



---

**MP-60**  
**COMPUTER SYSTEM**

**UTILITY**  
**REFERENCE MANUAL**

---



# LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Rev.	Page	Rev.	Page	Rev.	Page	Rev.
Cover Page	B						
ii thru viii	B						
1-1/1-2	A						
2-1	A						
2-2	B						
2-3	A						
2-4	A						
2-5	B						
2-6	B						
2-7 thru 2-11/2-12	A						
3-1 thru 3-4	A						
3-5	B						
3-6 thru 3-9	A						
3-10	B						
3-11/3-12	A						
4-1 thru 4-4	A						
5-1 thru 5-4	B						
Comment Sheet	B						



## PREFACE

---

The purpose of this manual is to provide users of UTIL, FMP, and COPYL with a description of the external features and operating instructions of each package.



# CONTENTS

---

Section		Page
1	INTRODUCTION	
2	FILE MAINTENANCE PACKAGE	
	Control Statements Processed by FMP .....	2-1
	*FMP .....	2-2
	*FLD List File Label Directory .....	2-3
	*DLD List Device Label Directory .....	2-4
	*ENTER Write Device Label and Format .....	2-5
	*DUMP Write File(s) From Disk To Magnetic Tape .....	2-6
	*LOAD Write File(s) From Magnetic Tape to Disk .....	2-6
	*VERIFY Compare File(s) On Tape With Disk File(s) .....	2-7
	*OUT Terminate the *FMP Call .....	2-8
	FMP Error Messages and Recovery Procedures .....	2-8
	Job Structure Examples .....	2-10
3	INPUT/OUTPUT UTILITY PACKAGE (UTIL)	
	Control Statements Processed by UTIL .....	3-2
	Transfer Information .....	3-4
	Verify Requests .....	3-6
	Motion Control Requests Provided By UTIL .....	3-7
	UTIL Error Messages and Recovery Procedures .....	3-8
	Job Structure Examples .....	3-9
4	LOAD FILE EDITOR (COPYL)	
	Control Statements Processed By Load File Editor .....	4-1
	Generate New Load File .....	4-1
	Generate New Load File Example .....	4-2
	COPYL Error Messages .....	4-3

CONTENTS (Cont.)

Section		Page
5	FILE CATALOG LISTING (CATLIST) .....	5-1
	Control Statements Processed by CATLIST .....	5-1
	CATLIST Output .....	5-2
	CATLIST Examples .....	5-3



# INTRODUCTION

1

---

This reference manual describes the use and operation of the MPX utility package, which consists of the following:

- File Maintenance Package (FMP)
- Input/Output Utility Package (UTIL)
- Load File Editor Program (COPYL)

The MPX utility programs are contained on the MPX library, with individual services being called into execution by the reading of card image control statements from the standard input device. Unless specifically noted, the standard input device is a card reader.

Operator familiarity with the MPX/RT operating system is assumed.

In the following descriptions, control statements begin with an asterisk in column 1 of the input card. The request code begins in column 2 and is followed by the required parameter. All fields are separated by commas or parentheses.



# FILE MAINTENANCE PACKAGE

2

---

The FMP provides a comprehensive set of mass storage utility routines, which allow the operator to do the following:

- Maintain device labels
- Maintain file labels
- List selected portions or all of the file label directory
- List selected portions or all of the device label directory
- Write files from disk to magnetic tape
- Reload files from magnetic tape to disk

The FMP package is called by the MPX/RT job processor after reading an \*FMP statement from the standard input device. The package is loaded, and control statements are processed from the standard input device. Processing continues until an \*OUT control statement is read, at which time control is returned to the MPX monitor.

## CONTROL STATEMENTS PROCESSED BY FMP

The functions provided by FMP are as follows:

- \*FLD                    List file label directory
- \*DLD                    List device label directory
- \*ENTER                 Write a device label and format a disk pack
- \*DUMP                  Write file(s) from disk to magnetic tape
- \*LOAD                  Write file(s) from magnetic tape to disk
- \*VERIFY                Compare dump tape file(s) with disk file(s)

In the description of the FMP control statements, the following mnemonics will be used. Parameters not included in this list must be on the control statement as the alphabetic character specified.

- AK One to four characters specifying the access privacy key for a file identified by FN, ED, and OWNER
- DID One- to eight-character device identifier
- DT A one-character number specifying the type of disk:
  - 1 = 9425
  - 2 = 844
  - 3 = 9427
  - 4 = 9760 or 858
  - 5 = 640
- ED Two characters specifying the edition number of a file with the same FN and OWNER.
- FMPAK A master access privacy key, which will allow FMP routines access to files regardless of the AK of the files. Refer to the MPX-RT Systems Installation Handbook (Control Data publication No. 14064000) for this master access privacy key.
- FN One to 14 characters specifying a file name.
- LUN One- or two-digit logical unit number of a magnetic tape, which has been opened prior to execution of the \*FMP control statement.
- OWNER One to four characters specifying the file owner.

\*FMP

This control statement is read by the MPX operating system and causes the FMP to be loaded and placed in operation. Upon being placed in operation, FMP will recognize only those control statements listed in the previous paragraph. All MPX control statements must precede the \*FMP statement or follow the \*OUT statement.

All messages and errors will be reported on the standard output device.

\*FLD LIST FILE LABEL DIRECTORY

Listing of the FLD is possible by listing all or selected portions. The control statement options are as follows:

- \*FLD,A,FMPAK

List all file labels in the label file.

- \*FLD,O,OWNER,AK

List all file labels of files that have an owner and access key matching the OWNER and AK in the control statement. If FMPAK is used for the AK field, all file labels that match OWNER will be listed.

- \*FLD,F,FN,OWNER,ED,AK

List the file label for the file that matches the FN, OWNER, ED, and AK. If the field ED contains \*\*, a file label is listed for each edition.

The two FLD listing formats are fixed and variable length. The fixed length portion is first and is listed as follows:

FILENAME	FFF...FFF
EDITION	EE
OWNER	0000
USAGE	READ ONLY or READ/WRITE
NUMBER OF ALLOCATED BLOCKS	XXXXXX
HIGHEST BLOCK WRITTEN	XXXXXX
BLOCK SIZE IN WORDS	XXXXXX
NUMBER OF SECTORS PER BLOCK	XXXXXX
DEVICE TYPE	DDDD
NUMBER OF DEVICES	XXXX
NUMBER OF SEGMENTS	XXXX



## \*ENTER WRITE DEVICE LABEL AND FORMAT

The \*Enter control statement is a service function to prepare disk packs for use in the MPX operating system. Functions performed are as follows:

- Writing a device label on a new pack
- Relabeling an existing pack
- Clear by writing zeros on the pack

With the \*ENTER statement in execution, the FMP routine examines the device type and requests the operator (at the operator console) to mount the pack to be entered on the available disk drive. Upon acknowledgement from the operator, the function is performed on the assigned disk pack.

Of special note:

- Whenever the \*ENTER statement is performed, all previously written information on the disk pack is no longer available to the system.
- Whenever the C parameter (clear) is selected and an irrecoverable write occurs for a track, the track is downed and not available to the system.

The control statement and its functions are as follows:

\*ENTER,DT,DID,C

Write a device label, DID, on the disk pack of the device type specified by DT and clear previous information by writing zeros. If the C parameter is absent, zeros are not written. Previously written information is not available to the system.

## \*DUMP WRITE FILE(S) FROM DISK TO MAGNETIC TAPE

The primary purpose of the \*DUMP control statement is to retain system or user files on magnetic tape as backup to the disk pack. The magnetic tape to be used must be equipped and assigned to the system prior to the \*FMP control statement.

The \*DUMP statement writes the label of each file to be dumped on the specified magnetic tape unit. Following the header, each track of the file is dumped by blocks. When the entire file has been dumped, an end-of-file is written. The file and header are written in binary on the magnetic tape, with the header containing the file allocation parameters. This method allows a number of files to be dumped onto the same magnetic tape.

The control statement options are as follows:

- \*DUMP, O, LUN, OWNER, FMPAK, R

Dump all the files having the specified OWNER to tape. The parameter R is optional. If present, the file(s) is released after the dump is complete. The file(s) must be released if it is to be allocated by the \*LOAD control statement.

- \*DUMP, F, LUN, FN, OWNER, ED, AK, R, DWN

Dump the file specified by FN, OWNER, ED, and AK to tape. The parameter R is optional. If present, the file is released after the dump is complete. The file must be released if it is to be allocated by the \*LOAD control statement. The parameter DWN is optional. The parameter R must be present when DWN is specified. If DWN is specified, the bad tracks encountered during the dump are marked assigned in the labels of the device containing the bad tracks.

At the completion of the dump, the following information will be placed on the standard output device.

FID = file name, owner, edition

FILE DUMPED

\* IDENT IDENTIFY NEW DT AND/OR DID FOR \*LOAD STATEMENT

The primary purpose of the \*IDENT control statement is to identify a new device type and/or device identifier on which to allocate files during the \*LOAD processing. Files written in blocker-deblocker format may be moved across both device type and device identifier. Files not written in blocker-deblocker format, such as RESFILE, may only be moved across device identifiers.

The control statement format is as follows:

- \*IDENT, DT, DID

Allocate the files on the DT and DIDs specified. 1-8 DIDs may be specified separated by commas. Both parameters must be present.

- \*IDENT

This form of the statement restores \*LOAD to allocate files as specified on the dump tape.

\*LOAD WRITE FILE(S) FROM MAGNETIC TAPE TO DISK

The \*LOAD control statement transfers a file(s) to a mass storage file(s) from a magnetic tape produced by \*DUMP. The use of the R and NS parameters allows for releasing segmented files and reallocating them as non-segmented files.



The control statement options are as follows:

- \*LOAD, A, LUN, FMPAK, R, NS

Load all the files to disk from the tape unit specified by LUN. The parameter R is optional. If R is specified, the same existing files on disk are released and reallocated before loading begins. The parameter NS is optional and is meaningful only when R is specified. If NS is specified, the files are allocated with no segmentation allowed.

- \*LOAD, F, LUN, FN, OWNER, ED, AK, R, NS

Load the file specified by FN, OWNER, ED, and AK from the tape unit LUN to disk. When more than one file exists on the tape, the tape will be searched until the correct file is found. The parameter R is optional. If R is specified, the same existing file on disk is released and reallocated before loading begins. The parameter NS is optional and is meaningful only when R is specified. If NS is specified, the file is allocated with no segmentation.

At the completion of the load, the following information will be placed on standard output device.

FID = File name, owner, edition

FILE LOADED

\*VERIFY COMPARE FILE(S) ON TAPE WITH DISK FILE(S)

The \*VERIFY control statement compares a file(s) on magnetic tape, created from the \*DUMP control statement, with the corresponding file(s) on disk.

The control statement and its functions are as follows:

\*VERIFY, LUN

Verify all files on the dump tape as specified by LUN with the corresponding files on disk.

At the completion of the verify, the following information will be placed on the standard output device.

FID = File name, owner, edition

FILE VERIFIED

All errors are reported on the standard output unit.



<u>Error Message</u>	<u>Cause</u>	<u>Action</u>
IRRECOVERABLE ERROR OCCURRED WHILE READING/ WRITING BLOCK XXXXX FROM/TO DISK/TAPE	Error occurred on block XXXXX during a load or dump request.	Processing continues.
LABEL FILE READ ERROR, BLOCK NUM = XXXXX	Error occurred while reading the label file.	Processing continues.
LOAD TAPE IS NOT IN FMP FORMAT	Load tape was not created by FMP.	FMP skips to next control card.
NO FILES/DUMP/LOADED	No files specified by the load or dump request were found in the label file (dump) or on the load tape.	FMP skips to next control card.
FMP ABNORMALLY TERMINATED	An error causes FMP to abort processing.	FMP aborts processing.
BAD TRACK, TRACK NUM =	In processing the *ENTER request, an error occurs.	Mark track assigned. Processing continues.
CANNOT WRITE DEVICE LABEL	Error occurred in writing the device label.	FMP skips to next control card.
***OPEN ERRORXX	OCARM ERROR XX occurred on a dump or load request.	FMP aborts processing.
***CLOSE ERRORXX	OCARM ERROR XX occurred on a dump or load request.	FMP aborts processing.
***MODIFY ERRORXX	OCARM ERROR XX occurred on a load request.	FMP aborts processing.
***RELEASE ERRORXX	OCARM ERROR XX occurred on a dump or load request.	FMP aborts processing.
***ALLOCATE ERRORXX	OCARM ERROR XX occurred on a load request.	FMP aborts processing.

<u>Error Message</u>	<u>Cause</u>	<u>Action</u>
***DEVICE CANNOT BE PLACED ONLINE	Operator cannot place the device on-line, or a mass storage unit is not available.	FMP skips to next control card.

The OCARM ERROR codes (XX) are defined in the MP-60 MPX/RT Reference Manual, Control Data publication No. 14062300.

## JOB STRUCTURE EXAMPLES

FMP job structure examples are as follows:

- List the contents of the label file
  - \*JOB (...)
  - \*SCHED (...)
  - \*FMP
  - \*FLD, A, FMPAK
  - \*OUT
  - \*EOJ
- List the label of a specific file
  - \*JOB (...)
  - \*SCHED (...)
  - \*FMP
  - \*FLD, F, FILE-NAME, OWNER, 01, \*\*\*\*
  - \*OUT
  - \*EOJ

- List the device label directory

- \*JOB (...)

- \*SCHED (...)

- \*FMP

- \*DLD, A, FMPAK

- \*OUT

- \*EOJ

- Dump files from disk to tape and verify

- \*JOB (...)

- \*SCHED (...)

- \*EQUIP (10=MT)

- \*FMP

- \*DUMP, O, 10, OWNER, FMPAK

- \*VERIFY, 10

- \*OUT

- \*EOJ



## INPUT/OUTPUT UTILITY PACKAGE (UTIL)

3

---

The I/O utility package (UTIL) enables the operator to perform the following functions on peripheral hardware devices such as the disk, magnetic tape, card reader, line printer, and card punch.

- Transfer information between two devices
- Verify information between two devices
- Motion control

The UTIL program is called by the MPX/RT job processor on reading a \*UTIL statement from the standard input device. The utility routine accepts request statements from the standard input device. The program continues processing requests until terminated by an \*OUT statement, at which time control is returned to the MPX monitor.

When an end-of-file condition (\*EOJ) is detected on the card reader, disk files are closed and the UTILITY program returns to the job monitor. The user can use the \*ENDFILE control card, which is treated as end-of-file records when processing UTIL request statements.

Binary records are transferred to the printer as "BINARY RECORD." If no spacing control is requested, character 1 of each record is assumed to contain a print control character. When single or double spacing is specified, the first 135 characters of the output record are printed, preceded by the required print control character.

The messages "UTILITY IN" and "UTILITY OUT" are transferred to the printer at initialization and termination of UTILITY. Upon completion of each transfer and verify request, the number of files and records is outputted to the standard output device in the following message:

FILES NN RECORDS MM COMPLETE

UTIL uses BLKDEBLK for creating blocked tapes; refer to the MPX/RT Reference Manual, Control Data publication No. 14062300, for a description of the headers and data areas (page 6-2). The magnetic tapes created are compatible with ANSI standards. The user determines the format of unblocked tapes by specifying the record length.

## CONTROL STATEMENTS PROCESSED BY UTIL

The functions provided by UTIL are as follows:

- \*T Transfer information from one peripheral device to another
  - a) Card to card
  - b) Card to tape
  - c) Card to printer
  - d) Cards to disk file
  - e) Tape to tape
  - f) Tape to card
  - g) Tape to printer
  - h) Tape to disk file
  - i) Disk file to disk file
  - j) Disk file to card
  - k) Disk file to tape
  - l) Disk file to printer
- \*V Verify information, on a record-by-record basis, between the following peripheral devices.
  - a) Card versus tape
  - b) Card versus card ( two card readers required)
  - c) Card versus disk file
  - d) Disk file versus disk file
  - e) Disk file versus tape
  - f) Tape versus tape



- Motion is controlled by the following statements.
  - a) \*REW               Rewind magnetic tape or disk file
  - b) \*EOF               Write end-of-file
  - c) \*ADF               Advance logical unit
  - d) \*BSF               Backspace logical unit N files
  - e) \*BSR               Backspace logical unit N records

In the description of the UTIL control statements, the following mnemonics will be used.

- U1                    One- or two-digit input logical unit number
- U2                    One- or two-digit output logical unit number

The parameter fields for logical units U1 and U2 must be previously equipped to physical devices using EQUIP cards, or ALLOCATE and OPEN if the units are on disk, before UTIL is called.

Logical units U1 and U2 cannot be equal, and two different logical units cannot be equipped to the same physical unit in a UTIL request, except for disk.

If an output unit is not defined, it is interpreted as a dummy unit used to reposition the input file by the given number of records or files. Codes are as follows:

- B/blank               B = blocked type  
                           Blank or not B = unblocked tape
- Block length         Length of tape record or block
- Max records           Sets an upper limit on the number of records to be transferred. If N is omitted, records are transferred until the number of files in the next parameter has been reached. The upper limit of number of records to be transferred is in decimal. At the end of the transfer, the number of records and files encountered are outputted to the standard output device. Max records would be specified only when a given number of records of a file are to be transferred.

- Max files                    Sets an upper limit on the number of files to be transferred. If it is omitted, one file is transferred or the number of records specified by max records are transferred. The upper limit of the number of files to be transferred is in decimal. At the end of the transfer, the number of records and files encountered is placed on the standard output device. Max files would be specified only when a given number of complete files are to be transferred.
  
- Spacing                      Used only when the output unit is the printer; double space if field is D, and single space if field is S. If spacing is not specified, forms control is provided by the first character of each record.

#### TRANSFER INFORMATION

The following are the individual transfer request formats. Standard blocked tapes are generated using the system blocker/deblocker package. In all requests involving tapes, the tape may be unblocked with the record length given or blocked using a specified block length.

#### NOTE:

The system standard block length is 480 words.

- Card to tape

\*T, U1, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

Transfer information from card reader, logical unit U1, to magnetic tape, logical unit U2.

- Card to card

\*T, U1, U2, MAX RECORDS, MAX FILES

Transfer information from card reader, logical unit U1, to card punch, logical unit U2.

- Card to printer

\*T, U1, U2, MAX RECORDS, MAX FILES, SPACING

Transfer information from card reader, logical unit U1, to printer, logical unit U2.

- Tape to card

\*T, U1, B/BLANK, BLOCK LENGTH, U2, MAX RECORDS, MAX FILES

Transfer information from magnetic tape, logical unit U1, to card punch, logical unit U2.

- Tape to printer

\*T, U1, B/BLANK, BLOCK LENGTH, U2, MAX RECORDS, MAX FILES, SPACING

Transfer information from magnetic tape, logical unit U1, to printer, logical unit U2.

- Tape to tape

\*T, U1, B/BLANK, BLOCK LENGTH, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

Transfer information from magnetic tape, logical unit U1, to another magnetic tape, logical unit U2.

- Disk file to tape

\*T, U1, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

Transfer information from disk file, logical unit U1, to magnetic tape, logical unit U2.

- Disk file to card

\*T, U1, U2, MAX RECORDS, MAX FILES

Transfer information from disk file, logical unit U1, to card punch, logical unit U2.

- Disk file to printer

\*T, U1, U2, MAX RECORDS, MAX FILES, SPACING

Transfer information from disk file, logical unit U1, to printer, logical unit U2.

- Card to disk file

\*T, U1, U2, MAX RECORDS, MAX FILES

Transfer information from card reader, logical unit U1, to disk file, logical unit U2.

- Tape to disk file

\*T, U1, B/BLANK, BLOCK LENGTH, U2, MAX RECORDS, MAX FILES

Transfer information from magnetic tape, logical unit U1, to disk file, logical unit U2.

- Disk file to disk file

\*T, U1, U2, MAX RECORDS, MAX FILES

Transfer information from disk file, logical unit U1, to another disk file, logical unit U2.

## VERIFY REQUESTS

The verify request function permits the verification of information on a record-by-record basis. Some peripheral devices, such as the card punch and printer, cannot be used to verify information because they are incapable of input.

In the individual verify request formats that follow, the parameter field notations are the same as for transfer requests, except that U1 is defined as the logical unit read first and U2 as the logical unit read second.

The number of records and files verified are outputted to the standard output device at completion of the request. If two corresponding records do not compare, the file and record numbers are outputted to the standard output device.

- Card versus tape

\*V, U1, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

### NOTE

If tape is logical unit U1, then block format and length must directly follow the logical unit U1 parameter.

- Card versus card

\*V, U1, U2, MAX RECORDS, MAX FILES

### NOTE

Two card readers are required.

- Tape versus tape

\*V, U1, B/BLANK, BLOCK LENGTH, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

- Disk file versus tape

\*V, U1, U2, B/BLANK, BLOCK LENGTH, MAX RECORDS, MAX FILES

#### NOTE

If the tape is logical unit U1, then block format and length must directly follow the logical unit U1 parameter.

- Card versus disk file

\*V, U1, U2, MAX RECORDS, MAX FILES

- Disk file versus disk file

\*V, U1, U2, MAX RECORDS, MAX FILES

#### MOTION CONTROL REQUESTS PROVIDED BY UTIL

Motion control requests provided by UTIL are as follows:

- \*REW, U           REWIND

For magnetic tape, rewind logical unit U to the load point.

For disk, set the pointer to the first block of file U.

- \*EOF, U, N

For magnetic tape, write N end-of-file records on device U. If no N parameter is specified, 1 is assumed.

For disk, this is a no operation (NOP).

- \*ADF, U, N

For magnetic tape, advance logical unit U forward N files. If no N parameter is given, 1 is assumed. If from the load point, advance forward logical unit U to the beginning of file N.

For disk, set the pointer to the highest block written + 1. The N parameter is ignored.

- \*BSF, U, N

For magnetic tape, logical unit U should be backspaced N files. If no N parameter is given, 1 is assumed.

For disk, set the pointer to the first block of file U.

- \*BSR, U, N

For magnetic tape, logical unit U should be backspaced N records (blocks). If no N parameter is given, 1 is assumed.

For disk, set the pointer backward N blocks. If no N parameter is given, 1 is assumed.

There is no specialized request to advance a logical unit N records, but the function can be performed using a TRANSFER request and an output logical unit that is not assigned.

## UTIL ERROR MESSAGES AND RECOVERY PROCEDURES

All errors are reported on the standard output device.

<u>Error Message</u>	<u>Cause</u>	<u>Action Taken By Program</u>
INVALID REQUEST	Illegal request code or illegal parameter field within the request.	Message placed on the standard output unit, and then the program searches for the next request.
BLOCK TOO BIG	On requests that require a block length field, the number requested is larger than the buffer area provided for it within the utility program.	Message placed on the standard output unit, and then the program searches for the next request. (The buffer lengths allowed by the utility program could be enlarged through reassembly.)
ERROR LU XX	A read or write error was detected on logical unit XX.	The message is placed on the standard output unit. The program closes any unit opened for the request, and then searches for the next request.

<u>Error Message</u>	<u>Cause</u>	<u>Action Taken By Program</u>
PICK NN LU XX	Deblocker error number NN occurred on logical unit XX. Refer to the blocker/deblocker error codes for cause of this message.	The message is placed on the standard output unit. The program closes any units opened for the request and then searches for the next request.
PACK NN LU XX	Blocker error number NN occurred on logical unit XX. Refer to the blocker/deblocker error codes for cause of this message.	The message is placed on the standard output unit. The program closes any unit opened for the request and then searches for the next request.
COMPARE ERR - RECORDS NN FILES MM	Compare error occurred on a verify request on record NN of the verify request, in the MMth file compared. (The record count is not restarted at the start of each new file compared.)	The message is placed on the standard output unit. The verify request continues. (There is no way at present to limit the number of verify errors outputted on any verify request.)

## JOB STRUCTURE EXAMPLES

UTIL job structure examples are as follows:

- Card to print
  - \*JOB (...)
  - \*SCHED (...)
  - \*UTIL
  - \*T, 63, 62, , 1, S
  - \*OUT
  - \*EOJ

o Card to tape (unblocked)

\*JOB (...)

\*SCHED (...)

\*EQUIP (30=MT)

\*UTIL

\*T, 63, 30, , 20, , 1

DATA

\*END FILE

\*EOF, 30, 1

\*OUT

\*EOJ

o Disk to print

\*JOB (...)

\*SCHED (...)

\*OPEN (40...)

\*UTIL

\*T, 40, 62, 100, 1, D

\*OUT

\*EOJ

o Disk to tape (with verify)

\*JOB (...)

\*SCHED (...)

\*OPEN (...)

\*EQUIP (30=MT)



\*UTIL

\*T, 41, 30, B, 480, 500, 1

\*REW, 41

\*REW, 30

\*V, 41, 30, B, 480, 500, 1

\*OUT

\*EOJ



## LOAD FILE EDITOR (COPYL)

4

---

The load file editor package enables the user to do the following:

- Maintain program load files in the form of relocatable binary records
- Replace existing subroutine modules with new binary load modules
- Add new subroutine binary load modules to the load file

The load file editor is called by the MPX/RT job processor upon reading a \*COPYL statement from the standard input device. Only one copy/edit operation is allowed per \*COPYL statement.

### CONTROL STATEMENT PROCESSED BY LOAD FILE EDITOR

The control statement and its functions are as follows:

\*COPYL (U1, U2, U3)

The parameters are as follows (logical unit numbers of a file can consist of magnetic tape or disk):

- U1                    Logical unit number of an existing load file
- U2                    Logical unit number of a file that contains subroutine binary modules to be replaced or inserted on the new load file
- U3                    Logical unit number of the updated load file

### GENERATE NEW LOAD FILE

Selected subprogram binary modules may be replaced or inserted on a load file using an existing load file and an update binary file. The update file would typically be produced by preceding assemblies or compiles.

All files are initially rewound, and subroutine binary module names are extracted from U2. A copy operation of U1 to U3 is then performed, replacing those subroutine binary modules specified on U2 on the new load file U3. The modules on U2 need not be in order. If modules on U2 are not included on U1, they are inserted at the end of the new load file. Subroutine binary modules are read from U1 until an end-of-file is encountered. All files must be in MPX standard blocking format, with a block size of 480 words. At the completion of the copy operation, files U1 and U3 are not rewound; file U2 is rewound.

A listing of the contents of the new load file is produced on standard output in the following format:

DECKS LISTED IN ORDER OF THEIR OCCURRENCE ON THE NEW BINARY FILE

R DECKNAME INDICATES REPLACED DECK

A DECKNAME INDICATES ADDED DECK

Name 1            Name 2            Name 3            .....

.

.

\*\*\*\* COPYL COMPLETED\*\*\*\*

GENERATE NEW LOAD FILE EXAMPLE

The following example of a card deck illustrates the use of COPYL to generate a new load file from an existing load file, replacing selected subroutine binary modules.

\*JOB (ID=UPDATE)

\*SCHED

\*OPEN (10, BINARY-LGO, 01, \*\*\*\*, R)

\*ALLOCATE (BINARY-LGO, JFS, 02, \*\*\*\*, 480, 1000, RW, 2, SYSTEM01)

\*OPEN (20, BINARY-LGO, JFS, 02, \*\*\*\*, RW)

\*CMP (I, L, X)

source decks

\*FTN (I, L, X)

source decks

\*COPYL(10,57,20) replacement modules on LGO

\*CLOSE(10)

\*CLOSE(20)

\*RELEASE(BINARY-LGO,JFS,02,\*\*\*\*,R)

\*EOJ

### COPYL ERROR MESSAGES

<u>Error Message</u>	<u>Cause</u>	<u>Action Taken By Program</u>
ERROR ON COPYL CONTROL CARD	Illegal parameters on card.	Message placed on the standard output unit. Files remain in their current position. MPX gets control and searches for the next request.

The following errors are detected in the BLOCKER/DEBLOCKER programs, which pass the error code to COPYL where they are outputted to the standard output device.

\*\*\*\* LOGICAL UNIT XX BLOCKER/DEBLOCKER ERROR AS FOLLOWS:

On this line one of the following error messages will be placed.

- EOF OR REACHED ALLOC. LIMITS
- LOGICAL UNIT IS NOT OPEN
- WRITE ATTEMPTED ON A READ ONLY FILE
- BUFFER AREA ALREADY DEFINED
- BUFFER SIZE TOO LARGE
- BUFFER SIZE TOO SMALL
- BUFFER AREA NOT DEFINED
- RECORD SIZE LARGER THAN BUFFER SIZE
- BUFFER AREA ALREADY DEFINED
- CANNOT PERFORM SPECIFIED FUNCTION

- LOGICAL UNIT INVALID
- IRRECOVERABLE I/O ERROR
- BLOCK POINTER OUT OF BOUNDS

Action taken by the program is same as that described previously.

# FILE CATALOG LISTING (CATLIST)

5

---

The file catalog listing routine enables the user to produce a list of files within the mass storage system. This list is sorted into order by file owner. The list may be restricted to selected file owners or to selected disk devices.

## CONTROL STATEMENT PROCESSED BY CATLIST

The control statement and its functions are as follow:

\*CATLIST (L=X, O=X, U=X/X...X/X, P=X/X...X/XK-CATAK)

The parameters are as follow:

L=X                    X specifies the logical unit number which is to receive the listing output. If omitted, logical unit 62 is used.

O=X                    X specifies the format of the output listing.

<u>Entry</u>	<u>Indicates</u>
S	Short format
N	Normal format
E	Extended format

If omitted, short format is used.

U-X/X...X/X            Each X consists of 1 to 4 characters specifying a file owner to appear on the output listing. The file owners must be speparated by the character /. If omitted, files of all owners are listed.

P=X/X...X/X            Each X consists of 1 to 8 characters specifying a device label to appear on the output listing. The device labels must be separated by the character /.

If omitted, files on all devices are listed.

K=CATAK                The appearance of this parameter enables the access keys of all files to be listed. CATAK is an installation parameter. If omitted or if CATAK is specified incorrectly, the access keys are not listed.

## CATLIST OUTPUT

The CATLIST output listing contains all information stored in the system label file. The columns headings are as follow:

LBN	Block number of the label in the label file
OWNER	Identity of the owner of the file
FILE NAME	Identifies the file
ED	Edition number which identifies different versions of the file
CR-DATE	Creation date
LU-DATE	Last used date
DU-DATE	Last dumped date
ALLO	Number of blocks allocated to the file
USED	Number of blocks used by the file
SIZE	Number of words per block..
SECT	Number of sectors per block
SC	Number of segments in the file
TYPE	Device type on which this segment resides
DEVICE ID	Identification of the devices on which this segment resides
LSL	Lower sector address: sector address at which this segment begins
SL	Segment length: number of sectors in this segment



\*CATLIST (U=MPXL/MPXR)

CATALOG FILE LISTING ON 10/12/79 AT 13/24/15

PAGE 1

LBN	OWNER	FILE NAME	ED	CR-DATE	LU-DATE	DU-DATE
3	MPXL	MPX-LIBRARY	20		10 9 79	
8	MPXL	MPX-LIBRARY	21	10 9 79	10 12 79	
322	MPXL	MPX-LIBRARY	22	10 11 79	65 65 65	
230	MPXL	MPX-LIBRARY	BW	10 9 79	10 9 79	
248	MPXR	MPX-RESIDENT	01	10 12 79	10 12 79	10 12 79
2	MPXR	MPX-RESIDENT	17			
9	MPXR	MPX-RESIDENT	18	10 12 79	10 12 79	
252	MPXR	MPX-RESIDENT	CE	10 11 79	10 12 79	10 12 79
247	MPXR	MPX-RESIDENT	CN	10 11 79	10 12 79	10 12 79
275	MPXR	MPX-RESIDENT	CP	10 11 79	10 12 79	10 12 79
10	MPXR	MPX-RESIDENT	NU	10 9 79	10 9 79	

\*CATLIST (O=S,U=SAVE)

CATALOG FILE LISTING ON 10/12/79 AT 13/23/46

PAGE 1

LBN	OWNER	FILE NAME	ED	CR-DATE	LU-DATE	DU-DATE
134	SAVE	ASMBACK	00	10 9 79	10 11 79	10 11 79
40	SAVE	BATCH-DEFAULT	00	10 11 79	10 11 79	10 11 79
36	SAVE	CATLIST	00	10 11 79	10 12 79	10 11 79
132	SAVE	DELFILE	00	10 9 79	10 11 79	10 11 79
35	SAVE	EDIT	00	10 11 79	10 12 79	10 11 79
136	SAVE	HELP	00	10 9 79	10 11 79	10 11 79
39	SAVE	JOB-	00	10 11 79	10 11 79	10 11 79
4	SAVE	LABBACK	00	10 9 79	10 11 79	10 11 79
130	SAVE	LABBKG	00	10 9 79	10 11 79	10 11 79
127	SAVE	LOAD	00	10 9 79	10 11 79	10 11 79
131	SAVE	LOAD, JOB	00	10 9 79	10 11 79	10 11 79
267	SAVE	POSBKGND	00	10 12 79	10 12 79	10 12 79
259	SAVE	POSBKGNP	00	10 12 79	10 12 79	10 12 79
129	SAVE	PRINT	00	10 9 79	10 12 79	10 11 79
135	SAVE	RELEGAY	00	10 9 79	10 11 79	10 11 79
128	SAVE	T66-SAV1	00	10 9 79	10 11 79	10 11 79
133	SAVE	T66-SAV2	00	10 9 79	10 11 79	10 11 79

\*CATLIST (O=N, U=SAVE)  
CATALOG FILE LISTING ON 10/12/79 at 13/23/55

PAGE 1

LFN	OWNER	FILE NAME	ED	CR-DATE	LU-DATE	DU-DATE	ALLO	USED	SIZE	SECT
134	SAVE	ASMBACK	00	10 9 79	10 11 79	10 11 79	12	11	480	3
40	SAVE	BATCH-DEFAULT	00	10 11 79	10 11 79	10 11 79	12	1	480	3
36	SAVE	CATLIST	00	10 11 79	10 12 79	10 11 79	12	1	480	3
132	SAVE	DELFILE	00	10 9 79	10 11 79	10 11 79	12	2	480	3
35	SAVE	EDIT	00	10 11 79	10 12 79	10 11 79	12	1	480	3
136	SAVE	HELP	00	10 9 79	10 11 79	10 11 79	12	1	480	3
39	SAVE	JOB-DEFAULT	00	10 11 79	10 11 79	10 11 79	12	1	480	3
4	SAVE	LABBACK	00	10 9 79	10 11 79	10 11 79	12	1	480	3
130	SAVE	LABBKG	00	10 9 79	10 11 79	10 11 79	12	1	480	3
127	SAVE	LOAD	00	10 9 79	10 11 79	10 11 79	12	1	480	3
131	SAVE	LOAD-JOB	00	10 9 79	10 11 79	10 11 79	12	1	480	3
267	SAVE	POSBKGN	00	10 12 79	10 12 79	10 12 79	12	1	480	3
259	SAVE	POSBKGN	00	10 12 79	10 12 79	10 12 79	12	1	480	3
129	SAVE	PRINT	00	10 9 79	10 12 79	10 12 79	12	1	480	3
135	SAVE	RELEGAY	00	10 9 79	10 11 79	10 11 79	12	1	480	3
128	SAVE	T66-SAV1	00	10 9 79	10 11 79	10 11 79	42	1	480	3
133	SAVE	T66-SAV2	00	10 9 79	10 11 79	10 11 79	12	1	480	3

\*CATLIST (O=E, U=SAVE)  
CATALOG FILE LISTING ON 10/12/79 at 13/24/05

PAGE 1

LFN	OWNER	FILE NAME	ED	CR-DATE	LU-DATE	DU-DATE	ALLO	USED	SIZE	SECT	SC	TYPE	DEVICE ID	LSL	SL
134	SAVE	ASMBACK	00	10 9 79	10 11 79	10 11 79	12	11	480	3	1	640	SYSTEM01	24156	36
40	SAVE	BATCH-DEFAULT	00	10 11 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	5796	36
36	SAVE	CATLIST	00	10 11 79	10 12 79	10 11 79	12	1	480	3	1	640	SYSTEM01	5724	36
132	SAVE	DELFILE	00	10 9 79	10 11 79	10 11 79	12	2	480	3	1	640	SYSTEM01	24084	36
35	SAVE	EDIT	00	10 11 79	10 12 79	10 11 79	12	1	480	3	1	640	SYSTEM01	5832	36
136	SAVE	HELP	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	24228	36
39	SAVE	JOB-DEFAULT	00	10 11 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	5760	36
4	SAVE	LABBACK	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	2448	36
130	SAVE	LABBKG	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	24012	36
127	SAVE	LOAD	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	23904	36
131	SAVE	LOAD-JOB	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	24048	36
267	SAVE	POSBKGN	00	10 12 79	10 12 79	10 12 79	12	1	480	3	1	640	SYSTEM01	48708	36
259	SAVE	POSBKGN	00	10 12 79	10 12 79	10 12 79	12	1	480	3	1	640	SYSTEM01	42984	36
129	SAVE	PRINT	00	10 9 79	10 12 79	10 11 79	12	1	480	3	1	640	SYSTEM01	23976	36
135	SAVE	RELEGAY	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	24192	36
128	SAVE	T-66SAV1	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	23940	36
133	SAVE	T-66SAV2	00	10 9 79	10 11 79	10 11 79	12	1	480	3	1	640	SYSTEM01	24120	36

## COMMENT SHEET

MANUAL TITLE MP-60 Computer System UTILITY Reference Manual

PUBLICATION NO. 14063800 REVISION B

**FROM:** NAME: \_\_\_\_\_  
BUSINESS ADDRESS: \_\_\_\_\_

### COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

PRINTED IN U.S.A.

AA3419 REV. 11/69

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.  
FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

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

FIRST CLASS  
PERMIT NO. 8241  
MINNEAPOLIS, MINN.



CUT ALONG LINE

POSTAGE WILL BE PAID BY  
**CONTROL DATA CORPORATION**  
Aerospace Division  
Box 609  
Minneapolis, Minnesota  
55440

FOLD

FOLD

