MPE T-MIT
INTERNALS UPGRADE TRAINING

HEWLETT PACKARD

19447 PRUNERIDGE AVENUE, CUPERTINO, CA 95014

This package contains the internals upgrade training for MPE-Mighty Mouse, the first HP software to run on the HP3000 Series 37. MPE-Mighty Mouse will run on all systems that support MPE V/E, i.e. Series 39/40/42/44/48/64/68.

CONTENTS

SECTION 1:
USER INTERFACE

SECTION 2:
SYSDUMP/ INITIAL

SECTION 3:
MISCELLANEOUS MPE MIGHTY MOUSE TOPICS

APPENDIX A:
ERROR MESSAGES

NEW

NEW MPE COMMANDS

AND

SYSTEM STARTUP STATE CONFIGURATOR

MOUSE 13

Programmatic Creation of Sessions

:STARTSESS command

STARTSESS intrinsic

Restrictions:

- Hardwired terminal
- Free
- Job accepting
- Available to system

"Basically, a terminal that someone could walk up to and log on.

Note: User that executes command must have PS capability

Envisioned Uses:

- Startup state configurator
- UDC's
- Applications

Command Syntax:

:STARTSESS ldev;logonstring

Example:

:STARTSESS 21;KEN.OSE;HIPRI

:STARTSESS 30;KEN/PASSWORD.OSE,MMFIXES

Intrinsic Syntax:

STARTSESS(ldev,logonstr,jsid,jsnum,err);

value ldev;

logical ldev;

byte array logonstr;

integer jsid;

double jsnum;

integer array err;   << 2 words, err# in first word >>

Two New Commands

o   :LISTEQ command

o   :LISTFTEMP command

Enhancement to :WELCOME

:WELCOME from a file

:WELCOME command

o   Allows a welcome message to be taken from a file
    or still may be used as currently invoked.

o   Syntax

    :WELCOME [welcomefile]

o   Example

    Create an EDITOR file containing the message:

    ****

    Blondie - Series 64

    ****

    and kept to the file WELCFILE

    :WELCOME WELCFILE

    ****

    BLONDIE - Series 64

    ****

    :

:LISTEQ command

o Displays the user's file equations

o Syntax

:LISTEQ [listfile]

o Example

:FILE DUMPTAPE;DEV=TAPE;DEN=6250
:FILE PRETTYPT;DEV=EPOC;ENV=LP2.ENV.OSE

:LISTEQ

FILE EQUATIONS

FILE DUMPTAPE;DEV=TAPE;DEN=6250
FILE PRETTYPT;DEV=EPOC;ENV=LP2.ENV.OSE
:

:LISTFTEMP command

o   Displays file information on the user's temporary file

o   Syntax

     :LISTFTEMP [fileset][,listlevel][;listfile]

o   Display format is similar to :LISTF

o   Example 1

     :SPLPREP SOURCE1
     :BUILD TEMPFILE;REC=-72,16,F,ASCII;CODE=99;TEMP

     :LISTFTEMP

     TEMPORARY FILES FOR LARRY.OSE,CARGNONI

     $OLDPASS.CARGNONI.OSE
     TEMPFILE.CARGNONI.OSE

     :

o **Example 2**

:BUILD GFILE.HP32414A.SUPPORT;REC=-80,,F,ASCII;TEMP

:BUILD ATEMP.HP32033C.SUPPORT;TEMP

:LISTFTEMP @.@.SUPPORT,1

TEMPORARY FILES FOR LARRY.OSE,CARGNONI

ACCOUNT= SUPPORT   GROUP= HP32033C

FILENAME	CODE	------------LOGICAL RECORD--------			
		SIZE	TYP	EOF	LIMIT
ATEMP		128W	FB	0	1023 (TEMP)

ACCOUNT= SUPPORT   GROUP= HP32414A

FILENAME	CODE	------------LOGICAL RECORD--------			
		SIZE	TYP	EOF	LIMIT
GFILE		80B	FA	0	1023 (TEMP)

JOB SCHEDULING

PURPOSE: Allows user to specify a future time and date
on which a :STREAM file will enter MPE'S WAIT state.

(It may then enter EXEC state, depending on
MPE's Job Limit, Job Fence, etc.)

COMMANDS: :SCHEDJOB
:SHOWSCHED
:ABORTSCHED

JOB SCHEDULING SYNTAX

:SCHEDJOB   FILENAME [, char]

           [;AT= hh:mm]

           [;DAY= {DAY/DOM/DEM} / ;DATE= mm/dd/yy]

DAY From End month

                    OR

           [;IN= [Days] [, [Hours] [, Minutes]]]

EXAMPLES

:SCHEDJOB   FILE; AT= 18:00;DAY=MON
:SCHEDJOB   FILE; DAY= 5
:SCHEDJOB   FILE; DAY= -10
:SCHEDJOB   FILE; DATE= 7/4/84;AT=12:00
:SCHEDJOB   FILE; IN= ,5,30

BACKUP AUTOMATION

NEW COMMANDS:

:FULLBACKUP [dumptape] [,auxlistfile]

:PARTBACKUP [dumptape] [,auxlistfile]

Dumptape default is   FILE DUMPTAPE;DEV=TAPE
Auxlistfile default is   FILE SYSDLIST;DEV=$STDLIST

PROGRESS MESSAGES:

Store operation is X% complete.
Store operation is X% complete.

.

.

.

Store operation is X% complete.

SYSTEM STARTUP STATE CONFIGURATOR

- AVAILABLE ON ALL SYSTEMS THAT
  RUN MPE MIGHTY MOUSE

- ALLOWS COMMANDS TO BE EXECUTED
  JUST AS IF THE SYSTEM MANAGER HAD
  EXECUTED THEM FROM THE CONSOLE

- PERFORMED WHEN SYSTEM COMES UP

- ENSURES THAT LIMITS ARE SET,
  STREAMS ENABLED, etc

- NOT ALL MPE COMMANDS CAN BE EXECUTED

- MECHANISM IS BASED ON A SPECIAL FILE;

  ## SYSSTART.PUB.SYS

  CREATOR OF THE FILE MUST BE MANAGER.SYS

- SIMILAR IF FORMAT TO A UDC FILE

- SYSSTART FILE MAY CONTAIN UP TO 6 BLOKS

- THE HEADERS OF EACH BLOCK MAY BE;
    STARTUP
    WARMSTART
    COOLSTART
    COLDSTART
    UPDATE
    RELOAD

- THE BODY OF EACH BLOCK CONTAIN
  MPE COMMANDS

- SEPARATE EACH BLOCK WITH *'s

- COMMANDS IN THE STARTUP BLOCK ARE
  EXECUTED FIRST FOLLOWED BY THE
  COMMNADS IN THE BLOCK CORRESPONDING
  TYPE OF START OR LOAD

- OUTPUTS GO TO THE SYSTEM CONSOLE

- IT MAY NOT HAVE A LOCKWORD

- IF AN ERROR OCCURS IT WILL BE PRINTED
  ON THE SYSTEM CONSOLE AND EXECUTION
  WILL CONTINUE WITH THE NEXT COMMAND

SAMPLE SYSSTART FILE

1	STARTUP
2	STREAMS 10
3	ALLOW @.@;COMMANDS=REPLY
4	ALLOCATE PROGRAM X.PUB.ACCT
5	LIMIT 4,16
6	JOBFENCE 4
7	OUTFENCE 5
8	OUTFENCE 12;LDEV=14
9	*****
10	WARMSTART
11	STARTSESS 20;MGR.ACCT;HIPRI
12	STARTSESS 21;USER.ACCT
13	STARTSESS 22;USER.ACCT
14	*****
15	COOLSTART
16	STARTSESS 20;MGR.ACCT;HIPRI
17	STARTSESS 21;USER.ACCT
18	*****
19	COLDSTART
21	STARTSESS 20;MGR.ACCT;HIPRI
22	STARTSESS 21;USER.ACCT
23	*****

MOUSE 17

MPE CHANGES THAT RELATE
TO SYSDUMP AND INITIAL

A — TIME OF CENTURY CLOCK

B — AUTOMATIC INITIALIZATION

C — NEW CONTROL B COMMANDS

D — TABLE LOOKUP

E — I/O MAP

F — BACKUP AUTOMATION

G — SYSTEM TABLE DIALOGUE IMPROVEMENTS

H — INITIAL AND SYSDUMP
    PROGRESS MESSAGES

TIME OF CENTURY CLOCK

- EXISTS ONLY ON MIGHTY MOUSE HARDWARE

- BACKED UP BY INTERNAL BATTERY

- ALLOW'S MPE TO SET DATE AND TIME
  AUTOMATICALLY AS THE SYSTEM COMES UP

- USER CAN CHANGE TIME IF DEFAULT
  IS NOT CORRECT

- CLOCK MAY HAVE TO BE RESET FOR
  CHANGES IN DAYLIGHT SAVINGS TIME,
  IF CPU BOARD IS CHANGED
  OR IF THE SYSTEM IS A NEW INSTALLATION

- TIME IS NOT RESET AFTER A POWER FAIL

TIME OF CENTURY CLOCK DIALOGUE

- AS MPE COMES UP FROM A LOAD OR START IT WILL PROMPT THE CUSTOMER TO VERIFY THE DATE AND TIME

- TUESDAY, SEP. 10TH, 1984, 9:30AM?

- IF DATE/TIME IS CORRECT TYPE Y OR "CR"

- IF NOTHING IS TYPED THE TIME IS ASSUMED CORRECT, SINCE THERE IS A 60 SECOND TIMED READ, AND THE SYSTEM CONTINUES IT'S LOAD OR START

- IF THE DATE AND TIME IS NOT CORRECT TYPE N AND THE TRADITIONAL DATE/TIME DIALOGUE WILL BE ENTERED

- THE TOC IS UPDATED WITH THE DATE/TIME

- THE SYSTEM CONTINUES COMING UP

AUTOMATIC INITILIZATION

- A WARMSTART IS INVOKED WITH NO
  DIALOGUE WHEN THE SYSTEM IS POWERED
  ON AND THE KEYSWITCH IS PLACED IN
  THE "NORMAL" POSITION

NEW CONTROL B COMMANDS

- CONTROL MODE COMMANDS

- TEST MODE COMMANDS

CONTROL MODE COMMANDS

AR	Retry Auto Restart.
COLdstart	Perform COLdload using Load device with no dialog.
COOlstart	Perform COOlstart using Start device with no dialog.
DISc	Perform Start using Start device. Same as Start with no parameters.
DISk	Same as DISc.
DUmp	Perform Dump using the indicated or Start device.
Help	Display Help messages.
Load	Perform Load from load device or specified device with dialog.
NEWsystem	Perform Reload using Load device (only for factory MIT).
RELOad	Perform Reload using Load device with no dialog.
RUN	Run system after Control B halt.
SPeed	Perform speed sensing.
STart	Perform Warm/Coolstart from Start device or specified device with dialog.
TApe	Perform Load using Load device.
TEst	Go to Test mode.
UPDAte	Perform Update using Load device with no dialog.
WARmstart	Perform Warmstart using Start device with no dialog.

TEST MODE COMMANDS

- A[LL]

- C[PU]

- E[XIT]

- I[OMAP]

- M[EMORY]

- CH[ANNEL]

- PON

- HE[LP]

TWO NEW QUESTIONS IN
SYSDUMP AND INITIAL DIALOGUES

- **LIST DEVICE DEFAULTS?**
- **DEVICE NAME?**

LIST DEVICE DEFAULTS?

- WILL PRINT THE DEFAULT INFORMATION
  ON ALL SUPPORTED I/O DEVICES

DEVICE NAME?

- ENTER A DEVICE NAME FROM THE DEFDATA
  FILE TO TAKE ADVANTAGE OF THE
  DEFAULT CONFIGURATION INFORMATION

	Step	Dialogue
	S	LIST I/O DEVICES?
	6	LIST CS DEVICES?
NEW	6.1	LIST DEVICE DEFAULTS?
	7	HIGHEST DRT = <XXX> (MIN=X,MAX=XXX)? Note that the maximum DRT value offered is CPU-dependent.
	8	LOGICAL DEVICE #? Operator gives some response.
NEW	8.1	DEVICE NAME? Operator enters a device name from the DEFDATA file to take advantage of default information. This name must have the "HP" before the product number (e.g. HP2680A).
X = Default	9	DRT# = X? Only Initial will supply this default. It will be found by polling the physically connected devices through the IOMAP mechanism. The default DRT number will be displayed only for devices *not already* in the configuration.
X = Default	10	UNIT# = X? This default value will be offered only if the DRT default was supplied.
X = Default	11	SOFTWARE CHANNEL = X?
X = Default	12	TYPE = X?
X = Default	13	SUBTYPE = X?
X = Default	34	RECORD WIDTH = X?
	36	ACCEPT JOBS OR SESSIONS = :N?
	37	ACCEPT DATA = YN?
	38	INTERACTIVE = YN?
	39	DUPLICATIVE = YN?
	40	INITIALLY SPOOLED = YN?
	42	INPUT OR OUTPUT = IO? Will only be asked if the device is initially spooled.
	42.1	AUTO REPLY = YN? This is a new prompt for device-specific information.
	43	DRIVER NAME = name?
	48	DEVICE CLASSES = class?

DEVICE NAME	ID CODE	C H A N	T Y P E	SUB TYPE	TERMINAL TYPE	TERMINAL SPEED	REC WIDTH	OUTPUT DEV	MODE	DRIVER NAME	DEVICE CLASSES
HP7902	!0081	0	2	0			128	0		HIOFLOP0	FLOP
HP9895	!0081	0	2	0			128	0		HIOFLOP0	FLOP
HP7906RP	!FF02	0	0	10			128	0		HIOMDSC1	DISC
HP7906FP	!FF02	0	0	11			128	0		HIOMDSC1	DISC
HP7906BP	!FF02	0	0	12			128	0		HIOMDSC1	DISC
HP7911	!0204	0	3	1			128	0		HIOMDSC2	DISC
HP7912	!0208	0	3	2			128	0		HIOMDSC2	DISC
HP7914	!020A	0	3	4			128	0		HIOMDSC2	DISC
HP7920	!FF03	0	0	8			128	0		HIOMDSC1	DISC
HP7925	!FF04	0	0	9			128	0		HIOMDSC1	DISC
HP7933	!0212	0	3	8			128	0		HIOMDSC2	DISC
HP7935	!0212	0	3	8			128	0		HIOMDSC2	DISC
HP7945	!0220	0	3	5			128	0		HIOMDSC2	DISC
HP7970E	!0183	0	24	0			128	0		HIOTAPE0	TAPE
HP7974	!0174	0	24	3			128	0		HIOTAPE2	TAPE
HP7976A	!0176	0	24	1			128	0		HIOTAPE1	TAPE
HP7978	!0178	0	24	2			128	0		HIOTAPE2	TAPE
HP9140	!0240	0	3	0			128	0		HIOCTAP0	CTAPE
HP9144	!0260	0	3	3			128	0		HIOCTAP1	CTAPE
HP2608A	!2001	0	32	4			66	0	S	HIOLPRT0	LP
HP2613	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2617	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2619	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2563A	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2565	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2566	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2608S	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2680A	!2004	0	32	8			66	0	S	HIOPPRT0	LP
HP2688A	!2004	0	32	8			66	0	S	HIOLPRT0	LP
HP2893A	!0101	0	8	0			40	0		HIOCRDR0	CARDRDR
HPLPADCC	!0010	0	32	14	22	960	66	0	S	HIOASLP2	LP
HPLPATP	!000F	0	32	14	22	960	66	0	S	HIOASLP0	LP
HPLPTIC	!0011	0	32	14	22	960	66	0	S	HIOALSP0	LP
HPTERMADCC	!0010	0	16	0	10	960	40	&	JAID	HIOTERM2	TERM
HPTERMATP	!000F	0	16	0	10	960	40	&	JAID	HIOTERM1	TERM
HPTERMTIC	!0011	0	16	0	10	960	40	&	JAID	HIOTERM1	TERM

IOMAP FACILITY

– PROVIDES INFORMATION ABOUT THE PHYSICAL I/O CONFIGURATION OF THE SYSTEM DURING INITIAL(ONLY)

Step   **Dialogue**

4   I/O CONFIGURATION CHANGES? <u>Y</u>

4.1   IOMAP?
             Enter YES for a list of physically connected devices. The format is:

IMB	CHAN	DEV	DRT	UNIT	DESCRIPTION	LABEL/STATUS
0	1	0	8	0-5	TIC Terminal Controller	
			8	7	TIC Terminal Controller	
	4	1	33	0	7914 Disc Drive	MH7914U0
		2	34	0	7935 Disc Drive	NOT READY
		4	36	0	2680 Page Printer	
	12	1	97	?	7970E Mag Tape	

SYSTEM TABLE DIALOGUE IMPROVEMENTS

– FOE MPE MIGHTY MOUSE THE MINIMUM AND MAXIMUM SIZES FOR SYSTEM TABLES IS PROVIDED IN THE PROMPTS THEMSELVES

Step	Dialogue
0	MEMORY SIZE = <XXXX> (MIN=256, MAX=4096)?
75	CST = <XXX> (MIN=80, MAX=2048)?
76	EXTENDED CST = <XXXX> (MIN=16, MAX=8192)?
77	DST = <XXXX> (MIN=70, MAX=4096)?
78	PCB = <XXX> (MIN=12, MAX=1024)?
79	I/O QUEUE = <XXX> (MIN=20, MAX=1300)?
80	DISC REQUEST TABLE = <XXX> (MIN=20, MAX=900)?
81	TERMINAL BUFFERS PER PORT = <XX> (MIN=1, MAX=99)?
82	SYSTEM BUFFERS = <XXX> (MIN=8, MAX=253)?
83	SWAP TABLE = <XXX> (MIN=128, MAX=4096)?
84	PRIMARY MESSAGE TABLE = <XXX> (MIN=10, MAX=1023)?
84.1	SECONDARY MESSAGE TABLE = <XXX> (MIN=10, MAX=1023)?
85	SPECIAL REQUEST TABLE = <XXX> (MIN=10, MAX=2048)?
86	ICS = <XXXX> (MIN=256, MAX=4096)?
87	UCOP REQUEST TABLE = <XXX> (MIN=1, MAX=1024)?
88	TIMER REQUEST LIST = <XXX> (MIN=6, MAX=1023)?
89	BREAKPOINT TABLE = <XXX> (MIN=1, MAX=1024)?
90	MAX NUMBER OF USER LOGGING PROCESSES = <XX> (MIN=2, MAX=64)?
91	MAX NUMBER OF USERS PER LOGGING PROCESS = <XXX> (MIN=1, MAX=256)?
97	# OF RINS = <YYY> (MIN=XX, MAX=1024, USED=XX)?

INITIAL AND SYSDUMP PROGRESS
MESSAGES

– INITIAL NOW DISPLAYS PROGRESS
MESSAGES DURING A SYSTEM START/LOAD

```
 *Initializing of disc free space maps complete
**Recover lost disc space in progress
**Recover lost disc space complete
  Initializing of directory complete
  Loading of system files in progress
  Loading of system files complete
  Reload of user files in progress
  Reload of user files complete
  Part 1 of 6 complete - system table set-up
  Part 2 of 6 complete - SL binding
  Part 3 of 6 complete - I/O process creation
  Part 4 of 6 complete - Driver loading
  Part 5 of 6 complete - Table set up
  Part 6 of 6 complete - Process creation
```

– SYSDUMP NOW DISPLAYS HOW CLOSE TO
COMPLETION THE SYSDUMP IS. THIS MESSAGE
WILL BE DISPLAYED ON THE $STDLIST DEV.

```
Store operation is X% complete.
Store operation is X% complete.
               .
               .
               .
Store operation is X% complete.
```

The above messages will be printed every minute.

MISCELLANEOUS

CARTRIDGE TAPES

- ALL CARTRIDGE TAPES ORDERED FORM HP
  NOW COME CERTIFIED

88140LC	BOX OF 5	600ft
88140SC	BOX OF 5	150ft

- TO PREPARE A CERTIFIED CARTRIDGE FOR USE:

  :DOWN ldev#

  :VINIT

  >SERIAL ldev#

  A FOREIGN DISC.

  TO BE MADE SERIAL?(Y/N)Y

  >EXIT

  :UP ldev#

The most significant changes to MPE for Mighty Mouse have been in the User Interface. Many "friendliness features" have been incorporated in this part of MPE Mighty Mouse:

o   :WELCOME from a file
o   Programmatic creation of sessions
o   System reserved JCWs
o   :LISTEQ and :LISTFTEMP commands
o   Job scheduling capability
o   User Defined Command enhancements
o   System startup state configurator
o   Time of Century clock support

This section of the Internals Upgrade Training Package will cover each of these features. The externals of each feature will be discussed, followed by a description of its internal functioning. A list of facts to keep in mind ("gotchas") for each feature is included.

:WELCOME Message from a File

Currently, when a user of MPE wishes to add or replace the Welcome message on his system, he uses the :WELCOME command and is prompted to enter the new message. With MPE-Mighty Mouse, the user can specify an Editor file from which :WELCOME can take its message. This permits easy editing of the Welcome message before :WELCOME is used.

The new syntax of the :WELCOME command is:

        :WELCOME [welcomefile]

where welcomefile is the Editor file containing the new message. Line numbers, if included, are ignored.

If :WELCOME is used without welcomefile, it will function as it has in the past.

FACTS TO NOTE

o   After issuing a :WELCOME command with a specified welcomefile, that welcomefile will become the system Welcome message. Changes to the file will not affect the Welcome message until another :WELCOME command is issued.

o   File equations, implicit or explicit, are not allowed. For example, the user cannot do

    :WELCOME *A

    or

    :FILE A = welcfile
    :WELCOME A

INTERNALS

There are two data segments reserved for Welcome messages. DST %57 and %60 point to the two segments, and SYSGLOB %277 points to the DST entry which indicates the *current* Welcome message.

When :WELCOME welcomefile is used, the following events occur:

1. The command executor moves the welcome message from welcomefile to one of the Welcome message data segments. This segment will be the one that is not currently in use.

2. The pointer in SYSGLOB %277 will be changed from the current DST entry to the DST entry of the new Welcome message.

3. Subsequent Welcome messages will be issued from the new current data segment.

HEWLETT PACKARD

:WELCOME Message
From a File

SYSGLOB   DST   Data segment #1

%277   %57   current welcome

%60

Data segment #2

:WELCOME welcfile

welcfile   Data segment #2

"welcome to Mighty Mouse"   - - -> FREAD - - -->

SYSGLOB   DST   Data segment #1

%277   %57   old welcome

%60

Data segment #2

new current welcome
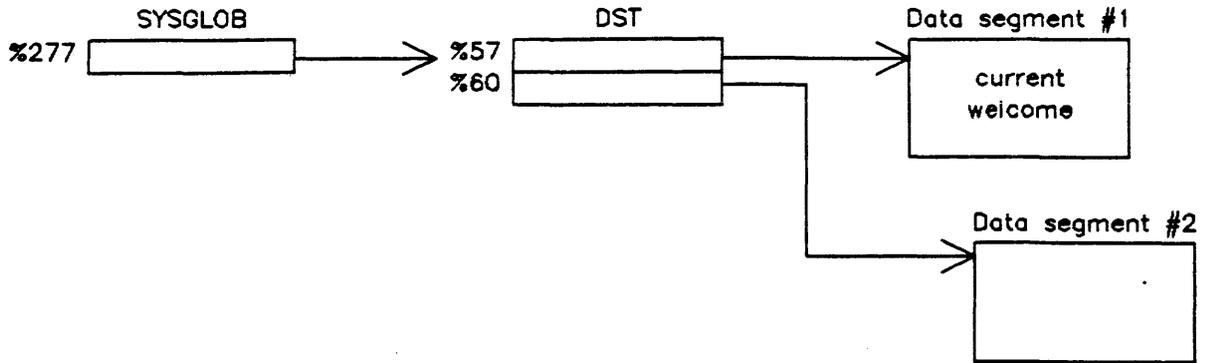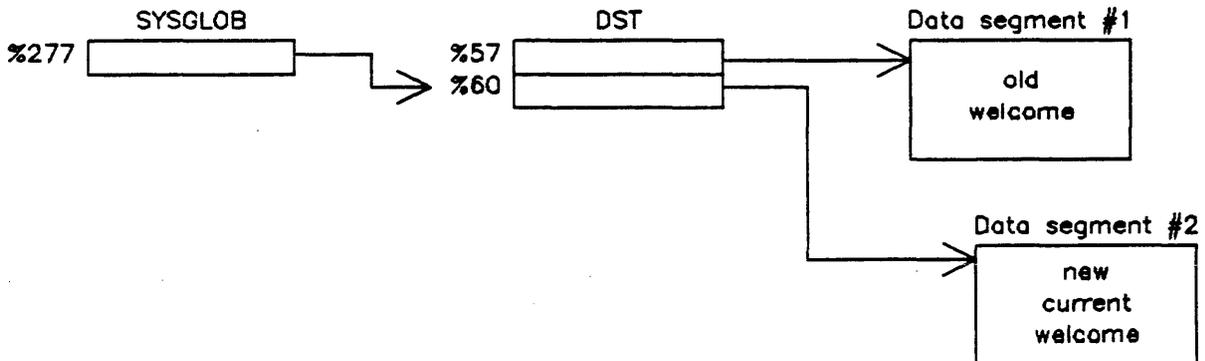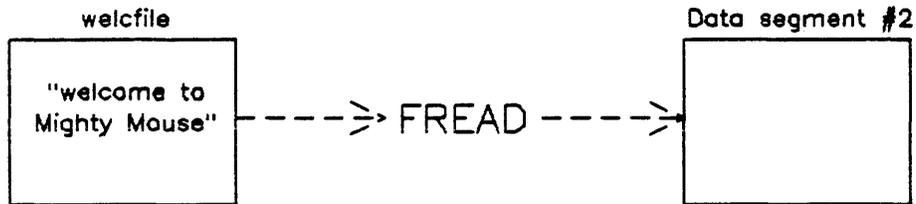
mmtring01

Programmatic Creation of Sessions

When **Programmatic Creation of Sessions** is used, a user can create sessions for himself or other users at any terminal on the system. The effect at the other terminal will be the same as if a user had logged on there himself.

Sessions may be created at other terminals either programmatically, through use of the new **STARTSESS** or **DEVLOGON** intrinsics, or through use of the new MPE command, :STARTSESS. Sessions may be aborted with the **ABORTSESS** intrinsic.

COMMAND AND INTRINSIC SYNTAX

```
              IV       BA       I      D       IA
STARTSESS ( ldev, logonstr, jsid, jsnum, errorstat );
             ->       ->      <-     <-      <-
```

The STARTSESS intrinsic will initiate a session on the specified terminal using the given logon string.

ldev   The logical device number of the terminal on which the session is to be created. This must be a valid terminal device. The device must be a physical terminal; virtual terminals will not be accepted.

logonstr   A character string holding the logon parameters in the same format expected in the :HELLO command. The first characters in the string should be the job/session name, if specified, or the user name if no job/session name is assigned. *Any passwords will have to be supplied here; there will be no prompting for passwords, so if any are required and not supplied, STARTSESS will fail.* logonstr must be terminated with a carriage return character (%15).

jsid,jsnum   Parameters returned from STARTSESS that identify the session created. The intrinsic JOBINFO requires these two parameters in this form. If STARTSESS is unsuccessful, zero values will be returned for jsid and jsnum.

                 jsid: Integer indicating the process type of the CI Main. The possible values for a job/session identifier are
                    1 = session;
                    2 = job.
                 With STARTSESS, jsid will always be assigned "1".

                 jsnum: 32-bit value which, when used with jsid, uniquely identifies the created session.

errorstat   A two-element integer array in which the status of the call is returned. The first element will contain a zero if no errors occured; otherwise, an error value is returned. Error values are listed in Appendix A. The second element is reserved for future use, and will always contain a zero.

```
              IV      BA      I      D      IA        LV
DEVLOGON (ldev, logonstr, jsid, jsnum, errorstat, waitillon );
              ->      ->     <-     <-     <-        ->
```

DEVLOGON is an *uncallable* procedure used to create sessions programmatically. It has more power than STARTSESS, and less stringent error checking. Most users should restrict their use to STARTSESS, though a few privileged applications will take advantage of DEVLOGON.

DEVLOGON has the same parameters with the same meanings as STARTSESS, with the addition of the waitillon parameter.

waitillon   A logical value that specifies when DEVLOGON will return control to the user. If waitillon is *true*, DEVLOGON will return to the caller when the created session enters the EXEC* state; if waitillon is *false*, DEVLOGON will return when the session is first introduced to the system. In either case, a detected error will cause DEVLOGON to return. A larger range of errorstat values will be available when waitillon is true.

```
             IV      DV       IA
ABORTSESS ( jsid, jsnum, errorstat );
             ->      ->       <-
```

The ABORTSESS intrinsic will abort the specified session. The parameters are similar to those of STARTSESS and DEVLOGON.

jsid   The job/session identifier in the format returned by STARTSESS and used by JOBINFO.

jsnum   The job/session number in the format returned by STARTSESS and used by JOBINFO.

errorstat   An error/status condition of the call in the same format used by STARTSESS. Possible error returns are listed in Appendix A.

:STARTSESS [ldev]; logonstring

This command will allow an MPE user to create a session on the specified terminal. The parameters have the same meanings as they have for STARTSESS, the intrinsic.

This command is programmatically executable, can be executed in break, cannot be broken, and is available in a job or session. Error conditions will receive CIERR messages and set the CIERROR Job Control Word.

FACTS TO NOTE

When using **Programmatic Creation of Sessions**, remember:

o   The creating user requires PS (Programmatic Session) capability.

o   The terminal on which the session is created must be available: no other user may be logged on, and the terminal must be turned on.

o   No speed sensing will be done for the target terminal, so it must be set at the configured baud rate.

o   When the session is created, nothing will be printed to the target terminal until its Carriage Return key is pressed.

o   Using the :STARTSESS command, the new System Startup State Configurator feature could be used to start up many sessions when the system is first brought up.

INTERNALS

When a user logs on to his own terminal, he first presses the Carriage Return key. This interrupts the terminal driver, which sets the terminal's entry in the Logical to Physical Device Table (LPDT) to "service requested" and wakes DEVREC. DEVREC sets the terminal's LPDT entry to "service granted" and sends a prompt to the terminal for the user's :HELLO string.

When a user creates a session on a terminal other than his own, the scenario is somewhat different.

Refer to slide MMTRNG03. When the STARTSESS or DEVLOGON intrinsic or the :STARTSESS command is used, DEVREC is bypassed. Instead, the following activities take place:

1. The command executor checks to make sure the target device is a terminal, that it is free and available, and that it is job/session accepting. The terminal's entry in the LPDT is set directly to "service granted" and STARTDEVICE is called.

2. STARTDEVICE will execute on the user's stack. STARTDEVICE parses the logon string, sets up a JMAT entry for the session, allocates IDD and ODD entries for the session, and calls SCHEDULEJMAT to assign a priority for the session. Then STARTDEVICE wakes UCOP.

3. UCOP checks the jobfence and session limits; if the limits are met and the new session is not being created HIPRI, it will not be allowed to start; an error will be returned. If the new session passes these limits, UCOP will allocate resources (job tables, DST entries, etc.) for a new Command Interpreter. UCOP will then call PROCREATE.

4. PROCREATE will create a Command Interpreter for the new session. It will create a stack for the new session and wake the new CI. At this point the session is in EXEC* state. *Control will return to the calling session.*

5. The CI will call INITJSMP to initialize the job tables for the new session. After initializing the tables, INITJSMP will wait until the Carriage Return key is pressed on the target terminal.

6. When Carriage Return is pressed, the new session will log on. It will enter the EXEC state, the welcome message will be printed, and any UDCs will be initialized.

7. From this point, the session is identical to any non-programmatically created session.

When STARTSESS is used, the user cannot use the waitillon parameter; STARTSESS will always function as if waitillon is true. Users of DEVLOGON can specify a value of true or false for waitillon.

If waitillon is true and an error occurs by the time UCOP is awakened, UCOP will send an error message to the caller via IPC. If an error occurs past UCOP, the CI will send an error message to the caller via IPC. Once the session is waiting for its Carriage Return on its own terminal, it cannot communicate with its creator.

If waitillon is false, DEVLOGON releases control of the session as soon as it is introduced. Should anything go wrong, there is no way to communicate errors to the caller.

HP HEWLETT PACKARD

Programmatic Creation of Sessions

STARTSESS (Idev,logonstr,jsid,jsnum,errorstat);
DEVLOGON (Idev,logonstr,jsid,jsnum,errorstat,waitillon);
:STARTSESS Idev;logonstr

```
COMMAND
EXECUTOR  ──────▶  LPDT          "service granted"
    │
    │
    ▼
STARTDEVICE       Parses command
                  Entries in JMAT, IDD, ODD
    │  └──▶  SCHEDULEJMAT    Gives priority
    │
    ▼
UCOP              Checks limits
                  Allocates resources
    │
    ▼
PROCREATE         Creates Command
                  Interpreter
    │
  EXEC*
    ▼
   CI  ──────▶  INITJSMP      Initializes
    │                          jobtables
    │                          $STDIN
    ▼                          $STDLIST
                               etc.
Logon
WELCOME
Initialize UDCs

You're on your own!
```
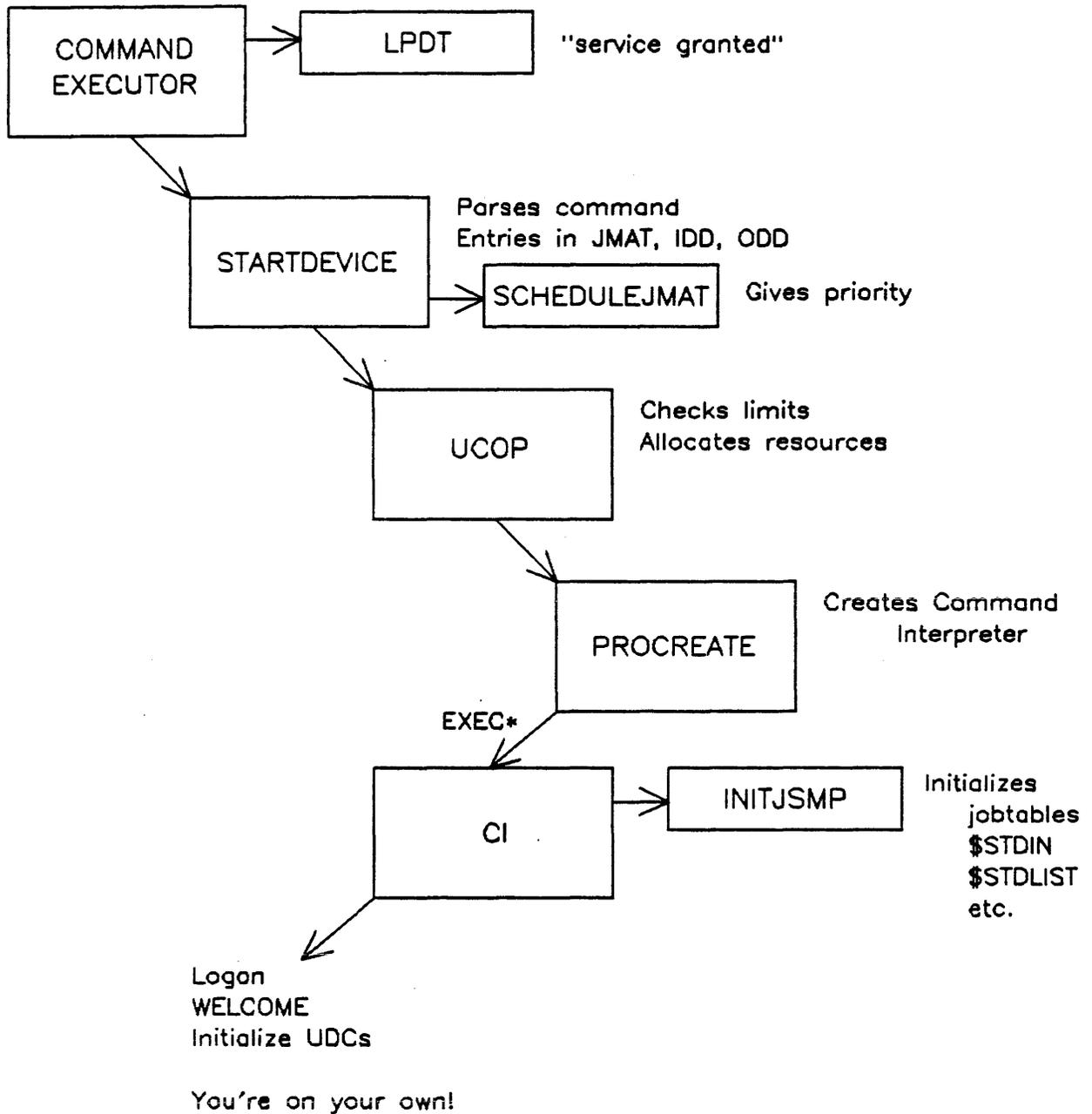
mmtrng03

System Reserved JCWs

Currently, there are two classes of Job Control Words (JCWs): **Regular JCWs**, which are named and given values by the user, and **System Defined JCWs**, which are named by the system and given values by either the system or the user. With MPE-Mighty Mouse, a new class of JCWs is introduced: **System Reserved JCWs**.

System Reserved JCWs are named and assigned values exclusively by the system; the user cannot assign values to these JCWs, nor can he have a regular JCW with the same name as a reserved JCW. The System Reserved JCWs included with MPE-Mighty Mouse are:

HPDAY   Indicates the day of the week. Possible values for HPDAY range from 1 to 7 inclusive, with 1 indicating Sunday and 7 indicating Saturday.

HPDATE   Indicates the day of the month. Possible values range from 1 to 31 inclusive.

HPMONTH   Indicates the month of the year. Possible values range from 1 to 12 inclusive, with 1 indicating January.

HPYEAR   Indicates the year of the century. The contents of HPYEAR will be displayed in two digits (e.g. HPYEAR = 84).

· HPHOUR   Indicates the hour of the day, using the military clock. Possible values range from 0 to 23 inclusive.

HPMINUTE   Indicates the minute of the hour. Possible values range from 0 to 59 inclusive.

FACTS TO NOTE

o   Users cannot assign values to System Reserved JCWs, but they can use the information these JCWs contain. Users can retrieve the values of these JCWs with the :FINDJCW intrinsic.

o   Users can display the contents of a System Reserved JCW with the :SHOWJCW command *only* if they specifically request it. For example:

```
:SHOWJCW
JCW = 0
CIERROR = 1945

:SHOWJCW HPDAY
HPDAY = 4
```

o   The :IF command can be used to test System Reserved JCWs. For example:

```
:SETJCW FRIDAY = 6
:IF HPDAY = FRIDAY THEN
:   FULLBACKUP
:ELSE
:   PARTBACKUP
:ENDIF
```

This example streams a full backup job on Fridays and a partial backup job on the other days of the week. Note that the new :FULLBACKUP and :PARTBACKUP commands are used; these are discussed in Section 2.

INTERNALS

The values in System Reserved JCWs come from two sources:

The values in HPHOUR and HPMINUTE come from the system clock.

The values in HPDAY, HPDATE, HPMONTH and HPYEAR come from the Timer Request List (TRL). The ALMANAC procedure (introduced with Native Language Support) formats the contents of the TRL and supplies the appropriate information to each JCW.

```
┌─────────────────┐
│      NOTE       │
└─────────────────┘
```

In other words, these System Reserved JCWs do not take up any space in the JCW subtable of the JDT.

System Reserved JCWs

HPHOUR
HPMINUTE
HPDAY
HPDATE
HPMONTH
HPYEAR
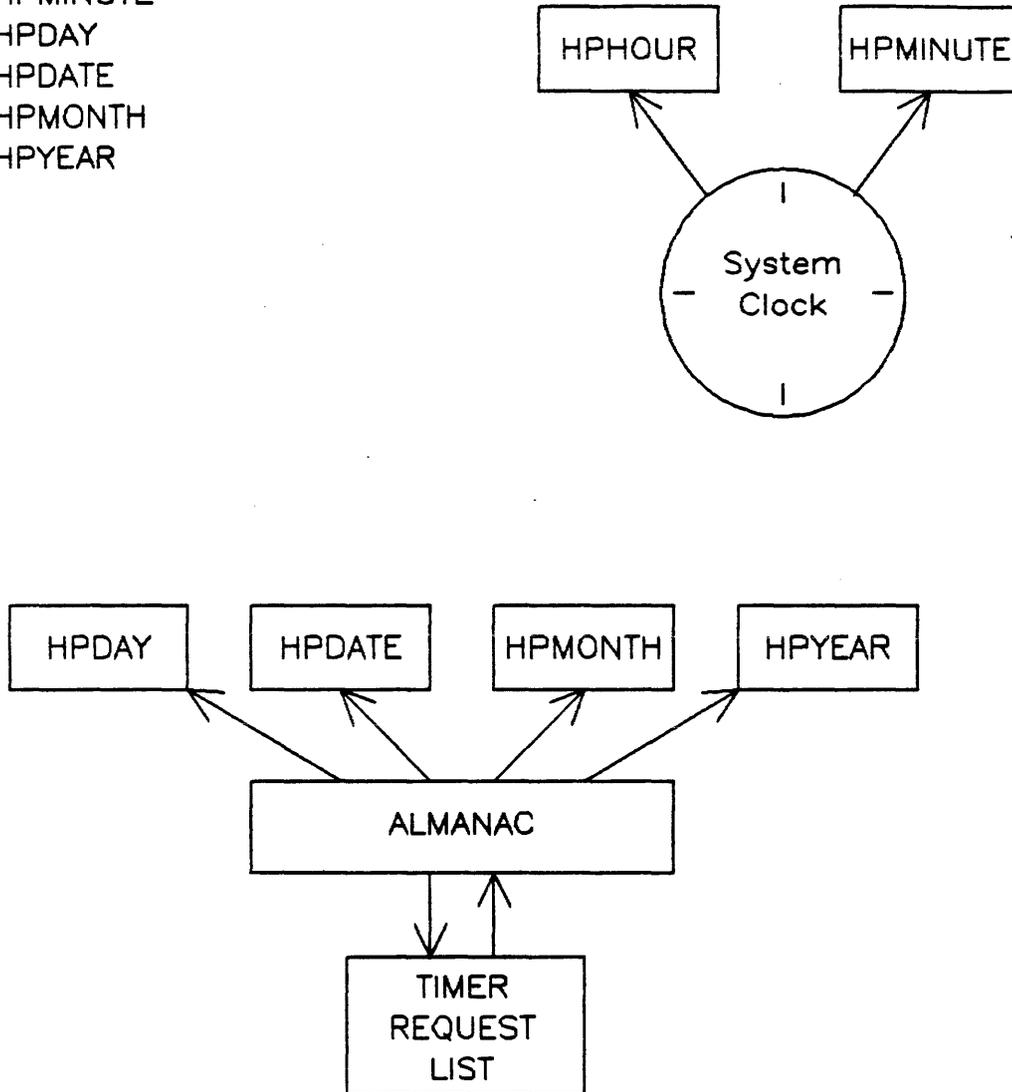
CI Commands Replace LISTEQ5

Currently, the LISTEQ5 utility is used to list file equations and temporary files. With MPE-Mighty Mouse, CI commands have been developed to provide these functions. These commands are :LISTEQ and :LISTFTEMP.

:LISTEQ

The :LISTEQ command will display all active :FILE equations for that job or session. The display format is similar to that of the LISTEQ5 utility for :FILE equations. Syntax for this command is:

```
:LISTEQ [listfile]
```

where listfile is a file to which output is directed. Default is to send output to $STDLIST.

:LISTFTEMP

The :LISTFTEMP command will display temporary files for a job or session. The command's syntax is similar to that of :LISTF:

```
:LISTFTEMP [fileset][,listlevel][;listfile]
```

where fileset, listlevel, and listfile are essentially the same as the corresponding parameters for :LISTF.

The display format for :LISTFTEMP will be similar to that of :LISTF with a few variations:

o   With :LISTF, the display format for listlevel 0 is six filenames per display line. With :LISTFTEMP at listlevel 0, fully qualified filenames are displayed one per line.

   Example:

```
:LISTFTEMP @.@.@

 TEMPORARY FILES FOR LARRY.OSE,CARGNONI

 $OLDPASS.CARGNONI.OSE
 AJ.CARGNONI.OSE
 AJJ.CARGNONI.OSE
 Z10Z414A.JORDANMM.OSE
 TEMPFILE.TEMPGROUP.SYS
```

o   With :LISTFTEMP at listlevels 1 and 2, the display will show "(TEMP)" at the end of the entry for the file.

Example:

```
:LISTFTEMP A@,1

TEMPORARY FILES FOR LARRY.OSE,CARGNONI

ACCOUNT = OSE                    GROUP = CARGNONI
FILENAME    CODE  ------------LOGICAL RECORD--------
                   SIZE  TYP       EOF      LIMIT

AJ      *          80B  FA         16        200    (TEMP)
AJJ     KSAM       80B  FA        414        414    (TEMP)
```

o   With :LISTFTEMP at listlevel -1, the display will show "(TEMPORARY FILE)" at the end of the filename. Remember that the user must have SM or AM capability to use listlevel -1.

FACTS TO NOTE

o   Both :LISTEQ and :LISTFTEMP are available in break and programmatically, via the COMMAND intrinsic.

o   As with the :LISTF command, users of :LISTFTEMP will need SM or AM capability to display any file with a listlevel of -1.

o   :LISTFTEMP will display the system file $OLDPASS only if all the files under a general qualification are listed; that is, if wildcard characters are used and $OLDPASS fits their specifications.

o   Both :LISTEQ and :LISTFTEMP run on the user's stack and require at least 8K words to run.

INTERNALS OF :LISTEQ

When a :LISTEQ command is issued, the following events take place:

1. The Job Directory Table (JDT) for this session is moved to the user's stack. To do this, first the JIR (Job SIR) is locked, then the JDT is pushed to the stack, then the JIR is released.

2. Word %3 of the copied JDT points to the Job File Equation Table (JFEQ). With this pointer, the JFEQ is located. Each JFEQ entry is formatted and sent to the listfile (usually $STDLIST).

3. When every JFEQ entry has been processed, :LISTEQ will come to the Job Line Equation Table (JLEQ). When this table is reached, :LISTEQ completes. The JDT will be popped from the user's stack.

HEWLETT PACKARD

CI Commands Replace LISTEQ2 — 1

:LISTEQ [listfile]

1.

Job SIR | JDT | user's stack | Job SIR
locked | | Q ... S | unlocked

2.

user's stack

JFEQ ptr

JFEQ

JLEQ

File Equation Entry -->> listfile

3.

user's stack

Q,S

— done —

JLEQ

mmtrng05
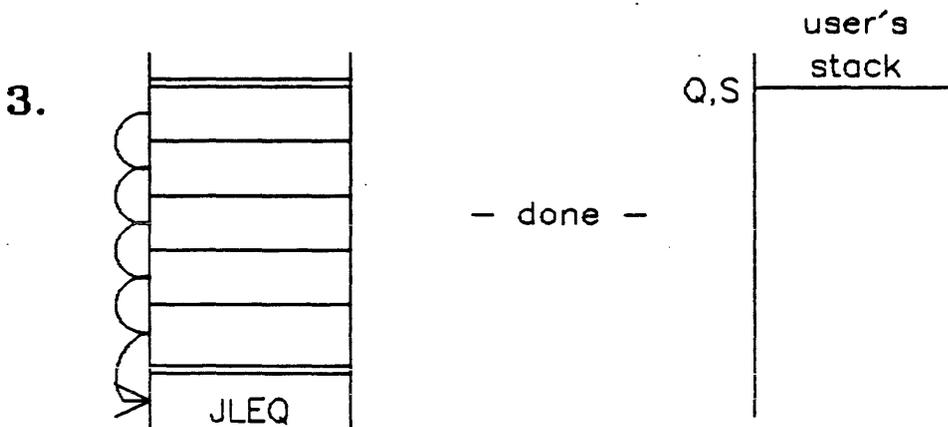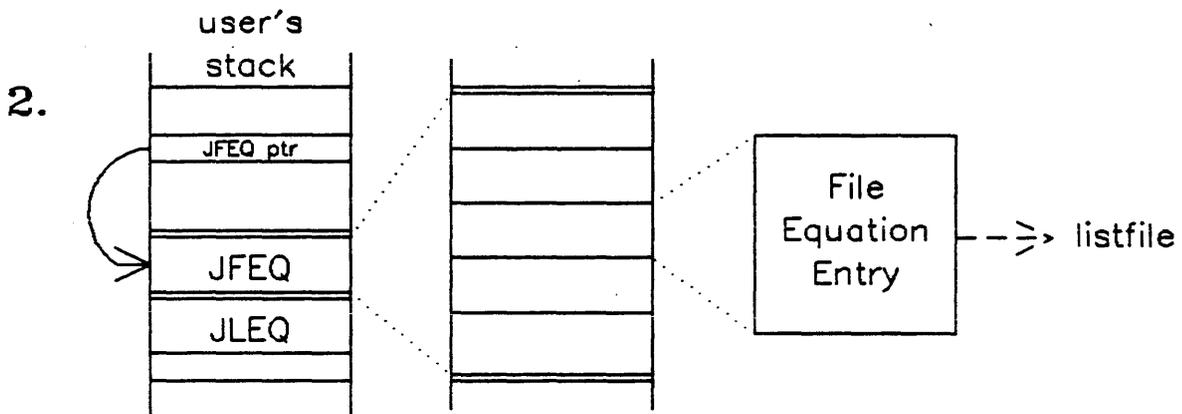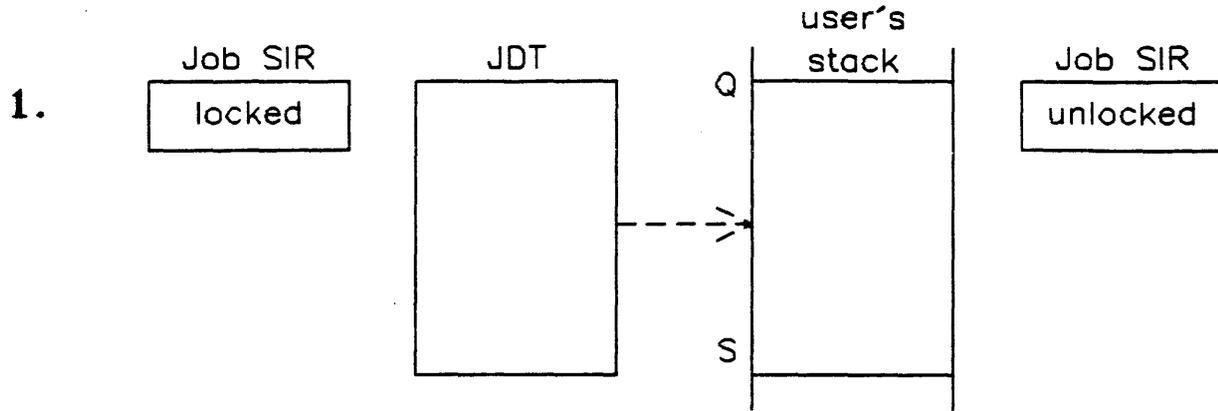
INTERNALS OF :LISTFTEMP

When :LISTFTEMP is used, the following actions are taken:

1. As with :LISTEQ, the JDT is moved to the user's stack. The Job SIR is locked before the transfer and unlocked when the transfer is complete.

2. Word %2 of the copied JDT points to the Job Temporary File Directory (JTFD). With this pointer, the JTFD is located.

   If :LISTFTEMP has been used with parameters, it will utilize the same directory compare routine as :LISTF. Each JTFD entry's filename, groupname and accountname will be compared against what :LISTFTEMP specifies, including wildcard and special characters.

3. For JTFD entries that match :LISTFTEMP's specifications, the directory will be used to locate the temporary file's file label. The label will be formatted and sent to the specified listfile (usually $STDLIST). When each qualifying entry in the JTFD has been processed in this way, :LISTFTEMP completes. The JDT will be popped from the user's stack.

HEWLETT PACKARD

CI Commands Replace LISTEQ2 – 2

:LISTFTEMP [fileset][,listlevel][;listfile]

1.

Job SIR
| locked |

JDT

user's stack
Q

S

Job SIR
| unlocked |

2.

user's stack
| |
| JTFQ ptr |
| JTFD |
| JFEQ |
| |

directory compare routine

3.

Match!
| | ---> listfile

user's stack
Q,S

– done –

mmlrng015

1-20
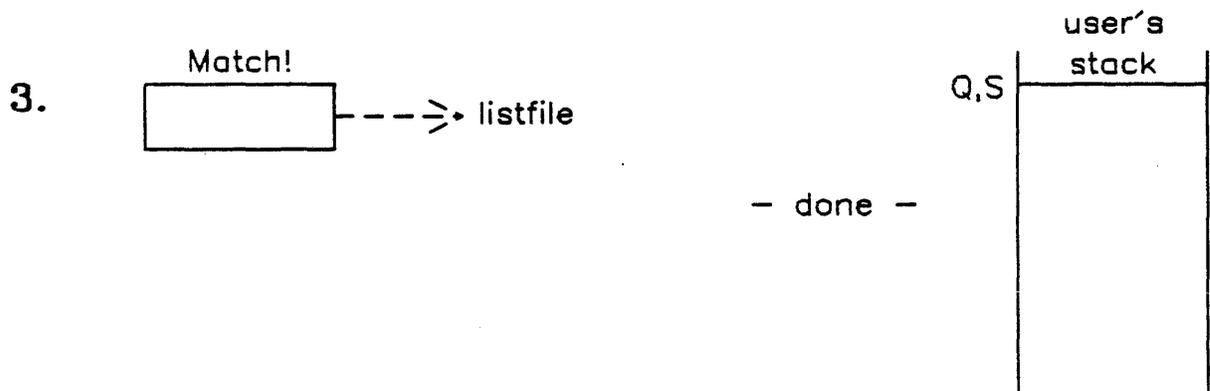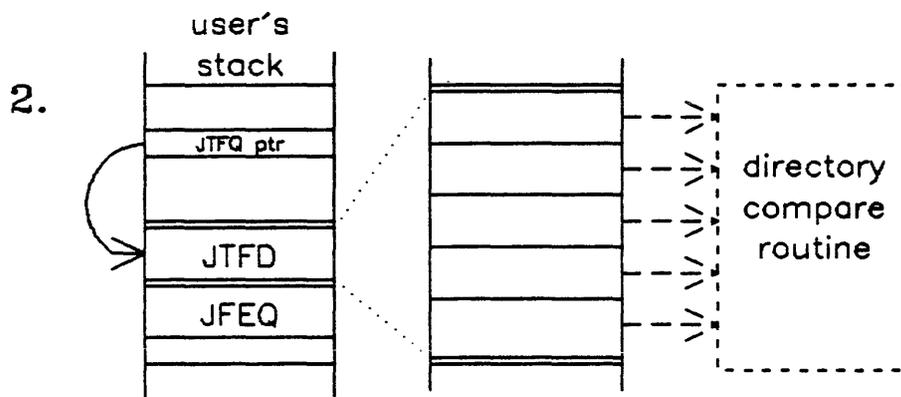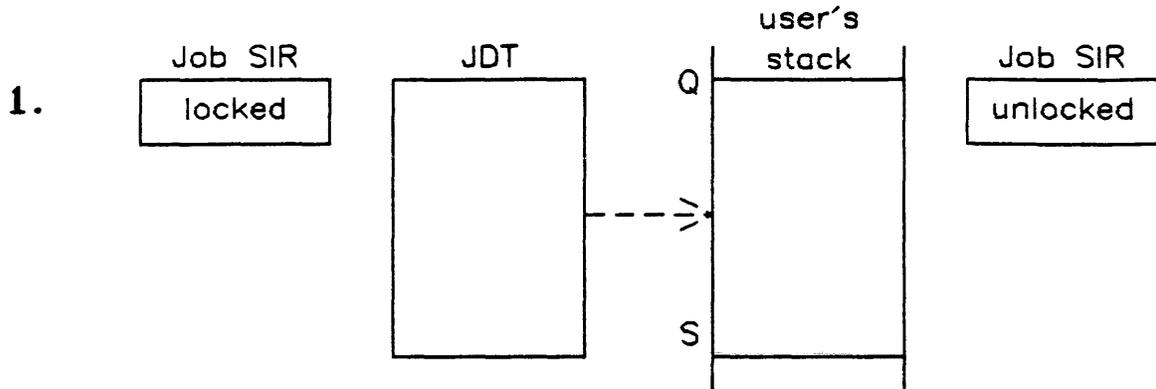
Job Scheduling

MPE-Mighty Mouse introduces a new feature: the **Job Scheduling** facility. By using the new job scheduling feature, a user can specify when a job is to be :STREAMed. When a user schedules a job in this way, he specifies the time a job will enter the WAIT state; if the job satisfies the various system parameters (e.g, Job Limit, Job Fence), it enters the EXEC state at that time.

Scheduled jobs can be aborted via the :ABORTJOB command. The :SHOWJOB command will show the jobs that are scheduled on the system.

SCHEDULING JOBS

The syntax of the :STREAM command for scheduling jobs is:

```
:STREAM filename [,char]
         [;AT=timespec]
         [;DAY={dow|dom|dem}    |    ;DATE=custdate]
```

                                  or

```
         [;IN=[days][,[hours][,minutes]]]
```

where the parameters have the following meanings:

filename   The Editor file containing the commands of the job, in :STREAM format. Note that the first character of the first record is assumed to be the replacement character for the expected colon (:) that identifies MPE commands. The user must have READ or EXECUTE access to the file.

char   The colon replacement character.

dow   Day-Of-Week. Possible values are:

            SUN[DAY]
            MON[DAY]
            TUE[SDAY]
            WED[NESDAY]
            THU[RSDAY]
            FRI[DAY]
            SAT[URDAY]

dom   Day-Of-Month. An integer between 1 and 31 inclusive. This indicates the calendar day of the month. If dom is greater than or equal to the current day's day-of-month, the current month is implied. Otherwise, the next month is assumed. An error message results if the day-of-month specified does not fit with the corresponding month (e.g. "31" is specified for February). If omitted, the current date is used.

dem   Day from End of Month. A negative integer between -31 and -1 inclusive. This indicates the calendar day from the end of the specified month on which the job will run. Thus, a "-1" value represents the last day of the month. If the specified day from the end of the month indicates a day earlier than the current day, the next

month is assumed. That is, if today is the seventh day from the end of the month and a "-8" value is used, the job will be scheduled for the eighth day from the end of the next month.

custdate   Date, specified in the format MM/DD/YY, where MM is the month (1 <= MM <=12), DD is the day (1 <= DD <= 31), and YY is the year. If omitted, the current date is used.

timespec   Time specification. This is the absolute time of day, in the format

HH:MM

where HH is the military hour of the day (0 <= HH <= 23) and MM is the minute of the hour (0 <= MM <= 59). If no DAY or DATE is used, timespec will be a time of the current day; if timespec is earlier than the current time, the next day will be used.

days   Days. A positive integer indicating the offset in days from the current date. If omitted, zero is used (specifying the current day).

hours   Hours. A positive integer (0 <= hours <= 23) indicating the offset in hours from the current time. If omitted, zero is used.

minutes   Minutes. A positive integer (0 <= minutes <=59) indicating the offset in minutes from the current time. If omitted, zero is used.

The time a job is to enter the WAIT state can be specified in two ways: in terms of *absolute* time or *relative* time.

absolute   The user supplies an exact time for the job, using the AT parameter of :STREAM.

relative   The user specifies a time offset from the current time using the IN parameter of :STREAM.

A time specification earlier than the current time will cause an error.

If no errors are detected, a **ScheduleID** will be written to the user's screen. This will have the same format as the JSNum issued by the :STREAM command. If more than one job is included in the command file, each job will receive its own **ScheduleID**, and each job will be scheduled for the same specified time. If the command file also contains :DATA files, these will be added to the Input Device Directory (IDD) immediately. There is a limit of 64 jobs and :DATA files in any one command file.

When a job is scheduled for a future time, it enters the SCHED state. When the specified time is reached, the job enters the WAIT state, and will be EXECuted when system variables allow.

ABORTING SCHEDULED JOBS

The :ABORTJOB command is used to remove scheduled jobs from the system. The syntax for accomplishing this is:

    :ABORTJOB { ScheduleID !   [jobname,] username.acctname }

If :ABORTJOB is used with username.acctname and more than one job is scheduled, :ABORTJOB will delete only one of the qualifying jobs, and its choice cannot be predicted. For this reason, it is recommended that ScheduleID be used with :ABORTJOB, since it uniquely specified a scheduled job.

A user's ability to abort a scheduled job depends upon the JOBSECURITY state, the username.acctname of both the scheduled job and the user, and the user's capabilities.

SHOWING SCHEDULED JOBS

The :SHOWJOB command is used to display all (or a specified subset of) jobs scheduled. The :SHOWJOB command has been enhanced to display scheduled jobs at the same time it displays other jobs. The STATE for scheduled jobs will be "SCHED". Note that a new field, "SCHEDULED-INTRO", has been added to the display to show when the jobs are due to be run.

Example:

```
:SHOWJOB

JOBNUM  STATE IPRI JIN  JLIST    INTRODUCED   JOB NAME

#S210   EXEC       20  20        WED 11:27P   OPERATOR.SYS
#J113   EXEC       10S EPOC      THU  9:04A   MAILROOM,MAILROOM.HPOFFICE
#J199   WAIT  D 1  10S EPOC      THU 10:20A   FIELD.SUPPORT

3  JOBS:
    0 INTRO
    1 WAIT; INCL 1 DEFERRED
    2 EXEC; INCL 1 SESSION
    0 SUSP
JOBFENCE= 6; JLIMIT= 8; SLIMIT= 50

CURRENT:  8/16/84 14:45

JOBNUM  STATE IPRI JIN  JLIST    SCHEDULED-INTRO   JOB NAME

#J202   SCHED  8   10S EPOC      8/16/84 15:30     SCHEDJOB,SCHED.OSE
#J201   SCHED  15  10S EPOC      8/17/84  1:00     SEESPOOL,RSPOOL.SYS

2 SCHEDULED JOB(S)

5 JOBS (DISPLAYED):
    3 SCHEDULED
    2 EXEC; INCL 1 SESSION
JOBFENCE= 6; JLIMIT= 4; SLIMIT= 44
```

The CURRENT line shows the current date and time. The last line (JOBFENCE...) shows the current values of the system parameters. Note that the scheduled jobs are listed in the order in which they will be introduced to the system.

FACTS TO NOTE

When using the Job Scheduling facility, keep the following in mind:

o Jobs typically proceed through the following states: INTRO, WAIT, INIT (or EXEC*), EXEC, and possibly SUSP. When a job is scheduled, it passes from INTRO to the SCHED state, and at the appropriate time enters the WAIT state.

o **Recovery**: Scheduled jobs will survive a WARMSTART. Any other system start will cause scheduled jobs to be deleted. If the system is being brought down for any reason, first perform a :SHOWJOB to show scheduled jobs, and reschedule them when the system is brought back up.

o A scheduled job uses an entry in the JMAT table. For this reason, and because of the limited recoverability of scheduled jobs, it is recommended that most jobs be scheduled no more than a few days in advance.

o If a user specifies a day or date for a job but does not specify a time, the job will *not* enter the WAIT state at midnight on the specified day, as might be assumed. Instead, it will use the current time *at the time :STREAM is used* and enter the WAIT state at *that* time on the given day.

o When used without job scheduling parameters, the :STREAM command will work exactly as it has in the past.

INTERNALS

The JMAT has been changed to accommodate job scheduling. In entry 0 of the JMAT, word %16 will now contain the JMATSCHEDHEAD pointer, which points to the *Scheduled Job Queue*. The Scheduled Job Queue is a linked list of JMAT entries for jobs in the SCHED state (indicated by %70 in word 0 [0:6] of each entry). This queue is linked separately from the other JMAT entries, which are in the WAIT state. (This WAIT queue is the same as prior to MPE Mighty Mouse.)

When job scheduling is issued, the :STREAM command executor performs the following activities:

1. Parses and checks the given parameters.

2. Introduces the job(s) into the system using SCHEDSTREAM, an entry point for the :STREAM command.

3. The SCHEDULESCHED procedure links the new JMAT entry(ies) into the Scheduled Job Queue.

4. Wakes UCOP.

Once the scheduled jobs are linked in the JMAT, they are "managed" by CHECKSCHED, a procedure within UCOP. This procedure runs whenever new jobs are scheduled, or whenever the time in the Timer Request List (TRL) comes up. CHECKSCHED:

1. Checks the Scheduled Job Queue for jobs that are due to run. If there are none, CHECKSCHED goes back to sleep.

2. For jobs that are ready, calls SCHEDULEJOB to link into the regular WAIT queue. Goes through all the jobs that are scheduled for the current time.

3. Aborts the current TRL. If there are more jobs scheduled, CHECKSCHED sets a new TRL, either for the time of the next scheduled job or for 24 days from the current time, whichever is first.

   The largest time offset that can be represented in 32 bits is 24 days. When the next scheduled job is less than 24 days away, the TRL will be reset to reflect the correct time.

Once CHECKSCHED places the scheduled jobs in the WAIT queue, UCOP can deal with them in the same way it deals with any other job.

Note that UCOP (specifically, CHECKSCHED) can be awakened by the :STREAM command executor, in case a job is scheduled for the current time, or by the TRL timer.

HEWLETT PACKARD

Job Scheduling

```
:STREAM filename [,char]
        ⎡ ⎡ [;AT = timespec]                               ⎤ ⎤
        ⎢ ⎢ [;DAY = {dow|dom|dem}  |   ;DATE = custdate]  ⎥ ⎥
        ⎢ ⎣                                               ⎦ ⎥
        ⎢                    or                             ⎥
        ⎢                                                   ⎥
        ⎣ [;IN = [days][,[hours][,minutes]]]               ⎦
```

JMAT

%3   HEAD POINTER → [job] — [job] — [job]   wait queue

%4   TAIL POINTER

%16 JMATSCHEDHEAD

STREAM

Introduces jobs.

Links JMAT entries in SCHED queue.

job

job

sched queue   job

0   5

%70

Checks SCHED queue.
For ready jobs, call
    SCHEDULEJOB to place in WAIT.
Abort current TRL; set new.

CHECKSCHED → TRL
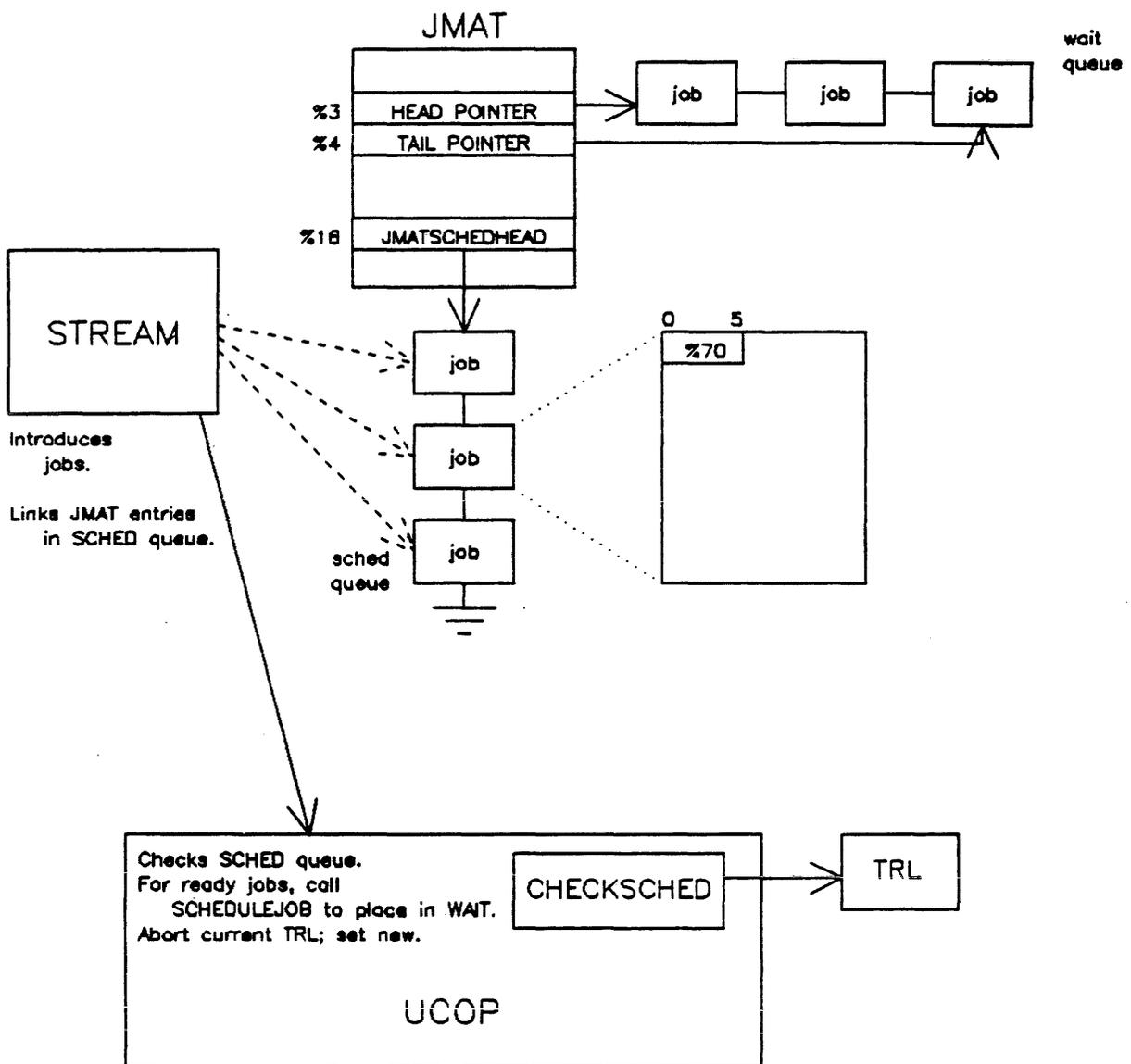
UCOP

UCOP deals with jobs in WAIT as before.

User Defined Commands Enhancements

In the past, account managers have occasionally set up LOGON UDCs for users in their accounts. These UDCs have typically run an application program and then logged the user off the system. This can be a problem, since once this kind of UDC is set up, it is very difficult for an account manager to disable it. The only workaround is to disable UDCs for the entire system by renaming COMMAND.PUB.SYS and then allow the user to log on and perform a :SETCATALOG.

With MPE-Mighty Mouse, an enhancement to :SETCATALOG permits account managers to assign or disable UDC catalogs for any user in their account; system managers can assign or disable catalogs for any user on the system. This can be done by using the new "USER =" parameter. The syntax of :SETCATALOG is now:

```
:SETCATALOG ...parms... [; USER = username [.acctname]]
```

A parallel change in the :SHOWCATALOG command will allow the account manager to list the UDC files for any user in the account, or the system manager to list the UDC files for any user in the system. The syntax of :SHOWCATALOG is now:

```
:SHOWCATALOG [listfile] [; USER = username [.acctname]]
```

FACTS TO NOTE

The following restrictions apply to the use of the :SETCATALOG and :SHOWCATALOG commands:

o Users of :SETCATALOG or :SHOWCATALOG require AM capability to set or display the UDC catalogs of users within their account, or SM capability to do this for any user within the system.

o When :SHOWCATALOG is used with the "USER =" parameter, it will display only the *names* of the specified user's UDC files, not the actual UDCs.

o The :SETCATALOG ... USER = ... command cannot be used with wildcard characters (e.g. "@"). :SHOWCATALOG may use a single ampersand (@) to represent the user or account name, but not combined with other characters as a wildcard character. When "@" is used in place of a user name, it indicates all account-level UDCs; "@.@" indicates system-level UDCs.

o If the "USER =" parameter is omitted, :SETCATALOG and :SHOWCATALOG function exactly as they have in the past.

INTERNALS

When :SETCATALOG and :SHOWCATALOG are used without the "USER =" parameter, they go through COMMAND.PUB.SYS for UDC file information. COMMAND.PUB.SYS lists the UDC file names associated with each user, as well as the names of account-level and system-level UDC files. Entries for UDC files include pointers to extra data segments (XDSs) which contain the actual UDC file, including UDCs themselves.

:SETCATALOG. This command will work essentially as it does today. With the "USER =" parameter, :SETCATALOG will scan through COMMAND.PUB.SYS. When it finds a user.account matching its specifications, it will delete UDC entries associated with that user.account; if the command specified new UDC file names, it will add them to COMMAND.PUB.SYS.

Note that the deletion or addition of UDC files will not take affect until the next time the affected user(s) log on.

:SHOWCATALOG. *Without* the "USER =" parameter, :SHOWCATALOG works as follows:

1. :SHOWCATALOG goes through the user's UDC directory (an extra data segment) to find pointers to the user's UDC files.

2. :SHOWCATALOG uses these pointers to find the files, and lists the name of each UDC file as well as the names of the UDCs contained within. The names of the files and UDCs can be listed to the user's screen or to a listfile.

When "USER =" is used with :SHOWCATALOG, it works slightly differently:

1. :SHOWCATALOG goes through COMMAND.PUB.SYS looking for the *specified* user.account.

2. When :SHOWCATALOG finds the correct user.account, it lists the UDC *file names* to the user's screen or listfile. :SHOWCATALOG never goes to the UDC files themselves, and will only list the information that it can get from COMMAND.PUB.SYS.

Note that, without the "USER =" parameter, :SETCATALOG and and :SHOWCATALOG function exactly as they have in the past.

HEWLETT PACKARD

UDC Control Enhancements

:SHOWCATALOG

UDC Directory

UDC file
UDC
UDC
UDC file
UDC

UDC file
UDC definition
UDC definition

UDC file
UDC definition
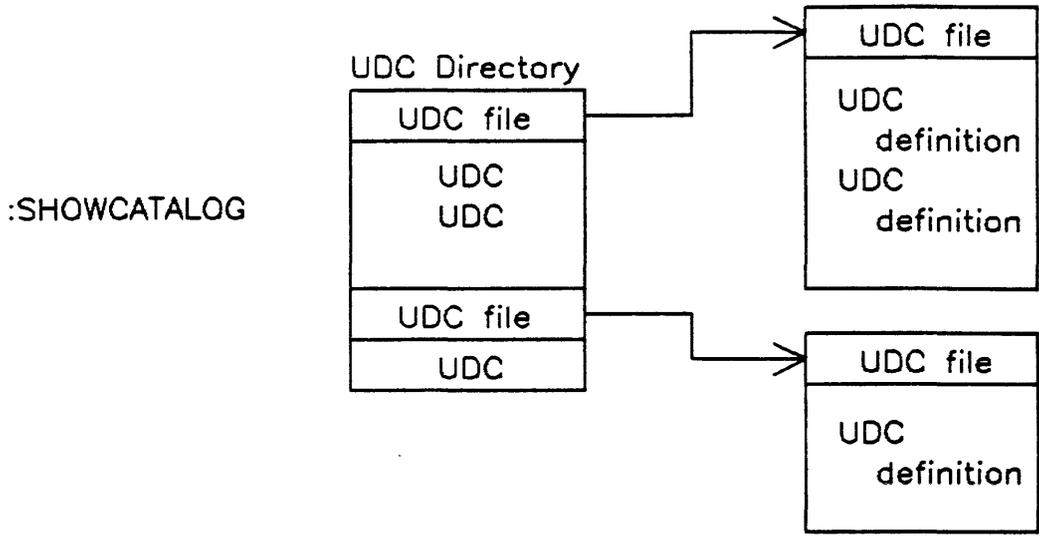
:SETCATALOG
:SETCATALOG ... USER = user.account
:SHOWCATALOG ... USER = user.account

COMMAND.PUB.SYS

user.account
UDC filename
user.account
UDC filename
UDC filename

mmtmg08

1-30

System Startup State Configurator

Prior to MPE Mighty Mouse, when MPE was started and the system came up for the first time, the system automatically assigned default values for various system parameters (e.g. JOBFENCE, Job LIMIT, OUTFENCE, etc.). Customers often had to change these values to fit their individual needs, by resetting system parameters and frequently by streaming special jobs.

To accomplish this, customers often had to assign LOGON UDCs to OPERATOR.SYS; unfortunately, this meant these UDCs were executed every time OPERATOR.SYS logged on. Having the operators go through a series of steps when the system started was clumsy.

MPE Mighty Mouse offers a new alternative: the *System Startup State Configurator*. This feature permits the customer to determine in advance what values his system will come up with, depending upon what type of startup is done.

SYSSTART.PUB.SYS

The operation of the System Startup State Configurator is similar to the operation of UDCs. The file corresponding to the UDC file is **SYSSTART.PUB.SYS**; the names of system starts correspond to the UDCs.

The following restrictions apply to the System Startup State Configurator:

o   The following *System Startup State Commands* will be supported:

STARTUP (for all system starts)
WARMSTART
COOLSTART
UPDATE
RELOAD*
COLDSTART

None of these commands will have any parameters.

o   No options are available with the above six commands. The commands will be echoed to the console (with passwords stripped), BREAK will be disabled, and the UDC options LOGON, LIST, etc. make no sense in this context.

*In SYSSTART.PUB.SYS, RELOAD is available only with options SPREAD or RESTORE. NULL and ACCOUNTS will not be available because the required files are not available to RELOAD during system startup.

o   The MPE commands available for execution in SYSSTART.PUB.SYS are:

ACCEPT	HEADOFF	STARTCACHE
ALTLOG	HEADON	STARTSESS
ALLOCATE	~~IMFCONTROL~~	STARTSPOOL
ALLOW	JOBFENCE	STOPCACHE
CACHECONTROL	JOBSECURITY	STOPSPOOL
COMMENT	JOBPRI	STREAM
CONSOLE	LDISMOUNT	STREAMS
DEALLOCATE	LIMIT	SUSPENDSPOOL
DISALLOW	LMOUNT	TAKE
DISCRPS	LOG	TELL
DOWN	MPLINE	TELLOP
DOWNLOAD	MRJECONTROL	TUNE
~~DSCONTROL~~	OUTFENCE	UP
FOREIGN	REFUSE	VMOUNT
GIVE		

Commands were chosen on the basis of being programmatically executable and requiring no user interaction. No other commands, UDCs, or other System Startup State Commands are allowed in the body of a System Startup State Command.

The file SYSSTART.PUB.SYS must be created by MANAGER.SYS. When the system is started, the appropriate System Startup State Command(s) will be executed if a valid SYSSTART.PUB.SYS is on the system.

A "valid" SYSSTART.PUB.SYS is an ASCII file which has one of the six Startup State Commands as its first record. The commands in the STARTUP block (if it exists) will be executed *first*, no matter what kind of startup is specified; the commands in the appropriate block are executed next (if a block corresponding to the specified type of startup exists). These executions occur immediately before OPERATOR.SYS is logged onto the console.

If no SYSSTART.PUB.SYS can be found, system startup works as it has in the past, and OPERATOR.SYS is allowed to log on. If SYSSTART.PUB.SYS is found but contains an error, an error message is printed to the console and execution continues.

The System Startup State Commands are not available after the system has been brought up.

SAMPLE SYSSTART.PUB.SYS

The System Startup State command file must be named SYSSTART, and must reside in PUB.SYS. This file must be a regular EDITOR file (i.e. 80-byte records, fixed, ASCII). The file can be numbered or unnumbered; data beyond 72 characters will be ignored.

The format for this file is the same as that for a UDC file, with the restrictions mentioned earlier. Blocks are terminated by a "*" in the first byte of a record or by the end-of-file. Records are continued onto another line by adding a "&" as the last non-blank character in the record. The maximum continued record length is 239 bytes.

The following is an example of a valid System Startup State command file:

```
1       STARTUP
2       STREAMS 10
3       ALLOW @.@;COMMANDS=REPLY
4       ALLOCATE PROGRAM APPLIC.PUB.ACCT
5       LIMIT 4,16
6       JOBFENCE 4
7       OUTFENCE 5
8       OUTFENCE 12;LDEV=14
9       ***
10      WARMSTART
11      STARTSESS 20;MGR.ACCT;HIPRI
12      STARTSESS 21;USER.ACCT
13      STARTSESS 22;USER.ACCT
14      STARTSESS 23;USER.ACCT
15      ***
16      COOLSTART
17      STARTSESS 20;MGR.ACCT;HIPRI
18      STARTSESS 21;USER.ACCT
19      STARTSESS 22;USER.ACCT
20      STARTSESS 23;USER.ACCT
21      ***
```

This example has defined a STARTUP command to be invoked every time the system is started. In addition, a WARMSTART or COOLSTART will cause sessions to be created for MGR.ACCT on logical device 20 (probably the console) and for USER.ACCT on three other terminals.

NOTE

In this example, there are not definitions for UPDATE, RELOAD or COLDLOAD. If this system comes up in any of these starts, only those commands listed in STARTUP will be executed. There is no requirement that any or all of the possible commands be present in SYSSTART.PUB.SYS.

FACTS TO NOTE

o If one of the System Startup Commands creates a session on the console (i.e. STARTSESS 20;USER.ACCT), the automatic logon of OPERATOR.SYS will fail.

o The System Startup Commands are not available to any user; they are invoked by PROGEN only during system startup.

o Once the system is up, the System Startup Commands will not be recognized.

o Errors: If any of the MPE commands in a System Startup Command block results in an error, then an error message will be generated and execution will continue at the next command in the block.

o The System Startup State Configurator facility can be disabled simply by renaming SYSSTART.PUB.SYS.

o Lockwords are not permitted on SYSSTART.PUB.SYS.

INTERNALS

The System Startup State Configurator will be available on all systems that support MPE-Mighty Mouse. The internal workings of this facility will differ between Mighty Mouse hardware and other systems, however.

On the Series 37:

1. The operator turns the Keyswitch to position 2 (CONTROL/LOCAL) and enters a Control B command. In this example, the operator enters the command, "WARM". (The operator could also simply turn the Keyswitch to position 1 [NORMAL] and a WARMSTART would be started automatically.)

2. The Series 37 microcode interprets this instruction and loads INITIAL.

3. INITIAL brings the system up as it has in the past. It calls PROGEN, specifying a WARMSTART by passing information through a communications extra data segment.

4. PROGEN finds SYSSTART.PUB.SYS and scans it, searching for System Startup State commands. It establishes pointers for the STARTUP and WARMSTART blocks.

5. The commands in the STARTUP block are executed first, followed by those in the WARMSTART block. The system comes up and OPERATOR.SYS is logged on (unless another session has been created on the console).

On non-Mighty Mouse hardware, startup will look similar to the way it is now:

1. At the console, the operator will enter, "START". The microcode for that system will interpret this command and load INITIAL.

2. INITIAL will prompt the operator for the type of startup, and the operator will respond as before.

3. From this point, system startup is the same as on Mighty Mouse hardware.

HEWLETT PACKARD

System Startup State Configurator

Mighty Mouse:
Control—B Command

>WARM

```
        ┌──────────┐
        │   XDS    │
        └──────────┘
┌──────────┐   ┌──────────┐   ┌──────────┐      SYSSTART.PUB.SYS
│   MM     │→  │          │→  │          │→   ┌─────────────────┐
│microcode │   │ INITIAL  │   │ PROGEN   │    │ * STARTUP       │
└──────────┘   └──────────┘   └──────────┘    │   .             │
                                              │   .             │
                                              │   ***           │
                                              │ * WARMSTART     │
                                              │   .             │
                                              │   .             │
                                              │   ***           │
                                              │   COOLSTART     │
                                              │   .             │
                                              │   ***           │
                                              │   etc.          │
                                              └─────────────────┘
```
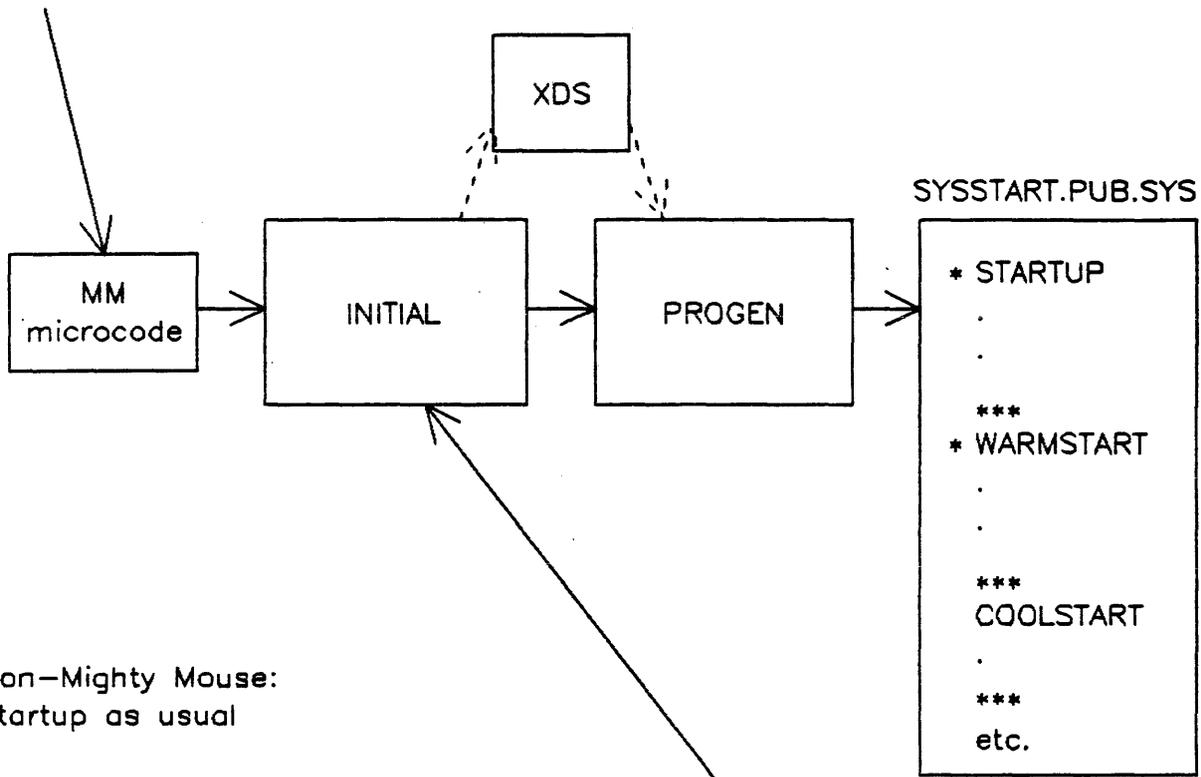
Non—Mighty Mouse:
Startup as usual

>START
WHICH OPTION <WARMSTART/COOLSTART>?   WARM

mmtmg09

Time of Century Clock Support

This feature is supported only on SERIES 37 HARDWARE!

The HP3000 Mighty Mouse processor contains a Time of Century clock which is set when the system first runs MPE and continues running even after the system is shut down or powered off. This clock allows MPE to determine the date and time as it is starting up, thus simplifying the startup procedure by eliminating the need for user intervention.

In order to correct for clock inaccuracies, shipment to other time zones or failure of the internal battery, the operator will be given an opportunity to reset the time and date. Just before the system comes up, the date and time will be printed to the console for the user to verify:

```
THU, JUL  5, 1984,  8:45 AM ? (Y/N)
```

When this message comes up, the operator has three options:

1. He can accept the given date or time and respond with "YES" (or "Y", or "y", or "yes") or a carriage return. The system will come up with the given date and time.

2. He can ignore the prompt. (He doesn't even have to be in the room.) After sixty seconds the read will time out and the system will come up with the given date and time.

3. He can reject the given date and time and enter "n" or "no". He will then enter the standard dialog for specifying the date and time he desires. The time of century clock will be updated to reflect this new time. The system will come up with the new date and time.

```
┌──────────────┐
│     NOTE     │
└──────────────┘
```

Operator intervention is required *only* if the given time and date are unacceptable; otherwise, no interaction is necessary.

FACTS TO NOTE

o   The time of century clock is a *hardware* feature. Microcode has been written to allow MPE to communicate with this clock through PROGEN. This clock will not be accessible to user programs or other system code.

o   The time of century clock is considerably less accurate than other system clocks: it is accurate to the nearest second, whereas the other clocks are accurate to within milliseconds. However, the time of century clock has its own battery backup, so it will survive powerfails and any kind of extended shutdown.

o   The time of century clock will survive shutdowns, but the other system clock will not be automatically corrected; a startup must be done to get any other clocks into synchronization.

INTERNALS

The time of century clock is a feature of the Series 37 hardware. PROGEN communicates with this clock via two new privileged machine instructions: RTOC and WTOC.

When the system is being brought up, the following events take place:

1. Near the end of startup activities, PROGEN reads the time of century clock: the ReadMMClock procedure issues an RTOC instruction to retrieve the 32-bit value from the clock, then does error checking and translates the value to the form that is used by the Startclock procedure of HARDRES.

2. Startclock is called with the time from the time of century clock; it starts any other clocks used by the system.

3. InitMMDateTime prints the time and date from the time of century clock and the operator is given the opportunity to accept or reject the time. A timed read is done from the console for the operator's response.

4. If the operator responds to the prompt with "YES" or a carriage return, *or* if the read from the console times out, the system comes up with the given time. Startup is complete.

5. If the operator rejects the time given by the time of century clock, the InitDateTime procedure is called. This is the old procedure that handles the time and date dialog for all non-Mighty Mouse HP3000s. The standard dialog will be used for the operator to enter a date and time to his liking.

6. InitDateTime takes the operator's input and sends it to two places: Startclock in HARDRES to change the process clock and system clock, and the WriteMMClock procedure to update the time of century clock.

7. WriteMMClock issues a WTOC machine instruction to change the time of century clock. Startclock sets the other clocks, and the system comes up. Startup is complete.

On all non-Mighty Mouse hardware, system startup will work as before.

Time of Century Clock Support

3. Okay?
   yes = done
   no =

TOC Clock

PROGEN

RTOC 1. → ReadMMClock → InitMMDateTime

WriteMMClock ← InitDateTime → Dialogue

WTOC

Battery Backup — *Run off Both.
all the Time*

Startclock

HARDRES

System Clock
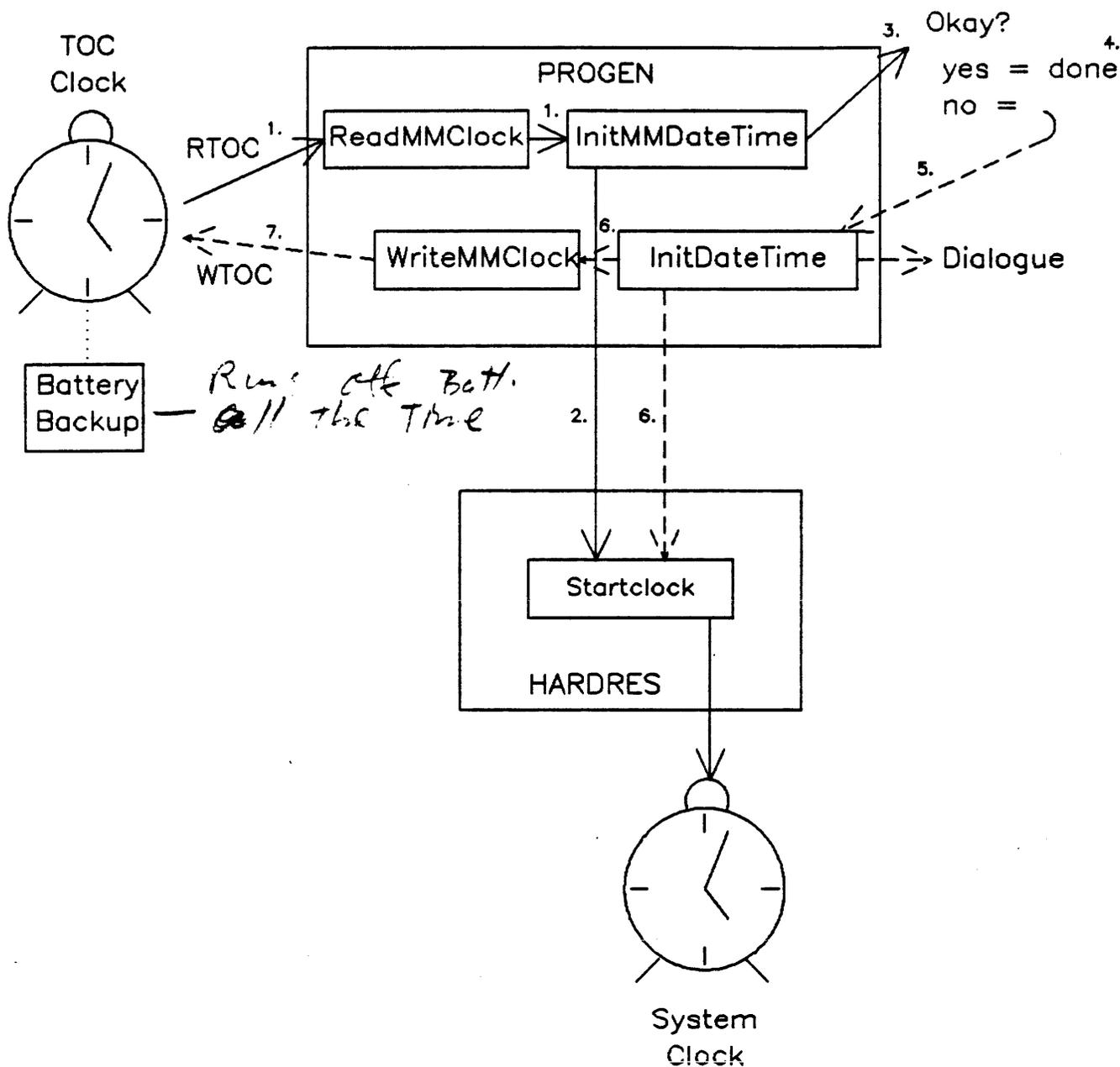
mmlmg10

Several enhancements have been made to Sysdump and Initial which will make these programs easier to use. Some hardware-dependent features will be available only on the Series 37, but the rest of these enhancements will be available for all systems using MPE Mighty Mouse. The enhancements include:

o   Automatic initialization, using the keyswitch and Control-B commands*
o   Table lookup for device defaults
o   I/O Map facility
o   Backup automation commands
o   System tables dialog improvement

This section will describe each of these new features. The internal workings of features will be described where appropriate.

---

*Available only on Series 37 hardware.

Automatic Initialization

This feature is supported only on SERIES 37 HARDWARE!

The Automatic Initialization features of Mighty Mouse enable an operator to bring up the system with little intervention between the time the power is turned on and the MPE prompt appears. These features are implemented by the new Four-position Keyswitch, Control-B Commands, and the Time of Century Clock (described earlier).

FOUR-POSITION KEYSWITCH

The positions of the keyswitch have the following meanings:

0   **Off.** Power off.

1   **Normal.** Local mode console, control-B disabled. When the keyswitch is turned to this position, a Warmstart is done automatically, without dialogue.

2   **Control/Local.** Local mode console, control-B enabled. When the keyswitch is turned to this position, control-B mode is entered and control-B commands may be used.

3   **Control/Remote.** Remote mode console, control-B enabled. When the keyswitch is turned to this position, an SE can switch to a remote console connected through the modem port on the back of the Series 37. Note that, with the keyswitch in any other position, this port may be used as a regular terminal port; when the keyswitch is turned to position 3, a regular terminal will be dropped so that a modem can be used. Commands and responses will be echoed to the customer's console as well as to the remote console.

When the key is turned from the "Off" position, microcode performs a series of selftests. These tests will take approximately 30 seconds. The operator has that long to put the keyswitch into the desired position. Another way to look at this: The LED on the front of the Series 37 will display the selftest currently underway. Any time before the LED displays "C" for "Card Test", the key position can be changed.

CONTROL-B COMMANDS

When the keyswitch is turned to position 2 (Control/Local) or position 3 (Control/Remote), Control-B mode is automatically entered. From here, any of the following startup options can be requested:

ST[art]   [chan, dev [,{P[erm] C[hange]}]]
              Asks, "WHICH OPTION <WARMSTART/COOLSTART> ?" Enters Initial dialogue.

DI[sc]   Same as START with no parameters.

DI[sk]   Same as START with no parameters.

L[oad]   [chan, dev [,{P[erm] C[hange]}]]
              Asks, "WHICH OPTION <COLDSTART/RELOAD/UPDATE> ?" Enters Initial dialogue.

TA[pe]   Same as LOAD with no parameters.

WAR[m]	Performs a WARMSTART with no dialogue.
COO[l]	Performs a COOLSTART with no dialogue.
UPD[ate]	Performs an UPDATE with no dialogue.
COL[dstart]	Performs a COLDSTART with no dialogue.
REL[oad]	Performs a RELOAD option SPREAD with no dialogue.
NEW[sys]	Performs a RELOAD option SPREAD with no dialogue. This command assigns a volume name and virtual memory to the system disc, and so *it is valid only for initial software installations on new systems*.
H[elp]	Displays the available commands, parameters, and effects. The HELP command does not display information about START or LOAD.
DU[mp]	[chan, dev] Performs a dump to the specified device.
AR	Retry auto restart.
RUN	Run the system after CNTL-B HALT.
SP[eed]	Allows the operator to change the console speed.
TE[st]	Go to selftest mode.

The *default* channel (chan) and device (dev) can be obtained by entering Control-B mode and typing TE[st], then IO[map]. This will show the channel, device and unit for the connected start, load and dump devices. These addresses are stored in the TOC register in CPU memory, and will be used by Initial for booting the system.

It is recommended that the operator check the default addresses of the start, load and dump devices before he boots the system. After checking, there are four possible courses of action to take. The following example uses the LOAD command, although the START command is equally valid.

LOAD	Loads the system from the current system default load device.
LOAD chan,dev	Loads from the specified load device; does not change the system default. The next load is not affected by this command.
LOAD chan,dev,C	Changes the system default load device by writing the given channel, device and unit to the TOC register. This command does not invoke a load; the next LOAD command, used without parameters, will load from this device.
LOAD chan,dev,P	Changes the system default load device by writing the given channel, device and unit to the TOC register, *and* loads from this device.

2-3

FACTS TO NOTE

o For the DUMP device, the operator can specify a device other than the default listed in the TOC register, but he cannot change the contents of the TOC register. The DUMP device specified in the TOC register will always be the same as the START device specified there; changing the channel and device for the START device will change the channel and device for the DUMP device as well.

INTERNALS

When the keyswitch is turned to position 2 (Control/Local), microcode immediately invokes Control-B mode. The following events take place:

1. The operator enters a startup command. Microcode picks up this command and stores its representation in location Qi-11 on the ICS. Microcode then invokes Initial.

2. Initial uses Qi-11 to set the variable STARTTYPE. Based on the value of STARTTYPE, Initial will go through different paths, with or without dialogue, from tape or disc, etc.

The representations for the various startup types are:

START DISC	%0	UPDATE	%11
		COLDSTART	%12
WARM	%1	RELOAD	%13
COOL	%2	NEW	%14
TAPE LOAD	%10	DUMP	%20

HEWLETT PACKARD

Control-B Commands

-> COOL $\dfrac{\text{micro}}{\text{code}}$

Qi-11   %2

ICS

INITIAL

STARTTYPE=COOL

Coolstart,
No Dialogue

Qi

- -

Startup from Disc

| START | DISC DISK | WARM | COOL |

Which option <warmstart/coolstart>?

warmstart, coolstart
w. dialogue

warmstart,
no dialogue

coolstart,
no dialogue

- -

Startup from Tape

| LOAD | TAPE | UPDATE | COLDSTART | NEW |

Which option <coldstart/reload/update>?

coldstart, reload, update
w. dialogue

warmstart,
no dialogue

coolstart,
no dialogue

reload,
no dialogue
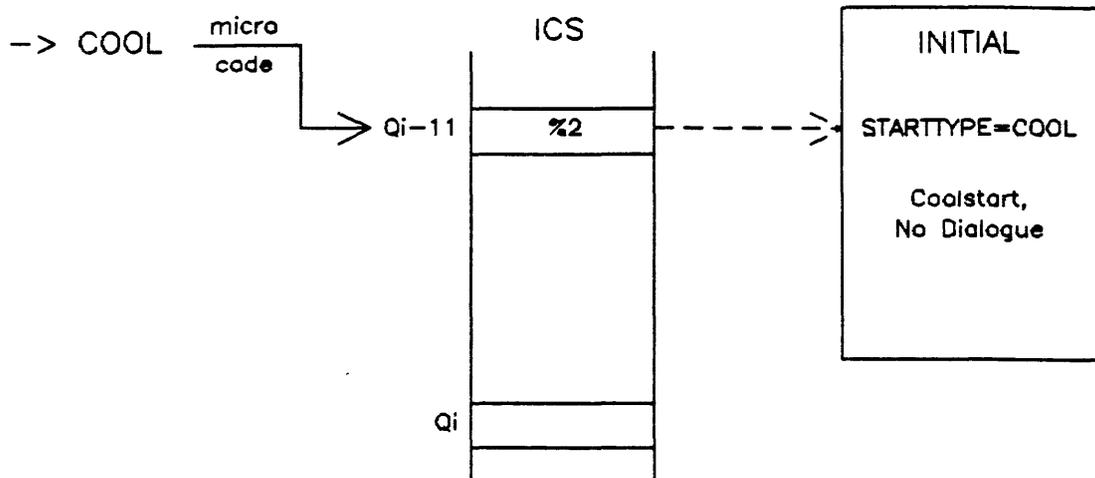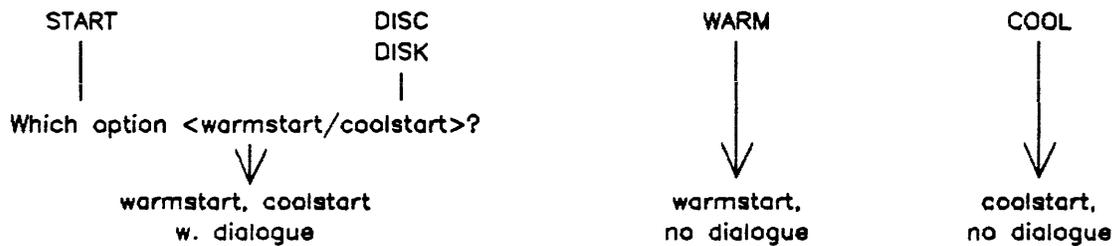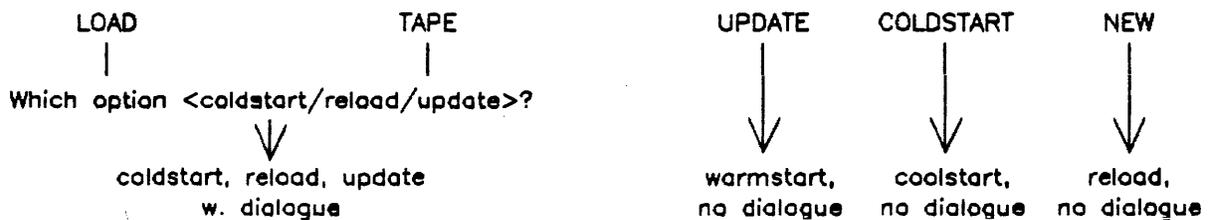
mmlmg11

Table Lookup for Device Defaults

The Sysdump and Initial dialogues are frequently used to add devices to a system's I/O configuration. Currently, an operator has to look up the devices' characteristics in a manual in order to specify them to Sysdump or Initial. With the release of MPE Mighty Mouse, however, a file is available that contains the default information for HP-supported devices. This file, DEFDATA.PUB.SYS, is available to Sysdump and Initial and eliminates the necessity for looking up default information every time a device is added to the system.

CHANGES TO SYSDUMP AND INITIAL DIALOGUES

The Sysdump and Initial dialogues have been modified to take advantage of the device default file. Two new questions, "LIST DEVICE DEFAULTS?" and "DEVICE NAME?" have been added. The first question allows the operator to see what defaults are available; the second new question allows the operator to take advantage of as many of the defaults as he wishes. If the operator supplies the name of the device, the next several questions will default to the system-supplied values. The operator may accept these values, or change as many as he pleases.

The affected part of the dialogue is now:

	Step	Dialogue
	5	LIST I/O DEVICES?
	6	LIST CS DEVICES?
NEW	6.1	LIST DEVICE DEFAULTS?
	7	HIGHEST DRT = \<XXX\> (MIN=X,MAX=XXX)? Note that the maximum DRT value offered is CPU-dependent.
	8	LOGICAL DEVICE #? Operator gives some response.
NEW	8.1	DEVICE NAME? Operator enters a device name from the DEFDATA file to take advantage of default information. This name must have the "HP" before the product number (e.g. HP2680A).
X = Default	9	DRT# = X? Only Initial will supply this default. It will be found by polling the physically connected devices through the IOMAP mechanism. The default DRT number will be displayed only for devices *not already* in the configuration.
X = Default	10	UNIT# = X? This default value will be offered only if the DRT default was supplied.
X = Default	11	SOFTWARE CHANNEL = X?
X = Default	12	TYPE = X?
X = Default	13	SUBTYPE = X?

X = Default   34   RECORD WIDTH = X?

                 36   ACCEPT JOBS OR SESSIONS = YN?

                 37   ACCEPT DATA = YN?

                 38   INTERACTIVE = YN?

                 39   DUPLICATIVE = YN?

                 40   INITIALLY SPOOLED = YN?

                 42   INPUT OR OUTPUT = IO?
Will only be asked if the device is initially spooled.

                 42.1 AUTO REPLY = YN?
This is a new prompt for device-specific information.

                 43   DRIVER NAME = name?

                 48   DEVICE CLASSES = class?

THE DEFDATA FILE

The default information for Sysdump and Initial comes from a new system program file called DEFDATA.PUB.SYS. It is not a file as such, but a table that is brought into the system during a coldstart. The cold load information table will provide the address of DEFDATA.PUB.SYS.

When a user of Sysdump or Initial lists device defaults, this is the information they will see from DEFDATA:

DEVICE NAME	ID CODE	C H A N	T Y P E	SUB TYPE	TERMINAL TYPE	SPEED	REC WIDTH	OUTPUT DEV	MODE	DRIVER NAME	DEVICE CLASSES
HP7902	!0081	0	2	0			128	0		HIOFLOP0	FLOP
HP9895	!0081	0	2	0			128	0		HIOFLOP0	FLOP
HP7906RP	!FF02	0	0	10			128	0		HIOMDSC1	DISC
HP7906FP	!FF02	0	0	11			128	0		HIOMDSC1	DISC
HP7906BP	!FF02	0	0	12			128	0		HIOMDSC1	DISC
HP7911	!0204	0	3	1			128	0		HIOMDSC2	DISC
HP7912	!0208	0	3	2			128	0		HIOMDSC2	DISC
HP7914	!020A	0	3	4			128	0		HIOMDSC2	DISC
HP7920	!FF03	0	0	8			128	0		HIOMDSC1	DISC
HP7925	!FF04	0	0	9			128	0		HIOMDSC1	DISC
HP7933	!0212	0	3	8			128	0		HIOMDSC2	DISC
HP7935	!0212	0	3	8			128	0		HIOMDSC2	DISC
HP7945	!0220	0	3	5			128	0		HIOMDSC2	DISC
HP7970E	!0183	0	24	0			128	0		HIOTAPE0	TAPE
HP7974	!0174	0	24	3			128	0		HIOTAPE2	TAPE
HP7976A	!0176	0	24	1			128	0		HIOTAPE1	TAPE
HP7978	!0178	0	24	2			128	0		HIOTAPE2	TAPE
HP9140	!0240	0	3	0			128	0		HIOCTAP0	CTAPE
HP9144	!0260	0	3	3			128	0		HIOCTAP1	CTAPE
HP2608A	!2001	0	32	4			66	0	S	HIOLPRT0	LP
HP2613	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2617	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2619	!200A	0	32	2			66	0	S	HIOLPRT2	LP
HP2563A	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2565	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2566	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2608S	!2101	0	32	9			66	0	S	HIOCIPR0	LP
HP2680A	!2004	0	32	8			66	0	S	HIOPPRT0	LP
HP2688A	!2004	0	32	8			66	0	S	HIOLPRT0	LP
HP2893A	!0101	0	8	0			40	0		HIOCRDR0	CARDRDR
HPLPADCC	!0010	0	32	14	22	960	66	0	S	HIOASLP2	LP
HPLPATP	!000F	0	32	14	22	960	66	0	S	HIOASLP0	LP
HPLPTIC	!0011	0	32	14	22	960	66	0	S	HIOALSP0	LP
HPTERMADCC	!0010	0	16	0	10	960	40	&	JAID	HIOTERM2	TERM
HPTERMATP	!000F	0	16	0	10	960	40	&	JAID	HIOTERM1	TERM
HPTERMTIC	!0011	0	16	0	10	960	40	&	JAID	HIOTERM1	TERM

The DRT# and UNIT# for devices are unique to particular systems, so they will not be supplied with these device defaults. The default DRT# and UNIT# for devices on a given system will be found by polling the physically connected devices during Initial, using the IOMAP procedure discussed in the next section.

The default output device number will be the configured logical device number for terminals or zero for all other devices. For terminals, the default listing will show the output device as "&".

FACTS TO NOTE

o   Despite its name, DEFDATA.PUB.SYS is *not* a file, but a table. It is not easily modified. The only possibility for modifying DEFDATA is to use the utility DISKED2 and then coldstart. Therefore, it is recommended that the file be left alone; if any user is unhappy with the defaults, they can be overridden during the Sysdump or Initial dialogues.

2-9

INTERNALS

The structure of the DEFDATA file is shown in the next two slides. Detailed information can be found from the MPE Tables Manual, but a few facts are useful to know:

o   Entries follow the header in order, and the file is physically contiguous. There are no links between the header and entry 0, or between the entries themselves.

o   Word 3 of the header will always have value "1". Entries in this table are of variable size, and this word is included only for consistency with other MPE tables.

o   In each entry, word 19.(6:1) indicates the significance of word 12. If word 19(6:1) = 1, then word 12 contains a pointer to the default output device name. If word 19(6:1) = 0, then word 12 contains the number of the output device.

o   After the driver name in each entry (words 27 to 30), values are arranged randomly and are found through pointers in words 10, 11 and 12.

o   In word 18, bits have the following definitions:

   J = Job Accepting
   A = Data Accepting
   I = Interactive
   D = Duplicative
   Sp.St. = Spool State

   In word 19, bits have the following definitions:

   CR = Core Resident
   DS = DS device
   SQ = Spool Queues
   CL = Indicates whether the output device given is a device class.
   AI = Default auto increment (DRT or Unit)

   In word 20, bit 7 has the following definition:

   AR = Auto reply

o   The last part of each entry is the CS LDTX area. This is still undergoing revision. It is a variable length list, which causes the entire entry to be variable in length.

HEWLETT PACKARD

DEFDATA Table Lookup File

HEADER FORMAT

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	CHECKSUM															
1	VERSION															
2	TOTAL TABLE SIZE (in words)															
3	ENTRY SIZE (set to 1)															
4	# OF TABLE ENTRIES															

ENTRY FORMAT

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0			D													
1			E													
2			V													
3			I													
4			C										N			
5			E										A			
6													M			
7													E			
8	Total Device Entry Size (in words)															
9	Number of Device Classes for this Device (currently set to 1)															
10	Device Class Name list pointer (entry relative)															
11	Terminal Descriptor file name pointer (entry relative)															
12	Default output device or pointer to class name (entry relative)															
13	CS LDTX entry pointer (currently set to 0)															
14	RESERVED FOR FUTURE USE															
15	Device ID Code															
16	RESERVED FOR FUTURE USE															

mmtrng12

This page intentionally left blank.

HEWLETT PACKARD

DEFDATA Table Lookup File (Cont.)

|		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17		RESERVED FOR FUTURE USE															
18		Device Type					Subtype				J	A	I	D	Sp.St.	
19		Chan #			CR	DS	SQ	CL	AI	Record Width						
20		Default term type					AR	RESERVED							
21		Term Speed															
22		RESERVED FOR FUTURE USE															
23		RESERVED FOR FUTURE USE															
24		RESERVED FOR FUTURE USE															
25		RESERVED FOR FUTURE USE															
26		RESERVED FOR FUTURE USE															
27		Driver Name															
		Terminal Descriptor File Name															
		Terminal Descriptor Group Name															
		Terminal Descriptor Account Name															
		Output Device Class Name															
		Device Class Name															
		CS LDTX Area (to be formalized at a later date)															

mmtmg16

The DEFDATA file is loaded onto the system during a coldload. The address of DEFDATA can be found by consulting the disc cold load information table, as shown in the next slide. DEFDATA information begins at location FAEFTR+75.

Sysdump and Initial make use of DEFDATA when I/O devices are being added to the system configuration. If a user of either program answers "yes" to "LIST DEVICE DEFAULTS?" then the program will scan DEFDATA, format the default information from each entry, and list this information to the user's screen.

If a user of Sysdump or Initial provides a device name (in format HPxxxxn) when prompted with "DEVICE NAME?" the program will scan DEFDATA, find the correct device entry, and use its default information in the prompts for device-specific information.

HEWLETT
PACKARD

Table Lookup

Disc
Cold Load
Information
Table

Initial/Sysdump
Communication
Record

segment size

memory address

disc address

Table
Lookup
Buffer

FAEFTR + 75

Initial's Segments
start here

FAEFTR + 80

Sysdump

LIST DEVICE DEFAULTS?

DEVICE NAME?

DEFDATA

specified entry

Initial

LIST DEVICE DEFAULTS?

DEVICE NAME?

mmtrng17

2-15
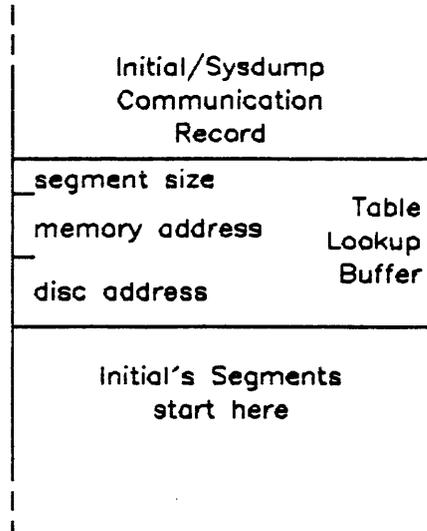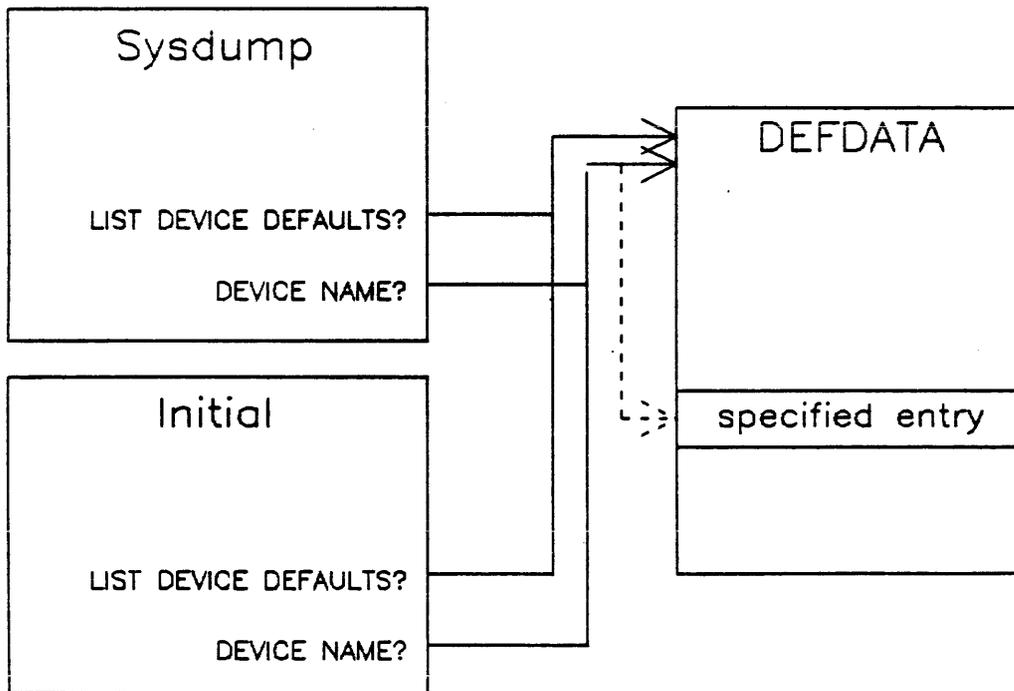
I/O Map Facility

The I/O Map facility supplies information about the physical I/O configuration of the system during Initial. This facility describes the physical arrangement of devices on the system to aid in the logical configuration of I/O devices in Initial.

To examine this map, a step has been added to the Initial dialogue:

Step	Dialogue
4	I/O CONFIGURATION CHANGES? Y
4.1	IOMAP? Enter YES for a list of physically connected devices. The format is:

IMB	CHAN	DEV	DRT	UNIT	DESCRIPTION	LABEL/STATUS
0	1	0	8	0-5	TIC Terminal Controller	
			8	7	TIC Terminal Controller	
	4	1	33	0	7914 Disc Drive	MH7914U0
		2	34	0	7935 Disc Drive	NOT READY
		4	36	0	2680 Page Printer	
	12	1	97	?	7970E Mag Tape	

FACTS TO NOTE

o   The HP7970E tape drive will only show on the I/O Map if it is loaded and online. This is a function of the controller. If the I/O Map can "see" the 7970E controller, but there are no units loaded and online, a question mark ("?") will be displayed for a unit.

o   The DRT number supplied with the I/O Map can be used as input to Initial.

7970E NOT supported currently on S/37 9/4/84

INTERNALS

When an I/O Map is requested in the Initial dialogue, the IOMAP procedure of Initial identifies the I/O hardware units on the system by taking the following steps:

1. IOMAP polls the IMBs on the system by issuing SMASK and RMASK assembly instructions.

2. When an IMB is located, IOMAP performs a rollcall to find the channels connected to the IMB.

3. Channel programs identify the type of each controller on the channel.

4. From the controller level, protocols vary. The units connected to each controller are identified and printed in the I/O Map listing as they are found.

The units connected to the system will be polled and listed in the following order:

   All the units on the first controller on the first channel on the first IMB;

   All the units on the second controller on the first channel on the first IMB;

   . . . . .

   All the units on the last controller on the last channel on the last IMB.

When the last unit has been listed, the Initial dialogue will continue.

HEWLETT PACKARD

I/O Map Facility

I/O CONFIGURATION CHANGES?   yes
IOMAP?   yes
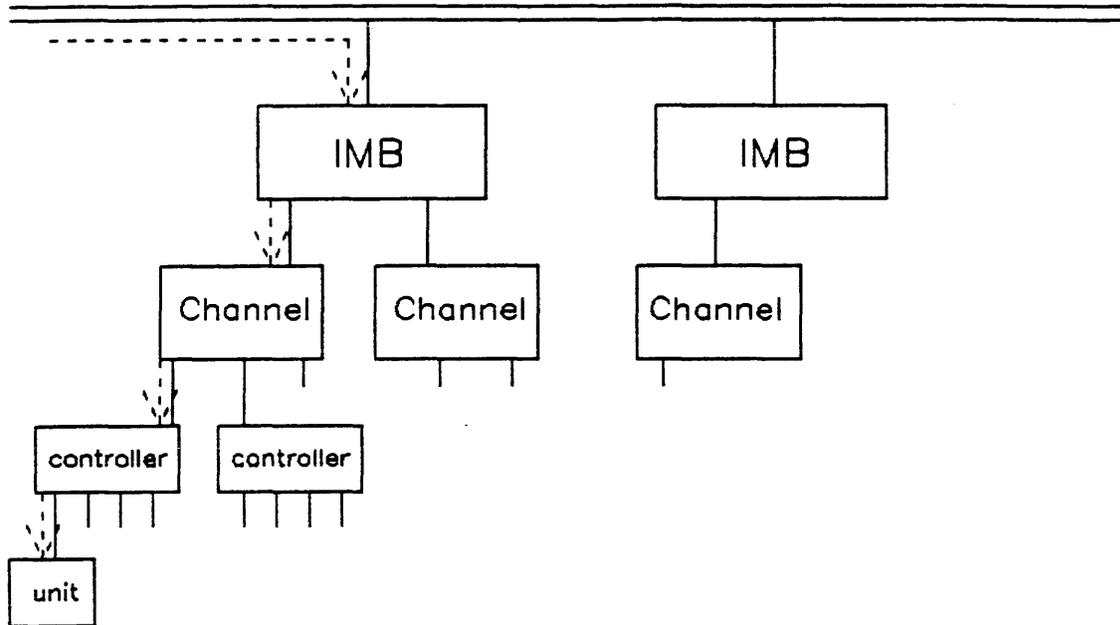
IMB	CHAN	DEV	DRT	UNIT	DESCRIPTION	LABEL/STATUS
0	1	0	8	0–11	ATP Terminal Controller	
			8	24–36	ATP Terminal Controller	
	2	0	16	0–7	ADCC Terminal Controller	
	3	0	24	0–3	ADCC Terminal Controller	
	5	1	41	0	7970E Mag Tape	
			41	2	7970E Mag Tape	
	6	1	49	0	7920 Disc Drive	DISC1
			49	1	7925 Disc Drive	DISC2
			49	2	7906 Disc Drive	NOT READY
			49	7	7905 Disc Drive	READ ERROR

mmimg13

Backup Automation

Two new commands are being introduced with MPE Mighty Mouse that will simplify system backup procedures. These commands, :PARTBACKUP and :FULLBACKUP, provide an alternative to going through the Sysdump dialogue. The only user interaction required with these commands is to mount tapes and reply to tape requests; even these replies are unnecessary if the tape drives are configured as *auto-reply*.

The format of the commands is:

```
:PARTBACKUP [dumptape][,auxlistfile]

:FULLBACKUP [dumptape][,auxlistfile]


Dumptape default is :FILE DUMPTAPE;DEV=TAPE

Auxlistfile default is :FILE SYSDLIST;DEV=$STDLIST
```

:FULLBACKUP will store all files on the system. Upon successful completion of a full backup, the current date is stored in the CONFDATA file for future use by :PARTBACKUP.

:PARTBACKUP will store all files that have been modified since the last full backup. The full backup date stored in CONFDATA will be compared against the modify date in each file's file label; when the modify date is later than the full backup date, that file will be stored.

The command executor for :FULLBACKUP and :PARTBACKUP will check for users on the system when a backup is initiated. If there are users on the system a warning will be displayed:

** WARNING ** USERS ON SYSTEM, SOME FILES MAY NOT BE STORED

FACTS TO NOTE

o   System default information: The two new backup commands will behave as if the operator had responded, "NO", to the Sysdump prompt, "ANY CHANGES?" Whatever the current system configuration is, it will remain the same.

INTERNALS

:PARTBACKUP and :FULLBACKUP are essentially entry points into the Sysdump code. No major changes have been made to MPE code or tables for these friendliness features.

System Tables Dialogue Improvement

Currently, when a user goes through the Sysdump dialogue to change the size of system tables, he has to know the minimum and maximum sizes that are allowed. The only way to find these minimums and maximums is to look them up in a manual. With the introduction of MPE Mighty Mouse, the minimum and maximum sizes for tables are provided in the prompts themselves.

DIALOGUE CHANGES FOR SYSDUMP

The following questions in the Sysdump dialogue have been modified:

Step	Dialogue
0	MEMORY SIZE = <XXXX> (MIN=256, MAX=4096)?
75	CST = <XXX> (MIN=80, MAX=2048)?
76	EXTENDED CST = <XXXX> (MIN=16, MAX=8192)?
77	DST = <XXXX> (MIN=70, MAX=4096)?
78	PCB = <XXX> (MIN=12, MAX=1024)?
79	I/O QUEUE = <XXX> (MIN=20, MAX=1300)?
80	DISC REQUEST TABLE = <XXX> (MIN=20, MAX=900)?
81	TERMINAL BUFFERS PER PORT = <XX> (MIN=1, MAX=99)?
82	SYSTEM BUFFERS = <XXX> (MIN=8, MAX=253)?
83	SWAP TABLE = <XXX> (MIN=128, MAX=4096)?
84	PRIMARY MESSAGE TABLE = <XXX> (MIN=10, MAX=1023)?
84.1	SECONDARY MESSAGE TABLE = <XXX> (MIN=10, MAX=1023)?
85	SPECIAL REQUEST TABLE = <XXX> (MIN=10, MAX=2048)?
86	ICS = <XXXX> (MIN=256, MAX=4096)?
87	UCOP REQUEST TABLE = <XXX> (MIN=1, MAX=1024)?
88	TIMER REQUEST LIST = <XXX> (MIN=6, MAX=1023)?
89	BREAKPOINT TABLE = <XXX> (MIN=1, MAX=1024)?
90	MAX NUMBER OF USER LOGGING PROCESSES = <XX> (MIN=2, MAX=64)?
91	MAX NUMBER OF USERS PER LOGGING PROCESS = <XXX> (MIN=1, MAX=256)?
97	# OF RINS = <YYY> (MIN=XX, MAX=1024, USED=XX)?

```
98     # OF GLOBAL RINS = <YYY> ( MIN=XX, MAX=1024, USED=XX)?

99     # SECONDS TO LOG ON = <XXX> (MIN=10, MAX=600)?

100    MAX # OF CONCURRENT SESSIONS = <XXX> (MIN=1, MAX=500)?

101    MAX # OF CONCURRENT RUNNING JOBS = <XXX> (MIN=1, MAX=500)?

102    DEFAULT JOB CPU TIME LIMIT = <XXXXX> (MIN=0, MAX=32767)?

113    LOG FILE RECORD SIZE(SECTORS) = <X> (MIN=1, MAX=8)?

114    LOGFILE SIZE(RECORDS) = <XXXXX> (MIN=16, MAX=32767)?

116    DIRECTORY USED = <YYY> (MAX=65000)?

127    MAX # OF SPOOLFILES KILOSECTORS = <XXX> (MIN=0, MAX=256)?

128    # OF SECTORS PER SPOOLFILE EXTENTS = <XXXXX> (MIN=128, MAX=32767)?

130A   MAX # OF CONCURRENT RUNNING PROGRAMS = <XXX> (MIN=1, MAX=511)?

131    MAX CODE SEG SIZE = <XXXXX> (MIN=1024, MAX=16384)?

132    MAX # OF CODE SEGMENTS/PROCESS = <XX> (MIN=1, MAX=255)?

133    MAX STACK SIZE = <XXXXX> (MIN=256, MAX=31232)?

134    MAX EXTRA DATA SEG SIZE = <XXXXX> (MIN=0, MAX=32767)?

135    MAX # OF EXTRA DATA SEGMENTS/PROCESS = <XXX> (MIN=0, MAX=255)?

136    STD STACK SIZE = <XXXX> (MIN=256, MAX=4096)?
```

DIALOGUE CHANGES FOR INITIAL

The following questions in the Initial dialogue have been modified:

Step   Dialogue

```
93     MAX # OF SPOOLFILES KILOSECTORS = <XXX> (MIN=0, MAX=256)?

94     # OF SECTORS PER SPOOLFILE EXTENT = <XXXXX> (MIN=128, MAX=32767)?
```

INTERNALS

No significant changes to MPE code or tables were required for this friendliness feature. All changes were implemented by changing the messages generated by Sysdump and Initial.

The major changes and enhancements to MPE for the Mighty Mouse MIT have been covered in the last two chapters. Other portions of MPE have been added or modified to support this MIT, and this chapter will deal with these.

The topics covered in this chapter are:

o   Modifications to Memory Logging

o   New Buffalo driver

Memory Logging

The changes to Memory Logging for MPE Mighty Mouse are fairly superficial. The memory logging module, MEMLOGAN, always checks which CPU it is running on; code has been added to MEMLOGAN to recognize the Series 37 CPU.

For other HP3000 systems, the default sampling interval for memory logging is one hour. For the Series 37, the default sampling interval will be ten minutes.

Buffalo Cartridge Tape Support

The Buffalo cartridge tape is being introduced at approximately the same time as the Mighty Mouse system. Hewlett-Packard expects to sell Buffaloes with most Mighty Mouse systems (indeed, Buffalo cartridge tapes are included in the "standard software configuration" that can be automatically installed with MPE Mighty Mouse) as well as for larger HP3000 systems. Here are a few facts about the Buffalo and how to configure it into the system:

o Product number: HP9144

o 67 Mb streaming cartridge tape drive

o Driver: HIOCTAP1

o Installation instructions:

   Buffalo will be installed as a serial disc device. It is a high speed device with an average data rate handshake capability of 660 Kbytes/second. Buffalo must be connected to a high speed GIC and should not share the primary GIC with the system disc. It is recommended that Buffalo not share a GIC with other discs on the system.

   In the Sysdump or Initial dialogue, configure Buffalo as follows:

```
TYPE                   3
SUPTYPE                3
RECORD WIDTH           128
OUTPUT DEVICE          0
ACCEPT JOBS/SESSIONS   NO
ACCEPT DATA            NO
INTERACTIVE            NO
DUPLICATIVE            NO
INITIALLY SPOOLED      NO
DRIVER NAME            HIOCTAP1
CLASSES                SDISC
                       or User's choice
```

o Interface: through the file system

FACTS TO NOTE

o A cartridge tape may only be used 128 times. On the 128th use of the cartridge, a message will be written to the console indicating that the cartridge is write-protected. The unit will no longer be able to access the cartridge. The user should be sure to make a backup copy of the cartridge before this point.

Error Codes for STARTSESS Intrinsic

This is a list of error conditions that are detected by the STARTSESS intrinsic and returned in the first element of the *errorstat* parameter. A negative error number indicates a warning-level error; the session should still log on. A positive error number indicates an error that prevented the session from logging on.

0   Successful call.

7000   LDEV out of range.

7001   LDEV must not be virtual.

7002   The device specified by LDEV is not a terminal.

7003   LDEV is not free.

7004   LDEV is not job accepting.

7005   LDEV in diagnostic mode.

7006   The LDEV terminal is DOWN.

7007   DOWN pending on LDEV.

1444   A password is required but not specified in LOGONSTR.

1424   Missing or invalid user name in LOGONSTR.

1426   Missing or invalid account name in LOGONSTR.

1438   No such user in the account.

1437   No such account in the system.

1436   No such group in the account.

1431   Specified session user lacks IA capability.

-1458 Bad *termtype* specified in LOGONSTR, default used.

-1459 Invalid PRI specified in LOGONSTR, default used.

-1479 Invalid TIME specified in LOGONSTR, no time limit is used.

-1461 HIPRI then INPRI specified, INPRI used.

-1464 INPRI then HIPRI specified, HIPRI used.

-1462 INPRI too low, lowest valid INPRI used.

-1463 INPRI too high, highest valid INPRI used.

-1465 OUTCLASS specified, ignored.

-1473 RESTART specified, ignored.

-1452 Unknown parameter found in LOGONSTR, ignored.

1412 Logon failed due to JMAT overflow.

1411 Logon failed due to IDD overflow.

-1451 Ignored delimiter.

7042 Group not specified, and there is no default home group.

The following error codes are returned in the ERRORSTAT parameter of ABORTSESS:

0   Successful call.

1   Caller does not have ":ABORTJOB" rights to the job/session, or :JOBSECURITY is low.

2   The session was not found.

3   Cannot abort a session in INTRO state.