

HEWLETT  PACKARD

IIB v IIB v III

That is The
Question!



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

MPE IID

	Page
• UNIFIED COMMAND LANGUAGE	3
• MISCELLANEOUS ENHANCEMENTS	22
• NEW MESSAGE SYSTEM	28
• TAPE LABELS	39
• PRIVATE VOLUMES & SERIAL DISC	65
• CONVERSION	Appendix F

HELP SUBSYSTEM

- All MPE commands except =Console Operator commands reside in a file (CICAT.PUB.SYS, currently 2765 sectors).
- User can obtain quick on-line information about any command. Information includes syntax, description, example.

```
:HELP [ HELP  
tablecontents  
command[,keyword]  
ALL  
EXIT ]
```

EXIT, Y^c OR BREAK TERMINATES HELP
S^c STOPS LISTING
Q^c RESUMES LISTING
TO D;SABLE HELP,PURGE CICAT.PUB.SYS

HELP SUBSYSTEM (continued)

To enter "subsystem" mode, enter:

:HELP

For example,

:HELP

INFORMATION IS AVAILABLE ON THE FOLLOWING CLASSES OF COMMANDS:

RUNNING SESSIONS

RUNNING JOBS

MANAGING FILES

RUNNING SUBSYSTEMS AND PROGRAMS

SYSTEM MANAGEMENT, STATUS, AND ACCOUNTING

UTILITY FUNCTIONS

FOR MORE INFORMATION, ENTER A KEYWORD. YOU CAN ALSO ENTER ANY COMMAND NAME AS A KEYWORD. ENTER 'HELP' FOR INFORMATION ON HELP. ENTER 'EXIT' TO LEAVE HELP.

KEYWORDS: SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY

> SESSIONS

RUNNING SESSIONS. FOLLOWING ARE THE COMMANDS USED:

() COMMAND LOG ON

ABORT

BYE

DSLIN

EOD

EOF

HELLO

HELP

REMOTE HELLO

RESUME

YOU CAN USE ANY COMMAND AS A KEYWORD.

KEYWORDS: SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY

> ABORT

:ABORT

ABORTS CURRENT PROGRAM OR OPERATION.

SYNTAX

:ABORT

KEYWORDS: PARMS,OPERATION,EXAMPLE

> EXIT



- Provides more descriptive error messages that suggest corrective action.

MPE IIA ERROR MESSAGES

:FILE A;REC=.,G;DISC=10000,16,1;SAVE
ERR 22,5 L
ILLEGAL PARAMETER

MPE IIB ERROR MESSAGES

:FILE A;REC=.,G;DISC=10000,16,1;SAVE
^
EXPECTED RECORD FORMAT OF F,V OR U. (CIERR 273)

PROVIDES A MEANS OF EDITING AN ERRONEOUS COMMAND ENTRY.

VALID EDIT SUB-COMMANDS:

- R - Replace (Default)
- D - Delete
- I - Insert
- U - Undo - Cancels effect of previous sub-command. Entering U, CR, then another U cancels previous sub-commands and restores line to original form.

EXAMPLE:

:STOREFILE1,FILE2,FILE3,FILE4,FILE5,*T

UNKNOWN COMMAND NAME.(CIERR 975)

:REDO

STORE FILE1,FILE2,FILE3,FILE4,FILE5,*T

A TAPE FILE NAME IS REQUIRED.(CIERR 1012)

:REDO

STORE FILE1,FILE2,FILE3,FILE4,FILE5,*T

STORE FILE1,FILE2,FILE3,FILE4,FILE5,*T

D

STORE FILE1,FILE2,FILE3,FILE4,FILE5,T

STORE FILE1,FILE2,FILE3,FILE4,FILE5,;

U

STORE FILE1,FILE2,FILE3,FILE4,FILE5,T

U

STORE FILE1,FILE2,FILE3,FILE4,FILE5,*T

STORE FILE1,FILE2,FILE3,FILE4,FILE5,*T

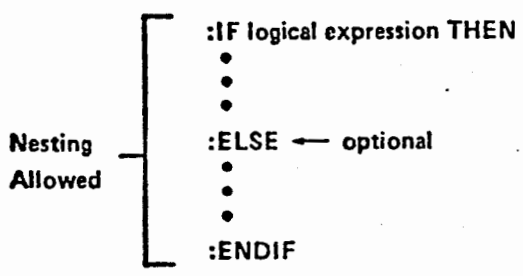


- Job control words allow two processes to communicate with each other, or a process to communicate with the Command Interpreter.
- MPE-IIB allows users to establish and manipulate user-defined Job Control Words (JCW's) and pass the system-defined Job Control Word "JCW".
- Allowing user to establish JCW's overcomes disadvantage of using system-defined JCW, which is that MPE may modify system JCW thus destroying whatever information user wished to pass.
- :IF, :ELSE, and :ENDIF commands used to control the execution sequence of steps within a job.

:IF, :ELSE, :ENDIF COMMANDS

BLOCK IF

Controls the execution sequence of steps within a job.



Relational Operators:

- >
- <
- >=
- <=
- =
- <>

Compound Expressions:

- AND
- OR
- NOT

Operands:

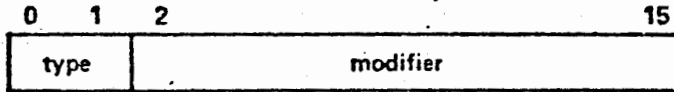
- JCW name
- OK, WARN, FATAL, SYSTEM
- Integer constants

:IF, :ELSE, :ENDIF COMMANDS (continued)

```
:RUN UPDATE1                (Run first data base update)
:IF JCW<WARN THEN           (Continue if OK so far)
:  SETJCW UPDATE1JCW:=JCW.   (Save current value of JCW)
:  RUN UPDATE2               (Run second data base update)
:  IF JCW<WARN THEN         (Continue if still OK)
:    SETJCW UPDATE2JCW:=JCW  (Save new value of JCW)
:    RUN DBSTATUS            (Report on current status of data base)
:    IF (UPDATE1JCW < 50) AND (UPDATE2JCW < 50) THEN
:      RUN DAILYRPT
:    ENDIF
:    IF ((UPDATE1JCW = 1) OR (UPDATE2JCW = 1)) THEN
:      RUN WEEKLYRPT
:    ENDIF
:  ELSE                       (Second update program failed)
:    RUN FIXUP                (Repair data base)
:  ENDIF
:ENDIF
```

JOB CONTROL WORDS (continued)

Job Control Word is a 16-bit logical.



- TYPE: 00 => OK
 01 => WARN
 10 => FATAL
 11 => SYSTEM

ABORT

*WHEN
WHERE*

System JCW — User can assign but cannot control value (MPE may modify value).
 Name is JCW.

User-defined JCW — User can assign and control value. Name is user-defined.

Accessible through intrinsics for interjob communication.

Accessible through commands for CI communication.

JCW INTRINSICS

LV
SETJCW (Word);

Allows user to establish bit content of the system Job Control Word (JCW).

L
jcw:=GETJCW;

Fetches contents of system Job Control Word (JCW).

BA I I
PUTJCW (jcwname,jcwvalue,status);

Puts value of user-defined JCW in JCW table.

BA I I
FINDJCW (jcwname,jcwvalue,status);

Searches JCW table for JCW specified in jcwname and returns value in jcwvalue.

JCW COMMANDS

:SETJCW jcwname delimiter jcwvalue

Searches JCW table for JCW specified by jcwname. If the JCW exists in the table, its value is updated. If the JCW does not exist, its name and value are added to the table.

jcwvalue can be one of the following:

1. Octal number between 0 and %177777.
2. Decimal number between 0 and 65535.
3. A name which has been defined as being equivalent to certain numeric values, as follows:

OK = 0

OK1 = OK + 1 = 1

OK100 = OK + 100 = 100 = %144

WARN = 16384 = %40000 (= OK16384)

WARN100 = WARN + 100 = 16484 = %40144

SYSTEM = 49152 = %140000

SYSTEM200 = SYSTEM + 200 = 49352 = %140310

FATAL = 32768 = %100000 (=WARN16384 = OK32768)

FATAL85 = FATAL + 85 = 32853 = %100125

4. The name of an existing JCW.

(OK < OK16383 < WARN < WARN16383 < FATAL < FATAL16383 < SYSTEM)

JCWNAME MAY BE @ TO SET ALL
JCWNAMES TO JCWVALUE

USER-DEFINED COMMANDS

- ALLOW USERS TO NAME, DEFINE AND EXECUTE SETS OF MPE COMMANDS.
- ALLOWS USERS TO REDEFINE MPE COMMANDS.
- NO SPECIAL CAPABILITY REQUIRED.
- USERS CREATE AND KEEP UDC CATALOGS IN THEIR OWN EDITOR FILES.
- PARAMETERS CAN BE INCORPORATED INTO USER'S COMMAND.
- SPECIAL LOGON EXECUTION/NOBREAK FEATURES.
- TABLE OF UDC USERS AND CATALOGS KEPT IN FILE COMMAND.PUB.SYS.
- UDC'S CAN BE DISABLED BY PURGING COMMAND.PUB.SYS.



Used by user-defined commands as a directory of all user-defined command catalogs in the system. This file is not part of MPE. If purged, it must be rebuilt by the System Manager. User-defined commands require that it have a record size of 20 words.

:BUILD COMMAND;REC=20

1023 records default.

EXAMPLE:

UDC DEFINITION

HEADER	{ SORT PARM1, OPTION LIST,BREAK	PARM 2 = \$STDLIST
BODY	{ FILE INPUT = !PARM1 FILE OUTPUT = !PARM2 RUN SORT.PUB.SYS	

UDC USE

```
:SORT DATAIN  
> KEY 1,10  
> END
```

old file DATAIN is sorted;output file is \$STDLIST

CREATING UDC's

Use the Editor to create UDC's.

```
:EDITOR
HP32201A.7.OH EDIT/3000 WED, MAR1, 1978, 8:49 AM
(C) HEWLETT-PACKARD CO. 1976
/A
```

- 1 EQ
- 2 OPTION LIST
- 3 FILE LP;DEV=LP
- 4 FILE T;DEV=TAPE
- 5 **** ← Separates UDC's
- 6 P P1,P2=SNULL,P3=SNULL,P4=SNULL
- 7 PURGE !P1
- 8 PURGE !P2
- 9 PURGE !P3
- 10 PURGE !P4
- 11 ****
- 12 L
- 13 OPTION LOGON,NOBREAK
- 14 RUN STUDENT
- 15 BYE
- 16 //
- ...
- /K UDCA
- /E

* MUST BE FIRST CHARACTER

*Restricts
student to
runny success
only then log off*

END OF SUBSYSTEM

EXECUTING UDC's

1. Use the :SETCATALOG command to set the UDC file into the command catalog.

```
:SETCATALOG [catname[,catname[,...[,catname]]]] [;SHOW]
```

```
:SETCATALOG UDC4
```

2. Enter the UDC name.

```
:EQ
```

```
FILE LP;DEV=LP
```

```
FILE T;DEV=TAPE
```

Once "set", a UDC file cannot be purged or kept under same name in Editor until released by enter :SETCATALOG with no udcname parameter.

```
:SETCATALOG UDC1
```

```
:PURGE UDC1
```

```
EXCLUSIVE VIOLATION: FILE BEING ACCESSED (FSERR 90)
```

```
UNABLE TO PURGE FILE UDC1.PUB.MPETEST. (CIERR 384)
```

```
:SETCATALOG
```

```
:PURGE UDC1
```

- A common UDC could be set up. Eg in CAT.PUB.SYS
Each user would enter SETCATALOG CAT.PUB.SYS
Bits set in JIT, JDT indicating UDC user

- XDS allocated as file system buffers for use
by CI to read UDC file

- NON SUPPORTED UTILITY UDCWHO2 TO PRINT
COMMAND FILE

:SHOWCATALOG [listfile]

Lists all UDC'S in each UDC catalog file

:SHOWCATALOG

UDC4.PUB.MPETEST

EQ

P

L

:HELP EQ

USER DEFINED COMMAND:

EQ

OPTION LIST

FILE LP;DEV=LP

FILE T;DEV=TAPE

UNCL REVIEW

NEW MPE COMMANDS:

```
:HELP [ HELP
        tablecontents
        command [,keyword]
        ALL
        EXIT ]
```

```
:REDO
:IF logical expression THEN
:ELSE
:ENDIF
:SETJCW jcwname delimiter jcwvalue
:SHOWJCW [jcwname]
:SETCATALOG [catname[,catname[,...[,catname]]]] [SHOW]
:SHOWCATALOG [listfile]
```

NEW MPE INTRINSICS:

```
          BA      I      I
PUTJCW (jcwname,jcwvalue,status);

          BA      I      I
FINDJCW (jcwname,jcwvalue,status);
```



MISCELLANEOUS ENHANCEMENTS

- **STREAM FILES MAY BE EITHER NUMBERED OR UNNUMBERED.**
- **NEW COMMAND (:SHOWME) DISPLAYS JOB/SESSION STATUS.**
- **NEW INTRINSICS (FMTCALENDAR, FMTDATE, FMTCLOCK) TO FORMAT TIME AND DATE INFORMATION.**
- **EDITOR**
- **FCOPY**
- **GENERIC NAMES**
- **STORE/RESTORE**
- **LISTDIR2.PUB.SYS**
- **SYSDUMP (file subsets can be specified)**

:SHOWME

USER: #S18,MGR.MPETEST,PUB (NOT IN BREAK)

MPE VERSION: HP 32002B.00.84

CURRENT: THU, MAR 2, 1978, 7:56 AM

LOGON: THU, MAR 2, 1978, 7:56 AM

CPU SECONDS: 1 CONNECT MINUTES: 1

\$\$STDIN LDEV: 41 \$\$STDLIST LDEV: 41

.....

MORRIS

.....

LV BA

FMTCALENDAR (date,string);

Formats the calendar date obtained from the CALENDAR intrinsic.

DV BA

FMTCLOCK (time,string);

Formats the time of day obtained from the CLOCK intrinsic.

LV DV BA

FMTDATE (date,time,string);

Formats date and time.

GENERIC NAMES

@, #, and ? can be used as "wild card" characters.

@ — zero or more alphanumeric characters.

— one numeric character.

? — one alphanumeric character.

Examples:

:LISTF n@.@.@ — list all files starting with character n.

:LISTF n#@.@.@ — list all files starting with character n and ending with a number.

:LISTF n?.@.@ — list all two character files starting with n.

Generic names can be used with the following commands:

:LISTACCT

:LISTF

:LISTGROUP

:LISTUSER

:LISTVS

:REPORT

:RESTORE

:STORE



;CLEAR

WHEN ONE LISTS A FILE WITH THE **.;CHAR** OPTION ANY UNPRINTABLE CHARACTERS WILL BE REPLACED WITH A PERIOD SYMBOL UNLESS THIS OPTION IS INCLUDED IN WHICH CASE THE ACTUAL CHARACTER WILL BE PRINTED. FOR EXAMPLE ONE MIGHT USE THIS OPTION WITH AN HP2640 TERMINAL WITH DISPLAY FUNCTIONS ENABLED.

;DEBLOCK=LOGICAL RECORD LENGTH (+=WORDS!-=BYTES)

ODD BYTE LENGTH RECORDS WILL BE DEBLOCKED BY FCOPY

;EBCDICIN=(A'B;X;Y;...),EXCLUDE

WHERE A,B IMPLIES A IS STARTING POSITION AND B IS COUNT
 WHERE A:Y IMPLIES X IS STARTING POSITION AND Y IS ENDING POSITION
 WHERE EXCLUDE IMPLIES THE SPECIFIED FIELDS ARE NOT CONVERTED
 FOR EXAMPLE FCOPY CAN CONVERT MIXED PACKED DECIMAL AND EBCDIC RECORDS

;TO=(DFILE,KFILE);NEW

FCOPY WILL COPY A KSAM FILE TO A KSAM FILE WHICH IS EITHER **;NEW** OR **OLD**. IT IS NOT NECESSARY TO USE **KSAMUTIL** IN SOME CASES

;SUBSET=

SELECTS LOGICAL RECORDS UNLESS PHYSICAL RECORD LENGTH IS EQUAL TO LOGICAL RECORD LENGTH

;SKIPOF=[+N/N OR N] [, +N/N OR N]

ALLOWS POSITIONING AT EOFs BACKWARD OR FORWARD FROM CURRENT POSITION AS WELL AS ABSOLUTE EOF POSITION' CAN'T BE USED WITH LABELED TAPES

;NOUSERLABELS

WILL NOW COPY USER LABELS(DISC OR TAPE) UNLESS THIS OPTION IS USED

;SUBSET=(1,30;100;200;...;3000,15)

MULTIPLE SETS OF RECORDS(SUBSETS) CAN NOW BE DEFINED

:STORE/:RESTORE

- All volumes of a multi-volume :STORE set need not be scanned in order to :RESTORE a file that resides on the last volume of the set (this was necessary with MPE IIA).
- Each volume carries a directory. Volumes after volume 1 carry a list of files stored on preceding volumes. Last volume carries list of all files stored on each volume.
- If SHOW parameter used with :STORE, list of files stored, by volume, is displayed.
- MPE IIA data structure:

Volume 1:**//Header/Directory/File/File/File/.../File/Trailer///****Succeeding Volumes:****Header/File/File/File/.../File/Trailer///**

- MPE IIB data structure:

Volume 1**//Header/Directory/File/File/File/.../File/Trailer///****Succeeding Volumes:****Header Directory/File/File/File/.../File/Trailer///**

MPE MESSAGE SYSTEM

ELEMENTS

CATALOG.PUB.SYS

Contains all MPE messages.

CICAT.PUB.SYS

Contains Help subsystem command descriptions.

MAKECAT.PUB.SYS

Program used to build a message catalog.

FEATURES

- CATALOG CAN BE CHANGED BY SYSTEM MANAGER.
- USERS CAN ACCESS CATALOG WITH GENMESSAGE INTRINSIC.
- USERS CAN BUILD OWN CATALOGS WITH MAKECAT PROGRAM.

CATALOG.PUB.SYS

This catalog may be changed by the System Manager. Users may access this catalog through the GENMESSAGE intrinsic. Users may also build their own message catalogs with the program MAKECAT and access their catalogs with GENMESSAGE.

CATALOG.PUB.SYS is not part of MPE and resides on system disc. It is not protected and can be purged by the System Manager. MPE continues to run but has no vocabulary. Currently, CATALOG.PUB.SYS is 530 sectors. It MUST be 1 extent.

FEATURES:

- NUMBERED EDITOR FILE CONTAINING ALL SYSTEM MESSAGES.
- CAN HAVE 1-20 SETS OF MESSAGES.
- A MESSAGE SET DIRECTORY IS KEPT IN A DATA SEGMENT AND AS A USER LABEL.
- SETS MUST BE NUMBERED IN ASCENDING ORDER (BUT NOT NECESSARILY CONTIGUOUS).
- MESSAGES WITHIN A SET MUST BE NUMBERED IN ASCENDING ORDER (BUT NOT NECESSARILY CONTIGUOUS).
- FORMAT:
 - "\$SET n" DENOTES A NEW SET
 - "\$Δ" DENOTES A COMMENT
 - MESSAGES MUST BEGIN WITH A MESSAGE NUMBER
 - "%" DENOTES MESSAGE CONTINUED AFTER ISSUING C.R'
 - "&" DENOTES MESSAGE CONTINUED IMMEDIATELY
 - "!" DENOTES A PARAMETER (UP TO 5)

CATALOG.PUB.SYS (continued)**Current Message Sets:**

- \$\$SET 1 – SYSTEM MESSAGES (FORMER MESSAGE CATALOG)**
- \$\$SET 2 – CIERROR MESSAGES**
- \$\$SET 3 – MISCELLANEOUS ABORT MESSAGES**
- \$\$SET 4 – PROGRAM ERROR ABORT MESSAGES**
- \$\$SET 5 – INTRINSIC ABORT MESSAGES**
- \$\$SET 6 – RUN-TIME ABORT MESSAGES**
- \$\$SET 7 – CI GENERAL MESSAGES**
- \$\$SET 8 – FILE SYSTEM ERROR MESSAGES**
- \$\$SET 9 – LOADER ERROR MESSAGES**
- \$\$SET 10– CREATE ERROR MESSAGES**
- \$\$SET 11– ACTIVATE ERROR MESSAGES**
- \$\$SET 12– SUSPEND ERROR MESSAGES**
- \$\$SET 13– MYCOMMAND ERROR MESSAGES**
- \$\$SET 14– LOCKGLORIN ERROR MESSAGES**
- \$\$SET 15– PRIVATE VOLUME ERROR MESSAGES**
- \$\$SET 16– DS/3000 ERROR MESSAGES**
- \$\$SET 17– HELP ERROR MESSAGES**
- \$\$SET 18– *NOT USED***
- \$\$SET 19– SERIAL DISC ERROR MESSAGES**

CICAT.PUB.SYS

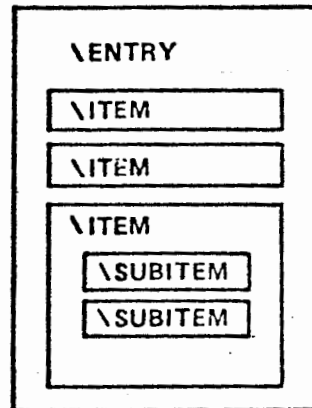
- NUMBERED EDITOR FILE CONTAINING ALL HELP MESSAGES.
- A KEYWORD ENTRY DIRECTORY IS KEPT AS A USER LABEL.
- ENTITIES ARE BROKEN DOWN INTO ENTRIES, ITEMS, AND SUBITEMS.

•FORMAT:

- A) \ENTRY= < ENTRY KEYWORD > < ITEM KEYWORD >,
 (required)
 < ITEM KEYWORD > ... < SUBITEM KEYWORD >, ...

TEXT

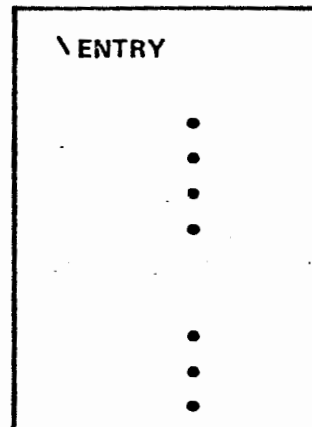
•
•
•



- B) \ITEM= < ITEM KEYWORD >

TEXT

•
•
•



- C) \SUBITEM= < SUBITEM KEYWORD >

TEXT

•
•
•

- D) \ALL TERMINATES CICAT



- E) \STOPHELP SKIPS SUBSEQUENT LINES

- F) \STARHELP STARTS DISPLAYING AGAIN

- G) \SUBSET AT BEGINNING OF FILE,
 ENABLES \STOPHELP \STARHELP PAIRS

MAKECAT.PUB.SYS

Program used to build a message system catalog. This file is part of MPE, resides on system disc, and is protected from being purged.

FEATURES:

- READS A SPECIFIED FILE (FORMALDESIGNATOR = INPUT) AND BUILDS A MESSAGE DIRECTORY INTO THE USERLABEL.
- HAS FOUR ENTRY POINTS: (RUN MAKECAT.PUB.SYS, _____)
 - NON SPECIFIED
 - CREATES A MESSAGE DIRECTORY FROM INPUT
 - BUILDS TEMP FILE,CATALOG
 - CAN BE RUN BY ANYONE TO CREATE A USER MESSAGE CATALOG
 - BUILD
 - MUST BE LOGGED ON UNDER PUB.SYS WITH SM OR OP CAPABILITY
 - CREATES A MESSAGE DIRECTORY FROM INPUT
 - BUILDS A NEW CATALOG.PUB.SYS
 - PUTS DST AND DISC ADDRESS IN SYSGLOB
 - DIR
 - MUST BE LOGGED ON UNDER PUB.SYS WITH SM OR OP CAPABILITY
 - READS FROM CATALOG.PUB.SYS ONLY
 - DOES NOT CREATE A DIRECTORY, ASSUMES ONE ALREADY BUILT
 - PUTS DISC ADDRESS IN SYSGLOB
 - HELP
 - CREATES A HELP DIRECTORY FROM INPUT
 - BUILDS A PERMANENT FILE, HELPCAT

MODIFYING CATALOG.PUB.SYS

1. Text CATALOG.PUB.SYS into the Editor.
2. Make the desired changes.
3. Keep the file under a new name and exit the Editor.
4. Log on as MANAGER.SYS and perform one of the following:
 - a. :FILE INPUT = catname,group.account
:RUN MAKECAT, BUILD
**NEW CATALOG INSTALLED
 - b. Perform a SYSDUMP (SYSDUMP runs MAKECAT, replaces CATALOG).
5. MPE keeps the old catalog as CATn where n is an archival number, CAT1, CAT2, etc. This file may be purged if it is no longer of any use.

To re-install the message catalog if MPE is printing "MISSING MSG.SET=mm.
MSG=nn", enter:

```
:HELLO MANAGER.SYS  
:FILE INPUT=CATALOG  
:RUN MAKECAT, DIR  
**NEW CATALOG INSTALLED
```



ACCESSING CATALOG.PUB.SYS WITH
GENMESSAGE INTRINSIC

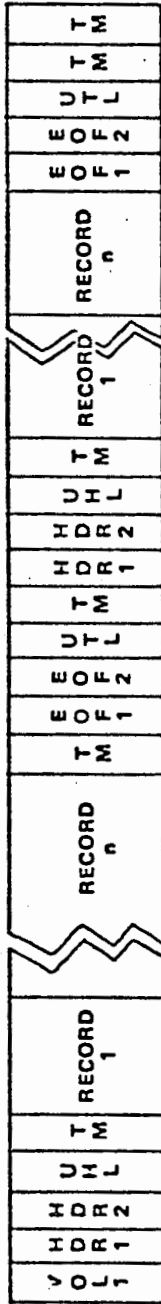
```

      I          IV   IV   IV BA   IV
msglen:=GENMESSAGE (filenum,setnum,msgnum,buff,buffsize,
                    LV   LV   LV   LV
                    parm1,parm2,parm3,
                    LV   LV   IV   I   O-V
                    parm4,parm5,msgdest,errnum);
    
```

- CCE - OK
- CCG - File system error
- CCL - Missing or invalid parameter, etc.

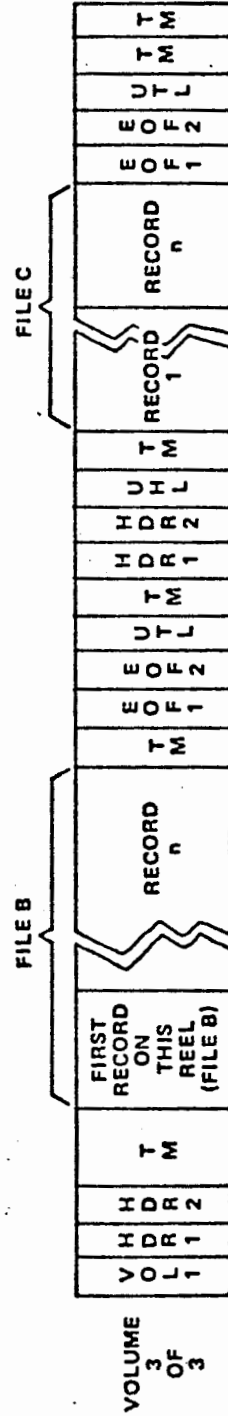
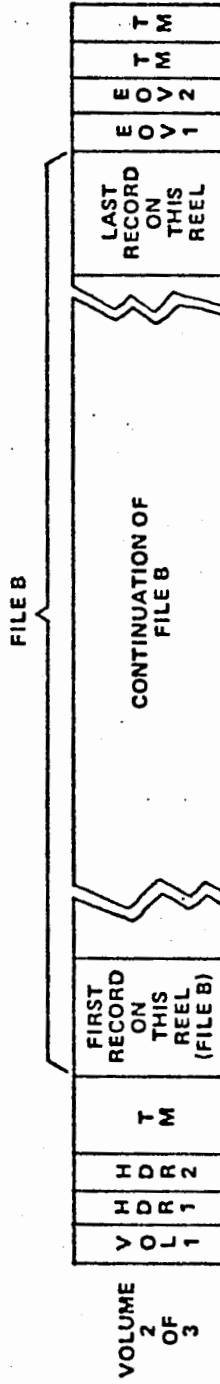
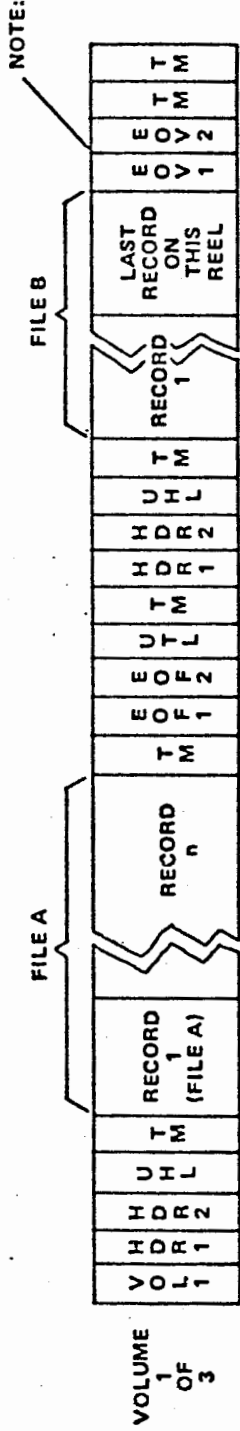
File being accessed by GENMESSAGE must be opened with foptions 5 (old permanent, ASCII), and aoptions %420 (read only, multirecord, semi-exclusive access).

MPE TAPE LABELS CONFORMING TO ANSI-STANDARD



MULTIPLE FILES ON A SINGLE VOLUME

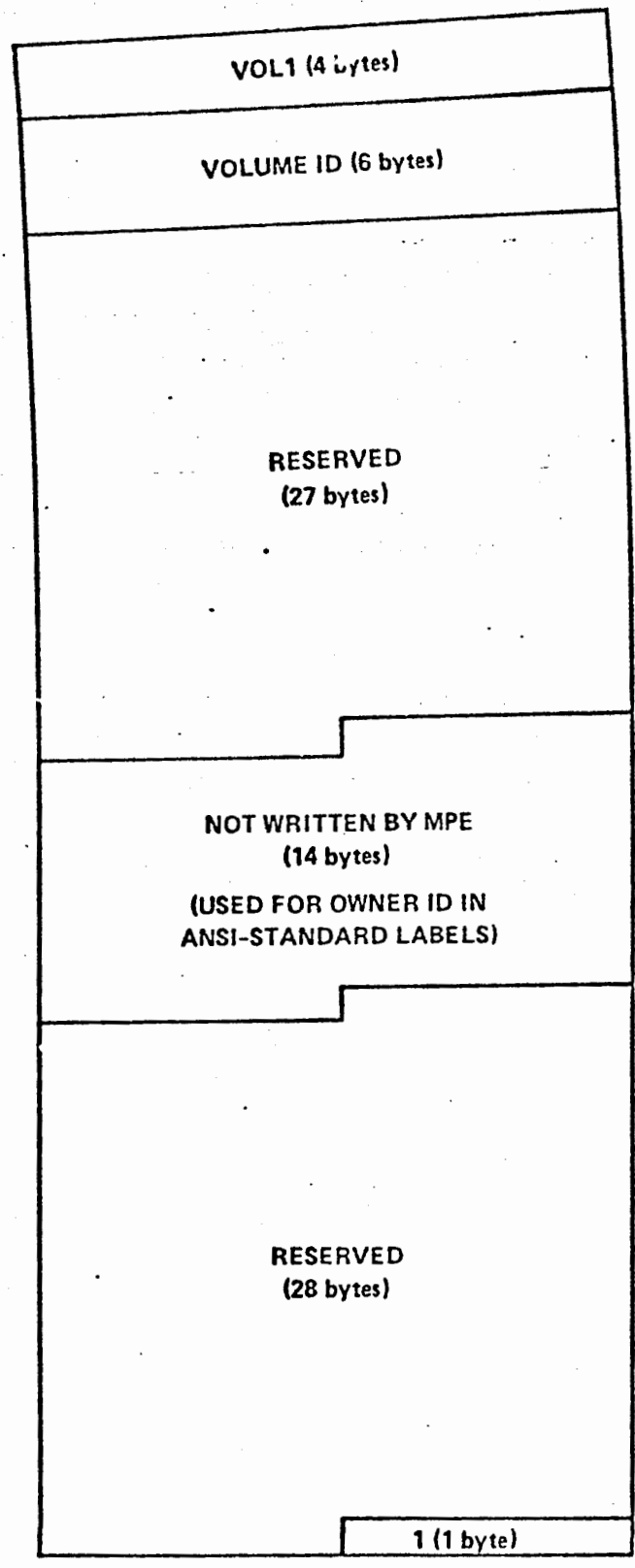
NOTE: WHEN THE FILE SPANS MORE THAN ONE VOLUME, EOF IS WRITTEN INSTEAD OF EOF.



MULTIPLE FILES ON MULTIPLE VOLUMES

VOLUME HEADER LABEL

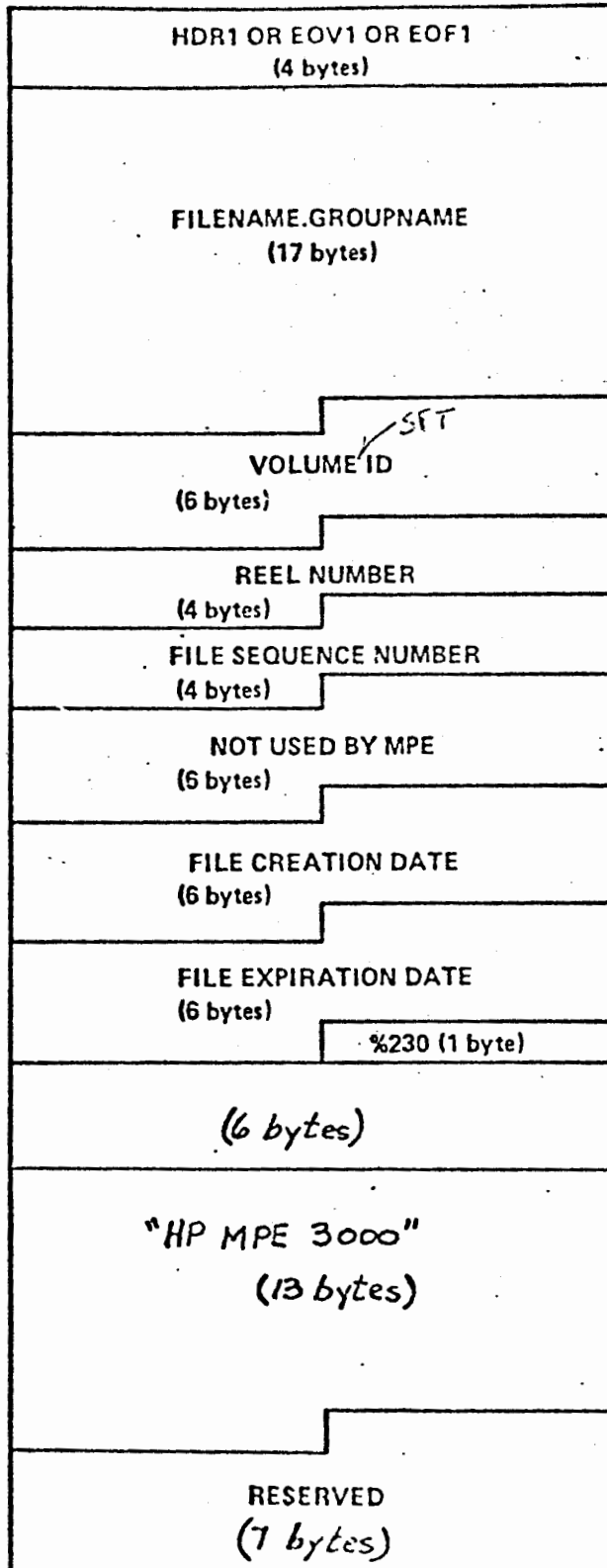
80 BYTES



Indicates that label conforms to ANSI.

HDR1/EOV1/EOF1 LABEL

80 BYTES



SFT

Indicates relative position of reel in a volume set.

Indicates relative position of file on a ~~reel~~ VOL SET

Reserved for generating data groups in ANSI-standard labels.

Indicates file created by MPE.

Not used if HDR1. Block count of data if EOVI or EOF1.

:FILE COMMAND CHANGES**:FILE formaldesignator[=filename[/lockword]]**

```
[;NOLABEL  
;LABEL={valid} [,type] [,expdate] [,seq]]]
```

- filename** Up to 17 characters, beginning with a letter. Can be fully qualified but anything over 17 characters (filename.groupname) is ignored.
- lockword** Supplied by creator to protect from illegal access. If supplied, must be specified when accessing file on tape. Ignored if non-HP created tape.
- NOLABEL** Specifies that file is not a labeled tape. (Default)
- LABEL** Specifies a labeled tape file.
- valid** Six alphanumeric characters identifying a labeled tape. (Must be exactly six characters!!!)
- type** Three characters identifying label type. ANS — ANSI-standard label
IBM — IBM-standard label.
Default is ANSI.
- expdate** Month/day/year, written in format mm/dd/yy. After this date, tape is not protected. Default is 00/00/00, meaning the file can be overwritten immediately.
- seq** Up to four numeric characters that specify the position of the file relative to other files on the tape, or one of the following:
- 0** — Causes a search of all volumes until the file is found.
 - ADDF** — Tape will be positioned so as to add a new file on the end of the volume (or last volume in a multi-volume set).
 - NEXT** — Tape will be positioned as next file on tape. If first open, then file positioned as first file on tape.

Default is NEXT.

:SHOWDEV.COMMAND CHANGES

:SHOWDEV TAPE

LDEV	AVAIL	OWNERSHIP	VOLID
7	AVAIL		
8	UNAVAIL	#S81: 1 FILES	SADSAM (ANSI)
9	AVAIL		
12	AVAIL		(NOLABEL)
13	AVAIL		TAPE01 (ANSI)
14	AVAIL		66666 (IBM)
15	AVAIL		

FOPEN INTRINSIC CHANGES

- foptions bit (6:1) — Labeled tape option.
1 — Labeled tape.
0 — No labeled tape.

formsmsg parameter used for tape label information.

[.valid] [,type] [,expdate] [,seq]

Valid must begin with a period to distinguish the parameter as specifying tape label information. Must be followed by exactly six alphanumeric characters.

Formsmsg array can contain more than 49 characters.

For Tape Label Information in a FORMSMMSG Parameter, all commas must be present.

FWRITE INTRINSIC CHANGES

CCL condition code not returned at end-of-tape for labeled tape. Instead, a mount message is sent to console requesting another reel be mounted.

FREAD INTRINSIC CHANGES

If file spans more than one volume of labeled tape, CCG not returned at end-of-tape. Instead, a mount message is sent to console requesting next reel be mounted. After reel is mounted, program will continue with the FREAD.

FCONTROL INTRINSIC CHANGES

Control Code	Operation for No Label	Operation for Labeled Tapes
5	Rewind File	Position at beginning of opened file. If the file also resides on a previous reel, a mount request for the previous reel is generated. (File is positioned at beginning of user header labels, if any.)
6	Write EOF	Not allowed for labeled tapes.
7	Space forward to tape mark	Space forward to beginning of user trailer labels, if any.
8	Space backward to tape mark	Space backward to beginning of user header labels, if any.
9	Rewind and unload tape	Not allowed for labeled tapes.



FWRITELABEL, FREADLABEL INTRINSIC CHANGES

Can write user labels on labeled tape file.

Must be 80 bytes long.

Suggest that first four bytes be UHLn or UTLn (n=number of the label).

USER HEADER LABELS

FWRITELABEL must be issued before first **FWRITE**.

FREADLABEL must be issued before first **FREAD** (first **FREAD** causes MPE to skip past any unread user header labels).

USER TRAILER LABELS

FWRITELABEL must be issued after last **FWRITE** and before **FCLOSE**.

FREADLABEL can be issued only after entire file has been read, or after the tape has been positioned at the beginning of UTL's.

FCHECK INTRINSIC CHANGES**ADDITIONAL ERRORS:**

- 116 INVALID TAPE LABEL FOPEN PARAMETERS**
- 117 ATTEMPT TO WRITE ON UNEXPIRED TAPE FILE**
- 118 INVALID HEADER OR TRAILER TAPE LABEL**
- 119 I/O ERROR POSITIONING TAPE FOR TAPE LABELS**
- 120 ATTEMPT TO WRITE IBM-STANDARD TAPE LABEL**
- 121 TAPE LABEL LOCKWORD VIOLATION**
- 123 END OF TAPE VOLUME SET**

FCLOSE INTRINSIC CHANGES

- 0** — Rewind and unload.
- 1** — Rewind and unload.
- 2** — If no label, tape is rewound and unloaded.
If labeled, tape is rewound to beginning of current file.
- 3** — No rewind. Position at next file.

- 1) Use :FILE command to provide tape label information for when file is opened (By FCOPY, a user PGM, etc.)

```
:FILE NEWTAPE1;DEV=TAPE;LABEL=FIL099,ANS,12/31/77,NEXT
```

- 2) Use FORMSMMSG parameter of FOPEN intrinsic for SPL or FORTRAN

```
BYTE ARRAY LABELID(0:79):= .FIL099,ANS,12/31/77,NEXT ;
```

```
FNO2:=FOPEN(FILID2,%1004,5,,DEV,LABELID);
```

The tape label will be written when the file is opened.

OPENING FILES ON LABELED TAPE

1. Must specify tape label information with a :FILE command (or the formsmg parameter of FOPEN intrinsic).
2. Must be an FOPEN/FCLOSE sequence for each file on a volume set. Simultaneous opens of files on labeled tapes are not allowed.

EXAMPLE:

To open file named INPUT on tape labeled TAPE01, use :FILE command as follows:

```
:FILE INDATA=INPUT;DEV=TAPE;LABEL=TAPE01,ANS,3
```

The file to be accessed will be the third file on the labeled tape.

NEW CONSOLE MESSAGES

1. hr:min/#jsnum/pin/VOL UNLABELED MOUNTED ON LDEV#n
2. hr:min/#jsnum/pin/VOL valid (ANSI) MOUNTED ON LDEV#n
3. hr:min/#jsnum/pin/VOL valid (IBM) MOUNTED ON LDEV#n

One of the above messages (depending on whether the tape is unlabeled, has an ANSI-standard label, or has an IBM-standard label) is displayed when a tape is recognized. Note that #jsnum is not displayed unless the mount was in response to a mount request (otherwise the jsnum would not be known).

4. ?hr:min/#jsnum/pin/MOUNT TAPE VOLUME valid

This message is displayed when a labeled tape is opened but has not been mounted. A reply of 0 will abort the job, otherwise the operator should mount the requested volume without any reply.

5. ?hr:min/#jsnum/pin/MOUNT VOL#n OF VOLUME SET valid ON LDEV#n

This message is displayed when a file spans more than one volume and it is necessary to switch volumes. No reply is necessary, the operator merely mounts the next volume on the specified device.

6. ?hr:min/#jsnum/pin/VOL ID FOR VOL#n OF VOLUME SET valid on LDEV#n (MAX CHARS.=6)?

This message is displayed when an unlabeled tape is mounted in response to message 5. The operator is expected to supply a six-character alphanumeric volume header identification which will be written on the tape by MPE.

7. ?hr:min/#jsnum/pin/VOLUME ID FOR filename (MAX CHARS.=6)?

This message is displayed when a labeled tape is opened and no volume header information is supplied. The operator is expected to reply with a six-character alphanumeric volume header identification.

8. ?hr:min/#jsnum/pin/OK TO WR ON UNEXPIRED VOL ON LDEV#n Y/N (MAX CHARS.=1)?

This message is displayed when a labeled unexpired tape is opened as unlabeled, or when a labeled unexpired tape is mounted in response to message 4 or 5.

PRIVATE VOLUMES

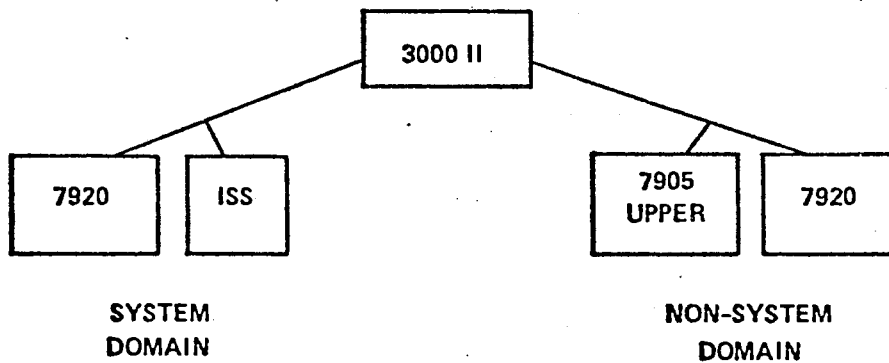
- **ALLOWS USERS TO ACCESS REMOVABLE DISC PACKS.**
- **CURRENTLY SUPPORTED ON HP 7905 (UPPER PLATTER) AND HP 7920.**
- **AUTO RECOGNITION OF DISC DRIVES.**
- **ON-LINE DISC FORMATTING.**
- **ON-LINE DISC CONDENSING.**
- **DISC-TO-DISC COPY.**
- **PRIVATE VOLUMES TRANSPORTABLE BETWEEN HP 3000 SYSTEMS.**

SYSTEM/NON-SYSTEM DOMAINS

- **ALL DISC DRIVES ARE ASSIGNED TO ONE OF TWO DOMAINS:**
 1. **SYSTEM DOMAIN** – ALL DRIVES WHICH WILL NOT (OR CANNOT) BE USED FOR PRIVATE VOLUMES. THIS INCLUDES LDEV 1.
 2. **NON-SYSTEM DOMAIN** – PRIVATE VOLUMES DISC DRIVES (HP 7905 [upper platter] AND HP 7920 ONLY).
- **DRIVES CAN BE SWITCHED BETWEEN DOMAINS AT RELOAD ONLY.**
- **AT RELOAD:**
 - SYSTEM DOMAIN DRIVES** – ON-LINE.
– IN VOLUME TABLE.
 - NON-SYSTEM DOMAIN** – OFF-LINE/ON-LINE
– NOT IN VOLUME TABLE.
– IF ON-LINE, MPE WILL ASK IF SYSTEM VOLUME.

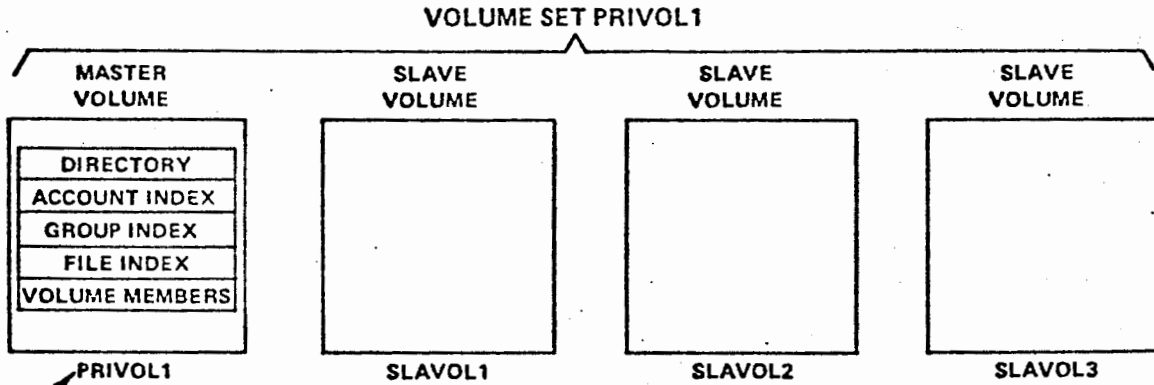


EXAMPLE:



VOLUME SET

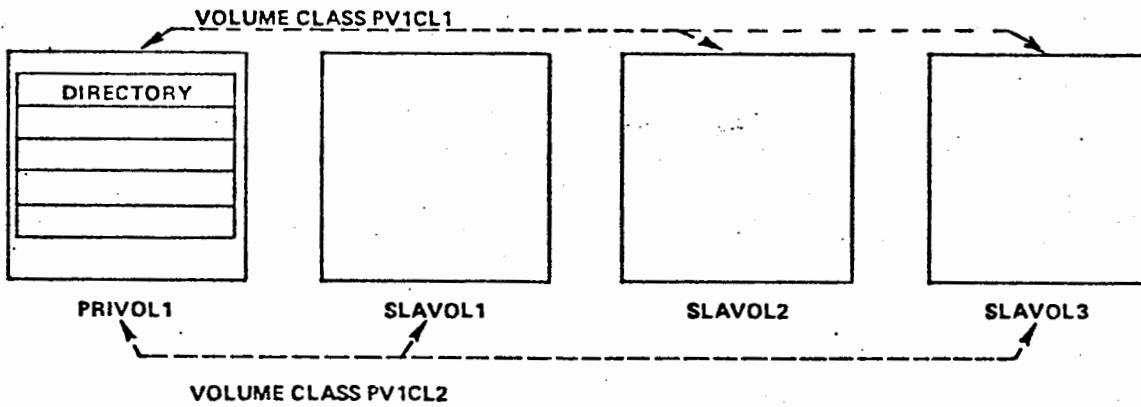
- Private volume packs can be combined into logical volume sets:
 - A volume set is a set of up to eight removable disc volumes which share a common file directory.
 - One of the member volumes of a volume set **MUST** have the same name as the volume set. This volume is the master volume.
 - The master volume contains a directory similar to the system directory, but reflecting only those accounts, groups, and files residing on this volume set.
- Information in the volume directory includes:
- A list of all accounts sharing the volume set.
 - A list of all groups sharing the volume set.
 - A list of all files residing on the volume set.



← Note that master volume has same name as volume set.

VOLUME CLASS

- Volume sets can be divided into subdivisions, called volume classes.
- One of the members of a volume class **MUST** be the master volume of the set. Thus, a single volume can be a volume class only if it is the master volume.
- A volume class is identified by a unique name and is accessed as a unit. It is the smallest volume unit that can be referenced by a user.



HOME VOLUME SET

- The home volume set is the set where the logon group's files reside.
- All files of a private volume group reside on that group's "home volume set."
- All files of a non-private volume group reside in the system domain (system volume set).
- A Volume set can contain Multiple accounts and groups.

The following steps must be performed to allow users to create/initialize private volumes:

- 1) RELOAD
- 2) Creating a volume set/class definition
- 3) Volume conditioning
- 4) Assigning private volumes capability

RELOAD (continued)

(PRINTS ONLY)

The applicable portions of the Initiator-User dialog, and the appropriate user responses, are shown below. User response is underlined.

STEP NO.	DIALOG AND USER RESPONSE
3.1	I/O CONFIGURATION CHANGES? <u>YES</u>
3.2	LIST I/O DEVICES? <u>YES</u>
3.5	LOGICAL DEVICE #? Enter the logical device number of the disc drive to be assigned to the non-system domain.
3.6	DRT #? Enter the DRT entry number of the disc drive to be assigned to the non-system domain.
3.7	UNIT # Enter the physical hardware unit number of the device, if the device shares its controller with other devices.
3.9	TYPE? <u>0</u>
3.10	SUB-TYPE? Enter 4 for HP 7905, 8 for HP 7920.
3.40	RECORD WIDTH? <u>128</u>

RELOAD (continued)

STEP NO.	DIALOG AND USER RESPONSE
3.41	OUTPUT DEVICE? <u>0</u>
3.42	ACCEPT JOBS OR SESSIONS? <u>NO</u>
3.43	ACCEPT DATA? <u>NO</u>
3.44	INTERACTIVE? <u>NO</u>
3.45	DUPLICATIVE? <u>NO</u>
3.46	INITIALLY SPOOLED? <u>NO</u>
3.50	DRIVER NAME? <u>IOMDISC1</u>
3.70	DEVICE CLASSES? <u>PVDISC</u>
3.71	IS PVDISC A SERIAL DISC CLASS? <u>NO</u>
5.3	DISC VOLUME CHANGES? <u>YES</u>
5.8 (r)	NON-SYSTEM VOLUME ON DEVICE # ADD TO SYSTEM VOLUME SET? <u>NO</u>

NOTE 3.71 YES Means serial only
 NO Means both (serial, PV)



CREATING A VOLUME SET/CLASS DEFINITION

1. Must have System Manager capability, or Account Manager and Create Volumes (CV) capabilities.
2. Use :NEWVSET command.

```
:NEWVSET vsname  
          [;MEMBERS=vname:type[,...[,vname:type]]]  
          [;CLASS=vcname:vname[,vname[,...[,vname]]]]
```

Only types allowed are HP 7905 and HP 7920. (Must include HP, not just number.)

EXAMPLE:

```
:NEWVSET PRIVOL1;MEMBERS=PRIVOL1:HP 7920, &  
:SLAVOL1:HP 7905,SLAVOL2:HP 7920;CLASS=PV1CL1:PRIVOL1,SLAVOL2
```

Creates a new entry in the system directory for

PRIVOL1.PUB.SYS

NOTE: PRIVOL1.PUB.SYS is merely an identifying name and does not imply that files on this volume set must belong to PUB.SYS (PUB.SYS only signifies the log-on group and account under which the definition was created.)

ALTERING A VOLUME SET/CLASS DEFINITION

1. Must have System Manager capability, or Account Manager and Create Volumes (CV) capabilities.
2. Use :ALTVSET command.

```
:ALTVSET vsname  
    [;ADDCLASS=vcname:vname[,vname[...[,vname]]]]  
    [;EXPANDCLASS=vcname:vname[,vname[...[,vname]]]]  
    [;EXPANDESET=vcname:type[,...[,vname:type]]]
```

EXAMPLE

```
:ALTVSET PRIVOL1;EXPANDESET=SLAVOL3:HP 7920,SLAVOL4:HP 7920, &  
:ADDCLASS=PV1CL2:PRIVOL1,SLAVOL3,SLAVOL4
```

PURGING A VOLUME SET/CLASS DEFINITION

1. Must have System Manager capability, or Account Manager and Create Volumes (CV) capabilities.
2. If a volume set includes volume classes, these must be purged individually (purging the volume set does not purge any volume classes associated with the set).
3. Use the :PURGVSET command.

```
:PURGEVSET vcsname
```

```
:NEWVSET PRIVOL1;MEMBERSE=PRIVOL1:HP7920,SLAVOL1:HP7905,SLAVOL2:HP7920;A
:CLASSE=PVILL1:PRIVOL1,SLAVOL2
:LISTVS
ACCOUNT=MPTEST          GROUPE=POB
```

```
VOLSET
-----
```

```
PRIVOL1          PVILL1(0)
:PURGEVSET PRIVOL1
VOLUME SET/CLASS DEFINITION PRIVOL1 TO BE PURGED? (YES/NO)YES
:LISTVS
```

```
:NEWVSET PRIVOL1;MEMBERSE=PRIVOL1:HP7920
:ADDVSET PRIVOL1;ADDCLASSE=PVILL1:PRIVOL1
DUPLICATE NAME IN DIRECTORY. (LIERK 900)
:PURGEVSET PVILL1
VOLUME SET/CLASS DEFINITION PVILL1 TO BE PURGED? (YES/NO)YES
:ADDVSET PRIVOL1;ADDCLASSE=PVILL1:PRIVOL1
:LISTVS
ACCOUNT=MPTEST          GROUPE=POB
```

```
VOLSET
-----
```

```
PRIVOL1          PVILL1(0)
:PURGEVSET PVILL1
VOLUME SET/CLASS DEFINITION PVILL1 TO BE PURGED? (YES/NO)YES
:LISTVS
ACCOUNT=MPTEST          GROUPE=POB
```

```
VOLSET
-----
```

```
PRIVOL1
```


VOLUME CONDITIONING

Once a volume set definition has been created, a disc volume must be conditioned for each member of the volume set.

To condition a disc volume:

1. Format the volume (if new or wiped out).
2. Initialize the volume (give the volume a name).

Both operations can be performed with the VINIT subsystem.

VINIT SUBSYSTEM COMMANDS (continued)

>INIT vname, ldev, vsname, groupname, acctname
[;GEN=genindex]

Initializes a volume. Writes the Volume Label, Defective Tracks Table, Free Space Table, and Volume Set Directory (if master volume).

1. Drive must be a non-system drive in the =DOWN state and volume must be SCRATCH.
2. Volume must have been formatted previously.

>SCRATCH ldev[;RESET]

Sets a volume to the scratch state. RESET parameter restores a scratch pack. Disc drive must be =DOWN.

>DSTAT [ldev
ALL]

Displays status of one or more disc drives. If both parameters omitted, status of all non-system disc drives is displayed.

>PDEFN [*
vsname groupname, acctname]

Displays volume set definition from the system directory.



VINIT SUBSYSTEM COMMANDS (continued)**>PLABEL Idev**

Displays the volume label of the volume mounted on Idev.

>PFSPACE Idev

Displays the free space table for the volume mounted on Idev.

>DTRACK Idev

Allows you to reassign defective tracks. Disc drive must be =DOWN.

>SERIAL Idev

Marks the volume label to indicate that the volume is a serial disc. Disc drive must be = DOWN and volume must be SCRATCH.

ASSIGNING PRIVATE VOLUMES CAPABILITY (continued)

- ACCOUNTS (cont'd)

To modify an existing account's capabilities to add CV and UV capabilities and to span the account, you could enter:

```
:ALTACCT OLDACCT;CAP=AM,AL,GL,SF,ND,UV,CV,IA,BA; &  
: VS=PRIVOL1.PUB.SYS:SPAN
```

The above :ALTACCT command added the CV and UV capabilities to the account named OLDACCT and created an account entry in PRIVOL1 volume set's directory. (The account still exists in the system directory, the SPAN parameter has no effect on this directory.)

Note that the only time you would use the SPAN parameter of the :ALTACCT command is when you plan to assign one or more of the account's groups to a private volume set, or when you plan to create a new group in the account and assign it to a private volume set.

An account may contain any mix of system domain groups and private volume groups.

The only time a group's files are assigned to a private volume set is when the group itself is assigned to that private volume set.

ASSIGNING PRIVATE VOLUMES CAPABILITY (continued)

- **GROUPS (cont'd)**

:NEWGROUP PVGROUP1;CAP=IA,BA;VS=PRIVOL1:SPAN

The above command marks the group entry for group PVGROUP1 in the system directory as a private volume group and creates a group entry for the group in PRIVOL1 volume set's directory.

The master volume of the volume set must be mounted when spanning a group into a volume set directory.

To span an existing group:

:ALTGROUP OLDGROUP;CAP=IA,BA;VS=PRIVOL1.PUB.SYS:SPAN

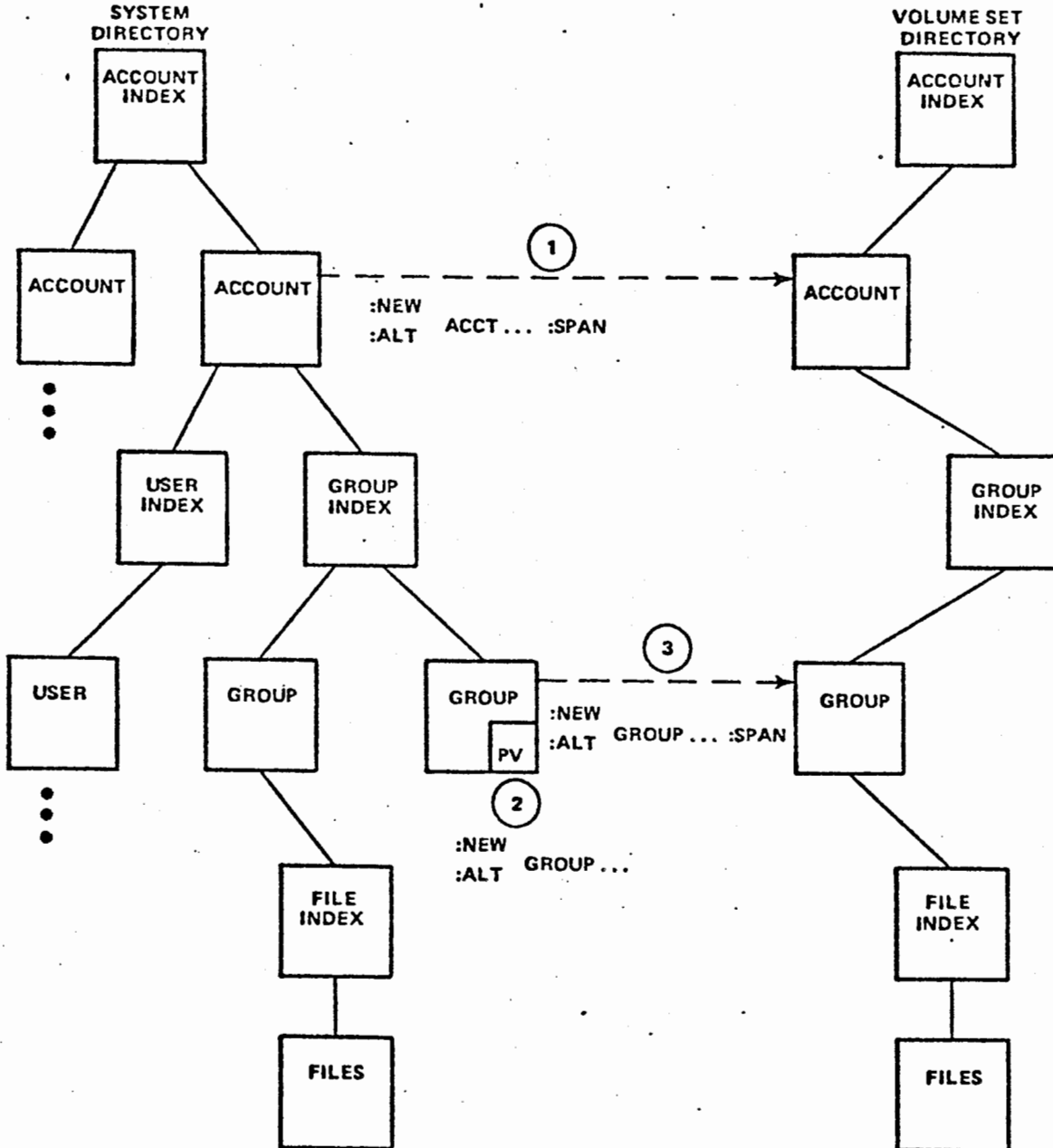
NOTE

No files are transferred to the private volume set.
An existing group which contains files cannot be spanned to a private volume set.

Before an existing group can be assigned to a private volume set with the :ALTGROUP command, the :ALTACCT command must have been entered with the SPAN parameter to create an entry for the account in the volume set's directory.

ASSIGNING PRIVATE VOLUMES CAPABILITY
(continued)

• SPANNING



- 1 Creates an account in the volume set directory.
- 2 Marks the group entry of the system directory as a private volume group.
- 3 Creates a group entry in the volume set directory.

NOTE

NO FILES ARE TRANSFERRED. AN EXISTING GROUP CONTAINING FILES CANNOT BE SPANNED.

USING PRIVATE VOLUMES

- **MOUNTING**

- PHYSICAL MOUNT**

- Volume is physically mounted on a disc drive.
 - MPE reads the volume label and prints a message on the console showing ldev and volume name.
 - A volume set is not considered to be mounted merely because one or all of its members is physically mounted. The drives are considered available (for volume switching) until logically mounted (allocated) with a user :MOUNT command or console = MOUNT command.

- LOGICAL MOUNT**

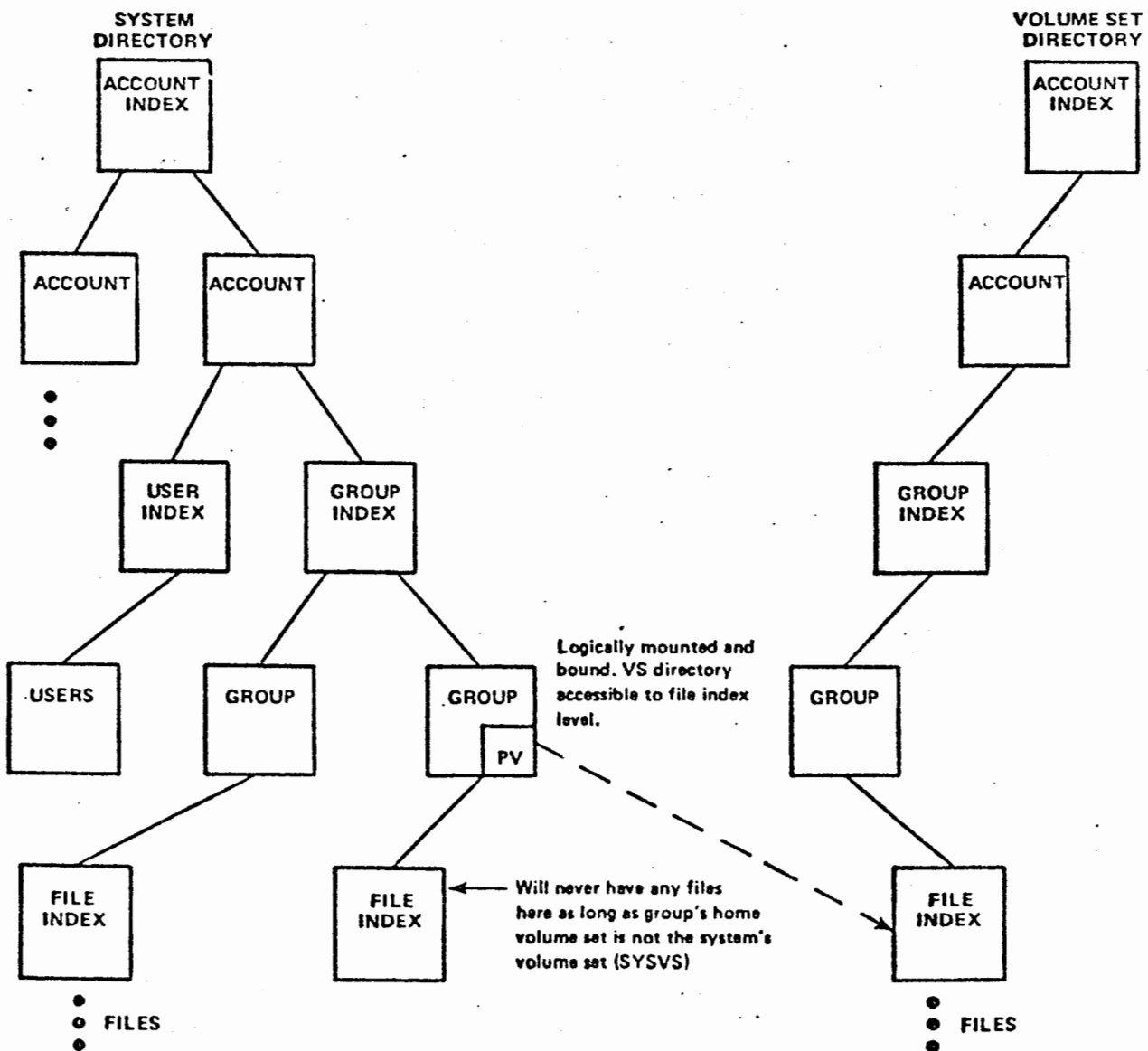
- All members of the volume set are physically mounted.
 - MPE can access the volume set directory.
 - An entry is created in the Mounted Volumes Table (MVTAB) marking the volume set as being logically mounted.
 - The drives are considered to be not available for volume switching (the drives are actually allocated to the volume set).

USING PRIVATE VOLUMES (continued)

• BINDING

When the File Index Pointer (in the system directory group entry) is set to the file index entry in the volume set directory, the volume set is said to be bound.

A volume set can be logically mounted but not have its file index linked to the system directory file index.



Logically mounted and bound. VS directory accessible to file index level.

Will never have any files here as long as group's home volume set is not the system's volume set (SYSVS)

USING PRIVATE VOLUMES (continued)

• COMMANDS

```
:MOUNT [ *  
        vcsname ] [.groupname[.acctname]]  
        [;GEN=[genindex]]
```

Mounts a volume set or class.

1. :MOUNT vcsname.groupname.acctname
 - a. Reserves the specified volume set.
 - b. Does not bind directories.
2. :MOUNT
or
:MOUNT*
 - a. Reserves the home volume set of the log-on group.
 - b. Binds directories for that group.
3. :MOUNT *.groupname.acctname
 - a. Reserves the home volume set of the specified group.
 - b. Binds directories for that group.

USING PRIVATE VOLUMES (continued)

- **EXPLICIT MOUNT REQUESTS**

Explicit mount requests are generated by the :MOUNT command or console =MOUNT command.

A :MOUNT or =MOUNT command reserves the volume set until explicitly released with a :DISMOUNT or =DISMOUNT command.

- **IMPLICIT MOUNT REQUESTS**

Various operations (FOPEN, :STORE, etc.) which require access to the home volume set will initiate mount requests. Such mounts are in effect only for the duration of the operation which requested the mount.

- **UNCONDITIONAL MOUNT REQUESTS**

Explicit mount requests, and those implicit mount requests which call FOPEN (such as a :BUILD command or a program which uses FOPEN), are unconditional. If the volume set is not physically mounted, a message is sent to the console requesting a mount, and the requesting operation suspends until the mount is completed.

- **CONDITIONAL MOUNT REQUESTS**

A conditional mount request depends on whether a volume set is physically mounted or not. If the volume set is physically mounted, a conditional mount request causes a logical mount.

If the volume set is not physically mounted, the conditional mount request is denied.

Command such as :STORE/:RESTORE generate conditional implicit mount requests.

USING PRIVATE VOLUMES (continued)**• AUTO RECOGNITION**

1. MPE automatically recognizes and reads the volume label from any disc switched on-line.
2. Console messages inform the operator:
 - a. Name of volume switched on-line.
 - b. Ldev of drive switched off-line.
 - c. Name and ldev of mounted volumes switched off-line.
 - d. Ldev of PV-available drives.
 - e. Which ldev(s) on which to mount a volume set.

EXAMPLES:

10:15/9/DISMOUNT ON LDEV# 2

10:15/9/PV01 OF PV01.PUB.SYS ON LDEV# 2

10:16/9/IN-USE VOLUME DISMOUNTED ON LDEV# 2

10:16/9/IN-USE VOLUME RE-MOUNTED ON LDEV# 2

CONSOLE OPERATOR COMMANDS

=VMOUNT { ON[,AUTO] } [;ALL]
 OFF

Enables or disables the Private Volumes Facility.

=VMOUNT OFF

Sets the Private Volumes Facility off. All requests, explicit and implicit, are rejected. Requests, even though they cannot be satisfied, are still printed on the console.

=VMOUNT ON

Sets the Private Volumes Facility on. All valid requests are allowed and the operator must reply to such requests.

=VMOUNT ON, AUTO

Sets the Private Volumes Facility on. All valid requests are allowed and MPE attempts to satisfy such requests without operator intervention (no message is sent to the console).

If the specified volume set/class is physically mounted, MPE logically mounts it. If the specified volume set/class is not physically mounted, the mount request is denied.

CONSOLE OPERATOR COMMANDS (continued)

=VMOUNT _____; ALL

Causes all private volume related messages to be printed on the console, including those not requiring operator intervention.

INITIAL DEFAULT IS VMOUNT OFF; ALL

=MOUNT { vcsname } .groupname.acctname[:GEN=genindex]

Logically mounts a volume set/class. No binding occurs with this command.

=DISMOUNT { vcsname } .groupname.acctname

Logically dismounts a volume set/class.

**=DSTAT [Idev]
[ALL]**

Displays current status of disc drives.



=VSUSER

Displays users of currently-mounted volume sets.

SERIAL DISC

- **NON-SYSTEM DOMAIN DISCS TREATED AS SERIAL DEVICES.**
- **:STORE/:RESTORE AND :SYSDUMP COMMANDS CAN REFERENCE DISC DRIVE AS WELL AS TAPE DRIVE.**
- **COMPATIBLE WITH ALL MPÉ COMMANDS AND INTRINSICS. ALL USER PROGRAMS DESIGNED TO USE MAGNETIC TAPE SHOULD RUN USING SERIAL DISC WITHOUT ANY MODIFICATION.**
- **INCREASED DATA TRANSFER RATE OVER MAGNETIC TAPE.**
- **LARGER STORAGE CAPACITY OF DISCS SUCH AS HP 7920 RESULTS IN A TIME SAVINGS BY ELIMINATING THE NEED FOR FREQUENT MOUNTING OF TAPE REELS.**
- **CURRENTLY SUPPORTED ON HP 7905 (UPPER PLATTER) AND HP 7920.**

- SERIAL DISC DRIVES ARE CONFIGURED INTO SYSTEM DURING SYSDUMP OR RELOAD.

3.71 Is classname A SERIAL DISC CLASS?

This question is asked (and it is asked for each class being implicitly created by this ldev change) if the device being configured is a moving-head disc, and classname is not the name of an existing device class.

Enter YES if classname is to be used exclusively for serial disc.

Enter NO for all other classes of system and private volume discs. (Classname (if not a system-domain disc) can be used for private volumes and serial disc if answer is NO.)

3.91.1 SERIAL DISC CLASS?

This question is asked if all devices in the class are direct-access devices.

Enter YES if classname is to be used exclusively for serial disc.

Enter NO for all other device classes.

- User request for serial disc relayed to console similar to tape request. Operator required to reply with logical device number of disc drive and whether a "WRITE RING" is to be considered present or not. Ldn only results in no write ring.
- Serial disc interface does not support NO-WAIT I/O.
- Auto-recognition mechanism identifies serial disc volume, when mounted, as serial disc.
- First file begins in sector zero of track one. Each file begins on sector boundary but individual records within the file are written without regard to sector boundaries.
- Each record has its length stored as a positive byte count in the word immediately before and after the record information.
/reclen/RECORD/reclen/reclen/RECORD/reclen/reclen/ . . .
- EOF mark consists of zeros for remainder of current sector.

OVERVIEW
OF
IMAGE ENHANCEMENTS



- o REMOTE DATA BASE ACCESS
- o PERFORMANCE IMPROVEMENT
(GLOBAL DATA BASE CONTROL BLOCK)
- o RECORD AND DATASET LOCKING
- o LOGGING AND RECOVERY

NOTE: THESE ENHANCEMENTS ARE SCHEDULED FOR RELEASE DURING 1978. ACTUAL RELEASE DATES HAVE NOT BEEN CONFIRMED. THE INFORMATION CONTAINED IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE.

REMOTE DATA BASE ACCESS

PURPOSE: PERMITS PROGRAMS EXECUTING ON ONE 3000/11 TO ACCESS ONE OR MORE DATA BASES ON OTHER REMOTE 3000/11's

FEATURES AND BENEFITS

- PROVIDES SAME CAPABILITIES FOR DATA BASES THAT ARE AVAILABLE TODAY WITH REMOTE FILE ACCESS UNDER DS/3000 FOR FILES
- TRANSPARENT INTERFACE TO REMOTE DATA BASES
 - NO PROGRAM CHANGES REQUIRED TO REDIRECT ACCESS TO REMOTE DATA BASE
 - NO ADDITIONAL USER KNOWLEDGE REQUIRED TO DEVELOP REMOTE DB APPLICATIONS
- DBUTIL NOW CONTAINS MORE INTERACTIVE FEATURES
- FILE EQUATIONS DYNAMICALLY DIRECT ACCESS TO LOCAL AND REMOTE DATA BASES
- DATA BASE ACCESS FILE AVAILABLE TO STORE "CANNED" REMOTE DATA BASE LOCATIONS

REMOTE DATA BASE ACCESS
(CUSTOMER IMPACT)

o ENVIRONMENT (CUSTOMER NEEDS)

- TWO OR MORE 3000/IIs
- DS/3000 HARDWARE AND SOFTWARE ON EACH SYSTEM
- IMAGE/QUERY

o USER INTERFACE CHANGES

- IMAGE INTRINSICS RETURN NEW ERROR CODES TO ACCOUNT FOR DATA COMMUNICATION PROBLEMS
- FILE COMMANDS CAN BE USED TO IDENTIFY LOCATION OF DATA BASE
- DBUTIL CHANGES
 - o USES BOTH ENTRY POINTS AND INTERACTIVE COMMAND DIALOGUE
 - o USED TO ACTIVATE AND DEACTIVATE A DATA BASE ACCESS FILE
- QUERY MAY ACCESS REMOTE DATA BASES

NOTE: TO IMPROVE PERFORMANCE IT IS BETTER TO REMOTELY LOG ON TO THE MACHINE AND EXECUTE QUERY THERE.

- NO CHANGES TO APPLICATION PROGRAMS REQUIRED WHEN A LOCAL DATA BASE IS MOVED TO A REMOTE COMPUTER

o REMOTE DATA BASE PERFORMANCE

- EXECUTES SLOWER THAN LOCAL DATA BASE ACCESS
- SET BY DS/3000 PERFORMANCE AND IMAGE PERFORMANCE

USES

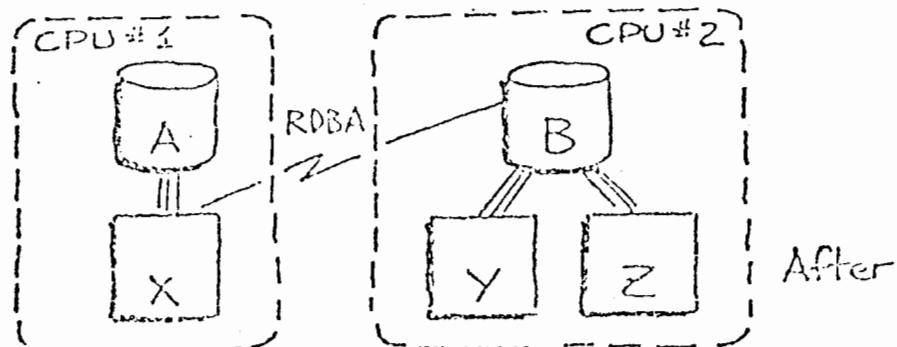
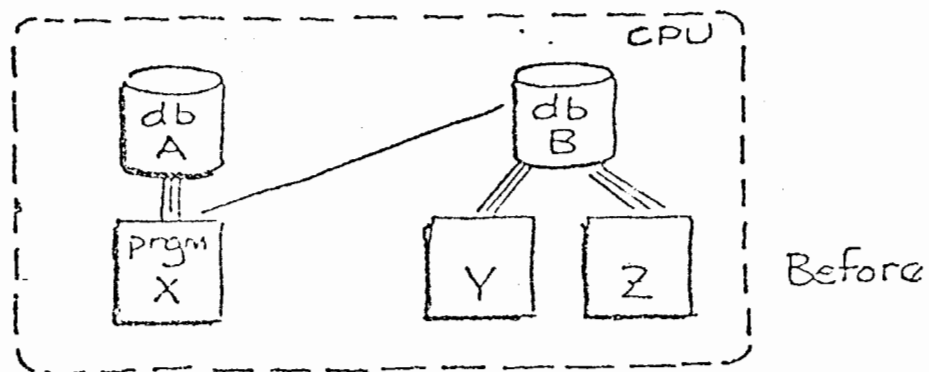
FOR OPTIMUM PERFORMANCE IN A MULTI-3000 DATA BASE ENVIRONMENT, THE DATA BASE(S) AND ACCESSING APPLICATION PROGRAMS SHOULD RESIDE AND EXECUTE ON THE SAME MAINFRAME.

WHEN TO UTILIZE REMOTE DATA BASE ACCESS

* DATA BASE LOAD SHARING



- IDENTIFY THE HIGH DEMAND APPLICATION PROGRAMS AND DATA BASE COMBINATIONS WHICH MUST EXECUTE CONCURRENTLY
- MOVE THESE COMBINATIONS TO SEPARATE MAINFRAMES TO SHARE THE LOAD
- MAY BE NECESSARY FOR SOME APPLICATIONS TO OCCASIONALLY REFERENCE REMOTE DATA BASES

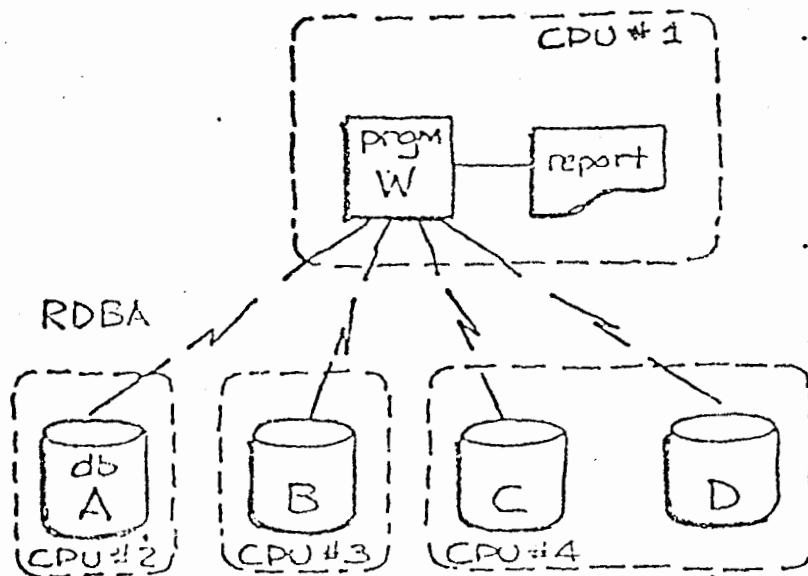


REMOTE DATA BASE ACCESS

USES (CONT.)

* CENTRALIZED ACCESS

- o SUMMARY REPORTS MAY REQUIRE INFORMATION WHICH IS STORED IN GEOGRAPHICALLY DISTRIBUTED COMPUTER SYSTEMS
- o MAIN PROGRAM "POLLS" EACH MAINFRAME IN THE NETWORK TO COLLECT A PORTION OF THE DATA NEEDED TO PRODUCE THE SUMMARY REPORT
- o CENTRAL PROGRAM MAY UPDATE SEVERAL REMOTE DATA BASES



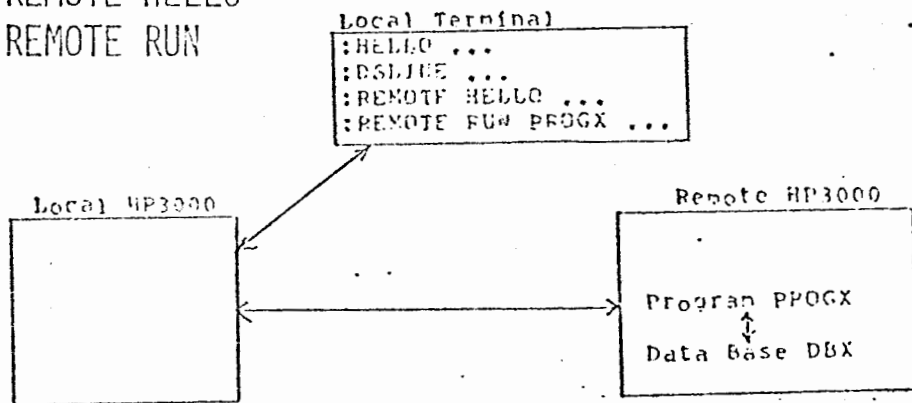
REMOTE DATA BASE ACCESS IMPLEMENTATION

* FOUR METHODS

- 1 REMOTE HELLO/REMOTE PROGRAM FILE/REMOTE DATA BASE
- 2 LOCAL APPLICATION PROGRAM/FILE COMMANDS/REMOTE DB
- 3 USING COMMAND INTRINSICS
- 4 USING A DATA BASE ACCESS FILE

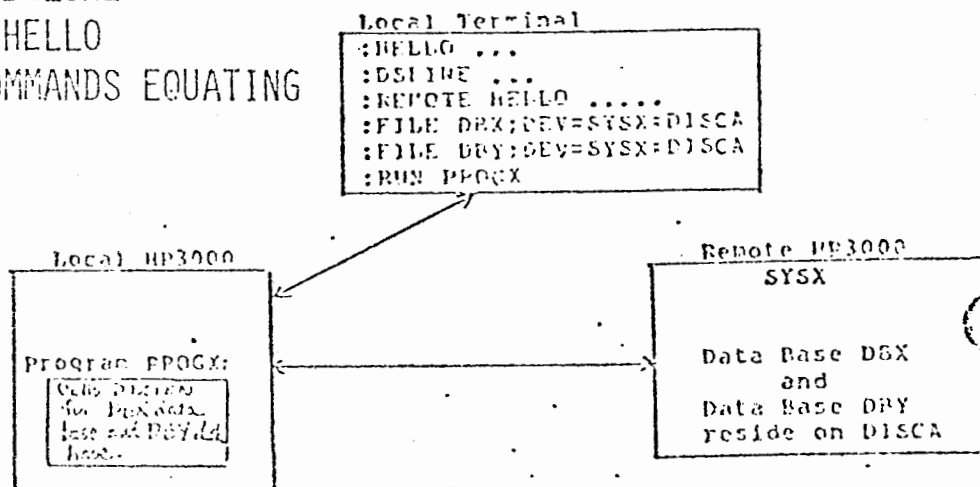
* - METHOD 1. (REMOTE HELLO/REMOTE PROGRAM/REMOTE DATA BASE)

- LOG ON TO LOCAL SYSTEM
- SET UP DSLINE (COMMUNICATIONS LINK)
- REMOTE HELLO
- REMOTE RUN



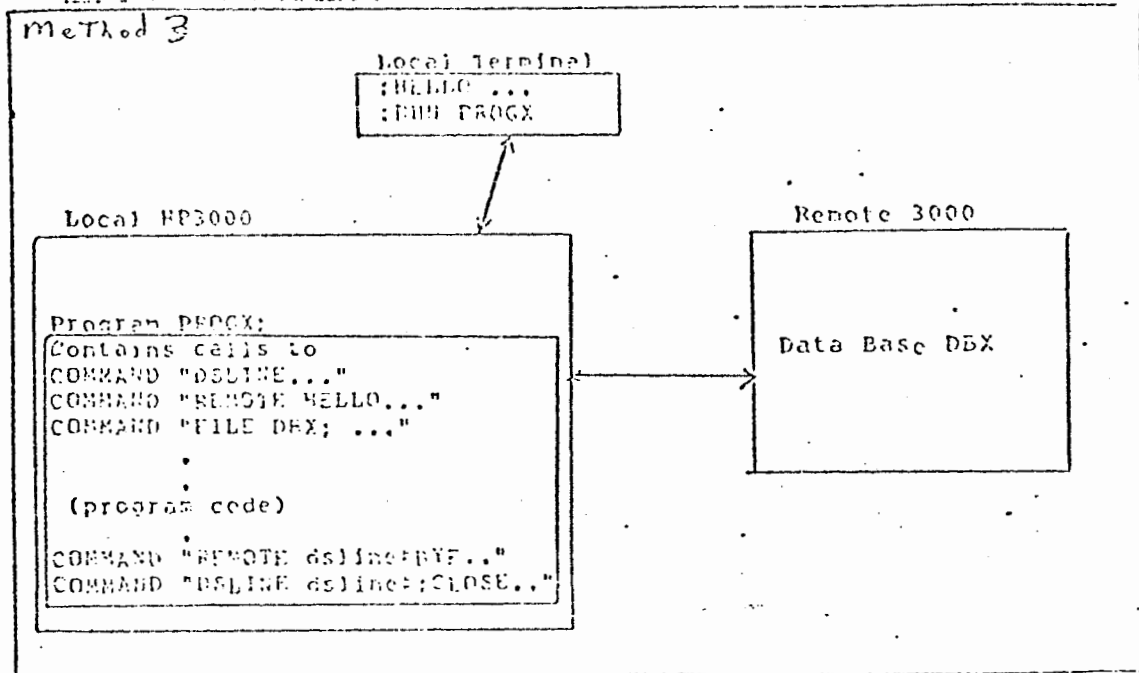
* - METHOD 2 (LOCAL PROGRAM/REMOTE DATA BASE/FILE COMMANDS)

- LOG ON TO LOCAL SYSTEM
- SET UP DSLINE
- REMOTE HELLO
- FILE COMMANDS EQUATING

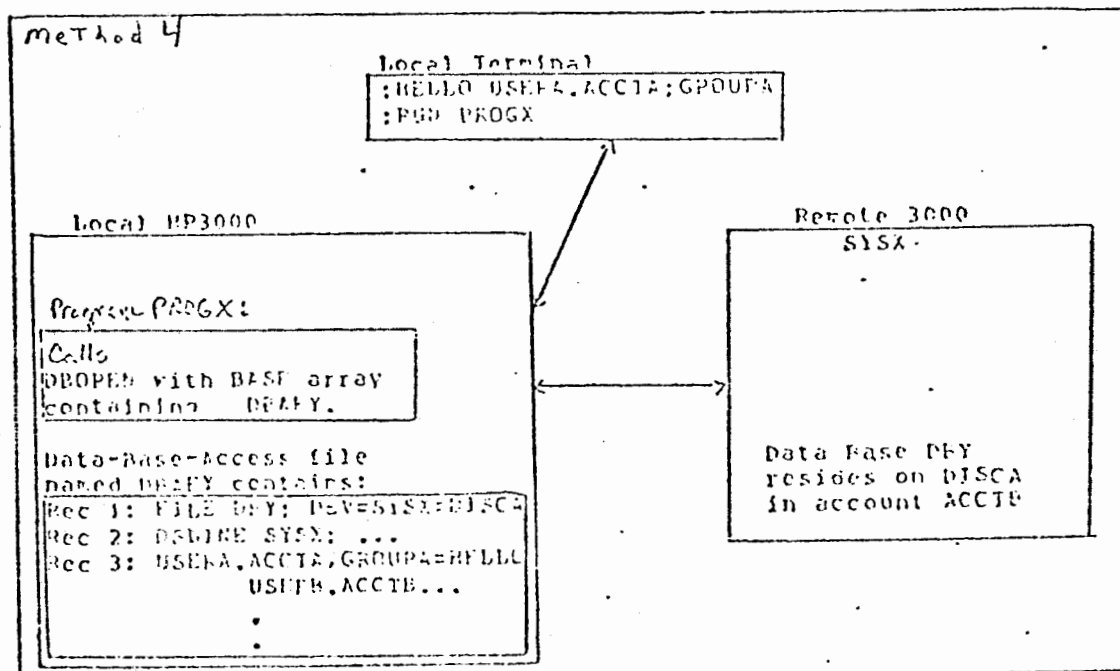


* - METHOD 3 (USING THE COMMAND INTRINSIC)

- MPE COMMAND INTRINSIC USED IN APPLICATION PROGRAM
- IF PROGRAM CODED IN COBOL, RPG, OR BASIC, A SPL OR FORTRAN PROGRAM PROCEDURE MUST BE CALLED



NOTE: ANY CHANGE IN THE DATA BASE NAME, ACCOUNT, OR PASSWORD INFORMATION REQUIRES MODIFICATION OF APPLICATION PROGRAM.





THE UNIVERSITY OF MELBOURNE
 DEPARTMENT OF COMPUTER SCIENCE
 487 PAVILLION ROAD
 PARKVILLE VIC 3010
 AUSTRALIA
 TEL: (03) 9346 3100
 FAX: (03) 9346 3101
 WWW: WWW.COMPSCI.MUNM.EDU.AU

THE UNIVERSITY OF MELBOURNE
 DEPARTMENT OF COMPUTER SCIENCE
 487 PAVILLION ROAD
 PARKVILLE VIC 3010
 AUSTRALIA
 TEL: (03) 9346 3100
 FAX: (03) 9346 3101
 WWW: WWW.COMPSCI.MUNM.EDU.AU

THE UNIVERSITY OF MELBOURNE
 DEPARTMENT OF COMPUTER SCIENCE
 487 PAVILLION ROAD
 PARKVILLE VIC 3010
 AUSTRALIA
 TEL: (03) 9346 3100
 FAX: (03) 9346 3101
 WWW: WWW.COMPSCI.MUNM.EDU.AU

DATA BASE UTILITY CHANGES

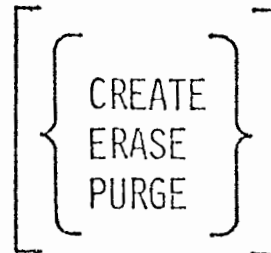
(DBUTIL)

*DBUTIL NOW USES BOTH ENTRY POINTS AND INTERACTIVE COMMAND DIALOGUE TO PERFORM AVAILABLE FUNCTIONS

*ENTRY POINTS

o FORMAT

:RUN DBUTIL.PUB.SYS,



(SAME AS ORIGINAL VERSION)

o IMAGE PROMPTS YOU FOR DATA BASE NAME AND OPTIONALLY, FOR THE MAINTENANCE WORD

*INTERACTIVE

o FORMAT

:RUN DBUTIL.PUB.SYS

>> COMMAND

o PROMPTS FOR A DBUTIL COMMAND

HELP - PROVIDES A DESCRIPTION OF ALL OTHER DBUTIL COMMANDS

CREATE - CREATES AND INITIALIZES EACH DATA SET

ERASE - REINITIALIZES DATA BASE TO EMPTY STATE

PURGE - PURGES ENTIRE DATA BASE INCLUDING ROOT FILE

ACTIVATE - PREPARES A DATA BASE ACCESS FILE TO BE USED

DEACTIVATE - DEACTIVATES A DATA BASE ACCESS FILE BEFORE MAKING CHANGES TO IT

VERIFY - USED TO DETERMINE WHETHER OR NOT A DATA BASE ACCESS FILE IS ACTIVATED OR DEACTIVATED

SET - CHANGES OR REMOVES THE MAINTENANCE WORD

EXIT - TERMINATES DBUTIL EXECUTION

NOTE: THE ABOVE COMMANDS MAY BE ABBREVIATED TO THE FIRST THREE CHARACTERS

PERFORMANCE IMPROVEMENT
(GLOBAL DATA BASE CONTROL BLOCK)

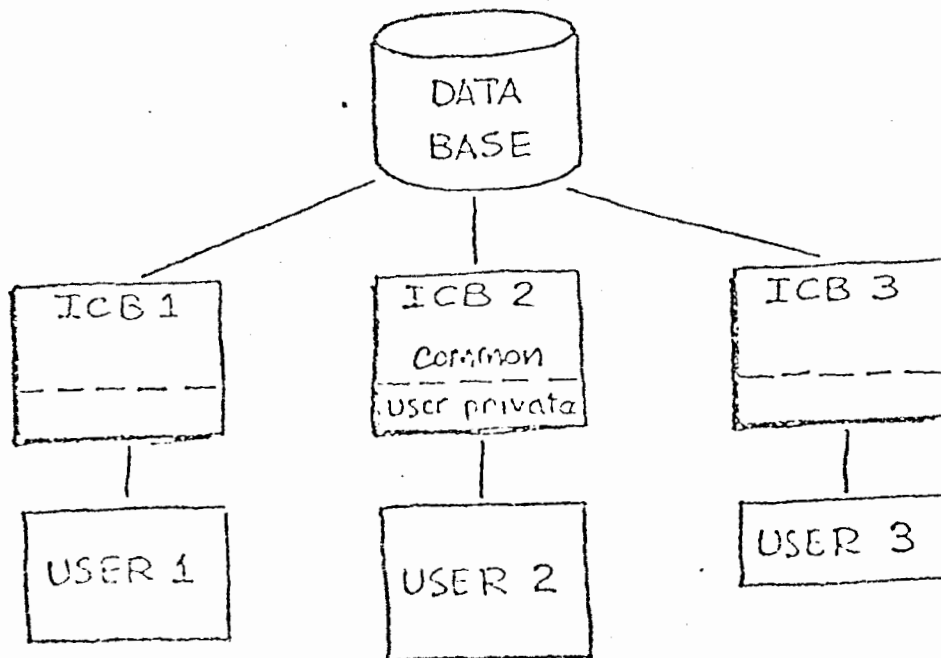
FEATURES AND BENEFITS

- o IMPROVED IMAGE PERFORMANCE
 - BETTER RESPONSE TIME FOR ANY MIX OF DATA BASE USERS
 - MORE USERS ARE POSSIBLE WITH SAME RESPONSE TIME
- o LESS LOAD ON MPE; LESS INTERFERENCE WITH CONCURRENT NON-IMAGE USERS
- o NO PROGRAMMING CHANGES OR ADDITIONAL USER KNOWLEDGE REQUIRED TO RECEIVE THESE BENEFITS

PERFORMANCE IMPROVEMENT
(GLOBAL DATA BASE CONTROL BLOCK)

*CURRENT IMPLEMENTATION

- EACH PROCESS THAT OPENS A DATA BASE HAS ITS OWN PRIVATE DBCB (EXTRA DATA SEGMENT)
- SIZE (IN WORDS) = ROOT LENGTH + 4 (BUFFER LENGTH) + TRAILER
- MUCH OF THE CONTENTS OF THE DBCB'S ARE DUPLICATED AMONG USERS (SUCH AS DATA BASE STRUCTURE INFORMATION, CURRENT NUMBER OF ENTRIES IN EACH DATA SET, ETC.)
- NO GENERAL AND EFFICIENT WAY OF COMMUNICATING AMONG PROCESSES



*NEW IMPLEMENTATION

*USES TWO STRUCTURES INSTEAD OF ONE

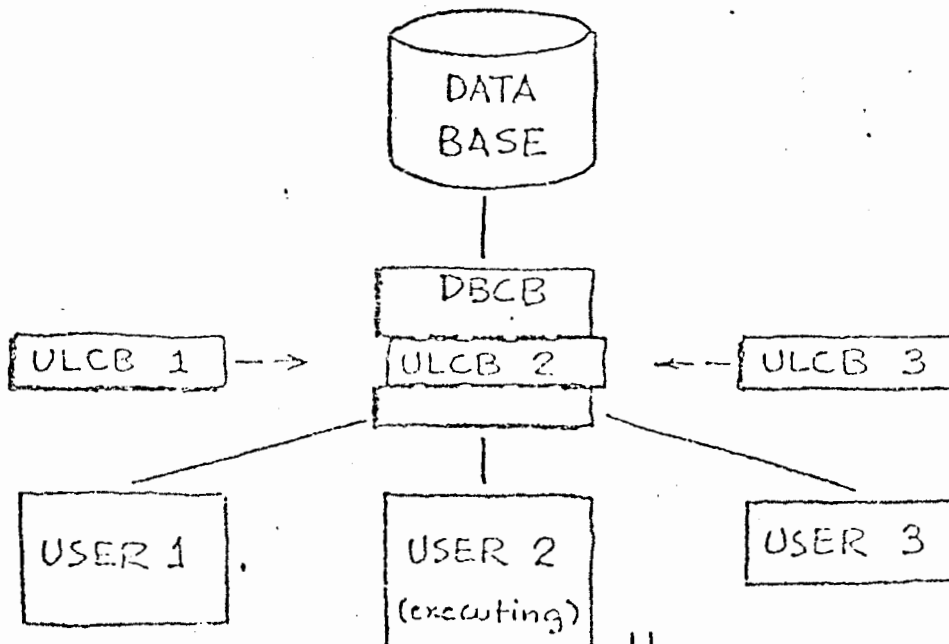
- ULCB (USER LOCAL CONTROL BLOCK)
- DBCB (GLOBAL DATA BASE CONTROL BLOCK)

*ULCB

- PROCESS PRIVATE (ONE FOR EACH OPEN DATA BASE)
- CONTAINS INDIVIDUAL USER'S DATA SUCH AS:
 - CURRENT RECORD POSITION IN EACH DATA SET
 - CURRENT LIST USED FOR EACH DATA SET
 - SECURITY INFORMATION, ETC.

*DBCB

- ONE FOR EACH OPEN DATA BASE
- CREATED WHEN FIRST USER OPENS DATA BASE
- DELETED WHEN LAST USER CLOSES DATA BASE
- CONTAINS ALL GLOBAL DATA (FIXED AND DYNAMIC)
- INCLUDES A COMMON POOL OF BUFFERS (CONTROLLED BY DATA BASE ADMINISTRATOR)



- o REDUCES TOTAL MEMORY USAGE IN MULTI-TERMINAL APPLICATIONS
 - ROOT INFORMATION WHICH WAS REDUNDANTLY KEPT FOR EACH USER IS NOW KEPT ONCE, IN THE DBCB.
 - BUFFERS FOR ALL USERS KEPT IN THE DBCB
 - SPECIFIC USER DATA KEPT IN USER'S ULCB
 - REDUCES LOAD ON MEMORY MANAGER
 - TOTAL MEMORY REQUIREMENTS ARE DECREASED (REDUCES SWAPPING)
 - ULCBs ARE SMALL, THEREFORE EASIER TO PLACE IN MEMORY
 - DBCBs MAY BE "CORE RESIDENT" IF ALL USERS ACCESSING THE SAME DATA BASE

- o REDUCES EXPLICIT DISC I/O PERFORMED BY IMAGE
 - DBCB CONTAINS THE ONLY SET OF BUFFERS FOR THAT DB
 - DBCB PROPERLY REFLECTS THE CURRENT CONTENTS OF THE DB

- o PROVIDES FOR AUTOMATIC INTER-PROCESS COORDINATION
 - UTILIZES NEW RECORD LEVEL LOCKING MECHANISM
 - IMAGE AUTOMATICALLY RESOLVES (WITHIN INDIVIDUAL INTRINSIC CALLS) THE POTENTIAL CONFLICTS THAT USED TO BE RESOLVED BY LOCKING
 - DBLOCK NO LONGER NEEDED TO PERFORM DB UPDATES

- o MORE BUFFERS AVAILABLE TO THE INDIVIDUAL USER



CUSTOMER IMPACT

*NEW VERSION OF IMAGE/3000 WILL ONLY RUN UNDER MPE-IIB

MPE-IIB
←

*MINIMAL USER INTERFACE CHANGES (A FEW NEW ERROR CODES)

*SOFTWARE IMPLICATIONS

- NO CHANGES REQUIRED TO EXISTING APPLICATION PROGRAMS
- IF DESIRED, DBLOCK AND DBUNLOCK CALLS MADE FOR THE PURPOSE OF PRESERVING DATA BASE STRUCTURAL INTEGRITY MAY BE REMOVED
- DBLOCKS AND DBUNLOCKS BRACKETING A SERIES OF OTHER IMAGE CALLS FOR THE PURPOSE OF MAINTAINING LOGICAL INTEGRITY OF THE DATA BASE SHOULD REMAIN

LOCKINGFEATURES AND BENEFITS

- o ALLOWS USERS TO GAIN EXCLUSIVE CONTROL OF A SMALL PIECE OF AN IMAGE DATA BASE
 - o PROGRAMS CAN LOCK ANY OR ALL OF THE FOLLOWING:
 - THE WHOLE DATA BASE
 - ONE OR MORE WHOLE DATA SETS
 - ONE OR MORE RECORDS IN NAMED DATA SET(S)
 - COMBINATIONS OF THE ABOVE
 - o DEADLOCK PROTECTION
 - IMAGE GUARANTEES THAT PROGRAMS WITH NORMAL CAPABILITY CLASSES WILL NEVER DEADLOCK (MULTIPLE RIN - MR SHOULD NOT BE SPECIFIED)
 - A PROGRAM MAY NOT CALL DBLOCK MORE THAN ONCE (UNLESS MR CAPABILITY - THEN IMAGE NO LONGER GUARANTEES DEADLOCK PROTECTION)
 - RECOVERY FROM A DEADLOCK REQUIRES A COOL START
 - o MULTIPLE LOCKS
 - CALLS TO DBLOCK MAY PASS ANY NUMBER OF LOCK DESCRIPTORS (PREDICATES)
 - A PROGRAM MAY GAIN CONTROL OF MANY GROUPS OF RECORDS SCATTERED ACROSS MULTIPLE DATA SETS
 - DBLOCK IGNORES ANY LOCKS THAT DUPLICATE ONES THAT ARE ALREADY SPECIFIED
 - IF USER HAS MR CAPABILITY, DBLOCK MAY BE CALLED REPEATEDLY UNTIL ALL LOCKS SUCCEED
- (DEFINITION) PREDICATE LOCKING - WHERE DATA ENTRIES TO BE LOCKED ARE DESCRIBED BY THEIR CONTENTS. (IMAGE REFERS TO A PREDICATE AS A LOCK DESCRIPTOR)

LOCKING LOGIC

- o LOCKING IS BASICALLY A MEANS OF COMMUNICATION AND CONTROL TO BE USED BY MUTUALLY COOPERATING USERS
 - IMAGE DOES NOT IN ANY WAY PREVENT A CALLING PROGRAM FROM ACCESSING A LOCKED DATA BASE ELEMENT
 - IMAGE ADVISES A PROGRAM THAT THE ENTITY IS LOCKED AND PREVENTS THE PROGRAM FROM LOCKING IT AT THE SAME TIME
 - IT IS THE RESPONSIBILITY OF THE APPLICATION PROGRAMMER TO DETERMINE THE STATUS OF THE ENTITY TO BE LOCKED
 - PROGRAM SHOULD CALL DBLOCK AND WAIT UNTIL IMAGE INDICATES THAT THE LOCK IS SUCCESSFUL BEFORE PROCEEDING

- o LOCKING DOES NOT INVOLVE READING OR WRITING TO THE DATA BASE ELEMENT TO BE LOCKED

- o IMAGE KEEPS A RECORD OF EVERYTHING THAT IS LOCKED BY ALL PROCESSES THAT HAVE THE DB OPENED (IMAGE MAINTAINS A GLOBAL LIST OF LOCK DESCRIPTORS)

- o IMAGE COMPARES THE NEWLY SPECIFIED LOCK DESCRIPTORS WITH THOSE IN THE GLOBAL LIST
 - IF CONFLICT -- RETURNS STATUS CODE AND/OR WAITS
 - NO CONFLICT -- IMAGE ADDS NEW LOCK DESCRIPTOR TO THE LIST

- o CONDITIONAL OR UNCONDITIONAL LOCKING IS AVAILABLE AT THE BASE, SET, AND RECORD LEVEL

DBLOCK (BASE, QUALIFIER, MODE, STATUS)

FUNCTION: APPLIES A LOGICAL LOCK TO A DATA BASE, A DATA SET, OR ONE OR MORE DATA ENTRIES

*BASE - NAME OF THE BASE PARAMETER USED WHEN OPENING THE DATA BASE

*QUALIFIER - CONTAINS THE NAME OF:

- AN INTEGER REFERENCING THE DATA SET NUMBER
OR
- AN ARRAY CONTAINING A DATA SET NAME
OR
- AN ARRAY CONTAINING THE LOCK DESCRIPTORS

*MODE - CONTAINS AN INTEGER INDICATING THE TYPE OF LOCK DESIRED

- MODE = 1
 - APPLIES AN UNCONDITIONAL LOCK TO THE WHOLE DATA BASE
 - RETURNS TO CALLING PROGRAM AFTER LOCK IS SUCCESSFUL
 - QUALIFIER IS IGNORED
- MODE = 2
 - APPLIES A CONDITIONAL LOCK TO THE WHOLE DATA BASE
 - RETURNS IMMEDIATELY A CONDITION CODE
 - o ZERO - INDICATES YOU HAVE LOCKED THE DB
 - o ~~≠~~ - INDICATES THE DB (OR PART OF IT) IS ALREADY LOCKED
- MODE = 3
 - APPLIES UNCONDITIONAL LOCK TO A DATA SET
 - QUALIFIER PARAMETER MUST CONTAIN:
 - o ARRAY CONTAINING LEFT JUSTIFIED NAME OF DATA SET
OR
 - o AT-SIGN (@) IN THE LEFT MOST BYTE (WHOLE DB)
OR
 - o NAME OF AN INTEGER REFERENCING THE D.S. NUMBER

DBLOCK (CONT.)

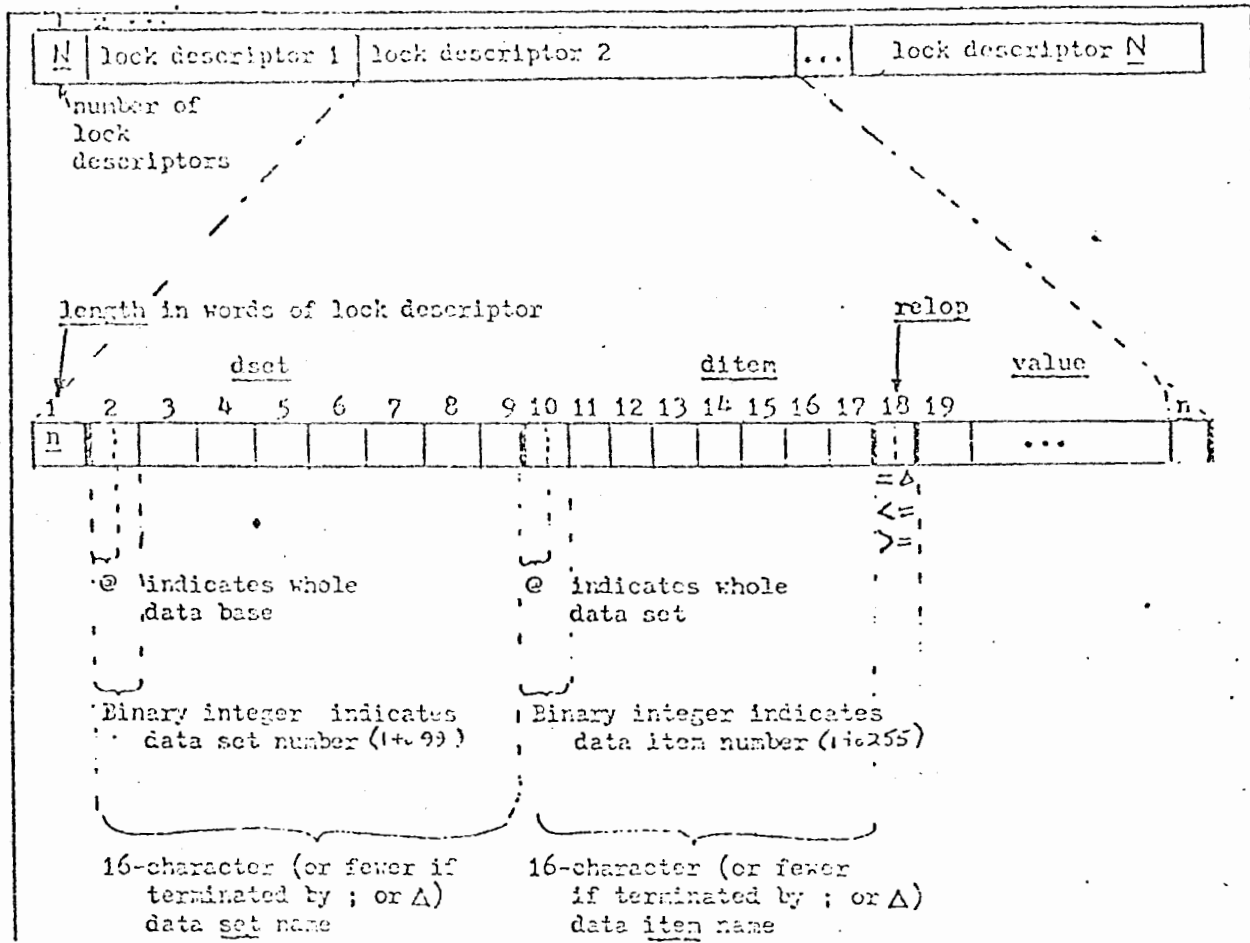


- MODE = 4
 - APPLIES A CONDITIONAL LOCK TO A DATA SET
 - QUALIFIER - SAME AS MODE 3
 - RETURNS CONDITION CODE TO CALLING PROGRAM IMMEDIATELY
 - o ZERO - DATA SET IS LOCKED
 - o ~~#~~ - DATA SET CANNOT BE LOCKED AT THIS TIME
- MODE = 5
 - APPLIES UNCONDITIONAL LOCKS TO THE DATA ENTRIES SPECIFIED BY THE LOCK DESCRIPTORS
 - QUALIFIER IS THE NAME OF AN ARRAY OF LOCK DESCRIPTORS
 - RETURNS ONLY WHEN ALL THE LOCKS HAVE BEEN ACQUIRED
- MODE = 6
 - APPLIES CONDITIONAL LOCKS OF THE
 - QUALIFIER - SAME AS MODE 5
 - RETURNS WHEN DBLOCK ENCOUNTERS A LOCK DESCRIPTOR THAT IT CANNOT APPLY
 - ALL LOCKS THAT HAVE BEEN APPLIED UNTIL THAT POINT ARE RETAINED

*STATUS - NAME OF A TEN WORD ARRAY IN WHICH IMAGE RETURNS STATUS INFORMATION

- WORD 1 = CONDITION WORD
- WORD 2 = ORDINAL NUMBER OF FIRST NON-LOCKABLE LOCK DESCRIPTOR IN MODE 6

LOCK DESCRIPTOR FORMAT



N - NUMBER OF LOCK DESCRIPTORS

LENGTH(N) - PHYSICAL LENGTH IN WORDS OF THE LOCK DESCRIPTOR
 - INCLUDES THE LENGTH FIELD (1 WORD)

DSET - ALWAYS 8 WORDS LONG

- IF FIRST CHARACTER IS A BLANK, SEMI-COLON, OR BINARY ZERO THE WHOLE LOCK DESCRIPTOR IS IGNORED

DITEM - ALWAYS 8 WORDS LONG

- IF '@' IS SPECIFIED, THE RELOP AND VALUE FIELDS IGNORED
- NEED NOT BE A SEARCH ITEM

RELOP - ONE WORD AND CONTAINS ONE OF THREE RELATIONAL OPERATORS

- o $<=$ (LESS THAN OR EQUAL TO)
- o $>=$ (GREATER THAN OR EQUAL TO)
- o $=\Delta$ (EQUAL - RIGHT CHARACTER MUST BE BLANK)

VALUE - VALUE OF THE DATA ITEM TO BE LOCKED

ADDITIONAL CONSIDERATIONS

PERFORMANCE

- OBJECTIVE IS TO PERFORM RECORD LOCK IN ABOUT THE SAME TIME AS A CURRENT DATA BASE LOCK (4-10ms)
- TIME REQUIRED TO APPLY A LOCK IS NOT DEPENDENT ON THE NUMBER OF RECORDS LOCKED
- RECORDS THEMSELVES ARE NOT TOUCHED
- A TABLE OF LOCKED ENTITIES IS KEPT IN MEMORY
- LOCKING ALL EMPLOYEES WHERE SEX = MALE, TAKES NO LONGER THAN LOCKING ONE SINGLE EMPLOYEE

CUSTOMER IMPACT



*ENVIRONMENT

- REQUIRES NEW VERSION OF IMAGE (GLOBAL DBCB's)
- REQUIRES MPE-IIB ✓ MPE-3

ALSO > 192K. Memory.

*USER INTERFACE

- ONLY EXTERNAL CHANGE TO IMAGE IS THE ADDITIONAL OF 4 NEW MODES TO DBLOCK INTRINSIC
- FIRST 2 MODES RETAIN ORIGINAL DEFINITIONS
- PROGRAMS ALREADY WRITTEN WILL REQUIRE NO CHANGES
- IF SET OR RECORD LOCKING DESIRED - PROGRAMS MUST BE CHANGED
- DATA BASE DESIGNER SHOULD ESTABLISH LOCKING STRATEGY

*OPERATIONAL CHANGES

- NONE
- ONLY PROGRAMMATIC INTERFACE IS AFFECTED

DATA BASE LOGGING AND RECOVERY

PURPOSE

- PROVIDES A MECHANISM TO LOG DATA BASE TRANSACTIONS TO TAPE OR DISC
- USES LOG (TAPE OR DISC) AS INPUT TO DB RECOVERY PACKAGE
- DB ADMINISTRATOR USES DBUTIL TO ENABLE OR DISABLE LOGGING
 - o GLOBAL FUNCTION
 - o NOT CONTROLLED BY INDIVIDUAL USERS
- RECOVERY REQUIRES:
 - o RESTORATION OF BACKUP COPY OF DATA BASE
 - o EXECUTION OF IMAGE RECOVERY SYSTEM AGAINST LOG FILE
- DATA BASE IS RESTORED BY RE-EXECUTING ALL COMPLETE TRANSACTIONS
- PARTIAL TRANSACTIONS WHICH FAILED TO COMPLETE ARE NOT RECOVERED
- UTILIZES MPE-IIB USER LOGGING SYSTEM
- RUNS ONLY ON SERIES II

FEATURES AND BENEFITS

- LOGGING EASILY ENABLED USING NEW INTERACTIVE DBUTIL
- LOGGING FUNCTIONS TRANSPARENT TO EXISTING APPLICATIONS
- LOG FILES PROVIDE A PERMANENT RECORD OF DB MODIFICATIONS (MAY BE USEFUL FOR AUDIT PURPOSES)
- MULTIPLE DATA BASES MAY LOG TO SAME LOG FILE SIMULTANEOUSLY
- "TRANSACTIONS" CONSISTING OF MULTIPLE IMAGE CALLS CAN BE IDENTIFIED (DBBEGIN - DBEND)
- ONLY COMPLETE TRANSACTIONS ARE RESTORED
- A RECOVERY/APPLICATION INTERFACE PERMITS THE PROGRAM DESIGNER TO NOTIFY END USERS OF RESULTS OF RECOVERY AND WHERE TO RESUME DATA ENTRY OPERATIONS
- MULTIPLE DATA BASES MAY BE RECOVERED SIMULTANEOUSLY
- ANY ADDITIONAL USER INFORMATION MAY BE LOGGED