.

The new output file is separated from the master and transaction files by the left arrow symbol (←).

Command Codes

The function to be performed by FUDGE2 is selected by including one of the following command codes at the end of the command string. Command codes are enclosed within parentheses (or preceded by a slash) and one (and only one) must appear in every command string.

FUDGE2 Command Codes

Command	Meaning
A	Append one or more programs from the transaction file(s) to the master file and write out the new file. The command string is as follows: new-file ← master-file,transaction-file,.....(A) \$
D	Delete one or more programs from the master file and write out the new file. The files (and programs) to be deleted are listed after master-dev:.. The command string is as follows: new-file ← master-file<file(s) to be deleted>(D) \$
E	Extract the specified files (and programs) from one or more input files and create a new output file. If program names are not specified for a file, the entire file is extracted. The command string is as follows: new-file ← masterfile<file(s) to be extracted>(E) \$

FUDGE2 Command Codes

Command	Meaning
I	Insert programs from one or more transaction files onto the master file and write out the new file. The programs from the transaction file(s) are inserted immediately before the specified programs on the master file. The command string is as follows: new-file ← master-file<file(s) to be inserted before>, transaction-file(s) (I) \$
L	List all relocatable programs within a file and print the listing on the output device, which must be either TTY: or LPT: The command string is as follows: listing-device ← file(L) \$
R	Replace the named program(s) on the master file with the named program(s) from the transaction file, and write out the new file. The command string is as follows: new-file ← master-file<file(s) to be replaced>, transaction-file<replacement file(s)>(R) \$
NOTE	
Only one operation can be specified per command string. Thus, to delete a file and replace some other one, two command strings are required.	

Examples

```
.R FUDGE2)
*LPT:←DTA1:LIB40(L) $)
```

List all relocatable programs (.REL) from the file LIB40, located on DTA1 on the line printer.

```
*DTA2:LIB4AA ←DTA1:LIB40<EXP.2>(D)$)
```

Delete the program EXP.2 from the file LIB40 on DTA1; write the new file on DTA2 and call it LIB4AA.REL.

```
*DSK:LIB4BB←D7A2:LIB4AA <EXP.3,EXP.3C>,)
D7A1:F1<EXP.3A,EXP.3B>(R) $)
```

Replace programs EXP.3 and EXP.3C located in file LIB4AA on D7A2, with programs EXP.3A and EXP.3B in File F4 on D7A9; write out the new LIB4AA file on disk and call it LIB4BB.

```
*PTP:←DSK:LIB4BB,DTA4:SCIENC<COSRTE>/A $)
```

Append the program COSRTE, located in file SCIENC on DTA4, to the file LIB4BB on disk; write out the updated LIB4BB file on the paper tape punch.

```
*DTA1:NFILE+DSK:MFILE<M1,M2,M3,M4> )
```

```
DTA3:TFILEA<TA1,TA2> )
DTA4:TFILEB<TB1,TB2>/I $ )
```

```
*DTA1:NFILE+DSK:MFILE<M1,M2,M3,M4> )
DTA3:TFILEA )
DTA4:TFILEB/I $ )
```

```
*+C )
```

Insert into MFILE the programs TA1 and TA2 from TFILEA and TB1 and TB2 from TFILEB. Create NFILE with the following order:
TA1,M1,TA2,M2,TB1,M3,TB2,M4

Insertion is on a one-to-one basis. If there are more programs to be inserted than specified programs before which they are to be inserted, the extra files are ignored.

However, in this example (where TFILEA and TFILEB contain the programs TA1 and TA2 and TB1 and TB2, respectively) create an NFILE with the following order:
TA1,TA2,M1,TB1,TB2,M2,M3,M4

Return to the Monitor.

Switches

Switches are used to manipulate file directories and to position magnetic tape. They are either preceded by a slash or enclosed in parentheses and can appear anywhere in the command string.

FUDGE2 Switch Options

Switch	Meaning
B	Backspace magnetic tape one file.
K	Advance magnetic tape one file.
W	Rewind magnetic tape.
Z	Clear directory of destination device (DTAn: only).

Examples

```
*R FUDGE2 )
*DTA2:TESTA+MTA1:(WK),MTA2::(ZA) $ )
```

```
*+C )
```

Clear the directory of DTA2; rewind MTA1 and advance the tape one file; append the first two program files from MTA2 to the second file on MTA1 and write out the resultant file on disk, calling it TESTA.

Return to the Monitor.

Diagnostic Messages

FUDGE2 Diagnostic Messages

Message	Meaning
?CANNOT DO I/O AS REQUESTED	Input cannot be performed on one of the devices specified for input (it is an output only device) or output cannot be performed on the device specified for output.
?DEVICE ERROR ON OUTPUT DEVICE	A write error has occurred on the output file.
?DIRECTORY FULL ON OUTPUT DEVICE	No more files can be added to the file directory on the output device (the directory is full).
?ENTRY BLOCK TOO LARGE, PROGRAM xxxxxx	The entry block of program xxxxxx is too large for the FUDGE2 entry table, which allows for 32 entry names. FUDGE2 can be reassembled with a larger table.
?FUDGE SYNTAX ERROR	The command string is illegal (e.g., the left arrow was omitted, a program name was specified for the output file, or some meaningless command was entered).
?x IS AN ILLEGAL CHARACTER	An illegal character has been encountered in the command string.
?x IS AN ILLEGAL SWITCH	An illegal or otherwise meaningless switch has been encountered in the command string.
?dev NOT AVAILABLE	The device either does not exist or has been assigned to another job.
?NOT ENOUGH ARGUMENTS	An insufficient number of files of one type or another has been specified.
?dev filename.ext progname NOT FOUND	Either the filename.ext or the program name was not found on the device (or in the file) specified. If a program name is printed, this may indicate that the program names in the command string appear in a sequence different from their sequence within the file; thus, the program may actually exist in the named file but was missed because of the incorrectly entered sequence in the command string.
?PROGRAM ERROR WHILE RESETTNG MASTER DEVICE	Either FUDGE2 cannot find the master device or cannot find the program name on the master device.

FUDGE2 Diagnostic Messages

Message	Meaning
?TOO MANY FILE NAMES OR PROGRAM NAMES	More than 40 program names or file names were given in a command string. Break the job into several segments and rerun.
?TRANSMISSION ERROR ON INPUT DEVICE dev	A transmission error has occurred while reading data from device dev.
?UNEQUAL NUMBER OF MASTER AND TRANSACTION PROGRAMS	An unequal number of master and transaction programs (or files) has been specified with a Replace request.

CROSS-REFERENCE LISTING (CREF) (VERSION CREF.V32 AND LATER)

CREF produces a sequence-numbered assembly listing followed by one to three tables, one showing cross references for all operand-type symbols (labels, assignments, etc.), another showing cross references for all user defined operators (macro calls, OPDEFs etc.), and another (if the proper switch is specified) showing the cross references for all op codes and pseudo-op codes (MOVE,XALL, etc.). A number sign (#) appears on the definition line of all symbols. The input to CREF is a modified assembly listing file created during a Macro-10 assembly or FORTRAN IV compilation when the /C switch is specified in the command string.

CREF provides an invaluable aid for program debugging and modification.

Requirements

Minimum Core: 2K
Additional Core: Takes advantage of any additional core available, as necessary.
Equipment: One input device (normally disk) which contains the modified assembly listing file; one output device (normally the line printer) for the listing.

Initialization

_R CREF)	Loads the Cross-Reference Listing program into core.
*	The program is ready to receive a command.
-	

Commands

General Command Format

output-dev: ← input-dev:filename.ext

output-dev: The device on which the assembly listing and cross-reference tables are to be printed (LPT: is assumed if device is not specified).

input-dev: The device on which the modified assembly listing was written during Macro-10 assembly (DSK: is assumed if device is not specified).

filename.ext (DSK: or DTAn:only) The filename and filename extension of the modified assembly listing file (CREF.LST is assumed if filename.ext is not specified).

← The output device and the input device are separated by the left arrow symbol.

Disk File Command Format

DSK:filename.ext [proj,prog]

[proj,prog] Project-programmer number assigned to the disk area to be searched for the source file if other than the user's project-programmer number.

Examples

```
._R MACRO )
```

Loads the Macro-10 Assembler into core.

```
*PTP: ,/C+DTA1:TXCALC )
THERE ARE NO ERRORS )
```

Assembles the program TXCALC from DTA1; writes the object program coding on the paper tape punch; writes a modified assembly listing on DSK: (assumed) and assigns it the filename CREF.LST.

```
PROGRAM BREAK IS 003771 )
```

```
7K CORE USED )
```

```
*rC )
```

Return to the Monitor.

```
._R CREF )
```

Loads CREF into core.

```
* )
```

Selects the default assumptions of:

```
output-dev:  LPT:
input-dev:   DSK:
filename.ext CREF.LST
```

```
*rC )
```

Return to the Monitor.

```
.
```

Switches

Switches are used to specify such options as magnetic tape control and list selection. All switches are preceded by a slash (/).

CREF Switch Options

Switch	Meaning
A	Advance magnetic tape reel by one file. /A may be repeated.
B	Backspace magnetic tape reel by one file. /B may be repeated.
K	Kill listing of references to basic symbols (labels, assignments, etc.).
M	Suppress listing of references to user-defined operators (Macro calls, OPDEFs, etc.).
O	Allow listing of references to machine and pseudo-operation codes (MOVE, XALL, etc.).
R	Requests (by typing out RESTART LISTING AT LINE:) the line number at which the listing is to Restart. (Such action might be necessary if the line printer ran out of paper, or jammed, etc.) The user types the line number followed by a carriage return.
S	Suppress program listing (list only the selected tables).

CREF Switch Options

Switch	Meaning
T	Skip to logical end of magnetic Tape.
W	ReWind magnetic tape.
Z	Zero the DECtape directory (DECtape must be output only).

Examples

```
.R CREF )
```

Loads CREF into core.

```
*/M+MTA1:/W )
```

Rewind MTA1 and process the first file, listing only the cross references for operand-type symbols (labels, assignments, etc.).

```
*DTA5:SAVE1/Z+ )
```

Process the file named CREF.LST in the user's area of disk; write the program listing and operand-type cross references on DTA5 and call the file SAVE1.

```
*+C )
```

Return to Monitor

```
.
```

Diagnostic Messages

CREF Diagnostic Messages

Message	Meaning
?dev NOT AVAILABLE	Device is assigned to another job.
?CANNOT ENTER FILE fme _l .ext	DTA or DSK directory is full; file cannot be entered.
?CANNOT FIND FILE fme _l .ext	The file cannot be found on the device specified.
?COMMAND ERROR	Error in last command string entered.
?DATA ERROR DEVICE dev:	READ or WRITE error.
?ERROR READING COMMAND FILE	Disk data error while reading nnnCRE.TMP (see below).
?IMPROPER INPUT DATA	Input data not in CREF format.
?INPUT ERROR ON DEVOCE dev:	READ error has occurred on the device.
?INSUFFICIENT CORE	Additional core is required for execution but none is available from Monitor.

Monitor Commands

CREF-format listing files generated by COMPILE, LOAD, EXECUTE, and DEBUG commands (using the /CREF switch) can be printed on the line printer by typing

```
._CREF )
```

The CREF command will print out all listing files that are specified in the CCL command file, nnn CRE.TMP (where nnn is the user's job number). After completion of this operation, nnn CRE.TMP is deleted to preclude the listing files being listed again by the next CREF command.

GLOBAL SYMBOL CROSS REFERENCE LIST (GLOB) VERSION #002 OR LATER

GLOB reads multiple binary program files produced by Macro and F40 and generates an alphabetic cross-referenced list of all global symbols encountered.

Requirements

Minimum Core: 2K
 Additional Core: Requests additional core from the Monitor as required.
 Equipment: An input device for each binary file to be scanned for global symbols and one or more listing devices for output.

Initialization

.R GLOB)	Loads the Global Cross-Reference Listing program.
*	The program is ready to receive a command.
-	

Commands

Input Command

dev:filename.ext,...filename.ext,dev:filename.ext,...filename.ext,...)

dev:	The device(s) containing the binary program files to be scanned.
MTAn:	(magnetic tape)
DTAn:	(DECtape)
DSK:	(disk)
PTR:	(paper tape reader)

filename.ext (DSK: and DTA: only)

The filename and filename extension of each binary program which resides on either disk or DECTape.

Output Command

dev: +\$

dev:

The device on which the global symbol listing is to be printed.

LPT: (line printer)

TTY: (Teletype)

Other output devices can be specified if desired.

More than one output command can be given if it is desired to produce several types of listings on several different devices. Each new output command is typed after the previous request has been completed.

Examples

.R GLOB

*DSK:F1,F2,DTA3:CALC1,CALC5

*LPT: + \$

*+C

:

The binary program files to be scanned are F1 and F2 on DSK, and CALC1 and CALC5 on DTA3.

All global symbols in these programs are to be listed on the printer. Printed with each symbol are its value, the name of the program in which it was defined, and the names of all the programs in which it was referenced (i.e., declared external).

Return to the Monitor.

Switches

The switches available in GLOB are used to determine the types of global symbols to be listed on each of the specified output devices. If no switches are typed, all global symbols are listed. There are also three separate switches (L, M, and X) which act independently.

All switches are either preceded by a slash or enclosed in parentheses and can appear anywhere in the output command string. However, only the most recently specified switch (except L, M, or X, which always take effect) is in effect at any given time.

GLOB Switch Options

Switch	Meaning
A	All global symbols are to be listed (assume if no switch is given).
E	List erroneous (multiply defined or undefined) symbols only.
F	List fixed (nonrelocatable) symbols only.
L	Turn on Library Search Mode (that is, only scan programs if they contain globals previously defined and not yet satisfied).
M	Turn off Library Search Mode.
N	List only those symbols which are never referred to.
R	List relocatable symbols only.
S	List multiply specified (i.e., symbols defined in more than one program, but with non-conflicting values) only.
X	Omit printing of listing title when output is other than TTY. Include printing of listing title when output is TTY.

NOTE

Normally, the title is printed on all devices except the Teletype.

Examples

```

.R GLOB )
*DTA1:TEST1.REL,SUBRTE,DSK:ARITH1, )
*SCIENC,RETEST )

```

The binary programs to be scanned are files TEST1.REL and SUBRTE on DTA1, and ARITH1, SCIENC, and RETEST on disk.

```

*LPT:+/R $ )
*TTY:+/E $ )

```

List only relocatable symbols on the printer.
Printer listing is completed. Enter command to print all erroneous symbols on the Teletype.

```

U EXTSYM          SUBRTE

```

(U = Undefined; EXTSYM is the undefined symbol; SUBRTE is the program in which EXTSYM appears.)

```

*+C )

```

Return to the Monitor.

Diagnostic Messages

GLOB Diagnostic Messages

Message	Meaning
?COMMAND SYNTAX ERROR	An illegal command string has been entered.
?DESTINATION DEVICE ERROR	An I/O error has occurred on the output device.
?DIRECTORY FULL	No more files can be added to the directory of the output device.
?dev NOT AVAILABLE	The device either does not exist or has been assigned to another job.
?filename.ext NOT FOUND	The filename.ext cannot be found in the directory on the device specified.
?TABLE OVERFLOW - CORE UUU FAILED TRYING TO EXPAND TO xxx	GLOB requested additional core from the Monitor, but none was available.

GLOB Error Flags

Flag	Meaning
M	Multiply defined symbol (all values are shown).
N	Never referred to (i.e., was not declared external in any of the binary programs).
S	Multiply specified symbol (i.e., defined in more than one program, but with non-conflicting values). In the listing, the name of the first program in which the symbol was found is followed by a plus sign.
U	Undefined symbol.

SOURCE COMPARE (SRCCOM) (VERSION 013)

SRCCOM compares, line by line, two versions of a source file coded as lines of ASCII characters and outputs any differences.

Requirements

Minimum Core: 2K

Additional Core: The minimum core allows for comparing files with minimal differences. SRCCOM automatically requests more core from the Monitor when it needs it. Major differences can usually be handled in 3K, but for comparing two completely different files, enough core is required to store all of both files simultaneously.

Equipment: *User teletype for control; two input devices for the two files to be compared; one output device for listing the differences.

Initialization

_R SRCCOM)

Loads the Source Compare routine.

*

Source Compare is ready to receive a command.

Commands

General Command Format

`list-dev:filename.ext ←input 1-dev:filename.ext, input2-dev:filename.ext`

list-dev: The device on which the differences are to be listed.

LPT:	(line printer)	(Any device that can output ASCII characters)
TTY:	(teletype)	
MTAn:	(magnetic tape)	
DTAn:	(DECtape)	
DSK:	(disk)	

input -dev: The devices on which the two source files to be compared are located.

MTAn:	(magnetic tape)	(Any device that can input ASCII characters)
DTAn:	(DECtape)	
DSK:	(disk)	
PTR:	(paper tape reader)	

filename.ext(DSK: and DTAn: only)

The filename and extension of either of the input source files.

The filename and extension to be assigned to the output list file. (SRCCOM.LST is assumed if no filename is specified.)

The output device is separated from the input source file devices by the left arrow symbol.

Default Conditions

TTY: is assumed as the output device if no other device is specified.

DSK: is assumed as both input devices if no other devices are specified.

A dot is necessary in filename #2 to explicitly indicate a null extension if the extension for filename #1 is not null.

Example:

LPT: ←DRAn:FORSE.MAC,DSK:FORSE.(FORSE has a null extension.)

The filename and extension for input file #1 is assumed to be the filename and extension for filename #2 unless another filename or extension is specified.

The filename and extension for the output file is assumed to be SRCCOM.LST unless another filename or extension is specified.

Switches

Switches are used to specify the manner in which the comparison is to be done. All switches consist of a single character preceded by a slash (/), anywhere in the command string.

Source Compare Switches

Switch	Meaning
/B	Enables the comparing of Blank lines. Normally blank lines are completely ignored.
/C	Comments (all text on a line after a semicolon) are ignored. /C will not cause a line consisting entirely of a comment to become a blank line which will be ignored. /S is also implied.
/S	Spacing (spaces and tabs) is ignored.
/n	(n=1, 2, ..., 9) A match consists of n lines. (n is normally 3) n successive lines must be found identical in the two input files for a match in the two files to be found. When a match is found, all differences between the current match and the previous match are listed. The first line of the match is also listed to make the location in the file easier to find.

Examples

```
*R SRCCOM )
*LPT: +DTA2:SOURCE.001,DTA3:
SOURCE.002 )
```

Compare the source file SOURCE.001 on DTA2 with the source file SOURCE.002 on DTA3 and list all differences on the line printer.

```
*LPT: +DSK:TRY1,DSK:TRY2 )
```

Compare the two files, TRY1 and TRY2, both of which are on the disk, and list the differences on the printer.

```
*+C )
```

Return to the Monitor.

Example of Source Compare Output

SRCCOM output	File 1 input	File 2 input
	page 1	page 1
FILE 1) FILE #1	FILE #1	FILE #2
FILE 2) FILE #2		
1)1† FILE #1	A	A
††1) A	B	B
****	C	C
2)1† FILE #2	D	G
† 2) A	E	H
*****	F	I
1)1† D	G	J
1) E	H	1
1) F	I	2
1) G	J	3
****	K	
2)1† G	L	
*****	M	
1)1† K		
1) L		
1) M		
1)2† N		
****	page 2	page 2
2)1† 1	N	N
2) 2	O	O
2) 3	P	P
2)2† N	Q	Q
*****	R	R
1)2† W	S	S
****	T	T
2)2† 4	U	U
2) 5	V	V
2) W	W	4
*****	X	5
	Y	W
	Z	X
		Y
		Z

†These numbers in the SRCCOM listing are page numbers referring to the input files.

††A line identical to both input files is listed to help find the location of the differences within the two files.

Diagnostic Messages

Source Compare Diagnostic Messages

Message	Meaning
?2K CORE NEEDED AND NOT AVAILABLE	SRCCOM needs 2K to initialize IO devices and the core is not available from the Monitor
?BUFFER CAPACITY EXCEEDED AND NO CORE AVAILABLE	The buffer is not large enough to handle the number of lines required for looking ahead and no more core is available from the Monitor.
?COMMAND ERROR	Error in last command string entered.
?DEVICE dev:NOT AVAILABLE	One of the input devices cannot be initialized; generally, the device either does not exist or has been assigned to another job.
?FILE 1 READ ERROR	An error has occurred on the first input device specified in the command.
?FILE 2 READ ERROR	An error has occurred on the second input device specified in the command.
?INPUT ERROR - filename .ext FILE NOT FOUND	The specified filename cannot be found.
?NO DIFFERENCES ENCOUNTERED	No differences were found between the two source files.
?OUTPUT DEVICE ERROR	An error has occurred on the output device.
?OUTPUT INITIALIZATION ERROR	The output device cannot be initialized; the device either does not exist or has been assigned to another job, the device is not an output device or the filename could not be entered on the device.

BINARY COMPARE (BINCOM)

BINCOM compares, word by word, two versions of a binary (.REL) program file and outputs any differences.

Requirements

Minimum Core: 1K if output device is other than DTAn:, MTAn:, or DSK;; otherwise 2K.

Additional Core: See Minimum Core.

Equipment: Two input devices for the two files to be compared; one output device for listing the differences. Both input files can be on disk.

Initialization

```

_R BINCOM )      Loads the Binary Compare routine.
*
-               Binary Compare is ready to receive
                a command.

```

Commands

General Command Format

```
list-dev:filename.ext ←input1-dev:filename.ext , input2-dev:filename.ext )
```

list-dev: The device on which the differences are to be listed.

LPT: (line printer)
 TTY: (teletype)
 MTAn: (magnetic tape)
 DTAn: (DECtape)
 DSK: (disk)

If list-dev:filename.ext ←is omitted, TTY: is assumed.

input1-dev: The devices on which the two binary files to be compared are located

DTAn: (DECtape)
 DSK: (disk)
 CDR: (card reader)
 PTR: (paper tape reader)
 MTAn: (magnetic tape)

filename.ext (DSK: and DTAn: only)

The filename and extension of either of the input binary files.

The filename and extension to be assigned to the output list file.

CommandFunction

NOTE

If .ext is omitted, null extension is assumed.

The output device is separated from the input binary file devices by the left arrow symbol.

Examples

```
.R BINCOM )
```

```
*LPT:-DSK:PROG1.REL,  
DTA1:PROG1.REL )
```

Compare the binary program file PROG1.REL in the user's area of the disk with a binary program file, PROG1.REL, on DTA1, and list all differences on the line printer.

```
NO ERRORS ENCOUNTERED )
```

No differences were found between the two files.

```
DTA1:BINA,DTA2:BINB )
```

Compare the binary program file BINA on DTA1 with the binary program file BINB on DTA2 and list all differences on the teletype.

```
[ loc file1-word file2-word XOR  
  :           :           :
```

NOTE

.REL is assumed as the extension name for both BINA and BINB.

```
*+C )
```

Return to the Monitor.

Diagnostic Messages

Binary Compare Diagnostic Messages

Message	Meaning
?COMMAND ERROR	Error in last command string entered.
?DEVICE dev: NOT AVAILABLE	Device has been assigned to another job or does not exist.
?END OF FILE PHASE ERROR	One input file is longer than the other.

Binary Compare Diagnostic Messages

Message	Meaning
?FILES BEING COMPARED ON SAME INPUT DEVICE	Files cannot be compared from the same input device unless that device is DSK:.
?INPUT ERROR filename.ext NOT FOUND	The file specified could not be found on the input device.
NO ERRORS ENCOUNTERED	No differences were found between the two binary program files.
?OUTPUT INITIALIZATION ERROR	The file cannot be entered.

Error Differences

Whenever a difference is encountered between the two files being compared, a line is printed on the listing device in the following format:

```
octal loc.      file1-word      file2-word      XOR of both words
```

TENDMP is a utility program, used to save and restore core images on DECtape. It is compatible with the PDP-10 time-sharing system's directory-structured DECtape format, and with the format of a SAVE file.

TENDMP operates in executive mode only, and will run on a PDP-6 or a PDP-10 computer with either a TD-10 or 551/136 DECtape system. (If a PDP-10 is used, the KE-10 Extended Order Code option is required.)

1. TENDMP FUNCTIONS

TENDMP has the following functions:

- 1) Selection of DECtape unit.
- 2) Listing of directory of DECtape.
- 3) Loading a program into core.
- 4) Zeroing of directory.
- 5) Merging a program from tape into core, leaving other areas untouched.
- 6) Dumping nonzero regions of core onto tape.
- 7) Deleting a particular file from a tape.
- 8) Specifying a starting address to be saved with a core image.
- 9) Starting a program loaded from tape.

All of the above functions can be performed by commands from the console Teletype, or by calling TENDMP as a subroutine and providing a command string in core.

2. COMMANDS¹

2.1 Unit Selection

To select a unit, type n (\$) where n is a number from 0 through 7 (unit 0 means 8; as usual).

¹In the command formats which follow, certain conventions apply.

- a. The symbol (\$) represents the ALTMODE character (ASCII code 033, 175, or 176). ALTMODE echoes back as a dollar sign (\$).
- b. Alphabetic characters in commands and filenames can be typed in either upper or lower case; they are considered to be in upper case.
- c. Filenames consist of a 6-character name and a 3-character extension. A space (not a period) separates the name from the extension. All printing characters are legal in filenames. If the space and extension are omitted, a null extension is assumed. SAVE files created by Monitor normally have the extension "SAV."
- d. The character RUBOUT or DELETE erases the entire command currently being typed.

This command causes TENDMP to read the directory block from DECtape unit n into core, and defines that unit as the current unit. This unit and directory are remembered in core, and need only be specified once during a sequence of operations, perhaps including running other programs, if the storage in core from 37177 through 37757 is not disturbed.²

2.2 Listing a File Directory

To list the file directory of the current unit, type

F\$

The name and extension of each file will be listed.

2.3 Loading and Starting a Program

To load and start a program, type the filename followed by a carriage return. If the extension is null, it may be omitted.

Examples:

file ext ↵

or

file ↵

This command causes core to be cleared from location 40 through location 37176, the program to be read from tape, and the program to be started at the address which was specified when the program was dumped (see below).

2.3.1 Variations on the Above Loading - To load a program without starting it, that is with control remaining in TENDMP, type:

L\$file ↵

This command clears core from 40 through 37176, and loads the program into core.

The clearing of core can be inhibited, and a program merged with existing core, as follows:

M\$file ↵

This causes only those areas for which information is present on tape to be modified. For instance,

M\$EDDT SAV ↵

would allow EXEC DDT to be used to examine a current core image, or to share core with a maintenance or diagnostic program.

Location 40 is cleared before the merge is done.

² All addresses given in this document assume the 16K version of TENDMP. Translation of these figures for other versions is a simple matter.

2.3.2 Starting the Program - The command to go to the current starting address (that last specified by loading or merging a program or set by the operator (see below)) is:

C(\$)

2.4 Zeroing a Directory

To clear the directory of the current unit, type

Z(\$)

To put a clear directory on a virgin DEctape, type

n(\$)

Z(\$)

The first command is necessary to specify the current unit, and it will read the contents of the directory block into core, even though the contents may be unspecified. However, the Z(\$) command will discard that information, create a legitimate, clear directory in core, and write it on the tape.

A clear directory reserves blocks 1, 2, and 144 (octal) with file code 36, and clears all filenames.

The last word in the directory, which is the tape identifier, is not cleared. If a tape does not have an identifier, one may be added manually by depositing it in location 37376, and then performing a Z(\$), D(\$), or K(\$) command.

2.5 Dumping a Program Onto Tape

When a program is dumped, a starting address is dumped with it. Therefore, a starting address must be specified before dumping.

The current starting address is set each time a program is loaded from tape; it may be left alone if the same address is to be dumped. Otherwise, the command

r(\$)

where n is the new current starting address is used.

Since this is the same format as the unit selection command, the restriction is imposed that the starting address must be greater than seven. In fact, it should be 20 or greater, since the accumulators are all used by TENDMP, and are not dumped with a file.

Dumping a file is accomplished as follows:

D(\$) file ext

This causes the contents of core, from location 20 through 37174, to be dumped with the name "file ext" on the current unit, and with the current starting address. Sequences of two or more zeroes are omitted

from the file. If a file of that name already exists, it is superceded. The correct core size information is put into the directory, but the contents of the date field are unpredictable. The directory is then written onto the tape. Control remains in TENDMP.

2.6 Deleting a File from the Tape

To delete a specific file, type

K\$ file ext

The specified file will be deleted (killed) and the directory will be written back out onto tape

3. DIAGNOSTIC MESSAGES

As core space in TENDMP is at a premium, there is only one error indication - the Teletype bell is rung. The user should then be able to diagnose the error by examining the last command typed and checking the list of possible errors given below.

- a. The directory may not have been read in from the DECTape as yet, or it may have been clobbered after being read in.
- b. The filename ext. specified in a K\$, L\$, or M\$ command cannot be found on the current DECTape.
- c. There is no room either in the directory or on the DECTape to dump another file. In the latter case, the directory in core may be in an intermediate state, so that rereading the directory from tape by an n\$ command is advisable.
- d. There are either no units or there is more than one unit dialed to the current unit number.
- e. The write-lock switch was on for a D\$, K\$, or Z\$ command.
- f. A tape read or write error occurred.
- g. The tape has run into the end zone (position the tape manually if this is the case).

4. TENDMP VERSIONS

The user should be aware that there are several versions of TENDMP and should check the label of the paper tape version to be used for the following parameters:

- a. Which DECTape control is used - TD-10 or 551.
- b. What memory size is handled. 16K or 32K are the usual cases.
- c. What paper tape format is used - HRI (Hardware Readin Mode - also called RIM10B) or RIM (Readin Mode - read by a special subroutine located in the shadow accumulators on the PDP-6). HRI is preferred for PDP-10.

The creation of these versions is explained in "Assembly Instructions" below.

5. ASSEMBLY INSTRUCTIONS

Assembling a copy of TENDMP from the source is straightforward. The output is normally in RIM10B format, and is, of course, absolute.

The normal assembly is for a 16K machine, and for the TD-10 DECTape control.

Conditional assemblies are provided to modify the above standards.

The normal assembly command string to MACRO is:

```
*PTP:,LPT:←TTY:,DTAx:TENDMP.MAC
```

To modify the standards, add some of the following at assembly time:

```
MODE=1 ;for 551 control version.
```

MODE is normally zero, representing a TD10 TENDMP.

```
CORE=1 ;for an 8K version
```

```
CORE=4 ;for a 32K version
```

CORE is normally 2, and is the number of 8K blocks of core for this version of TENDMP.

```
DEFINE RIM10B RIM ;for PDP-6 paper tape format
```

Example:

```
*PTP:,LPT:←TTY:,DTAx:TENDMP.MAC
CORE=4
↑Z
.END OF PASS 1
↑Z
```

6. STORAGE ALLOCATION

TENDMP, when loaded, occupies the upper end of available memory. The figures below are given for a 16K memory; translation of these figures for other core sizes should be obvious.

<u>Locations</u>	<u>Contents</u>
37175, 37176	Cleared
37177 through 37376	Directory of the current DECTape
37377 through 37757	Actual coding and temporary storage areas
37760 through 37776	Area reserved for command string (not modified)
37777	Byte pointer to the ASCII command string which may be in locations 37760 through 37776

In addition to these locations, TENDMP also modifies the contents of all accumulators (location 0 through 17).

Since the actual code occupies locations 37400 through 37757, TENDMP can fit into two DECTape blocks and can therefore be located in and bootstrapped from blocks 0 and 1 of a properly prepared DECTape, using the Hardware ReadIn feature of the PDP-10/TD-10.

TENDMP's starting address is 37400.

7. CALLING TENDMP AS A SUBROUTINE

TENDMP can be called from a program by the following procedure:

Place a series of commands, in ASCII format, in the reserved core area (37760-37776), omitting line feeds after carriage returns. Place a byte pointer to this command string in location 37777 such that an ILDB 37777 will retrieve the first character. Transfer to 37401. The commands will be executed. If the last command does not cause a transfer out of TENDMP, a RUBOUT should be the last character. This causes a carriage return line feed to be typed out, and control to be switched to the Teletype.

If an error occurs during these commands, the Teletype bell is rung, and control switches immediately to the Teletype.

TENDMP Command Summary

Operation	Command	Comments
Select a DECTape unit, read in its directory, and designate it as the current DECTape.	n\$	n must be in the range 0 to 7.
Zero the directory of the current DECTape. ¹	Z\$	
Dump nonzero areas of core and the current starting address onto the current DECTape. ¹	D \$ file ext ↵	The core image file is assigned the name "file ext."
Specify a new starting address prior to dumping or before giving the G \$ command.	n\$	n is at least greater than 7 but should be greater than 17 ₈ .
Clear core, load a program from the current DECTape, and start it.	file ext ↵	"file ext" is the name of the core image file to be loaded.
Clear core, load a program from the current DECTape, but do not start it.	L\$ file ext ↵	
¹ The new directory is written onto the DECTape.		

TENDMP Command Summary

Operation	Command	Comments
Merge a program from the current DECTape; leave the remainder of core undisturbed.	M(\$ file ext)	
Go to the current starting address.	G(\$)	
Delete (kill) a file from the current DECTape. ¹	K(\$ file ext)	
List the file directory of the current DECTape.	F(\$)	
¹ The new directory is written onto the DECTape.		

CREDITS:

The basic structure of TENDMP was suggested by MACDMP, a program written at the Project MAC PDP-6 installation at MIT. This guidance is gratefully acknowledged.

ADDENDUM

Post Assembly Instructions to Generate a Self Starting TENDMP

When a paper tape TENDMP is assembled with MACRO (version 008) it is not self starting, i.e., after reading the paper tape the operator must depress the "CONTINUE" switch to commence operation at the CTY.

In order to generate a self starting version from the assembler output:

1. Load a monitor with exec DDT
2. Then load assembler paper tape of TENDMP (either 16K or 32K)
3. Enter exec DDT (start at 141) and type

```
$L                               $ = ALTMODE
37400<37757 1R                   1R = Control R
37400$J
```

The above instructions pertain to 16K TENDMP, for the 32K version the first digit of each number will be 7 instead of 3.